UNIVERSIDAD MIGUEL HERNÁNDEZ
Doctorado en Tecnologías Industriales y de Telecomunicación

View-based SLAM in the $2\frac{1}{2}$D Space with Omnidirectional Images and Feature Point Information

Autor: David Valiente García
Directores: Dr. Ing. Óscar Reinoso García
Dr. Ing. Arturo Gil Aparicio

Tesis Doctoral presentada en la Universidad Miguel Hernández para la obtención del título de Doctor del Programa de Doctorado en Tecnologías Industriales y de Telecomunicación

2016

La presente Tesis Doctoral está sustentada por un compendio de trabajos previamente publicados en revistas de impacto, indexadas según *JCR Science Edition*. El cuerpo de dicha tesis queda constituido por los siguientes artículos, cuyas referencias bibliográficas completas se indican a continuación:

- *Creación de un modelo visual del entorno basado en imágenes omnidireccionales*. [46]
  A. Gil, D. Valiente, O. Reinoso, J.M. Marín
  Revista Iberoamericana de Automática e Informática Industrial, RIAI. Vol 9. pp. 441-452. 2012
  ISSN: 1697-7912. Ed. Elsevier.
  JCR-SCI Impact Factor: 0.475, Quartile **Q4**.

- *A modified stochastic gradient descent algorithm for view-based SLAM using omnidirectional images*. [136]
  D. Valiente, A. Gil, L. Fernández, O. Reinoso
  Information Sciences. Vol 279. pp. 326-337. 2014.
  ISSN: 0020-0255. Ed. Elsevier.
  JCR-SCI Impact Factor: 3.364, Quartile **Q1**.

- *A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images*. [135]
  D. Valiente, A. Gil, L. Fernández, O. Reinoso
  Robotics and Autonomous Systems. Vol 62. pp.108-119. 2014.
  ISSN: 0921-8890. Ed. Elsevier.
  JCR-SCI Impact Factor: 1.618, Quartile **Q2**.

AUTORIZACIÓN DE PRESENTACIÓN DE TESIS DOCTORAL POR UN CONJUNTO DE PUBLICACIONES

Directores: Dr. Ing. Óscar Reinoso García y Dr. Ing. Arturo Gil Aparicio

Título de la tesis: ***View-based SLAM in the $2\frac{1}{2}$D Space with Omnidirectional Images and Feature Point Information***

Autor: David Valiente García

Departamento de Ingeniería de Sistemas y Automática
Universidad Miguel Hernández de Elche

Los directores de la tesis reseñada AUTORIZAN SU PRESENTACIÓN EN LA MODALIDAD DE CONJUNTO DE PUBLICACIONES.

En Elche, a de de 2016.

Fdo: Dr. D. Óscar Reinoso García

Fdo: Dr. D. Arturo Gil Aparicio

AUTORIZACIÓN DE PRESENTACIÓN DE TESIS DOCTORAL

Directores: Dr. Ing. Óscar Reinoso García y Dr. Ing. Arturo Gil Aparicio

Título de la tesis: ***View-based SLAM in the $2\frac{1}{2}$D Space with Omnidirectional Images and Feature Point Information***

Autor: David Valiente García

Departamento de Ingeniería de Sistemas y Automática
Universidad Miguel Hernández de Elche

Los directores de la tesis reseñada CONFIRMAN QUE HA SIDO REALIZADA BAJO SU DIRECCIÓN POR D. David Valiente García en el Departamento de Ingeniería de Sistemas y Automática de la Universidad Miguel Hernández de Elche y autorizan su presentación.

En Elche, a de de 2016.

Fdo: Dr. D. Óscar Reinoso García Fdo: Dr. D. Arturo Gil Aparicio

PROGRAMA DE DOCTORADO EN TECNOLOGÍAS INDUSTRIALES
Y DE TELECOMUNICACIÓN

Dr. D. Ignacio Moreno Soriano, Coordinador del Programa de Doctorado en Tecnologías Industriales y de Telecomunicación de la Universidad Miguel Hernández de Elche.

Certifica

Que el trabajo realizado por D. David Valiente García titulado ***View-based SLAM in the $2\frac{1}{2}$D Space with Omnidirectional Images and Feature Point Information*** ha sido dirigido por el Dr. D. Óscar Reinoso García y el Dr. D. Arturo Gil Aparicio y se encuentra en condiciones de ser leído y defendido como Tesis Doctoral ante el correspondiente tribunal en la Universidad Miguel Hernández de Elche.

Lo que firmo para los efectos oportunos en Elche, a de de 2016.

Fdo.: Dr. D. Ignacio Moreno Soriano
Coordinador del Programa de Doctorado en Tecnologías Industriales y de Telecomunicación

DEPARTAMENTO DE CIENCIA DE MATERIALES, ÓPTICA Y TECNOLOGÍA ELECTRÓNICA
Universidad Miguel Hernández de Elche. Ed. Torrevaillo. Avda. de la Universidad s/n, 03202 Elche
Telf: 96 665 8498

Abstract

Mobile robotics has experienced an important proliferation in the recent days, with many fields of application available. A great variety of different mobile robots are present in different sectors of society, most of which are designated as autonomous. This term implies that the robot manages to operate itself, without any special supervision. To that purpose, the robot must be enabled to gather information from the environment in order to build its own understanding, which yields a map estimation. The scope of this thesis is focused on this aspect: the map building process with visual information from the environment. This process entails a non-trivial task, since it poses a challenge when it comes to obtaining a simultaneous estimation of the localization of the robot, and also of the map. This leads to one of the most essential paradigms in such context: the problem of SLAM (Simultaneous Localization And Mapping).

Different sort of information can be acquired by a set of well known sensors boarded on the robot, such as laser, sonar, GPS, etc. However, digital cameras have arisen as a promising alternative. They provide low consumption, low cost and lightness. Moreover, these visual sensors represent a potential tool for encoding large amounts of information within an only image. Thus in this work we propose a new map model, embedded in a visual SLAM approach, which is solely based on the use of omnidirectional images acquired with a monocular camera. An important strength of this camera resides on its particular wide field of view. In addition, we process the information extracted from feature points, as physical landmarks which are visually detected on the images. This idea differs from traditional approaches, which basically concentrate on the accumulative scheme for the incremental re-estimation of all the landmarks in the map.

Regarding the core algorithms under this context of visual SLAM, this thesis proposes several improvements to the robustness of the standard algorithm models. In particular, we present a customized offline model, which is capable of reducing those harmful effects associated with non-linear noise, as those introduced by catadioptric cameras. Many of the most accepted approaches are highly sensitive to this effects and fail to provide convergence assurance for the final estimation.

Moreover, another recognized drawback of former approaches is the management of the uncertainty of the system. This is usually originated by the same non-linear sources. Consequently, the estimation may be severely impaired as errors dramatically compromise its convergence. In this sense, this thesis contributes to the achievement of a robust model for uncertainty reduction, which is dynamically devised.

As a general commitment along all this thesis, we establish an experimental framework for all the different approaches and contributions made as a result of the research conducted in this context. Thus both simulated and real dataset experiments are repeatedly presented along this document.

Resumen

Actualmente dentro del campo de la robótica móvil se ha experimentado una importante proliferación de aplicaciones. Encontramos una gran variedad de robots móviles presentes en diversos sectores de nuestra sociedad, muchos de los cuales son aceptados como autónomos. Este término implica que el robot es capaz de operar por sí mismo, sin ningún tipo de supervisión especial. Para tal efecto, el robot debe ser habilitado para recoger información del entorno, de modo que pueda construir su propio entendimiento del mismo, tal como es una estimación de un mapa. El ámbito de esta tesis se concentra en este aspecto: el proceso de construcción de mapas con información visual del entorno. Este proceso implica una tarea de resolución no trivial, ya que plantea un reto en lo que se refiere a la obtención simultánea de la localización del robot, pero además del mapa. Esto último dirige hacia uno de los paradigmas más esenciales en este contexto: el problema de SLAM (*Simultaneous Localization And Mapping*).

Distintos tipos de información pueden adquirirse mediante un conjunto bien conocido de sensores embarcados en el robot, tales como láser, sónar, GPS, etc. Sin embargo, las cámaras digitales emergen como una prometedora alternativa. Proporcionan bajo consumo, bajo coste y ligereza. Además, estos sensores visuales suponen una potencial herramienta para codificar grandes cantidades de información en una única imagen. Así, en este trabajo proponemos un nuevo modelo de mapa, embebido dentro de una propuesta de SLAM visual, la cual está basada únicamente en el uso de imágenes omnidireccionales, adquiridas con una cámanra monocular. Una importante fortaleza de esta cámara radica en su particular amplio campo de visión. Además, procesamos la información extraída de puntos característicos, entendidos como marcas físicas, los cuales son detectadas visualmente sobre las imágenes. Esta idea difiere de las propuestas tradicionales, cuyo objeto se concentra en un esquema acumulativo para la reestimación de todas las marcas del mapa.

En cuanto al algoritmo núcleo dentro de este contexto de SLAM visual, esta tesis propone varias mejoras para la robustez de los modelos de algoritmos estándar. En particular, presentamos un modelo personalizado de tipo *offline*, el cual es capaz de reducir los efectos perjudiciales asociados con el ruido no lineal, tales como los introducidos por las cámaras catadióptricas. Muchas de las propuestas más aceptadas son altamente sentibles a estos efectos, y no logran asegurar la convergencia de la estimación final.

Por otra parte, otro de los inconvenientes reconocidos de las primeras propuestas es la gestión de la incertidumbre del sistema. Normalmente esto es debido a las mismas fuentes no lineales. Consecuentemente, la estimación puede verse severamente dañada, puesto que compromete dramáticamente su convergencia. En este sentido, esta tesis contribuye a la consecución de un modelo robusto para la reducción de la incertidumbre, la cual es concebida dinámicamente.

Como un compromiso general a lo largo de toda esta tesis, establecemos un marco experimental para todas las propuestas y contribuciones surgidas de los resultados de las investigaciones en este ámbito. De este modo, se presentan experimentos repetidamente a lo largo de este documento, tanto con conjuntos de datos simulados como reales.

Conclusiones

Como consecuencia del periodo de investigación bajo el marco de esta tesis, distintos resultados han sido obtenidos, los cuales han permitido realizar diversas aportaciones. Las más relevantes han sido comunicadas en revistas y congresos internacionales y nacionales. Además, el conjunto de artículos que sustentan el cuerpo de esta tesis, comprenden las más relevantes. En este sentido, a continuación se exponen las conclusiones extraídas de estas publicaciones y su relación directa con esta tesis, así como con los capítulos del documento. Hay que destacar que cada capítulo incluye un apartado dedicado a conclusiones parciales sobre el mismo. De modo similar, el último capítulo del documento, sintetiza todas las conclusiones, aportaciones y posibles trabajos futuros. Para más información, se sugiere su consulta detallada.

El primer artículo [46] expone el modelo principal definido en esta tesis. Se trata del nuevo modelo de mapa visual, conformado por un conjunto reducido de imágenes omnidireccionales. Los resultados de esta publicación demuestran la validez de la propuesta, la cual se denota como un modelo de mapa visual basado en un sistema EKF. El Capítulo 4 extiende los detalles de la explicación de esta aportación, así como sus resultados. En segundo lugar, la aportación hecha en la publicación [136], presenta un trabajo donde se adapta la estructura del algoritmo SGD a la geometría omnidireccional y al modelo propuesto anteriormente. También incluye resultados comparativos de eficiencia y precisión frente al método estándar de SGD, como validación de la propuesta para tratar los efectos indeseados de las fuentes de ruido no lineal. Los detalles relacionados con esta implementación son presentados en el Capítulo 5. En tercer lugar, la publicación [135] presenta un estudio exhaustivo para comparar dos de las aportaciones principales hechas en esta tesis. Se trata del modelo de mapa visual basado en un sistema EKF, apoyado en un conjunto de vistas omnidireccionales, y la anterior propuesta constituida por una modificación del algoritmo SGD, adaptado a la geometría omnidireccional. Los resultados confirman que el método SGD, pese a ser *offline*, demuestra ser una robusta alternativa para mitigar efectos no lineales indeseados, frente a los cuales el sistema EKF es especialmente sensible y ve como su convergencia y funcionamiento se ven seriamente afectados. Los Capítulos 4 y 5 extienden con mayor profundidad los desarrollos realizados a tal efecto, así como todos los resultados de dicha comparación.

Agradecimientos

Después de todo este largo camino resulta complicado expresar en unas pocas líneas todos los agradecimientos que me gustaría. En primer lugar, quiero dar las gracias a mis directores de tesis, Óscar Reinoso y Arturo Gil, por haberme brindado la oportunidad de desarrollar este trabajo y por todo su apoyo y confianza desde el primer momento. Sus directrices, recomendaciones y comentarios han sido fundamentales para dar forma a esta tesis.

Me gustaría también dar las gracias a todos los miembros del Departamento de Ingeniería de Sistemas y Automática y del grupo de investigación ARVC: Luís Payá, Luís Miguel Jiménez, David Úbeda, José María Marín, y especialmente a mis compañeros del laboratorio, Loren, Fran, Mónica y Miguel por su ayuda y colaboración, y en particular, por haberme permitido compartir tantos buenos momentos de amistad durante el día a día.

Agradecer también a Jaime Valls Miró y Maani Ghaffari Jadidi el tiempo que dedicaron a compartir sus ideas durante mi estancia en la University of Technology Sydney, las cuales fueron muy valiosas para el desarrollo de nuevas contribuciones dentro de esta tesis.

Por otro lado, quiero dar las gracias a todos mis amigos, por ser junto con mi familia, parte esencial de mi vida. Compartir momentos y experiencias con vosotros me ha hecho crecer como persona y ha supuesto una gran energía para seguir adelante y afrontar retos en todos los ámbitos, tanto personales, como académicos y profesionales.

De un modo muy especial, quiero dar las gracias a mi familia. A mi padre y a mi madre, mi agradecimiento infinito por todo vuestro esfuerzo, sacrificio y dedicación para hacernos salir adelante, por el apoyo y confianza que siempre me habéis dado y por los valores que me habéis transmitido. Todo lo que soy hoy como persona se lo debo a ellos. Junto con mi abuelo, han sido el modelo y guía durante toda mi vida. A mi hermana, por ser cómplice de tantas situaciones y sensaciones indescriptibles. A mis tíos, por ser otro pilar y referente dentro de mi familia. Y a mis primos, por ser también mis hermanos.

Finalmente, y en general, quiero expresar mi agradecimiento a todas las personas que durante algún momento me han acompañado en este camino, y que de un modo más o menos directo, han supuesto un apoyo para la consecución de este trabajo.

A todos y cada uno de vosotros: gracias.

A mis padres

List of Tables

List of Figures

List of Algorithms

Contents

1 Introduction

Robotics has shown an important proliferation in the recent days. Robots have been present in our society for several decades, however, the first generation were only introduced in the industrial sector, where the main purpose was the optimization of operators' tasks and their safety. Another common example in its early days is the humanoid robot, being at first instance a mere machine with human resemblance and appearance, in shape and movement. However, during the last decade remarkable advances have been achieved. The fusion of Artificial Intelligence within the field of robotics has allowed to provide powerful autonomous systems with an endless list of possible applications. Their incipience started at military applications, but they were rapidly extended to a diverse range of tasks which sustain the tremendous growth experienced by robotics nowadays. Some examples are: space robots, rescue robots, medical and assistance robots, domestic robots, education and entertainment robots, manufacturing and agricultural robots. Special mention must be made of the commercial boost in the sales of drones. These promising UAVs demonstrate capabilities to meet the requirements established by almost any of the possible field of applications mentioned above. Besides these, another interesting aspect that stands out is the recent trend of research towards total autonomous vehicles, as well known as self-driving cars. Figure 1.1 graphically synthesizes all these examples.

In general, any robot targeted at any of the possible field of applications should comprise an overall capability balanced between autonomy, environment perception, decision-making and adaptability. In this sense, the sensors of a robot represent a key aspect to finally determine its possible capabilities, assets and final purpose. Among the most common sensors, we can find: laser, sonar, encoders, pressure sensors, capacitive sensors, GPS, IMU, etc. In this thesis we have relied on another sort of sensors,

(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Figure 1.1: Different robotics applications. Figure 1.1(a) shows the *Mars Rover*, a NASA explorer. Figure 1.1(b) shows the *da Vinci* surgical system. Figure 1.1(c) shows a Samsung autonomous cleaning robot. Figure 1.1(d) shows a shipping drone used by Amazon. Figure 1.1(e) shows the self-driving car developed by Google. Figure 1.1(f) shows an agricultural robot by Bosch. Figure 1.1(g) shows the *Nitrofirex*, a firefighting UAV. Figure 1.1(h) shows a the AUV *Sirus*. Figure 1.1(i) shows the JJCR *H20*, a domestic mini-drone.

the visual systems. To date, most of the approaches have extensively used the former sensors, certainly due to their precision, robustness and maturity. However, more recently the tendency has turned to the use of visual information by means of digital cameras. Many applications benefit from the use of these sensors, whose characteristics outperform the previous sensors such as lasers in terms of the amount of usable information from the environment, but also due to their low cost, light weight and low consumption principally.

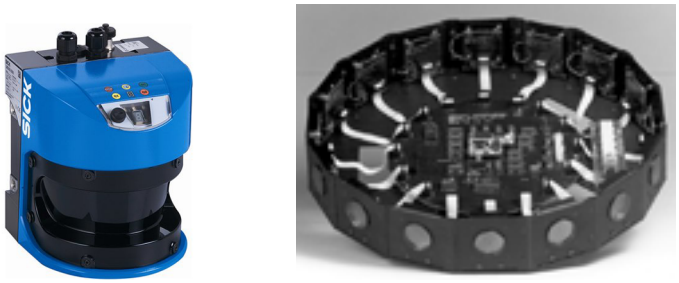
The next big challenge for an autonomous robot is to identify the paradigm and algorithms to process the information gathered. This is essential when it comes to decision and action-related situations. The robot must be autonomous in order to determine its own understanding of the environment, its current position inside this environment and the way to traverse it while interacting with the elements of such workspace. This exposition leads to the formulation of three fundamental concepts in mobile robotics: Localization, Mapping and Path Planning. The three of them directly translate the ideas exposed above. Their interrelation generates the most essential paradigm in such context: the problem of SLAM, (Simultaneous Localization And Mapping). Such problem poses a non-trivial challenge since it involves a laborious process to simultaneously deal with the Mapping and the Localization of the robot. This fact brings a challenge with regard to complexity, as the procedure is expected to work incrementally and to return a coherent representation of the environment. Moreover, the starting point assumes a void knowledge of the environment. Besides, the existence of noise sources becomes accountable for undesired effects which aggravate and jeopardize the final estimation. The Path Planning poses an extra challenge since it adds another decision stage to the previous process. As a result, different patterns arise. We can differentiate between Classic Exploration and Active Localization. The first one is referred to the navigation task conducted by the robot while it explores an environment and constructs a map simultaneously. The second term corresponds to the disambiguation needed when the robot already possesses a map estimation. Here, the robot keeps moving in order to get a more accurate localization with similar data measures from the environment. Figure 1.2 presents the integration of the previous terms and the resulting interrelations. Having presented these terms, we can situate the work conducted in this thesis within the research area of SLAM.

1.1 Scope

In the field of mobile robot applications, as already commented, the problem of SLAM is a crucial aspect to deal with, due to the necessity for a complete map of the environment which aids in the simultaneous localization of the robot. Thus this becomes the major expected capability in order to designate a robot as autonomous.

To date, SLAM approaches have been differentiated according to several factors, such as the way to estimate the representation of the map, the main algorithm for computing a solution and the kind of sensor to extract information from the environment. For instance, several map models were obtained thanks to the extensive use of laser data range [22] and sonar [54, 79]. Similar examples are extended to 3D [131]

Figure 1.2: Integrated exploration and its relations within the framework of SLAM.

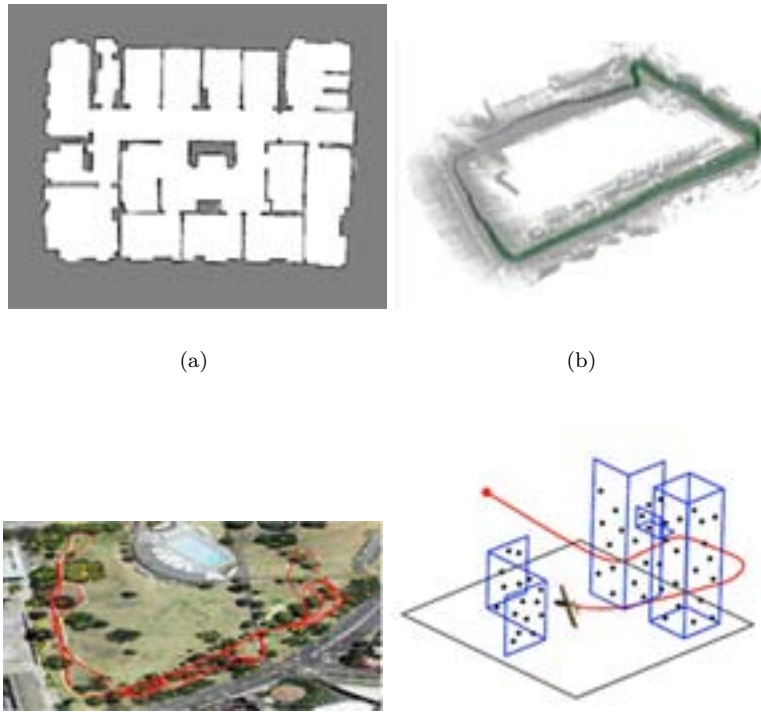


(a) (b)

Figure 1.3: Figure 1.3(a) shows an example of a laser sensor. Figure 1.3(b) shows an example of a sonar sensor.

and [3], where they use laser data in the first case, and laser combined with orientation data in the second. Laser approaches are commonly more precise than sonar [145] to these purposes. Figure 1.3 presents two examples of such sensors. In this sense, maps were principally generated by two representation models [127] and [94], corresponding, respectively, to 2D occupancy grid maps based on raw laser, and 2D landmark-based maps focused on the extraction of features, described from laser data measurements. Again, these models were rapidly extrapolated to 3D [36], [63]. Figure 1.4 shows several examples of these kind of maps.

More recently, visual sensors have reached a great emergence as the main tool



(a) (b)

(c) (d)

Figure 1.4: Occupancy and landmark maps. Figure 1.4(c) and Figure 1.4(d) present 2D and 3D landmark maps, respectively. Figure 1.4(a) and Figure 1.4(b) present 2D and 3D occupancy maps, respectively.

for collecting information for these map models. They represent a promising alternative to the classic sensors such as later or sonar in terms of the amount of usable information they can provide for the mapping tasks. Many approaches have concentrated on the use of two calibrated cameras, commonly known as stereo cameras, in order to extract sets of 3D visual landmarks with their visual description [43]. Other approaches merely exploit a monocular camera to estimate 3D visual landmarks [20, 69, 27]. They initialize the coordinates of each 3D landmark relying on an inverse depth parametrization, since there exists lack of scale on the distance to each landmark. Omnidirectional cameras, have especially proven the quality for collecting large amounts of visual information. They have been used solely [152, 106], and some others have even arranged two omnidirectional cameras, in order to take the best advantage of their wider field of view [112].

Another crucial aspect is the sort of estimation algorithm used as the kernel of the SLAM system. The most extensively used are online methods such as the

Extended Kalman Filter (EKF) [26], Rao-Blackwellized particle filters [94, 133, 93] and offline algorithms, such as the Stochastic Gradient Descent (SGD) [53], Multi-Level Relaxation [39] or Levenberg-Marquardt [87]. Over the last years, great efforts have been made on the study of the EKF-based SLAM methods sustained by visual sensors [26, 25, 20] and [43, 105, 55]. They all coincide on the position estimation of 3D visual landmark sets in a common reference system. The first cited group used monocular cameras with similar parametrization whereas the second group focused on larger layouts. These approaches are liable to encounter difficulties in assuring the convergence of the solution, particularly in the presence of non-linear errors.

A last mention has to be made of the applications focused on a different manner to process the visual information, that is, the appearance-based approaches. Contrarily to feature point methods (landmark-based), the latest group exploits the visual information in the form of an image as a whole. Each computed visual description consists of information processed from a specific image. This line is followed in approaches such as [37]. In general, these techniques reveal more efficient results, aiming at time reduction, but at a significant cost of precision.

Synthesizing, within the main field of mobile robotics, we can identify the scope for the major research of this thesis as the visual SLAM with feature point information provided by omnidirectional cameras.

1.1.1 Motivation

Having briefly introduced the scope of this thesis, a deeper contextualization level has to be exposed in order to justify the main reason which drove us to conduct the research under the framework of this thesis.

As commented above, we can assume three main factors which determine the final behaviour of a SLAM system. Therefore these factors become liable to be analyzed in depth so as to outline the motivation of this thesis:

1. Map model.
2. Sensors used.
3. Estimation algorithm.

Focalizing on these points, we encounter that one of the most precise solutions is provided by those map models consisting of a 3D estimation of visual landmarks [139], as physic points in the environment [59]. The visual approach for such maps demonstrate a better efficiency in terms of the final representation, since they can effortlessly encode the environment by means of a set of visual landmarks. In contrast, occupancy maps require more expensive computational resources to process and store the information, with high dependance on the resolution of the grid. Consequently, here we target the map model based on feature points information.

The key for such visual landmarks approaches relies on an accumulative estimation of the landmarks as long as the robot explores the environment [26]. Thus, whenever new information of the environment is discovered, it has to be incrementally added to the procedure by which the environment is re-estimated. In the end, the final map comprises the total set of estimated landmarks and the path followed by the robot referred to that map. Not to mention that the localization within this map has to be performed simultaneously. All together pose a real challenge, since the complexity of the problem escalates dramatically with the number landmarks, that is, with the number of variables. Moreover, some other aspects may intensify this complexity escalation. Acknowledged estimation algorithms such as the EKF evidence complexities with order $O(N^2)$, being N the number of variables representing the landmarks estimated in the map. This issue may be aggravated to the extent of considering a combination of diverse sensors with high computational requirements such as laser [56]. As a result, this issue affects negatively to the dimension of the map and the complexity of the entire process, which finally becomes critical when there is a high rate of re-estimation.

Despite the fact that complexity reduction and similar optimization contributions might be the obvious solution to work on, there are many authors who already concentrated on such approaches [57, 15]. Contrarily to this, here we find a challenging motivation to establish a research line on the definition of a novel map approach. We strongly believe that a compact representation of the environment can be proposed by means of a reduced set of images. The projection nature of omnidirectional cameras allows us to encode large amounts of visual information in a single image. Therefore this led us to devise a map composed by views, and consequently, a localization model in accordance to such map model. It is worth highlighting the relevance of a well-designed observation model in order to provide a feasible localization, especially with light computational requisites. The main purpose is to encode the environment with a reduced number of variables.

Regarding the kind of visual sensors, firstly we justify this election due to their powerful advantages in contrast to laser and sonar:

- Low weight and dimensions.
- Low cost.
- Low consumption.
- High amount of information in a single image.
- High level of processing tasks such as recognition.

Two approaches can be used in order to process the visual information provided by these sensors: feature-based and appearance-based. We opt for the extraction of feature points as they represent a mature and robust solution for the precision requirements we are dealing with in this thesis. We consider a sort of environments where there are distinctive physic details to be detected, as well as noticeable changes in

the point of view. Besides we intent to provide with a feasible observation model which relies on well-defined geometric relations, based on the information gathered extracted from the feature points. These facts suggest that feature methods are a robust option for such context. On the contrary, appearance-based methods, despite the fact that provide speeded-up results, are less precise under such circumstances. They encode a whole image in an only visual descriptor, thus dismissing valuable information about the scene. They also suffer from visual aliasing, that is, they fail to discern between similar visual information. However, on the feature methods downside, we have to deal with the calibration estimation and its associated errors, but also with problems such as the invariance to image changes, in terms of the type of visual descriptor chosen. In this sense, there are some methods which stand out from the others, as SIFT [86] and SURF [7]. When dealing with a general movement where relevant changes in the point of view are expected, and therefore a significant robustness in the detection is required, studies such as [44] demonstrates that SIFT fails to extract robust feature points when there are distance and orientation changes to these points. The results presented in [42] suggest that SURF provides a balanced solution in terms of efficiency and precision. According to this, SURF represents the visual description method employed in the framework of this thesis.

As for the estimation algorithm, as recently mentioned, the EKF has demonstrated its large acceptance in the community, as one of the best exponent of solver methods in the visual SLAM field. Nevertheless, these methods are troublesome in the presence of non-linear errors as they present difficulties in maintaining the convergence of the estimation. This situation normally appears in the presence of non-Gaussian errors introduced by the observation measurement, which usually causes data association problems [89]. Such errors are usually provoked by sensory input, in particular, omnidirectional sensors are significantly susceptible to cause this issue [137], due to its high non-linear nature. Other offline algorithms [151, 148, 128] emerge as alternatives to enhance stability under non-linear circumstances. Within this last group, specific techniques are defined in order to take advantage of different optimization techniques embedded in the core of the estimation algorithm [76, 33].

The main consequences in terms of non-linearities and instabilities, made us state a new research branch on the algorithm side. Here we seek to evaluate different estimation methods from the most typically employed. In this sense, we researched on a new estimation algorithm and devised several considerations when assuming the new map model sustained by omnidirectional images.

Another recognized drawback of former approaches is the management of the uncertainty of the system. This is generally derived from the same non-linear sources which mainly originate the sensor data. Taking no action on the uncertainty may severely compromise the convergence [64] of many estimators. This fact led us to outline a research line on the uncertainty reduction in compliance with real time oriented applications

1.1.2 Objectives

According to the motivation introduced in the previous section, the main objective of this thesis is to design a new map model, as the core of a view-based SLAM approach, based on a reduced set of omnidirectional images acquired with a single camera, from which certain visual information associated with a set of feature point is employed. To that final purpose, several goals have to be established in relation to:

- New map model proposal
  - Review the state of the art.
  - Study on the possibilities provided by monocular cameras, omnidirectional in particular, to represent the environment in a simpler and more compact manner with less number of variables.
  - Propose a new observation model to compute the motion transformation between poses, using omnidirectional views, as the key parts of the map.
  - Adapt the reference system to the omnidirectional geometry.
  - Extract preliminary visual measurements usable in a real time context.
  - Devise a reliable image initialization procedure for the variables of the map model.
  - Real data set acquisition and image processing.
  - Test the validity, efficiency, and general performance of the new map approach.
- Robustness against non-linearities
  - Overcome typical drawbacks presented by traditional map approaches.
  - Study on the different core estimation algorithms available.
  - Enhance the motion transformation computation.
  - Study on the improvements of the feature matching process.
  - Validation and comparative tests definition.
- Uncertainty reduction
  - Aid in the convergence assurance.
  - Study on the Bayesian techniques to obtain a probabilistic distribution of the visual information of the environment.
  - Study on the Information-based techniques to interrelate to the uncertainty of the system.
  - Propose a robust initialization view strategy based on the uncertainty reduction.
  - Validation and comparative tests, including larger scenarios.

1.2 Contributions

All the research conducted towards the achievement of the previous objectives resulted in several major contributions, which can be chronologically listed according to their development and implementation as follows:

- Implementation of a motion transformation model between poses at which omnidirectional images are acquired: performance results and a visual odometry approach were proposed.
- Definition and implementation of a new representation of the environment: map model based on a reduced set of omnidirectional views. This represents the basis for our view-based SLAM approach.
- Enhancement of the matching process within the observation model. Achievement through the propagation of the uncertainty of the system.
- Implementation of a modified SGD solver algorithm, adapted to the omnidirectional geometry of our view-based SLAM approach, so as to enhance robustness against non-linearities, in contrast to traditional solvers.
- Implementation of a new view initialization mechanism which accounts for information gain and losses within the SLAM system. As a result, the uncertainty of the system is reduced and the convergence of final estimation assured.

1.3 Set of Publications Supporting this Thesis

The major implementations and contributions made in this thesis are supported by a set of publications in journals ranked in the JCR Science Edition. The following journal papers support the work conducted in this document, which represent the result of the research under the scope of this thesis, with direct relation to the motivation and objectives already established:

- *Creación de un modelo visual del entorno basado en imágenes omnidireccionales*. [46]
  A. Gil, D. Valiente, O. Reinoso, J.M. Marín
  Revista Iberoamericana de Automática e Informática Industrial, RIAI. Vol 9. pp. 441-452. 2012
  ISSN: 1697-7912. Ed. Elsevier.
  JCR-SCI Impact Factor: 0.475, Quartile **Q4**.

- *A modified stochastic gradient descent algorithm for view-based SLAM using omnidirectional images*. [136]
  D. Valiente, A. Gil, L. Fernández, O. Reinoso
  Information Sciences. Vol 279. pp. 326-337. 2014.

ISSN: 0020-0255. Ed. Elsevier.
JCR-SCI Impact Factor: 3.364, Quartile **Q1**.

- *A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images*. [135]
  D. Valiente, A. Gil, L. Fernández, O. Reinoso
  Robotics and Autonomous Systems. Vol 62. pp.108-119. 2014.
  ISSN: 0921-8890. Ed. Elsevier.
  JCR-SCI Impact Factor: 1.618, Quartile **Q2**.

The first article focuses on the main proposal made in this thesis. The new map model, consisting of a reduced set of omnidirectional images is presented. The results of this publication confirm the validity of this approach which is established as an EKF view-based model. Chapter 4 comprises the extended explanation and further details of this contribution. Next, the second article presents all the work done on the adaption of the SGD structure to the omnidirectional geometry, but also to the new map representation proposed in this thesis, represented by the recently commented EKF view-based SLAM approach. Efficiency and accuracy comparison experiments with the standard SGD are presented to confirm the benefits provided by this contribution in terms of non-linearity mitigation. Additional details about this implementation are presented in Chapter 5. Then, the third article concentrates on a comparative study between two of the main contributions made under the framework of this thesis: the EKF view-based SLAM model which contains the new map model consisting of a reduced set of views and the modified SGD algorithm which was adapted to the omnidirectional geometry. Both contributions are presented in the two previous articles, and thus lead to the introduction of this third article. The results show that, despite the fact that the SGD is an offline method, it reveals to be a reliable alternative in order to deal with the non-linear effects which are very likely to compromise the EKF convergence. Chapter 4 and Chapter 5 provide further details about all these developments and the corresponding comparison results.

These articles presented above are appended in Appendix 8. The main contributions and conclusions extracted from the articles are related to the structure of this thesis along each specific chapter. Moreover, a brief synthesis is presented as a section in this introductory chapter. In addition, their contents coincide in terms of scope with other of the additional publications which are going to be listed in the next section. Some were produced either as continued or former work within the same lines of research.

Apart from the set of journal articles supporting this thesis, a special mention has to be made of two other two articles which complement the backbone of this thesis. The first presents the latest implementation of a new view initialization mechanism which accounts for information gain and losses in order to bound the uncertainty

of the system. The combination of Gaussian Processes with Information-based techniques demonstrate an improvement in our view-based SLAM approach to deal with the convergence issues. Here, several comparative results are presented to confirm the suitability of the novel approach but also its powerful advantages in contrast to the former approach, initially proposed in this thesis. All the fundamentals of these contributions are detailed in Chapter 6. Finally, the last manuscript presents an approach to visual odometry which benefits from the uncertainty propagation to the matching process. Here, the adoption of the epipolar constraint to the omnidirectional geometry is presented in order to define the motion transformation model, but also to design an enhanced matching process. As a result, this approach provides an improved and reliable feed-forward input for our view-based SLAM system. A set of real data experiments confirm the validity of this contribution and it assesses its performance. Further details are provided in Chapter 2.

- *Information-based view initialization in visual SLAM with a single omnidirectional camera*. [138]
  D. Valiente, M. G. Jadidib, J. Valls, A. Gil, O. Reinoso
  Robotics and Autonomous Systems. Vol 72. 2015.
  ISSN: 0921-8890. Ed. Elsevier.
  JCR-SCI Impact Factor: 1.618, Quartile **Q2**.

- *Visual odometry with a single omnidirectional camera for the view-based SLAM problem*.
  D. Valiente, A. Gil, L. Payá, D. Úbeda, O. Reinoso
  Submitted to **Information Fusion**.
  ISSN: 1566-2535. Ed. Elsevier.
  JCR-SCI Impact Factor: 4.353, Quartile **Q1**.

1.4 Additional Publications

Besides the main publications supporting this thesis, as a result of the research period, a list of additional publications which have been produced under the framework of this thesis is presented:

Journal Publications

- D. Valiente, L. Fernández, A. Gil, O. Reinoso. *Visual odometry through appearance- and feature-based method with omnidirectional images*. Journal of Robotics. Ed. Hindawi Publishing Corporation. ISSN:1687-9619. 2012.

Book Chapter Publications

- D. Valiente, A. Gil, L. Fernández, O. Reinoso. *Visual SLAM Based on Single Omnidirectional Views*. Informatics in Control, Automation and Robotics. Series: Lectures Notes on Electrical Engineering. ISBN:978-3-319-03500-0.

International Conference Publications

- F. Amorós, L. Payá, O. Reinoso, L. Fernández, D. Valiente. *Towards relative altitude estimation in topological navigation tasks using the global appearance of visual information*. VISSAP: International Conference on Computer Vision Theory and Applications. Lisbon, Portugal, 2014.
- D. Valiente, A. Gil, F. Amorós, O. Reinoso. *SLAM of View-based Maps using SGD*. ICINCO 2013 International Conference on Informatics in Control, Automation and Robotics. Reykjavik, Iceland, 2013.
- D. Valiente, A. Gil, L. Fernández and O. Reinoso. *View-based SLAM using Omnidirectional Images*. ICINCO 2012. International Conference on Informatics in Control, Automation and Robotics. Rome, Italy, 2012.
- L. Fernández, L. Payá, D. Valiente, A. Gil, O. Reinoso. *Monte Carlo Localization using the Global Appearance of Omnidirectional Images Algorithm Optimization to Large Indoor Environments*. ICINCO 2012. International Conference on Informatics in Control, Automation and Robotics. Rome, Italy, 2012.
- A. Gil, D. Valiente, O. Reinoso, L. Fernández, J. M. Marín. *Building Visual Maps with a single Omnidirectional Camera*. ICINCO 2011. International Conference on Informatics in Control, Automation and Robotics. Noordwijkerout, Netherlands, 2011.
- A. Gil, D. Úbeda, O. Reinoso, L. Payá, D. Valiente. *Creación de un laboratorio remoto para la docencia del lenguaje C/C++*. Conferencia Internacional de Ingeniería Mecánica y Energía 2010. Santiago de Cuba, Cuba, 2010.

National Conference Publications

- C. Parra, L. M. Jiménez, M. Ballesta, O. Reinoso, D. Valiente. *Localización de robots móviles con 4 gdl mediante visión omnidireccional*. XXXVII Jornadas Automática. Madrid, 2016.
- F. Amorós, L. Payá, D. Valiente, L.M. Jiménez, O. Reinoso. *Estimación de altura en aplicaciones de navegación topológicas mediante apariencia global de informaciŮn visual*. XXXV Jornadas Automática. Valencia, 2014.
- L. Fernández, L. Payá, O.r Reinoso, A. Gil, D. Valiente. *Visual Hybrid SLAM: An Appearance-Based Approach to Loop Closure*. First Iberian Robotics Conference Advances in Robotics. Madrid, 2013.

- D. Valiente, A. Gil, M. Juliá, L. Fernández, O. Reinoso. *Solución al problema de SLAM empleando SGD con imágenes omnidireccionales*. XXXIV Jornadas de Automática. Terrassa, 2013.
- A. Gil, A. Peidró, J. M. Marín, O. Reinoso, D. Valiente, L. Miguel Jiménez, M. Juliá. *Laboratorio Virtual y Remoto de robots paralelos*. XXXIV Jornadas de Automática. Terrassa, 2013.
- M. Juliá, A. Gil, L.M. Jiménez, D. Valiente, O. Reinoso. *Exploración teleoperada de entornos desconocidos mediante un conjunto de robots móviles*. ROBOT 2011, Robótica Experimental. Sevilla, 2011.
- D. Valiente, A. Gil, J. M. Marín, L. Fernández, O. Reinoso. *Construcción de mapas visuales con imágenes omnidireccionales*. XXXII Jornadas de Automática. Sevilla, 2011.
- M. Juliá, L. Payá, D. Valiente, L.M. Jiménez, D. Úbeda, O. Reinoso. *Laboratorio Virtual de exploración de entornos mediante sistemas multirobot*. XXXII Jornadas de Automática. Sevilla, 2011.
- L. Fernández, O. Reinoso, L. Payá, D. Úbeda, D. Valiente. *Creación de Mapas Topológicos Incrementales mediante métodos basados en apariencia global*. XXXI Jornadas de Automática. Jaén, 2010.

1.5 Framework

This thesis has been developed under a framework sustained by different research-related branches, such as grants, projects and collaborations.

1.5.1 Grants

The main support of this thesis has been a FPI grant given by the Ministry of Science and Innovation of the Spanish Government, with reference BES-2011-043482 and duration of 4 years. In addition, two other short-term grants were given by the Valencian Regional Government, the BAF/2011, and the *Santiago Grisolía* 2011 program, both for 3 months.

1.5.2 Research Stays and Collaborations

- A short research stay was supported by the grant given by the Ministry of Science and Innovation of the Spanish Government, with reference EEBB-I-14-08104. This grant made possible a 4 months stay in 2014 at the Centre for Autonomous Systems in the Faculty of the Engineering and IT at the University of Technology Sydney, Australia.
- A short research stay supported by the Teaching Staff Mobility Program given by the Miguel Hernández University. This grant allows to collaborate with Q-bot

Ltd, an Imperial College spin-off settled in London, UK. The research framework is strongly related to the topic of this thesis. The stay will take place in the last trimester of 2016.

1.5.3 Projects

Several projects have included the work of this thesis within their wider scopes and research contents. They were all connected by general objectives on the mapping and localization tasks through visual information acquired by cameras. The following list describes the projects given to the ARVC research group of the Miguel Hernandez University, where this thesis was hosted:

- Project: *Integrated Exploration of Environments by means of Cooperative Robots in order to build 3D Visual and Topological Maps intended for 6 DOF Navigation*.

  Supported by: CICYT Ministry of Science and Innovation

  Duration: 01/01/2011 to 31/12/2013

  Description: While a group of mobile robots carry out a task, they need to find their location within the environment. In consequence a precise map of a general and undetermined environment has to be known by the robots. During the last decade a series of methods have been developed that allow the construction of the map by a mobile robot. These algorithms consider the case in which the vehicle moves along the environment, constructs the map while, simultaneously, computes its location within the map. As a result, this problem has been named Simultaneous Localization and Mapping (SLAM). This research project focusses thus on the construction of visual maps in 3D general unknown environments by using a team of mobile robots equipped with vision sensors. In this sense, we propose to undertake, among others, the following lines: 6 DOF cooperative visual SLAM, in which the robots move following general trajectories in the environment (with 6 degrees of freedom) instead of the classical trajectories in which it is assumed that the robots navigate on a two-dimensional plane; integrated exploration, where the exploration paths of the robots consider to maximize the knowledge of the environment and, at the same time, take into account the uncertainty in the maps created by the robot(s); map alignment and map fusion of local maps created by different robots; and finally, the creation of maps using the information based in the visual appearance that allows the construction of high-level topological maps.

- Project: *Cooperative Mobile Visual Perception Systems as support for tasks performed by means Robot Networks*.

  Supported by: CICYT Ministry of Science and Innovation

  Duration: 1/10/2007 to 30/09/2010

  Description: Performing tasks in a coordinated manner by means of a team of robots is a topic of great interest and allows to improve the results compared

to the single-robot case. The current research project focuses on this particular field and proposes the need to use different vision systems distributed along the mobile agent network that gather a precise and complete description of the environment. To cope with the proposed goals it will be necessary to tackle with different research lines, in consequence, we worked on the following subjects: Cooperative map building and localization using particle filters, Visual landmark modeling: Improving data association in visual SLAM, development of cooperative exploration strategies using the information provided by each robot, cooperative reconstruction of environments using appearance based methods.

- Project: *Robotic Navigation in Dynamic Environments by means of Compact Maps with Global Appearance Visual Information*. Supported by: CICYT Ministry of Science and Innovation

  Duration: 01/09/2014 to 31/08/2017

  Description: Carrying out a task by a team of mobile robots that move across an unknown environment is one of the open research lines with a higher scope for a large development in the mid-term. In order to accomplish this task it has been proved necessary to possess a highly detailed map of the environment that will allow the localization of the robots as they execute a particular task. During the last years the proposer research team has worked with remarkable results in the field of SLAM (Simultaneous Localization and Mapping) with teams of mobile robots. The work has considered the use of robots equipped with cameras and the inclusion of the visual information gathered in order to build map models. So far, different kind of maps have been built, including metric maps based on visual landmarks, as well as topological maps base on global appearance-based information extracted from images. These maps have allowed the navigation of the robots in these maps as well as the performance of high level tasks in the environment. Nonetheless, there exists space for improvement in several areas related to the research carried out so far. Currently, one of the important problems consists in the treatment of the visual information and the updating of this information as the environment changes gradually. In addition, the maps should be created considering the dynamic and static part of the environment (for example when other mobile robots or people move in the environment), thus leading to the creation of more realistic models, as well as strategies to update the maps as changes are detected. A different research line considers the creation of maps that combine simultaneously the information about the topology of the environment, as well as semantic and metric information that will allow a more effective localization of the robot in large environments and, in addition, will enable a hierarchical localization in these maps. The proposed research project considers to tackle the aforementioned lines, thus considering the task of developing dynamic visual maps that will incorporate the semantic and topological structure of the environment, as well as the metric information when the robots perform trajectories with 6 degrees of freedom.

1.6 Structure

This document has been structured as follows:

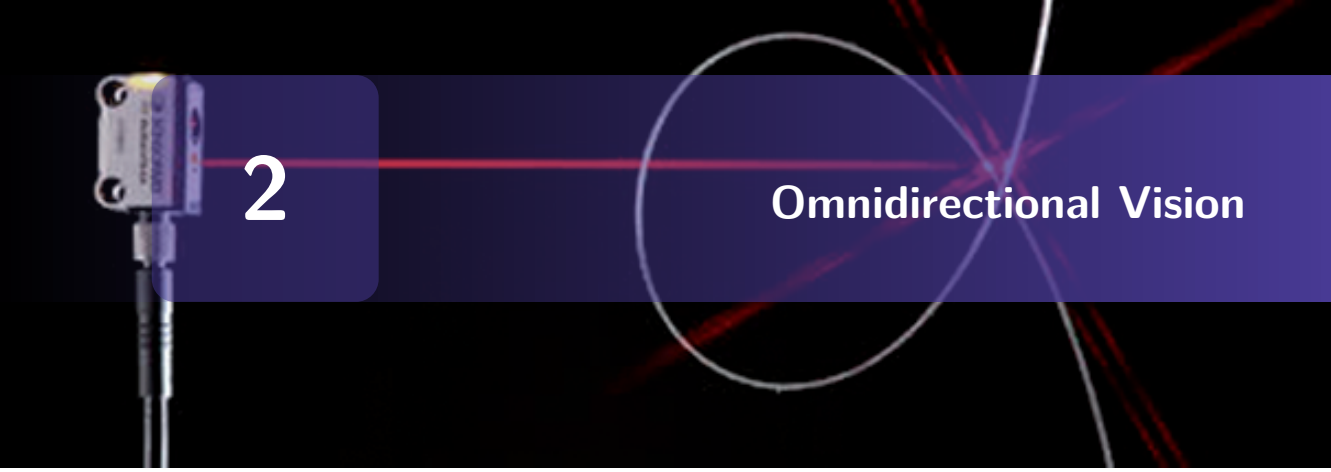
- Chapter 2 provides a general overview to the omnidirectional system and its calibration. The particular setup configuration and specification of the equipments are also presented. In addition, the essential of the epipolar geometry and its adaption to our omnidirectional reference system are described. Consequently, a motion transformation model is devised. This allows to propose a visual odometry approach, which represents a preliminary result, as a feed-forward input for a SLAM application. These results assess the performance and efficiency of the motion transformation proposed.

- Chapter 3 introduces the fundamentals of SLAM problem, as an approach to the theoretical background required in this thesis. Firstly, it concentrates on the Bayesian considerations. Next, it specifies on the several algorithm-specific methods, which are selected in this thesis to develop and implement new contributions to the framework of SLAM. Then an overview to Gaussian Processes (GP) is included, as a necessary tool for inference tasks. Similarly, a brief introduction to Information-based techniques is provided, as it is required for the work developed on the uncertainty reduction of the SLAM system. These two last techniques are combined in order to deal with the uncertainty of the system.

- Chapter 4 contains the development and implementation details of the first major contribution: a new map model based on a reduced set of omnidirectional views. This approach is sustained by an EKF-based method. Its modified structure is presented by the following division: map building process, view initialization, data association and observation model. Next, an enhancement of the matching process is described. Finally, real data results are included in order to validate the suitability and the benefits of this approach to work with real data scenarios.

- Chapter 5 describes the contribution to the robustness of the previous SLAM approach presented. Here a modified SGD algorithm, adapted to omnidirectional geometry is introduced. All the details regarding this implementation are presented. In addition, a comparative set of real data results is included in order to test the validity of the approach, but also to compare its efficiency and accuracy with the standard method, and with the previous contribution presented in this thesis.

- Chapter 6 presents the improvements devised in order to provide with a contribution aiming at the uncertainty reduction of the system. In this sense, a new view initialization model is described. All the basis of this approach is defined, in terms of the GP and the Information-based implementations. Several experimental sets are also presented in order to demonstrate the improvements against non-linear effects which cause uncertainty increases and thus convergence risks.



2 Omnidirectional Vision

In the field of mobile robotics, the incipient growth of visual sensors has led to an impressive increase in the use of cameras as the principal sensor in autonomous vehicles, especially in those aimed at navigation purposes. Catadioptric sensors have been on top of the trend. They consist of a combined system composed by a camera and a mirror which significantly outperform basic cameras due to its great ability to encode high amounts of information of the current scene of view [98]. The classification for this kind of visual systems is normally established by the physical characteristics of the mirror and the procedure to project the rays on the image frame.

The catadioptric sensor employed in the framework of this thesis is an omnidirectional sensor. The major potential of such sensors is that they allow to take profit of their favorable capability to gather scenes with 360° degrees. This is possible thanks to the high field of view that these sensors provide in comparison with planar cameras. Another promising aspect in contrast to traditional cameras is the invariance to orientation changes [83], which is highly valuable in order to avoid undesired obstruction effects on the image. Under the presence of obstructor elements, there is always information collected from the rest of relative angles to the vehicle. In addition to this, an omnidirectional sensor also delivers interesting features such as low battery consumption, good resolution, lightness, high acquisition rate, and low price in comparison with other sort of extensively used sensors such as laser [22] or sonar [17].

The approach here presented is cataloged as a catadioptric visual system since it builds an omnidirectional image from the reflection on an hyperbolic mirror. The process by which the image is generated may be observed in Figure 2.1. The rays coming from the acquired scene are reflected on the surface of the mirror and then

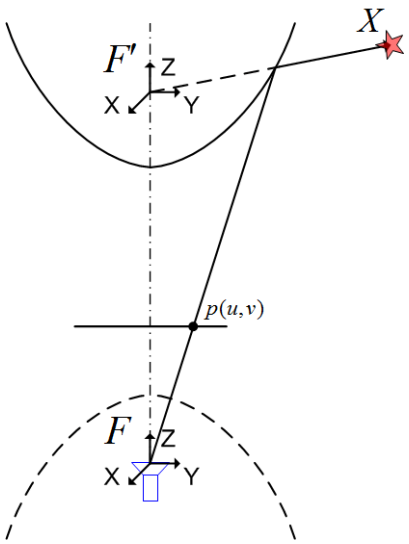


Figure 2.1: Generation of an image point $p(u, v)$ in pixels from its corresponding 3D point X. The center of projection of the camera coincides with the focus of the hyperboloid.

projected towards the image plane. Note that the center of projection of the camera coincides with the focus of the hyperboloid. Theoretically, this coincidence is the basis for the camera to focalize the 3D scene that is finally digitalized. This procedure was carried out for the first time in 1970, [113]. Other authors proposed conical mirrors [149], spherical [62], parabolic [99], eliptical [107] and hyperbolic [14]. In [5] diverse catadioptric systems with only one center of projection were presented, where the incident rays projected all onto that central point. Some others [10, 150, 50, 73] experienced with different catadioptric systems and published a wide variety of results. Notice that there is an evident necessity to define a new projection model, as for these cases, the pinhole model does no longer apply, since the non-linear transformation introduced by the geometry of the mirror has to be taken into consideration. This last consideration suggests that we present our catadioptric system and also that we deal with a series of definitions about its physics, its main characteristics and features, but especially its non-linear nature for the projection of a 3D point on the image plane. Consequently, in this chapter we proceed according to the following structure:

- We provide a general overview about the projection model of the omnidirectional systems and their calibration.
- Then we present the setup specifications, the obtained calibration and general information about the real omnidirectional equipment we used in this work.
- Next we introduce the essentials of the epipolar geometry as the fundamental tool to obtain a motion transformation model which only requires visual information as input data.

- In consequence with the previous point, we propose an adaption of the planar epipolar geometry to the non-linear model stated by our omnidirectional sensor. It is worth noting that epipolar geometry has been extensively studied in planar cameras by using the standard perspective model, but there is a wide field of research yet to be done in terms of omnidirectional models. We provide further detail about this contribution that allows us to define a motion transformation model which is crucial for the latter design of a robust observation model within our problem of SLAM.
- Finally we present some real results that assess the performance and efficiency of our motion transformation model. We exploit the implementation of a visual odometry approach to such purpose.

2.1 Catadioptric Projection

According to [60] any standard perspective camera model can be used to project points from a 3D general reference system named X, to an associated image reference system, seen as x on an image plane. The following equation reflects the process by which an optical ray draws an arrow through the optical center of the camera to an image point x.

$$\lambda x = PX \tag{2.1}$$

being $X = [X, Y, Z]$ and $x = [x, y, 1]$ the normalized image coordinates respectively, where $P \in \mathbb{R}^{3x4}$ is the projection matrix, that can be expressed as $P = [R|T]$, with R a rotation matrix $\in \mathbb{R}^{3x3}$ and $T \in \mathbb{R}^3$ expresses the translation between the camera reference system and the world reference system.

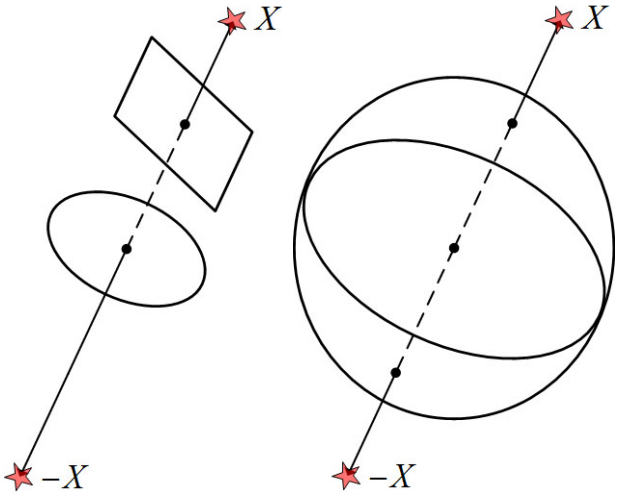
With the structure introduced in (2.1), scene points are always projected into their corresponding image points regardless they are in front or behind the camera. That is, only a half-space can be projected on the image.

On the contrary, an omnidirectional camera projects unequivocally points in front of and behind the camera to their proper points on the image. As a result, an omnidirectional camera can work with half-lines to represent image points in a spherical model. This behaviour may be noticed in Figure 2.2, where the two different central camera models are represented. The first corresponds to a standard perspective model and the second to a spherical model. The last one is commonly used to simplify the non-linear projection from 3D to 2D of catadioptric models such as the omnidirectional. Therefore, in the omnidirectional model, contrarily to the perspective model, an image point is the representation of all the scene points which coincide on a half-line that emerges from the camera center. According to this, equation (2.1) needs reformulation:

$$\lambda q = PX, \ \lambda > 0, \tag{2.2}$$

where $q = [x, y, z]$ encodes the image point, being $\|q\| = 1$.

In the same line, two new assumptions may be taken into account so as to deal with omnidirectional cameras:



(a) (b)

Figure 2.2: Representation of two central camera models. Figure 2.2(a) corresponds to a central camera model for the standard planar perspective model, which is not able to distinguish opposite points. Figure 2.2(b) corresponds to the spherical central camera model, valid for omnidirectional models, where opposite points are distinguishable by half-lines.

- The axis of the mirror supposes a symmetric rotation axis.
- This axis is orthogonal to the image plane.

Focusing on central omnidirectional cameras by using the general model (2.2), an observed scene point X directs the vector $p = (x''^T, z'')$ in the same direction as q, which is projected to u'' on the image plane, being collinear to x''. This development corresponds to the new projection formulation and it can be observed in Figure 2.3, where the projection procedure is depicted. Note that the sphere in the center of the hyperbolic mirror represents the spherical model which can be used in order to unify the notation of the projection vectors. Such spherical model permits to state a standard generalization for any catadioptric case, regardless the characteristics of the mirror and its non-linear function. Therefore, in order to move forward, it is required that the specific non-linearities associated with each mirror are considered. To that end, two new function expressions have to be defined, namely h and g, which are associated with the the non-linearities of each catadioptric system:

$$p'' = \begin{bmatrix} h(||u'||)u'' \\ g(||u'||) \end{bmatrix} \tag{2.3}$$

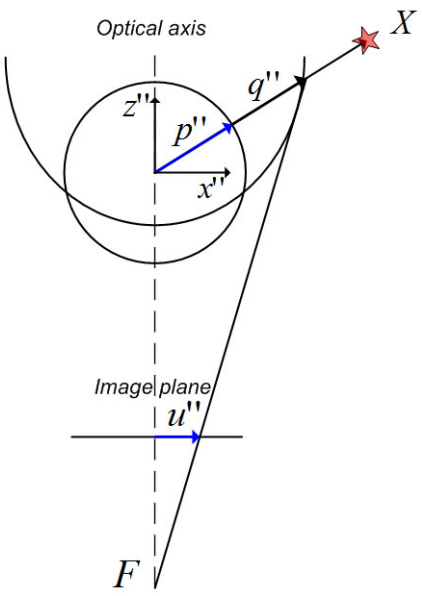


Figure 2.3: Mapping of a scene point X to the point u'' on the sensor plane, seen on the XZ plane.

2.1.1 Omnidirectional Calibration

Now it is worth introducing the calibration representation. This concept entails an improved terminology that unifies the expected information so as to model the projection behaviour of dioptric and catadioptric cameras. We focus on the last updated general expression (2.3) to reach the calibration of an omnidirectional system.

The unification by means of the term g/h relaxes the constraints and allows to force h=1, thus g verifies:

$$\lambda p'' = \lambda \begin{bmatrix} u'' \\ g(||u'||) \end{bmatrix} = PX \tag{2.4}$$

A generalized Taylor expansion can be performed on g so as to balance the misalignment effects that occur in the focus point of the mirror and in the camera optical center. This new general parametric form is feasible for any kind of camera sensor and it presents the following structure:

$$g(||u''||) = a_0 + a_1||u''|| + a_2||u''||^2 + \ldots + a_n||u''||^n \tag{2.5}$$

where $[a_0, \ldots, a_n]$ are the coefficients of a n-degree polynomial. It is important to mention that these coefficients act as the calibration of the model, as they encode the necessary information regarding the projection procedure. In addition to this, the previous expression can be further simplified if the following assumption for hyperbolic,

parabolic and elliptical mirrors is applied:

$$\frac{\partial g}{\partial \rho}\Big|_{\rho=0} \tag{2.6}$$

being ρ the module of the resulting vector in the image plane as ρ=$||u''||$. Then a_1=0 in equation (2.5) reduces the expression to:

$$g(||u''||) = a_0 + a_2||u''||^2 + \ldots + a_n||u''||^n \tag{2.7}$$

As a result, if we merge (2.7) with (2.4), the final expression for central omnidirectional cameras can be posed as:

$$\lambda p'' = \lambda \begin{bmatrix} u'' \\ a_0 + a_2||u''||^2 + \ldots + a_n||u''||^n \end{bmatrix} = PX \tag{2.8}$$

In order to obtain the final point projected on the digital image plane, the visual elements shown in Figure 2.3 have to be determined by proceeding in the following order:

- On the central projection of the scene, tracing the ray from point X to p'' .
- Applying the non-perspective mirror reflection in terms of the specific h and g functions that generates u'' from p'' for each particular mirror.
- Digitizing the transformation from u'' on the sensor plane to u' on the digital image plane.

2.2 Equipment

2.2.1 Omnidirectional System

Here we intend to present the real equipment that has been employed for the acquisition of images, and ultimately for the final purpose of SLAM in this thesis. The catadioptric system consists of a CCD camera, shown in Figure 2.4, with an hyperbolic mirror jointed by a specific assembly kit, as observed in Figure 2.5.

We have two different cameras, as well as two different mirrors. This fact allows us to defined several configurations for testing purposes. The cameras are manufactured by *Imaging Source*, with models: DMK-21BF04 [48] and DMK-41BF02 [47]. These models basically differ in the resolution. As for the mirrors, we mounted a Wide 70 [34], provided by *Eizho*, and a Super-Wide [1], provided by *Accowle*. They deliver high lateral angles of view which are especially useful to gather wider parts of the scene. The Wide 70 also provides a relevant feature that allows to vary the physical distance from the mirror to the center of projection. Figure 2.5 shows these two mirrors and Figure 2.6 two examples of images generated by them. In Table 2.1 we provide a series of specifications for these two cameras used in this work, and likewise in Table 2.2 for the mirrors.

Figure 2.4: Camera CCD FireWire DMK21BF04.

(a) (b)

Figure 2.5: Hyperbolic mirrors used in this work, assembled in their coupling systems. Figure 2.5(a) presents the Wide 70 manufactured by *Eizho*. Figure 2.5(b) presents the Super-Wide manufactured by *Accowle*.

2.2.2 Calibration

Now we can specifically concentrate on the omnidirectional model employed in the framework of this thesis. The ultimate calibration is obtained by means of an omnidirectional calibration software [118] that provides the required library components to estimate the coefficients $[a_0, \ldots, a_n]$ above presented.

This library carries out a last simplification to the expression described in (2.8). Now we assume that p expresses a pixel point on an image, being its coordinates (u, v) as it is drawn in Figure 2.7. This figure models the projection process of our omnidirec-

(a)

(b)

Figure 2.6: Example of two omnidirectional images captured with the hyperbolic mirrors: Wide 70 in Figure 2.6(a), and Super-Wide in Figure 2.6(b).

Camera specifications		
Model	DFK-21BF04	DFK-41BF02
Video format	UYVY/BY8	UYVY/BY8
Resolution	640x480	1280x960
Frame rate	60-3.75 fps	15-3.75 fps
Sensitivity	0.1 lx	0.15 lx
Dynamic range	8 bit	8 bit
CCD	Sony ICX098BQ	Sony ICX205AK
Pixel size	5.6x5.6 μm	4.65x4.65 μm
Connection interface	FireWire	FireWire
Power supply	8-30 VDC	8-30 VDC
Current consumption	200μA@ 12 VDC	200μA@ 12 VDC
Dimensions	50.6x50.6x56 mm	50.6x50.6x56 mm
Weight	265 g	265 g
Gain	0-36 dB	0-36 dB
Saturation	0-200%	0-200%
Shutter	1/10000 to 30 s	1/10000 to 30 s
Offset	0-511	0-511
White balance	-2 to +6 dB	-2 to +6 dB

Table 2.1: Camera specifications

Mirror specifications		
Parameters	*Eizoh* Wide 70	*Accowle* Super-Wide
Geometry	Hyperbolic	Hyperbolic
Diameter	70 mm	76 mm
Height	35 mm	43 mm
Upside angle of view	60°	55°
Downside angle of view	60°	65°
Mirror-camera distance	Variable - Optimum 165mm	Fixed - 122 mm
Weight	175 g	-

Table 2.2: Mirror specifications

tional system. It can be observed a 3D point $X(x, y, z)$ and its projection on the image $p = (u, v)$ in pixel coordinates with respect to the center of the omnidirectional image. P represents the 3D vector from X to the effective viewpoint in the mirror. Considering that the camera and mirror axes are aligned, then axis X and Y are proportional to u and v:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \alpha \begin{bmatrix} u \\ v \end{bmatrix} \tag{2.9}$$

It is worth noticing that the misalignment of these axes is one of the main causes why the image is out of focus since the camera center is not aligned with the focus. This issue is very likely to appear in almost any real application. Firstly because the camera image plane is never perpendicular to the sensor axis, and secondly because it

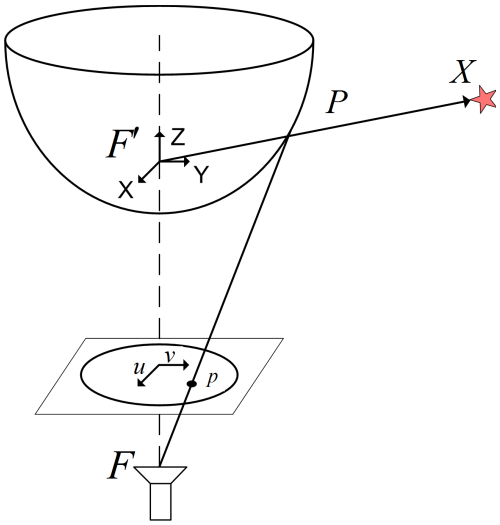


Figure 2.7: Projection model of an omnidirectional camera with an hyperbolic mirror.

is impractical to maintain the alignment where there always exists mechanical effects such as external vibrations. For all these reasons it is necessary to take certain actions in order to mitigate their harmful effects. Therefore we can state a function that may be estimated by means of calibration, and which transfers the detected point on the image frame as $p = (u, v)$ into its corresponding 3D vector P in the following terms:

$$P = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} u \\ v \\ f(u,v) \end{bmatrix} = \begin{bmatrix} u \\ v \\ f(\rho) \end{bmatrix} \tag{2.10}$$

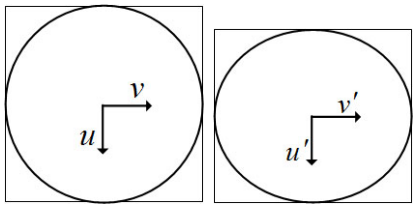
being $\rho = \sqrt{u^2 + v^2}$. Please note that P is a vector pointing to the direction of X, not a 3D point. This is due to the fact that we are dealing with only one image frame, and thus no information of any baseline is available, neither depth or scale values for 3D points. Furthermore, we can express f in terms of ρ, since the mirror is rotationally symmetric. That fact makes f only dependent on the distance of a point from the image center.

Finally the calibration has to seek the coefficients to estimate f in terms of the polynomial previously presented. In consequence, and according to (2.7):

$$f(\rho) = a_0 + a_1\rho + a_2\rho^2 + \ldots + a_n\rho^n \tag{2.11}$$

Moreover, here we pursue the refinement of the estimated calibration since an overall deviation is very likely to appear due to the sum of different sources of errors:

- Axes misalignment



(a) (b)

Figure 2.8: Misalignment effects caused on the image plane. Figure 2.8(a) represents the ideal case whereas Figure 2.8(b) represents the realistic case where there exists misalignment.

- Distortion
- Digitizing process: pixels are not perfectly squared

Figure 2.8 represents the effects of misalignment that have been commented above. Nevertheless, we can establish an affine transformation in order to estimate the relation between the distorted coordinates (u', v') and the ideal ones that do not present distortion, (u, v).

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{2.12}$$

2.2.2.1 Calibration Parameters

Once we have introduced the specific procedure to obtain the calibration of our omnidirectional camera, we can present the resulting calibration parameters.

First of all we show in Figure 2.9 the images captured, where the calibration pattern may be observed as a chessboard. Then we applied a corner detector to gather reliable points to input in the library of the calibration toolbox. The most relevant information provided by the calibration toolbox for our purpose is depicted by Figure 2.10. Figure 2.10(a) draws a representation of the reprojected calibration images. For each image, the reprojection of each calibration pattern may be observed. Figure 2.10(b) plots the function $f(\rho)$ retrieved by the calibration toolbox, which allows us to project/backproject points reciprocally between 2D and 3D spaces. Again, it is important to clarify that the backprojection only enables the retrieving of a vector that indicates the direction of a 3D point. This is justified by the lack of knowledge of the scale factor, since only one image is acquired at each step.

Table 2.3 presents the results obtained in the calibration of both mirrors. We present the expressions of the estimated polynomial $f(\rho)$, the associated error in the reprojected pixels, the camera center (x_c, y_c) and the affine transformation, $A = \begin{bmatrix} c & d \\ e & 1 \end{bmatrix}$, by which the misalignments effects can be mitigated as described in (2.12).

Calibration parameters	
Wide 70	
Projection func.	$f(\rho)$=-185+2.93 $\times 10^{-3}\rho^2 - 9.15 \times 10^{-6}\rho^3 + 1.84 \times 10^{-8}\rho^4$
Affine matrix	$A = \begin{bmatrix} 9.995 \times 10^{-1} & 2.7701 \times 10^{-4} \\ 4.1806 \times 10^{-4} & 1 \end{bmatrix}$
Error	0.658 pixels
Camera center	$(x_c, y_c) = (3.595\,781 \times 10^2, 3.591\,248 \times 10^2)$
Super-Wide	
Projection func.	$f(\rho)$=55.46+3.78 $\times 10^{-3}\rho^2 - 1.96 \times 10^{-6}\rho^3 - 1.12 \times 10^{-8}\rho^4$
Affine matrix	$A = \begin{bmatrix} 9.993 \times 10^{-1} & 8.7241 \times 10^{-4} \\ 7.5162 \times 10^{-4} & 1 \end{bmatrix}$
Error	0.453 pixels
Camera center	$(x_c, y_c) = (3.595\,781 \times 10^2, 3.591\,248 \times 10^2)$

Table 2.3: Calibration parameters for the Wide 70 and Super-Wide mirrors.

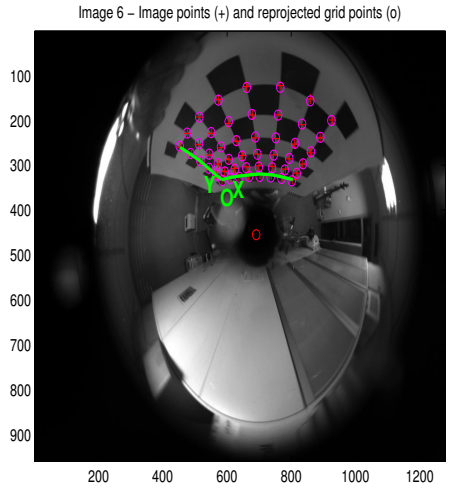
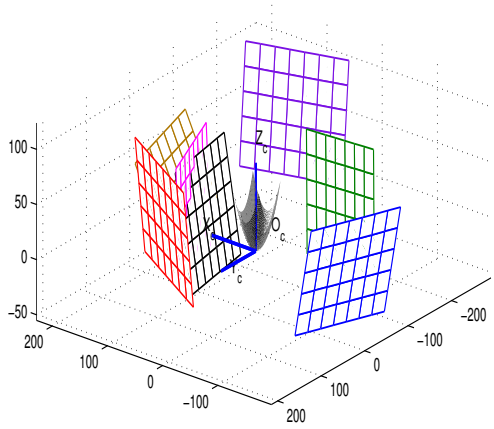


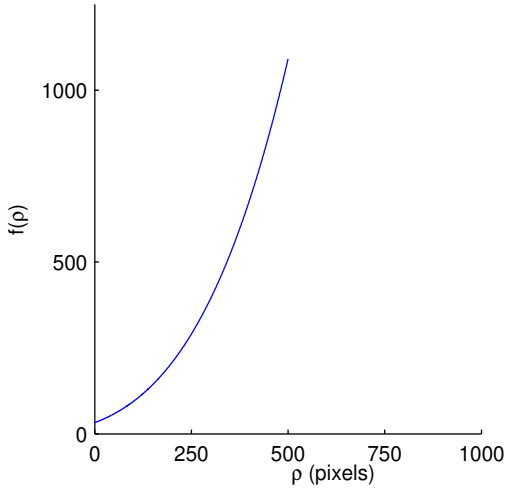
Figure 2.9: Chessboard pattern with corner points indicated. These corner points are the input for the calibration toolbox which returns the projection function f that characterizes our omnidirectional sensor.

2.2.2.2 Panoramic Conversion

The panoramic view is one of the most used models to present the information gathered by an omnidirectional camera system, as in [84] and [21]. This transformation provides a more natural view, since it is easier for the human eye to relate it to the planar view provided by a pinhole camera. Figure 2.11 presents the projection established by this type of view, where the information is projected onto a cylindrical surface. As a result, each panoramic projection depends on the specific hyperbolic mirror associated with



(a)



(b)

Figure 2.10: Figure 2.10(a) shows the reprojected chessboard patterns from which the corner points were detected for the calibration of the Wide 70 mirror. Figure 2.10(b) shows the estimated $f(\rho)$ obtained with the calibration toolbox for the same mirror. ρ is measured as the distance in pixels from the center of the omnidirectional image

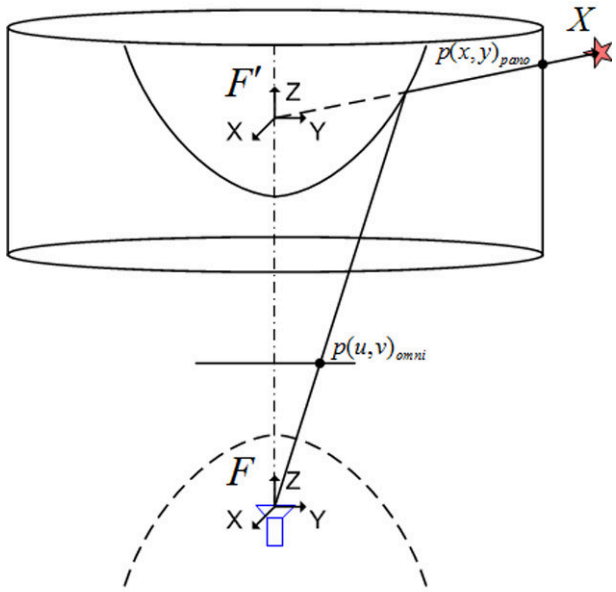


Figure 2.11: Projection model of the panoramic view. Point $p(u,v)_{omni}$ converts into $p(x,y)_{pano}$.

the omnidirectional system.

The procedure to obtain the panoramic view consists of converting the coordinates from the polar reference system (omnidirectional) to the cartesian reference system (panoramic). Then, a circular line in pixels on the omnidirectional image corresponds to an horizontal line on the panoramic view, and similarly a radial line corresponds to a vertical line. Figure 2.13 shows an example of this conversion from omnidirectional to panoramic. The projection of the hyperbolic mirror is non-linear, thus it projects differently the amount of visual information on different areas on the pixels of the image. Figure 2.12 presents two converted images with the two mirror available in the framework of this thesis, as introduced in Figure 2.5 and Figure 2.6. It can be noticed that Figure 2.12(a) induces a vertical distortion and it concentrates more information on the low areas (corresponding to low radius on the omnidirectional mirror). In Figure 2.12(b) this distortion is not appreciable. This is the main reason why our experiments are mainly conducted with the $Eizho$ Wide 70 mirror.

Finally, it is worth mentioning that the feature point extraction provides better results when the panoramic view is used in order to smooth the non-linear nature of the hyperbolic mirror. Several feature point detectors have been tested, and the best solution is provided by SURF, as suggested in [42]. This detector takes the most of the panoramic projection in order to compute multi-scaled areas to detect feature points. Despite this fact, the total number of points detected is reduced in

(a)

(b)

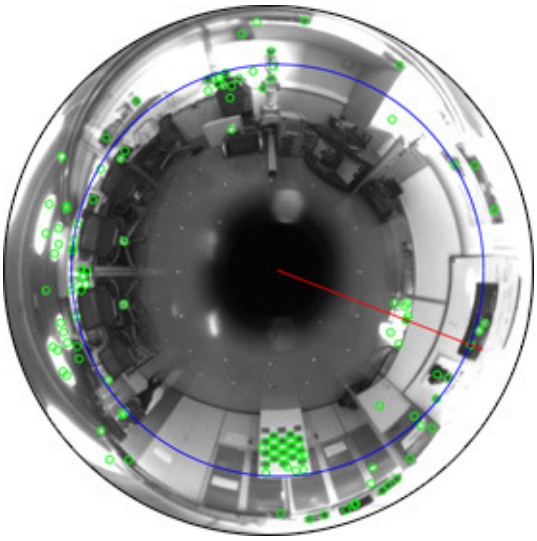
Figure 2.12: Panoramic images converted from the omnidirectional reference system. Figure 2.12(a) shows the image acquired with the *Eizoh* Wide 70 while Figure 2.12(b) shows the image acquired with the *Accowle* Super-Wide mirror.

the panoramic view. As a consequence, we force an expansion of the vertical axis, with a corresponding increase in the size of the image, so that more feature points are detected on the same image. Again, an interpolation is required. In our case, we used a *bicubic* resizing. Figure 2.13 shows an example of this situation, where the points detected on the panoramic image are back-converted to the omnidirectional reference system. Note that the vertical expansion applied is$\times 2$. In this example, the high and low radius of the omnidirectional image have been discarded as they represent areas where the mirror presents pronounced non-linear curvature but also there is irrelevant visual information.

2.2.3 Robot Pioneer P3-AT

The vehicle used in this work is a pre-assembled robot extensively known in this field of research: the Pioneer P3-AT [2], manufactured by *Mobile Robots*. It is a small four-wheel, four motor skid-steer robot, which we have boarded with our omnidirectional camera system, a laser sensor, LMS-200 [115], provided by the company *SICK*, an integrated sonar sensor and a PC. Figure 2.14 shows the robot used in this work with the mentioned equipment. The communication interface with the robot consists of a serial I/O bus to the hardware of the microcontroller, and an API framework to interact with the software of the robot through the PC.

In order to ease the communication process, we have also used the set of APIs libraries and SDKs provided at the open source project, ROS [125], in particular the set constituted by ROSARIA [126], which is a bridge between ROS and the API of



(a)

(b)

Figure 2.13: Conversion from omnidirectional to panoramic view. Feature points are detected on the panoramic, in Figure 2.13(b), and back-converted to the omnidirectional, in Figure 2.13(a).

the P3-AT, ARIA [114], provided by the manufacturer *Mobile Robots* and also open source. We establish access to the data sensors through ROS, so that we can define custom algorithms to compute a reliable ground truth. For that purpose we rely on the laser data to act as a reliable input for a gmapping algorithm [127, 51], that ultimately returns a ground truth estimation.

For further details about specifications of the Pioneer P3-AT, see the Table 2.4. Likewise, Table 2.5 provides more details about the specifications of the laser $SICK$ LMS-200.

Another factor to consider is the development of a model for the parametrization of the internal odometer of the robot. Each motor contains an encoder with a resolution of 100 ticks per revolution, which should provide enough accuracy for general

Figure 2.14: Robot Pioneer P3-AT used in this work for the acquisition of omnidirectional images, raw laser data and odometry data.

Robot specifications	
Physics	
Weight	12 kg
Dimensions	508x497x277 mm
Payload	Tile: 12 kg; Grass: 10 kg; Asphalt: 5 kg
Body	1.6 mm aluminum
Tires	Reinforced Pneumatic
Skid Steering Drive	
Turn Radius	0 cm
Swing Radius	34 cm
Max. Speed	0.7 m/s
Rotation Speed	140°/s
Max. Traversable Step	10 cm
Max. Traversable Gap	15 cm
Max. Traversable Grade	35%
Traversable Terrain	Asphalt, flooring, sand and dirt.
Power	
Run Time	2-4 hours (3 batteries without accessories)
Charge Time	12 hours
Available Power Supplies	5 V - 1.5 A and 12 V - 2.5 A
Batteries	
Number	Up to 3 at a time
Capacity	7.2 Ah (each)
Composition	Chemistry; lead acid

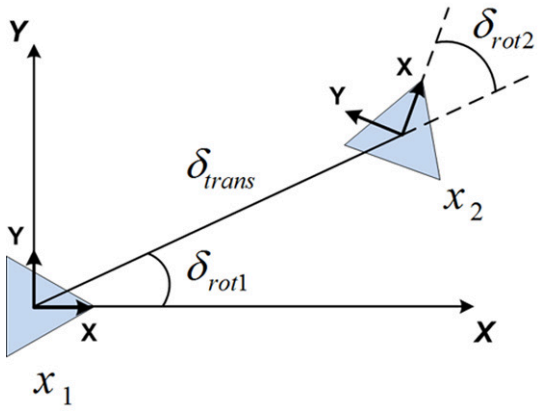
Table 2.4: Pioneer P3-AT specifications

Laser specifications	
Physics	
Weight	4.5 kg
Dimensions	156 x 155 x 210 mm
Features	
Field of application	Indoor
Version	Short Range
Light source	Infrared (905 nm)
Aperture angle	180°
Scanning frequency	75 Hz
Angular resolutions	0.25°; 0.5°; 1°
Operating range	0 - 80 m
Max. range with 10 % reflectivity	10 m
Performance	
Response time	13-53 ms
Resolution	10 mm
Systematic error	+/- 15 mm
Statistical error (1 sigma)	5 mm
Ambient operating temperature	0°- +50°C
Scanning range	80 m
Interface Data interface	RS-232; RS-422
Data transmission rate	9.6 / 19.2 / 38.4 / 500 kBps
Switching outputs	3 x PNP
Supply voltage	24 V DC +/- 15%
Power consumption	20 W

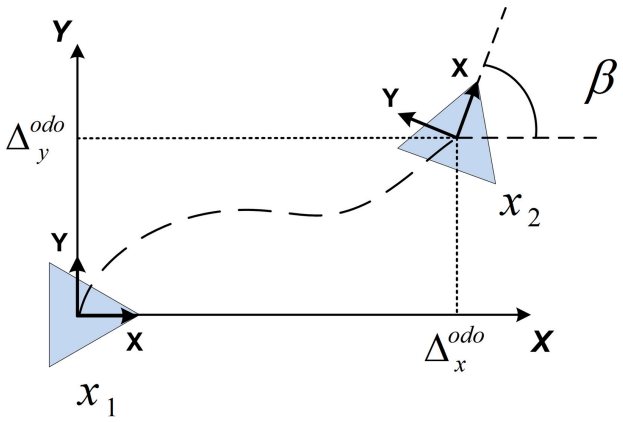
Table 2.5: LMS200 specifications

purposes. However, most of the experiments conducted in this work have been carried out at indoor scenarios where the wheels of the robot are prone to suffer steering, and consequently they tend to corrupt the measures with noise. This fact makes us define a parametrization for the odometry, so as to obtain a reliable model for its behaviour, that also permits to tune external noise parameters on this data. This will help out with the definition of worse case scenarios in terms of noise, that can be used to test the robustness of the contributions presented in this work.

There are several possible parametrizations to tackle with this point. The most widely known approaches are based on probabilistic motion models, sustained by some given proposal distributions [132, 35], and incrementally computed. Figure 2.15 depicts the corresponding diagrams for two odometry models. We have equally used both models in the framework of this work with satisfactory outputs for our goals. The analytical expressions to define the equations for these models look as follows:



(a)



(b)

Figure 2.15: Figure 2.15(a) represents the diagram for the Odometry model 1. Figure 2.15(b) represents the diagram for the Odometry model 2.

Odometry model 1

The equations that relate the prior pose (x_1, y_1, θ_1) and the new pose (x_2, y_2, θ_2) represent the incremental change as:

$$\begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} \cos(\hat{\delta}_{rot1}) & 0 & 0 \\ \sin(\hat{\delta}_{rot1}) & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \hat{\delta}_{trans} \\ \hat{\delta}_{rot1} \\ \hat{\delta}_{rot2} \end{bmatrix}$$

where $\hat{\delta}_{trans}$, $\hat{\delta}_{rot1}$ and $\hat{\delta}_{rot2}$ are the result of adding a gaussian, zero-mean random noise to the odometry readings as:

$$\hat{\delta}_{trans} = \delta_{trans} + \epsilon_{trans} \rightarrow \epsilon_{trans} \sim \mathcal{N}(0, \sigma^2_{trans}) \quad (2.13)$$

$$\hat{\delta}_{rot1} = \delta_{rot1} + \epsilon_{rot1} \rightarrow \epsilon_{rot1} \sim \mathcal{N}(0, \sigma^2_{rot1}) \quad (2.14)$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \epsilon_{rot2} \rightarrow \epsilon_{rot2} \sim \mathcal{N}(0, \sigma^2_{rot2}) \quad (2.15)$$

The rest of the parametrization is determined by the following approximated terms for the standard deviations required above:

$$\sigma_{rot1} = \alpha_1 |\delta_{rot1}| + \alpha_2 \delta_{trans} \quad (2.16)$$

$$\sigma_{trans} = \alpha_3 \delta_{trans} + \alpha_4 (|\delta_{rot1}| + |\delta_{rot2}|) \quad (2.17)$$

$$\sigma_{rot2} = \alpha_1 |\delta_{rot2}| + \alpha_2 \delta_{trans} \quad (2.18)$$

As an example of implementation, the following algorithm, specified as Algorithm 1, shows how a new pose, namely (x_2, y_2, θ_2), can be recalculated from a previous one, (x_1, y_1, θ_1), by overweighting with the noise parameters associated with the odometry, as described above $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$.

Algorithm 1 Odometry model 1 algorithm

function out = **OdoParam**$(pose_1, pose_2)$

Require: Inputs: two consecutive odometry readings and noise parameters
$(pose_1, pose_2)$=$\{(x_1, y_1, \theta_1), (x_2, y_2\theta_2)\}$
$param$=$(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$

1: $\delta_{rot1} = atan2(y_2 - y_1, x_2 - x_1) - \theta_1$
2: $\delta_{trans} = \sqrt{((x_2 - x_1)^2 + (y_2 - y_1)^2)}$
3: $\delta_{rot2} = \theta_2 - \theta_1 - \delta_{rot1}$
4: $\hat{\delta}_{rot1} = \delta_{rot1} - N(0, \alpha_1 \cdot \delta_{rot1} + \alpha_2 \cdot \delta_{trans})$
5: $\hat{\delta}_{trans} = \delta_{trans} - N(0, \alpha_3 \cdot \delta_{trans} + \alpha_4 \cdot (\delta_{rot1} + \delta_{rot2}))$
6: $\hat{\delta}_{rot2} = \delta_{rot2} - N(0, \alpha_1 \cdot \delta_{rot2} + \alpha_2 \cdot \delta_{trans})$
7: $x_2 = x_1 + \hat{\delta}_{trans} \cdot cos(\theta + \hat{\delta}_{rot1})$
8: $y_2 = y_1 + \hat{\delta}_{trans} \cdot sin(\theta + \hat{\delta}_{rot1})$
9: $\theta_2 = \theta_1 + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$
10: **return** (x_2, y_2, θ_2)

Odometry model 2

In the same manner, the following equations relate the prior and the new pose by means of an incremental change:

$$\begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} + \begin{bmatrix} \cos(\theta + \frac{\Delta_\phi^{odo}}{2}) & -\sin(\theta + \frac{\Delta_\theta^{odo}}{2}) & 0 \\ \sin(\theta + \frac{\Delta_\theta^{odo}}{2}) & \cos(\theta + \frac{\Delta_\theta^{odo}}{2}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta_x^{odo} \\ \Delta_y^{odo} \\ \Delta_\theta^{odo} \end{bmatrix}$$

For the covariance, we estimate the variances of the three variables of the odometry increment. We model them as independent, with zero-mean gaussian errors. These errors are composed by terms which are introduced by imperfections of the odometer, and also by the resolution of the encoders and by the potential drift effects.

Now we denote σ as the diagonal matrix that contains the three variances of the odometry, described as:

$$\sigma_{\Delta_x^{odo}} = \sigma_{\Delta_y^{odo}} = \sigma_{xy}^{min} + \alpha_1 \sqrt{(\Delta_x^{odo})^2 + (\Delta_y^{odo})^2} + \alpha_2 |\Delta_\phi^{odo}| \tag{2.19}$$

$$\sigma_{\Delta_\phi^{odo}} = \sigma_\phi^{min} + \alpha_3 \sqrt{(\Delta_x^{odo})^2 + (\Delta_y^{odo})^2} + \alpha_4 |\Delta_\phi^{odo}| \tag{2.20}$$

Therefore, the parametrization can be determined by the following terms:

$$\alpha_1 \quad (meters/meter) \tag{2.21}$$

$$\alpha_2 \quad (meters/degree) \tag{2.22}$$

$$\alpha_3 \quad (degrees/meter) \tag{2.23}$$

$$\alpha_4 \quad (degrees/degree) \tag{2.24}$$

$$\sigma_{xy}^{min} \quad (meters) \tag{2.25}$$

$$\sigma_\phi^{min} \quad (degrees) \tag{2.26}$$

2.3 Epipolar Geometry

In the framework of this thesis, the epipolar geometry represents a fundamental tool for determining motion transformations in the camera reference system. Particularly, we propose the extrapolation of the planar epipolar constraint to the omnidirectional camera reference system. We seek to describe a relation for the motion transformation between omnidirectional images, which can also be referred as the motion transformation between poses of the robot. The epipolar geometry is a tool of paramount importance to ultimately define a robust observation model that allows to perform efficient matching procedures in order to enhance our SLAM approach. As a result, we present our contribution to adapt the planar epipolar constraint to the omnidirectional case.

For planar camera reference systems, the epipolar geometry is an intrinsic projective geometry between two views which only depends on the camera calibration and

the relative poses. This intrinsic geometry is encapsulated by the fundamental matrix $F \in \mathbb{R}^{3x2}$ with $rank$=2. Considering a certain point in the 3D space X, which generates an image point x in a first view and x' in a second view, then F reflects the epipolar constraint to satisfy in the form:

$$x'^T F x = 0 \tag{2.27}$$

The fundamentals of the epipolar geometry lays on the capability to associate two views by a mutual epipolar plane and the intersection generated by this plane with the two image planes. Explicitly, for a X point in 3D space there exists an epipolar plane that is determined by the baseline axis (line joining the camera centers) and the two rays that project the point X into the two respective cameras. It is worth noting that in order to represent the entire 3D space, a beam of planes has to be considered around the baseline axis. Traditionally, this concept has been extensively exploited for matching purposes in stereo applications [120] since it facilitates the search of points in the second view.

Figure 2.16 comprises an example of epipolar geometry applied to a planar system. A point X in 3D has its projection on two image planes at x and x' respectively. The resulting epipolar plane, π, reveals the existing coplanarity between x, x', X, and the camera centers C and C'. This characteristic brings an important tool for searching a correspondence. When there is not any knowledge about x', the plane π constraints it to lie on the line of intersection l' that results from the intersection of π with the second image plane. As a result, the search for corresponding points can be beneficially restricted to a search over a line.

For a further comprehension of this geometry, it is required to complete the description of this terminology:

- Epipole, denoted as e, is the intersection of the baseline between the camera centers with the image plane. Analogously, it can be seen as the image point of the other camera center. That is, the relation between e and C'.
- Epipolar plane, denoted as π, is the plane containing the baseline. There exists a beam of epipolar planes that is parametrized by a linear factor which rotates around the baseline.
- Epipolar line, denoted as l, is the resulting line of intersection between a π plane with the image plane. It intersects with the baseline at the epipole. This epipolar line eventually represents the line where correspondences have to be found.

This last description is comprised by the general expression of the epipolar constraint in (2.27). Now we can move forward to introduce the essential matrix E [85], that is aimed at the specialization to the case of normalized image coordinates. The key point to establish a differentiation between F and E lays on the prior knowledge of a calibration for the cameras. The essential matrix is less conditioned since it has less degrees of freedom, but it needs a known calibration matrix K.

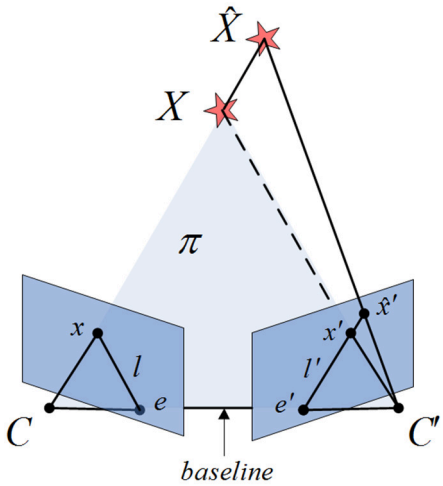


Figure 2.16: Epipolar geometry applied to the standard planar camera system.

The main consideration is that the camera matrix can be specified by $P = K[R|t]$, with R a rotation and t a translation. Hence the image point is $x = PX$. Since K is known, we can obtain the image point expressed in normalized coordinates as $\hat{x} = K^{-1}x$, that is $x = [R|t]X$. If we assume that the last expression corresponds to a camera matrix in the form $[R|t]$, where $K = I$, then $K^{-1}P = [R|t]$ is denoted as a normalized camera matrix. This abstraction leads to define the essential matrix as the application of the fundamental matrix to a corresponding pair of normalized cameras such as $P = [I|0]$ and $P' = [R|t]$, where $E = [t]_x R = R[R^T t]_x$. Finally considering that $\hat{x} = K^{-1}x$, leads to the final definition that states the essential matrix as:

$$\hat{x}'^T E \hat{x} = 0 \tag{2.28}$$

$$E = K'^T F K \tag{2.29}$$

The structure of E entails two degrees of freedom which are set by a rotation and three other degrees which are set by a translation. Nonetheless there exists an overall scale uncertainty. A decomposition of E can be achieved if we consider that:

$$E = [t]_x R = SR = \begin{bmatrix} 0 & 0 & \sin(\phi) \\ 0 & 0 & -\cos(\phi) \\ -\sin(\phi) & \sin(\phi) & 0 \end{bmatrix} \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.30}$$

with S skew-symmetric and R a rotation matrix. Then using the auxiliary matrices:

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} ; \; Z = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.31}$$

According to [60], being W orthogonal and Z skew-symmetric, E can be decomposed by SVD (Single Value Decomposition) as SVD(E)=$[U|S|V]$, and then the following

factorization can be verified:

$$S = UZU^T \tag{2.32}$$
$$R_1 = UWV^T \tag{2.33}$$
$$R_2 = UW^TV^T \tag{2.34}$$

Finally, this factorization allows to define four possible combination for the projection matrix of the second camera, and thus to obtain x'.

$$P_1 = [R_1|t_x] \tag{2.35}$$
$$P_2 = [R_1|-t_x] \tag{2.36}$$
$$P_3 = [R_2|t_x] \tag{2.37}$$
$$P_4 = [R_2|-t_x] \tag{2.38}$$

2.3.1 Computing Motion Transformation

Once the epipolar geometry has been presented we can focus on its application to the approach of this thesis. Assuming the theory for the planar case, we have applied it to our omnidirectional approach by adapting as well all the considerations presented above.

We can summarize the adaption to the framework assumed in this thesis by means of Figure 2.17. Note that here the camera lays on the focus F of the hyperbolic mirror, so that the concept of epipole does not correspond to the usual and commonly known in the planar case. Instead, we still have the epipolar plane π. In this case it is defined by the projection of the point, where rays coming from the 3D point to the effective viewpoint of the mirror are finally projected towards the camera centers C and C' respectively. It is also worth noticing that the epipolar lines are shaped into ellipses. This is again due to the fact that the intersection of π has to be calculated against the mirror surface, and then projected onto the image plane, which results at last instance into the epipolar lines l and l', turned into elliptical curves. Note that this elliptical lines are the result of the intersection of the epipolar plane with the hyperboloid of two sheets that models our hyperbolic mirror. Its projection on the image plane has the form represented by the dark lines. The infinite intersections generate the final image as expressed by the limits of the dark blue area. This new definition implies a contribution which will allow us to exploit several benefits in the following chapters of this work. The positive outcomes that can be extracted from the epipolar lines will be taken as important advantages for determining matched points between images. This aspect is also of paramount importance when dealing with the computation of motion transformation between poses of the robot, especially when the transformation is only sustained by visual measurements such as those generated by our omnidirectional system. This aspect turns to be essential when dealing with the necessity of robustness when it comes to the observation measurements within a SLAM approach.

In this context, the final goal is to obtain an only-visual observation model for the localization of the robot within the process of SLAM. In order to compute a motion

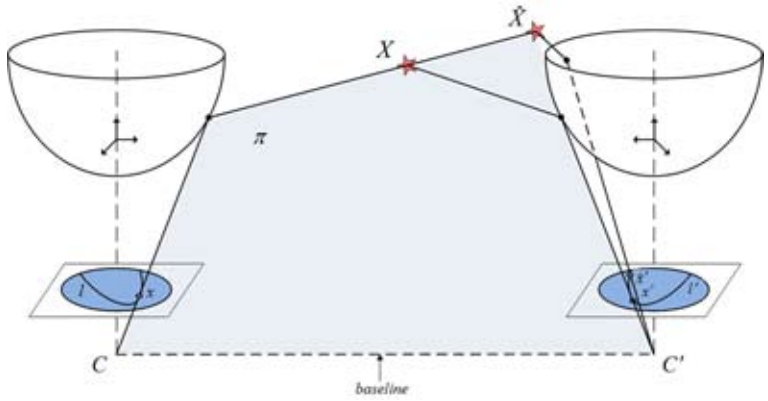


Figure 2.17: Epipolar geometry applied to the omnidirectional camera system.

transformation between poses we designed the procedure based on the epipolar constraints. Figure 2.18 presents the equivalence between two different poses of the robot and the corresponding images acquired at those poses. The motion transformation is produced by a certain rotation and translation, precisely determined by two relative angles: β and ϕ.

Assuming that (2.30) is accomplished, we can explicitly introduce the condition of a planar movement on the XY plane. Therefore the camera rotates on the Z-axis with β, being that axis orthogonal to the XY plane where the robot moves. So that $t_x = [\cos\phi, \sin\phi, 0]$, and finally we can extend the matrix into:

$$E = SR = \begin{bmatrix} 0 & 0 & \sin(\phi) \\ 0 & 0 & -\cos(\phi) \\ -\sin(\phi) & \sin(\phi) & 0 \end{bmatrix} \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The assumption of planar movement relaxes the problem and then the resolution of $x'^T E x = 0$ becomes less strict as now the essential matrix can be posed as:

$$E = \begin{bmatrix} 0 & 0 & e_1 \\ e_3 & 0 & e_2 \\ e_4 & 0 & 0 \end{bmatrix} \tag{2.39}$$

Therefore the epipolar equation (2.28) can be linearly expressed as $De = 0$, with $e = [e_1, e_2, e_3, e_4]$ being the variables of this linear system, and D the coefficients. In other words, D contains the coordinates of points in the two image systems, namely $x = (x_0, y_0, z_0)$ and $x' = (x_1, y_1, z_1)$ for two corresponding points between views. Note that the dimension of D is Nx4, with N the total number of correspondences found. Nevertheless, the minimum number of points to solve the problem is only $N_{min} = 4$. Each i row of D for each pair of correspondences has the following form:

$$D_i = \begin{bmatrix} x_0 z_1 & y_0 z_1 & z_0 x_1 & z_0 y_1 \end{bmatrix} \tag{2.40}$$

According to the decomposition of E by means of SVD, now we can extract the set of rotation and translations shown in (2.32) in the same manner by applying SVD(D). Then it is straightforward to recover the relative angles of the motion transformation from the inspection of the elements of E as follows:

$$\phi = \arctan \frac{-e_1}{e_2} = \arctan \frac{\sin(\phi)}{\cos(\phi)} \tag{2.41}$$

and thus the two possible translations as:

$$t_{x1} = [\cos\phi, \sin\phi, 0] \tag{2.42}$$

$$t_{x2} = [\cos\phi + \pi, \sin\phi + \pi, 0] \tag{2.43}$$

Likewise:

$$\beta = \arctan \frac{e_3}{e_4} + \arctan \frac{-e_1}{e_2} = (\beta - \phi) + \phi \tag{2.44}$$

and finally the two possible rotations as in [13]:

$$R_1 = \begin{bmatrix} \cos\beta & -\sin\beta & 0 \\ \sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.45}$$

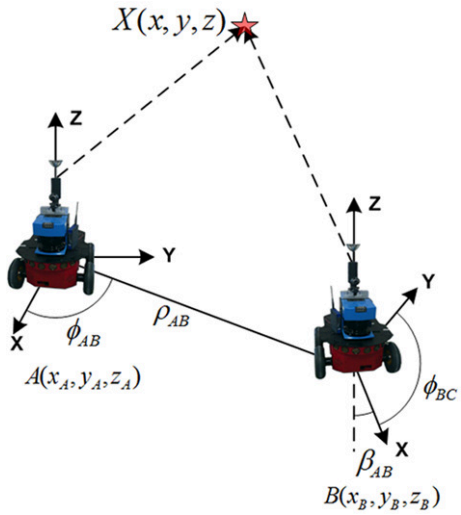
$$R_2 = \begin{bmatrix} 2\cos^2\phi - 1 & 2\cos\phi\sin\phi & 0 \\ 2\cos\phi\sin\phi & 2\sin^2\phi - 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} R_1 \tag{2.46}$$

A last mention has to be made of the other investigated techniques to estimate E. Diverse optimizers were also tested in this thesis, as Lagrange multipliers, Gauss-Newton and Levenberg-Marquardt. However, for our purpose, SVD provides the best balanced solution.

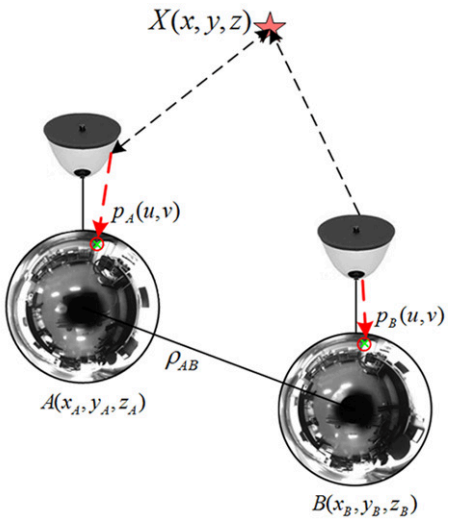
2.3.1.1 Selecting the Solution

The last step to take into consideration is the interpretation of the four possible solution pairs: In (2.35), (2.36), (2.37) and (2.38) these four candidates where compressed into a projection matrix. The interpretation of these solutions is sustained by the possibility of the existence of four different relative positions between cameras, all of which conform a set of angles that accomplish with (2.28) as they are either complementary or supplementary angles. In other words, the four coupled matrices $P = [R|t]$, as seen in (2.35), (2.36), (2.37), (2.38), which can be expressed by the combination of (2.42), (2.43), (2.45) and (2.46), and all satisfy the epipolar constraint.

It is obvious that the spatial interpretation comes from applying the four combination to the relative position between two camera poses, or two robot poses either way. Hence, in Figure 2.18, if we concentrate on Figure 2.18(b), we can distinguish the optical rays to later apply the four possible transformation, and finally select the valid solution amongst them. To do that, it is necessary to determine the combination that finds the point X in front of both cameras. This requires that we backproject x,



(a)



(b)

Figure 2.18: Motion transformation parameters between poses A and B, with relative angles indicated. Figure 2.18(a) shows the relative transformation, whereas Figure 2.18(b) shows the transformation in the camera reference system. A 3D point, $X(x, y, z)$ is indicated with its image projection on both cameras, denoted as $p_A(u, v)$ and $p_B(u, v)$.

which may also be seen as $p(u, v)$, to 3D in order to extract the sign of the rays (r and r'). In other words, we have to recover the direction of these rays so that we get their sign. We can recover the proper solution as that one which produces $r>0$ and $r'>0$ (positive sign implies intersection in front of both cameras) if we consider the motion transformation between poses, X and X', as:

$$X = t_x + RX' \tag{2.47}$$

$$rx = b + r_1 Rx' \tag{2.48}$$

where the second is upscaled and projected on the image frame. We can express it as a least square system in the form:

$$\begin{bmatrix} x_1 & -b \end{bmatrix} \begin{bmatrix} r/r' \\ 1/r_1 \end{bmatrix} = \begin{bmatrix} Rx' \end{bmatrix} \tag{2.49}$$

Then the correct positive pair $r >0$ and $r' >0$ ideally describes the correct rotation and translation which finally determines the real motion transformation between the two poses of the robot, as observed in Figure 2.18(a). However this process has to be carried out with more than one corresponding point. In our approach we set up an histogram with the matched points so that we avoid false correspondences at first instance, but also points that may have rays nearly parallel to the baseline and thus a negative solution. Figure 2.19 represents the four possible combinations. Figure 2.19(a) is the valid solution where both rays intersect in the positive half of both camera systems. Note that between them there exists a certain rotation and translation that we can consider as R_1 (2.45) and t_{x1} (2.42). Then it may be noted that Figures 2.19(a) and 2.19(b) are related by a translation t_{x2} (2.43), and Figures 2.19(c) and 2.19(d) by a rotation R_2 (2.46). Although all provide the same mathematical result that satisfy the epipolar constraint, the only valid is the one represented in Figure 2.19(a). Therefore, our algorithm retrieves it thanks to a process of histogram voting so as to avoid the false correspondence appearance.

2.3.2 Visual Odometry

Once we have presented in the previous section our proposal for computing a motion transformation, the most straightforward application we can think of is an approach to visual odometry. Therefore, in this section we intend to exploit the previous motion transformation model to define our custom application of visual odometry, which can also serve as a validation tool for such model. Besides, we set up an experimental set that permits to extract results which are useful for the establishment of a performance analysis of the motion transformation model.

Approaches to visual odometry have been extensively developed in the field of mobile robotics. They can be classified according to the kind of data sensor used to estimate the trajectory of the robot, such as [103, 95, 88, 101] with stereo cameras. However, monocular visual sensors, have also achieved successful results despite the fact that they can only recover a 2D relative motion [91, 102], and [129, 117] for omni cameras.

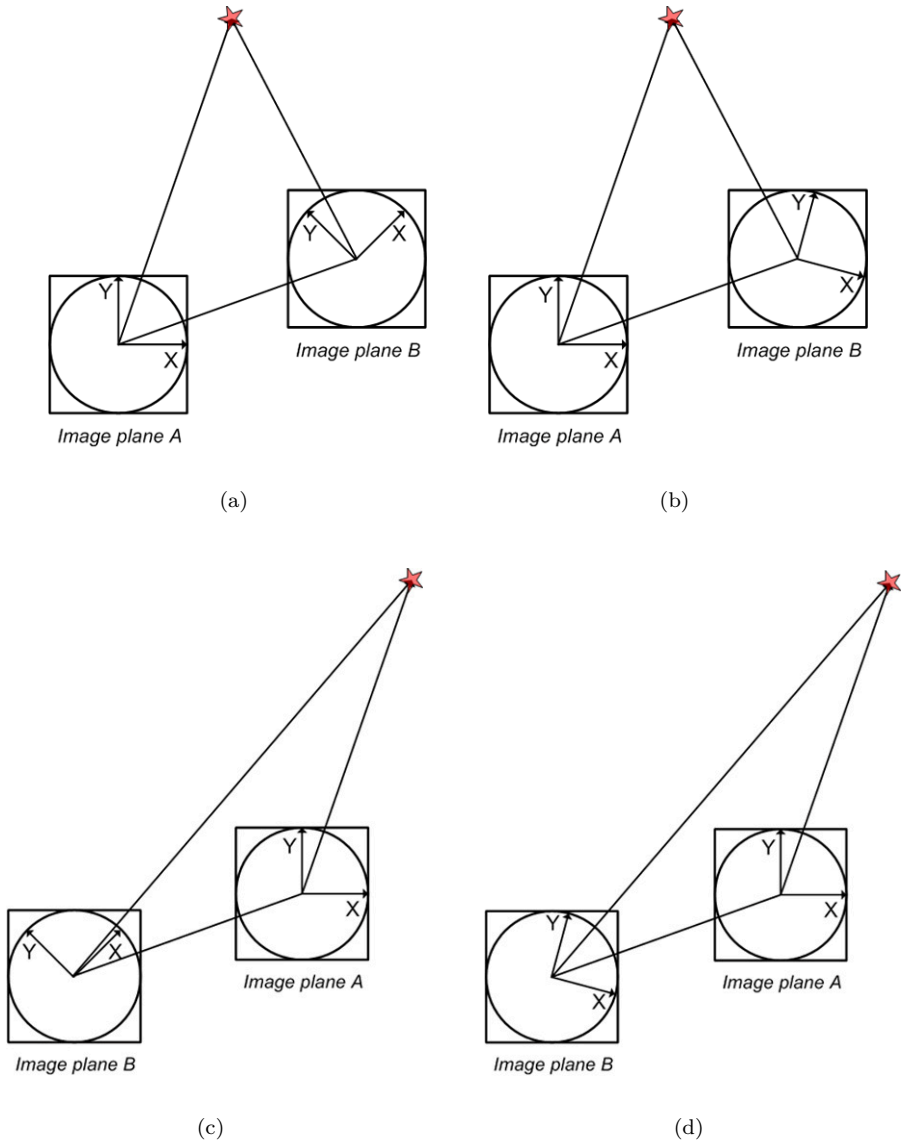


Figure 2.19: Interpretation of the four possible solutions on the plane XY, given a computed rotation R_1, and translation t_{x1}, after applying epipolar constraints. Figure 2.19(a) represents the valid solution where rays intersect in front of both cameras. For each figure, the relative pair of angles that determine the transformation between views is: (R_1, t_{x1}) in Figure 2.19(a), (R_2, t_{x1}) in Figure 2.19(b), (R_1, t_{x2}) in Figure 2.19(c) and (R_2, t_{x2}) in Figure 2.19(d).

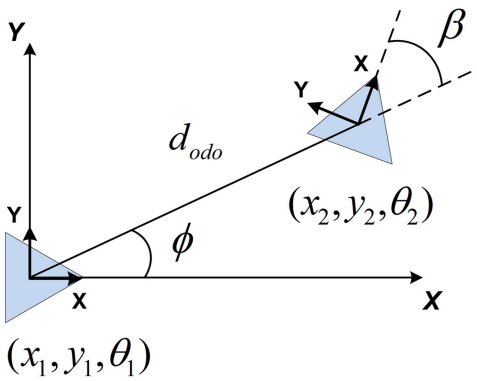


Figure 2.20: Diagram for the visual odometry approach.

Regarding the visual odometry that we define in this section, it entails a feature-based method, which makes use of the matched points between consecutive images at t and $t+1$. Some work in this field has been already conducted as in [23, 116, 90, 45].

The main idea lies on the extraction of a motion transformation from two consecutive omnidirectional views at t and $t+1$, with poses: (x_1, y_1, θ_1) and (x_2, y_2, θ_2) respectively. We can extract the relative angles β and ϕ from the set of matched points between images, as stated in (2.41) and (2.44) by following the procedure presented in Section 2.3.1. Then, assuming that the distance between poses is the value returned by the odometer, we can proceed similarly to the schemes presented in Figure 2.15(a) and Figure 2.15(b) to describe in the same manner the relations between (x_1, y_1, θ_1) and (x_2, y_2, θ_2), as it can be seen in Figure 2.20. Note that we have used a variation of Algorithm 1, where we can assume its motion parameters as:

- Pose at time t: (x_1, y_1, θ_1)
- Pose at time $t+1$: (x_2, y_2, θ_2)
- δ_{trans}: δ_{odo}; distance between consecutive poses at t and $t+1$.
- δ_{rot1}: ϕ
- δ_{rot2}: β
- α_1=α_2=α_3=α_4=0

2.3.2.1 Visual Odometry Results

First of all it is necessary to present the kind of environment where we have conducted the experiments. The defined scenarios correspond with real indoor environments, acquired at office-like spaces. In Table 2.6 we synthesize the main characteristics of these

Dataset characteristics				
Dataset	**No. images**	**Distance**	**Figures**	**Mockup**
Dataset 1	858	85.8 m	Figures: 2.23 and 2.24	Figure: 2.21
Dataset 2	121	48.4 m	Figures: 2.25 and 2.26	Figure: 2.22

Table 2.6: Dataset characteristics

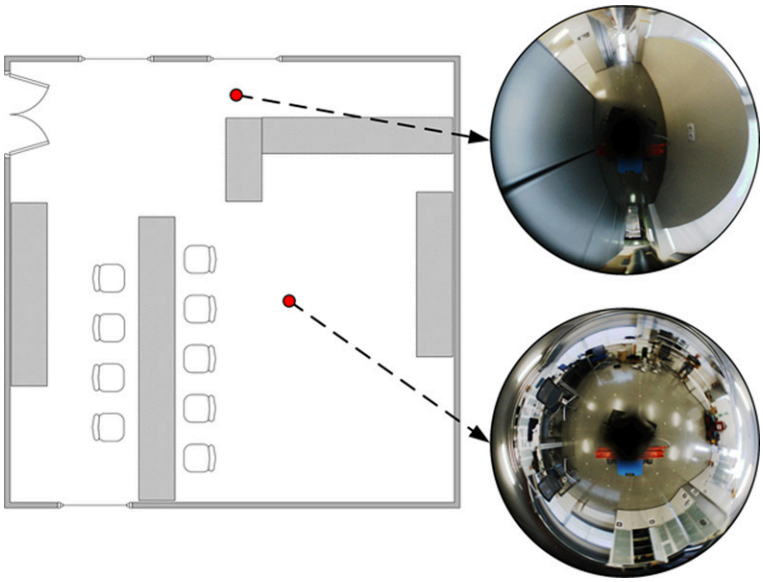


Figure 2.21: Mockup for the Dataset 1. Two examples of views of the environment are indicated.

environments and we associate them with a corresponding dataset of omnidirectional images. Note that we include the references to specific result figures for each dataset. There is also a column that refers to the synthetic mockups that depict the layout of each real scenario.

Dataset 1

The equipment used for the acquisition of data is the same presented in Section 2.2.3. The Pioneer P3-AT allows us to gather omnidirectional images with its corresponding odometry and ground truth. The last one is processed from the raw laser data by means of a gmapping algorithm [127, 51]. According to the results shown in [42], the feature points chosen to compute the motion transformation are SURF [7].

The Dataset 1 corresponds with one of the worst-case scenarios in mobile robotics. Figure 2.21 presents an approximation of the real layout of this scenario. The experiment was conducted while the robot was turning around its own position permanently. The intention was to accumulate a high error on the odometry. This sort of movement usually accomplishes that purpose. We force the robot to maintain a

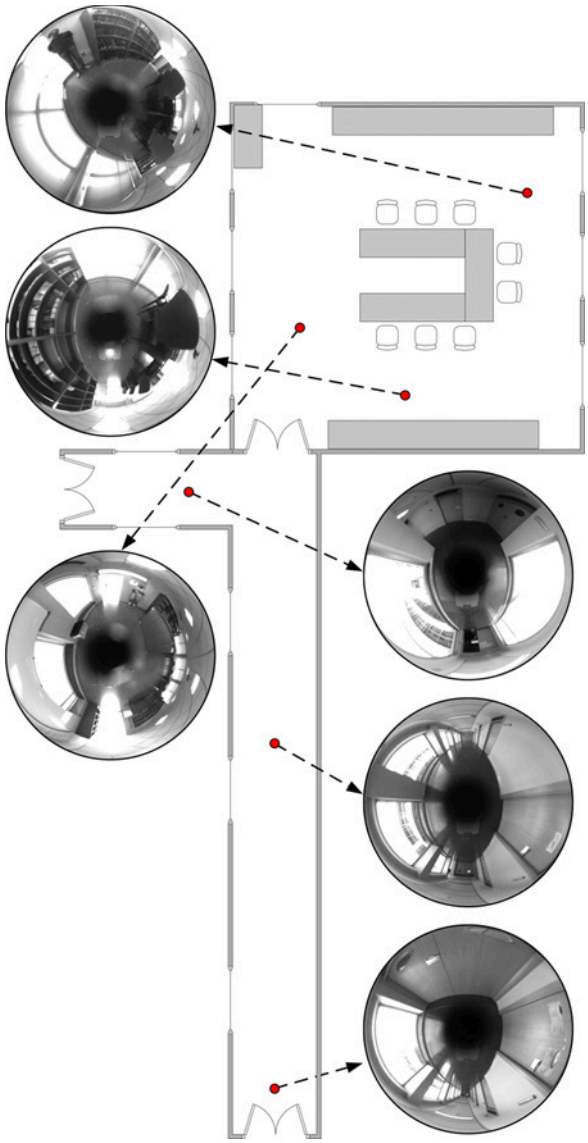


Figure 2.22: Mockup for the Dataset 2. Six examples of views of the environment are indicated.

constant turn that makes the wheels slide most of the time, and therefore the odometer introduces considerable errors. Figure 2.23 presents the visual odometry results for this scenario. The ground-truth is drawn in continuos line, the odometry in dashed line, and the visual odometry estimation in dash-dotted line. It can be observed how the estimation resembles the ground truth. Figure 2.24 presents the obtained errors. Figure 2.24(a) compares the error in X, Y and θ for the visual odometry estimation and the regular odometry of the wheels. Please note that compared to odometry, the best results are obtained by the presented approach. Odometry is prone to increase the error without bounds. Figure 2.24(b) represents the RMS (Root Mean Square) error that has been generated at the final pose of the robot against the number of matched points used to compute the motion transformation. It is important to point out that the computation was carried out 100 times, aiming at the retrieval of robust results. For this reason, the figure shows the mean value and the standard deviation. The tendency of the RMS reveals that the higher number of matched points, the more accurate results. It is obvious that low amounts of matched points may lead to harmful effects in case that few false positives are not filtered and input into the system.

Dataset 2

Similarly to the first dataset, the Dataset 2 consists of an office-like scenario with the addition of a corridor that introduces a changing effect on the lighting conditions. This also poses a challenge for the experiment. Figure 2.22 presents an approximation of the real layout of this scenario. In this experiment, we did not make use of the P3-AT robot. Here, an omnidirectional dataset was manually acquired over a grid conformed by 381 positions. The grid step is 40 cm. The goal is to prove the feasibility of the visual odometry estimation in a more realistic situation than in the previous dataset. Here the robot traverses a narrow corridor with windows on the one side until it enters a second room and it finally returns over the same trajectory. Following the same statements than in the previous dataset, the experiment has been repeated 100 times so as to ensure robust results in terms of error. Figure 2.25 presents the results. The ground-truth is drawn in dash-dotted line and the visual odometry estimation in continuos line. Note that there is not odometry values as the experiment has been developed by using a grid of images. Thus in this dataset we assume $\delta_{trans}{=}\delta_{odo}$ as the value extracted from the grid step. Again, the topologic shape of the estimation demonstrate high resemblance with the ground truth. Figure 2.26 presents the obtained errors for this scenario. Figure 2.26(a) compares the error in X, Y and θ for the visual odometry estimation and the regular odometry. Figure 2.26(b) represents the mean RMS error at the last pose of the robot over the 100 repetitions of the experiment. A similar conclusion may be extracted here, since the evolution of the RMS also proves that the more number of matched points, the more accurate results.

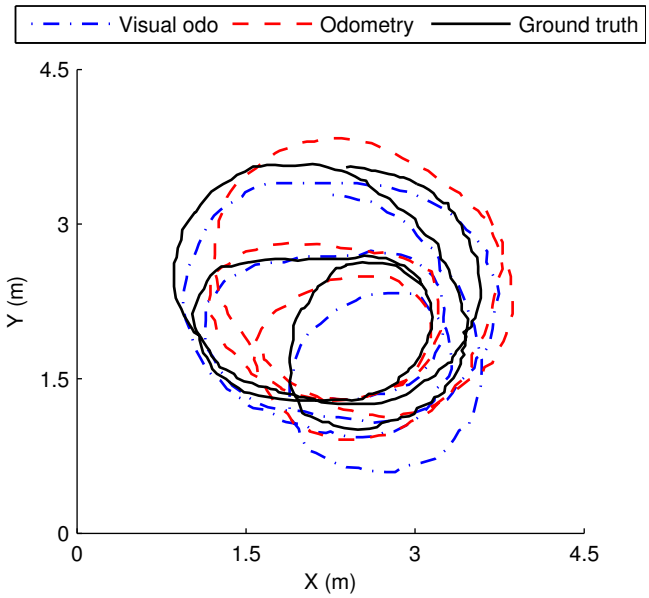


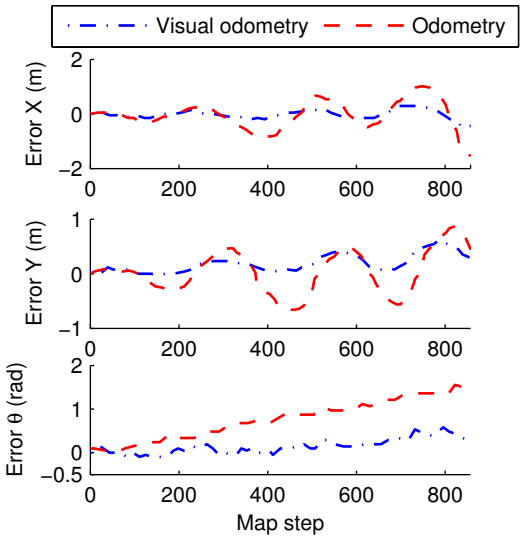
Figure 2.23: Results of visual odometry obtained in the Dataset 1. The estimated visual odometry is drawn in dash-dotted line, the odometry in dashed line and the ground truth in continuous line.

As a preliminary output extracted from these experiments, this visual odometry approach demonstrates that the relative angles β and ϕ obtained by means of the motion transformation, are valid for real applications in the field of mobile robotics.

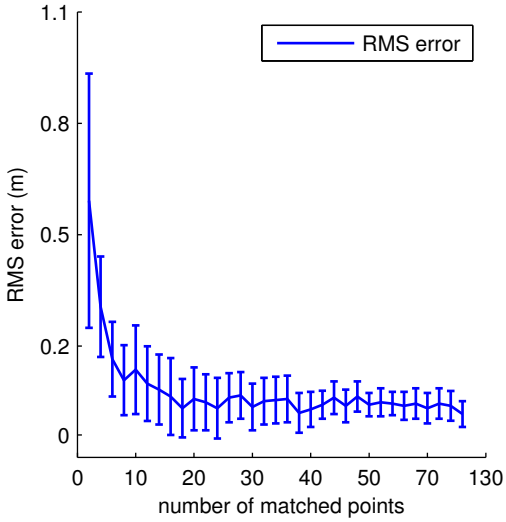
2.3.3 Performance

The definition of the previous experimental setup led us to consider a further study on the results in order to extract conclusions about the precision of the measurements. For this reason, we analyze the performance provided by the computation of the motion transformation described in Section 2.3.1. The accuracy on the extracted values (ϕ, β) needs to be studied under different conditions. The most relevant factors to take into account are the variation on the number of matched points and its dependency on the computational load.

Taking advantage of the datasets acquired in the previous section, we make use of the Dataset 1 and Dataset 2 in order to establish a series of performance experiments. We present the different results that have been obtained by using several variants of the solver algorithm embedded by the motion transformation computation. In particular, we have defined different kernels for the former SVD solver method, which was initially introduced to solve the equation system characterized by the coefficient matrix D, as expressed in (2.40). Note that a minimum number of 4 corresponding points between



(a)



(b)

Figure 2.24: Error results obtained in the Dataset 1. Figure 2.24(a) represents the error at each step in X, Y and θ. Figure 2.24(b) presents the mean RMS error and standard deviation against the number of matched points.

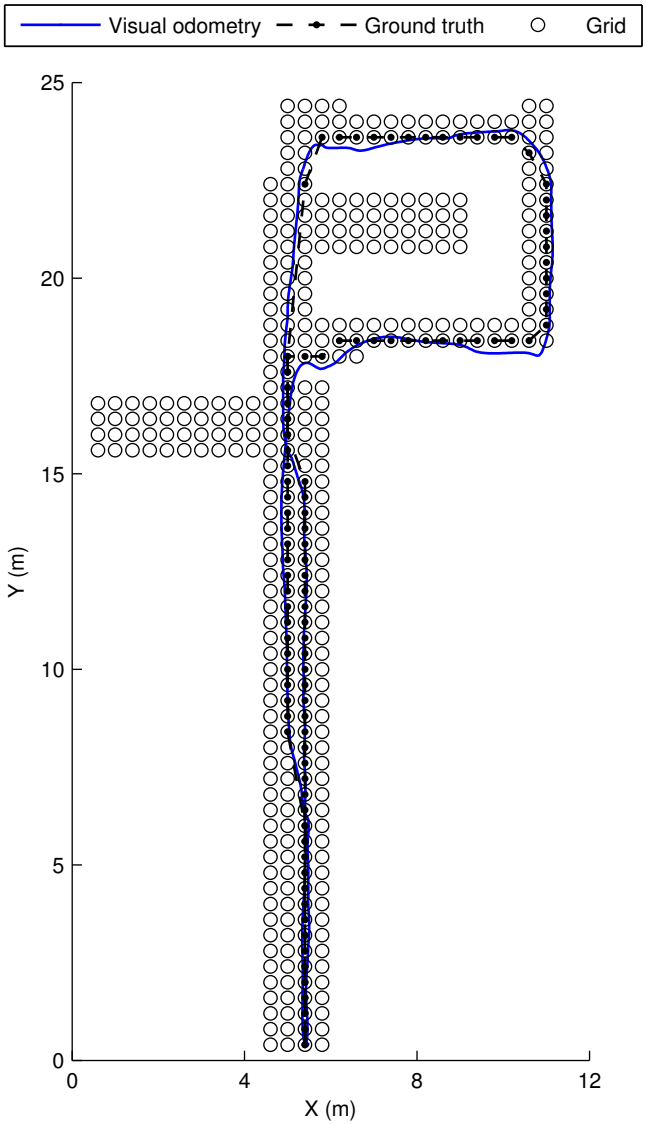
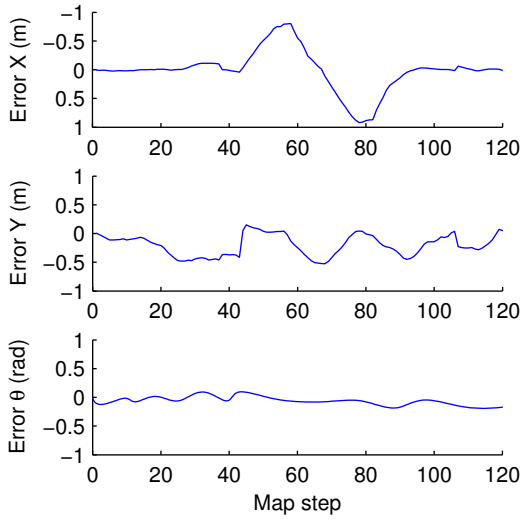
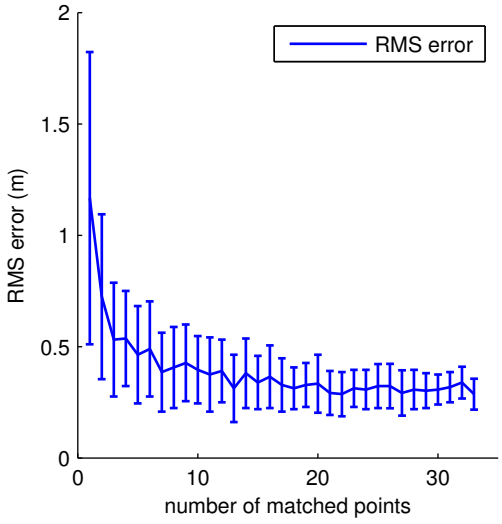


Figure 2.25: Results of visual odometry obtained in the Dataset 2. The estimated visual odometry is drawn in continuous line and the ground truth in dash-dotted line. The dark dots represent the rest of images that conform the grid.



(a)



(b)

Figure 2.26: Error results obtained in the Dataset 2. Figure 2.26(a) represents the error at each step in X, Y and θ. Figure 2.26(b) presents the mean RMS error and standard deviation against the number of matched points.

views is required to extract the motion transformation for a certain observation model, as $z_t(\phi, \beta)$. However we have set up three different schemes (Scheme 1, 2 and 3) to compute these values, which are defined as follows:

Scheme 1

A set with the total number of matched points detected between images, N, as the inputs for the SVD solver. Figure 2.27 depicts the block diagram for this scheme where I_1 and I_2 are two views to extract matches from, p and p', and D_{Nx4} the coefficient matrix that contains the input set for the SVD solver with the total N matched points found. Figure 2.30 presents results of accuracy for this first scheme, which is sustained by a method based on the SVD solver. The error is calculated as the absolute deviation from the real value in radians, though in the figure is converted to degrees for a simpler observation. Each bar represents a bin with the number of matched points that input the SVD solver. The frequency for each bin is represented by the height of the bar as a $\%$ out of the total number computed for the bins, that is:

$$\%_i = \frac{frequency_{bin_i}}{\sum_{i=1}^{n} frequency_{bin_i}}$$

For this method, the total number of matched points are introduced via the coefficient matrix D (2.40), as the only input set for the solver. That is, the final solution is obtained in a single step with the total number of matched points detected between images. Nevertheless the experiments have been repeated 100 times so as to avoid biased dependencies. As a result, we can plot mean values and standard deviations for the error on the computed measures.

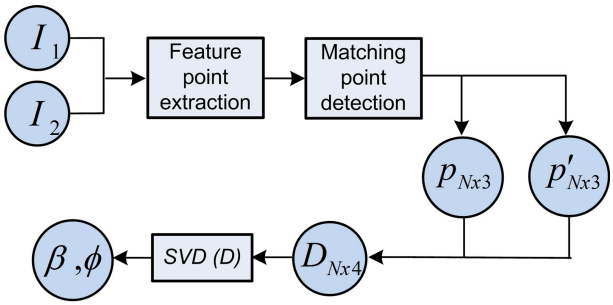


Figure 2.27: Block diagram for the Scheme 1.

Figure 2.30(a) presents the evolution on the error in β against the number of matched points and their frequency of repetition. Figure 2.30(b) presents the same terms for ϕ. Again, the mean values and standard deviation have been plotted. The precision on the estimated angles confirms the expected behaviour: the higher number of matched points the better retrieval of results. Despite this last fact, it also can be proved that the precision of these results is acceptable even when the number of matched points are relatively low. Notice that values obtained with 15 matched points provide a tolerable estimation for its use in a real application.

Scheme 2

Different n-subsets of matched points as the inputs for the SVD solver, being $n = N/k$, where N still represents the total number of matched points and k the desired size for the subsets. Thus the solution consists of n-pairs of values for β and ϕ, namely β^n and ϕ^n. Then an histogram voting with mean values is needed to reach a final solution. Note that each subset generates a matrix D^n with size of k-by-4. Figure 2.28 depicts the block diagram for this scheme, where I_1 and I_2 are two views to extract matches from, p and p', and D^n_{kx4} all the coefficient matrices into which the n-subsets are divided to input separately the SVD solver.

Figure 2.31 presents the same results of accuracy for the second variant of the solver. Here, we have divided the total number of matched points of each bin, into several different n-subsets as inputs for the SVD. In consequence, the number of solutions depend on the total number of matched points detected, N, and the value of k, which determines the size of D_{kx4} and also the number of subsets, since $n = N/k$. Then an histogram voting is applied in order to return a mean value for the final solution. Figure 2.31(a) presents the evolution on the error in β against the number of matched points and their frequency of repetition. Figure 2.31(b) presents the same terms for ϕ. In this case, the results provide a more accurate estimation than in the previous scheme. It is worth noticing that values obtained with 9 matched points can be acceptable for its use in a real application. The key point lays on the subdivision of the total number of matched points for the obtention of several n-solutions. This means that possible false positive that the system was not able to reject at first stages are now spread along the subsets, and their harmful effects attenuated. Then, false positive bias the solution only for a limited number of subsets, instead of the entire input as it would happen in the previous scheme. In this way, the effect of damaged solutions is diminished. Finally an histogram voting computes a mean value with all the possible solutions provided by all the subsets. However, it is evident that this scheme consumes more computation time. The next scheme even demands more time efforts.

Scheme 3

Different randomly permuted n-subsets of matched points as the inputs for the SVD solver: This strategy is quite similar to the previous one, but it uses a combinatorial histogram voting instead. This permits to randomize and obtain a considerable high number of possible combination for the n-subset of matched points to input the solver.

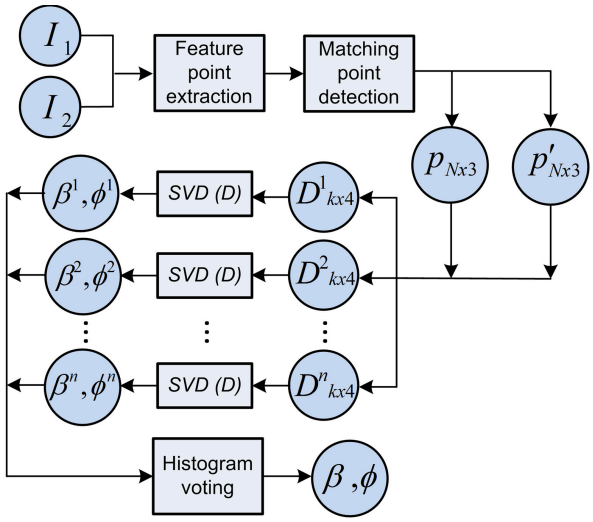


Figure 2.28: Block diagram for the Scheme 2.

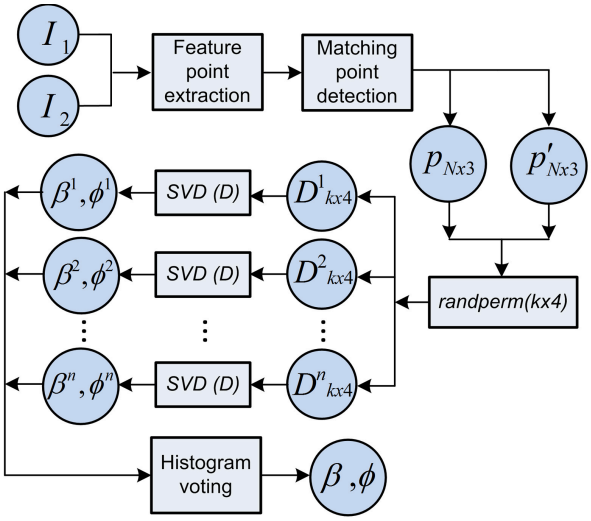


Figure 2.29: Block diagram for the Scheme 3.

Figure 2.29 depicts the block diagram for this scheme, where I_1 and I_2 are two views to extract matches from, p and p', and D^n_{kx4} all the coefficient matrices generated from the combinational randomization of the n-subsets to later input the SVD solver separately.

Finally, Figure 2.32 presents the results of accuracy for the third variant of the solver. Here the procedure is quite similar to the previous one. The difference is

introduced by a combinatorial permutation redistribution. For each bin, this technique randomly permutes the n-subset of matched points that input the solver, and allows to calculate all the possible combinations. Again, the final solution is extracted after applying histogram voting and a mean estimator.

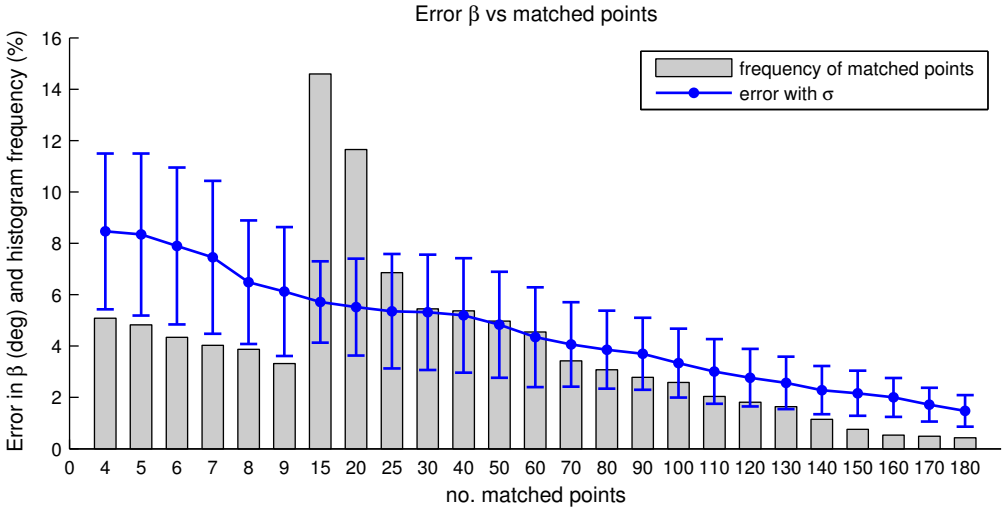
Similarly to previous figures, Figure 2.32(a) presents the evolution on the error in β against the number of matched points and their frequency of repetition. Figure 2.32(b) presents the same terms for ϕ. In this case, the results provide the most accurate estimation. Despite this fact, the results are very close to the previous ones and do not make a big difference. Here, we propose a randomization over the number of matched points to construct the different n-subsets. This allows us to reutilize the same data and to spread even more the possible presence of false positives. Therefore, computing a higher number of possible subsets makes the effect of wrong correspondences more irrelevant. Nonetheless, the time consumption may become totally inviable for a normal use in an application. Higher number of matched points implies a computation effort for the generation of the combinational permutation, which is definitely not worth it if we consider a balance between accuracy and time consumption.

Lastly, it is worth noticing that several selections of the value k that can be made. However, we conducted this analysis with the minimum number $k = 4$. Higher values imply that observations with a total number of matched points lower than k, cannot be evaluated with all the variant methods presented above, as some subsets would not have been filled due to the lack of points, and thus the comparison would have not be carried out.

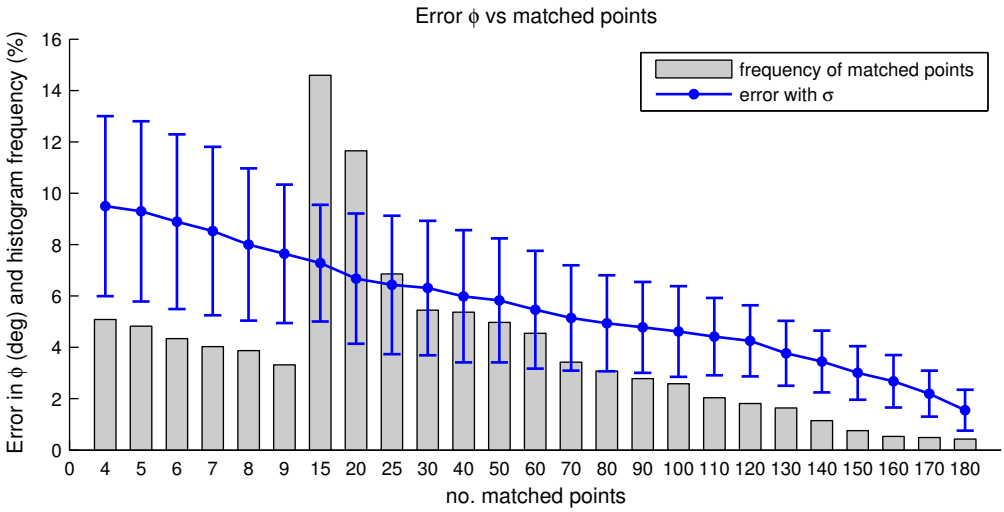
The experiments presented above suggested that the computational costs should be determined so as to provide a complete analysis of the schemes. Therefore, now we pursue to describe the time requirements for the motion transformation model presented in this chapter. To that end, Figure 2.33 divides the different contributions to the final time consumed in Scheme 1. In particular, Figure 2.33(a) represents the time spent by the matching process to detect points and the time spent by the SVD solver to provide the final solution. The sum of these two contributions is represented as the total time consumption. Note that the matching is a limiting stage in this sense. Even for higher number of matched points, and although the time consumed by the SVD grows with higher steep, it never reaches the time consumed by the matching process.

Figure 2.33(b) provides an outline for the comparison between the error obtained in β and ϕ with the total time consumption in the framework defined by the Scheme 1. Inspecting this figure makes even more evident that a high number of matched points to retrieve an accurate solution is not necessarily required. Similarly, Figure 2.34 and Figure 2.35 provide time results for the Scheme 2 and Scheme 3 respectively. These figures show the total time consumed.

The last set of results demonstrate the first evidences on the time consumption. It is clear that the more number of matched points, the more accuracy on the estimation. However, the solver methods proposed in Scheme 2 and Scheme 3 may be

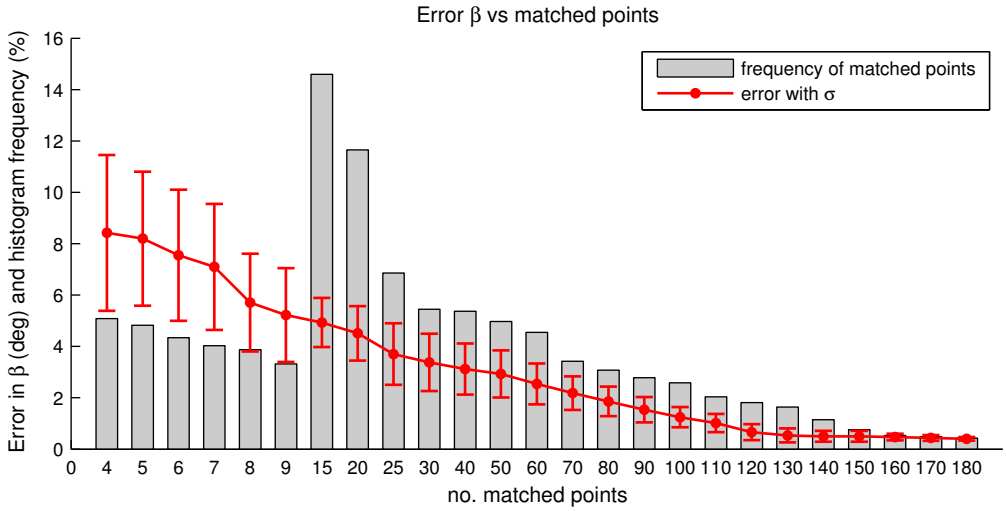


(a)

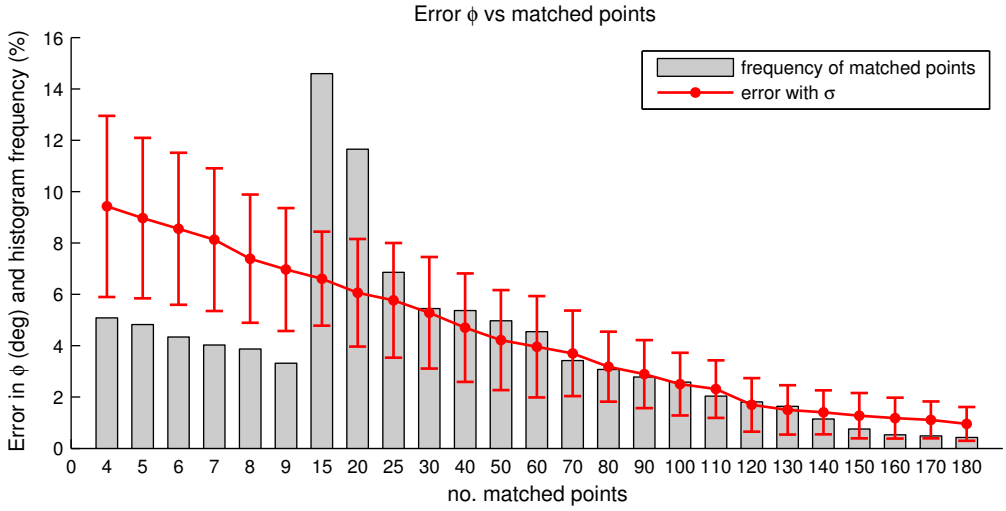


(b)

Figure 2.30: Scheme 1: Former SVD solver. Evolution of the error in β and ϕ (deg) against the number of matched points. The bins represent different subdivisions for the number of matched points detected. The frequency is presented as a % out of the total.

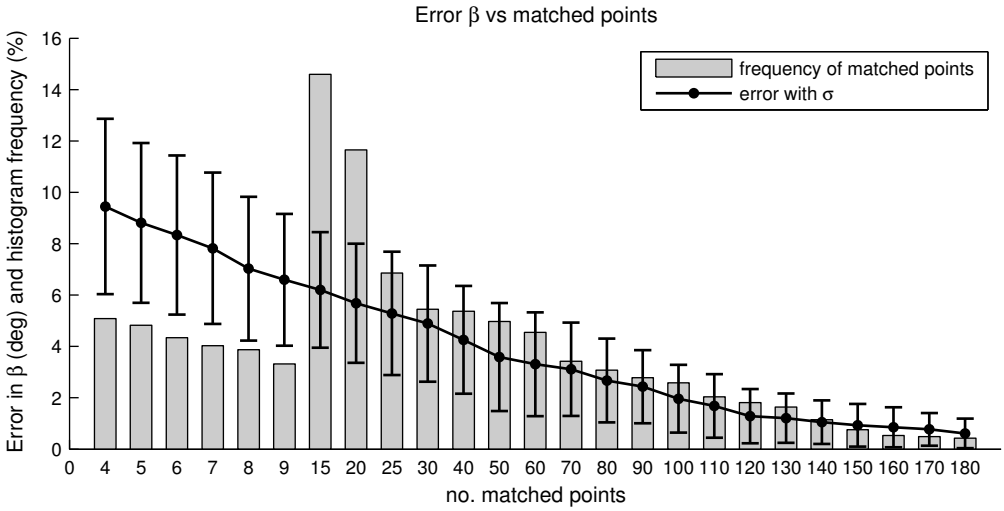


(a)

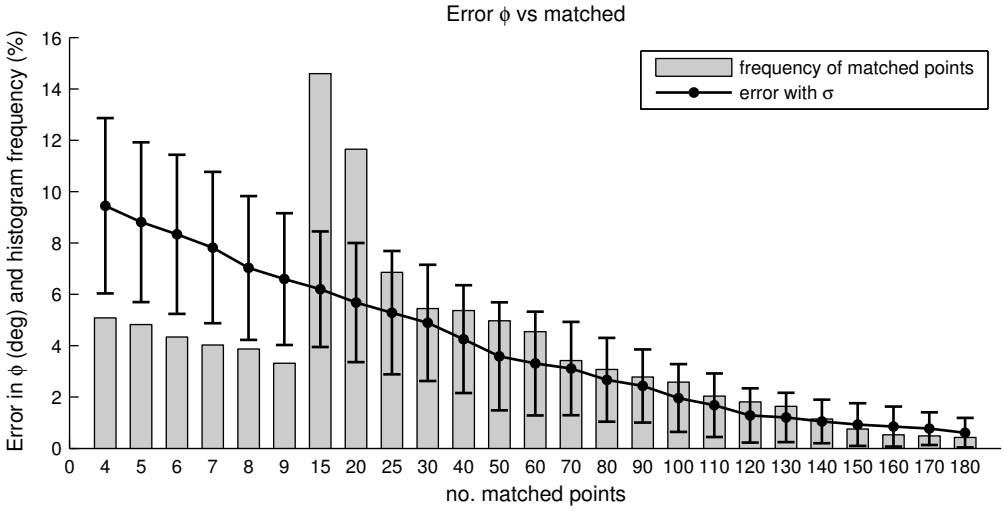


(b)

Figure 2.31: Scheme 2: SVD solver with n-subset inputs and histogram voting. Evolution of the error in β and ϕ (deg) against the number of matched points. The bins represent different subdivisions for the number of matched points detected. The frequency is presented as a % out of the total.



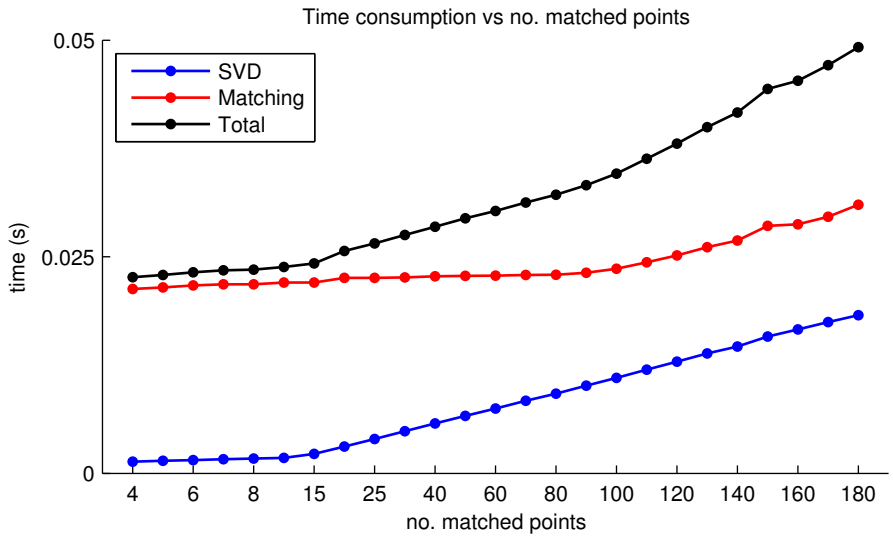
(a)



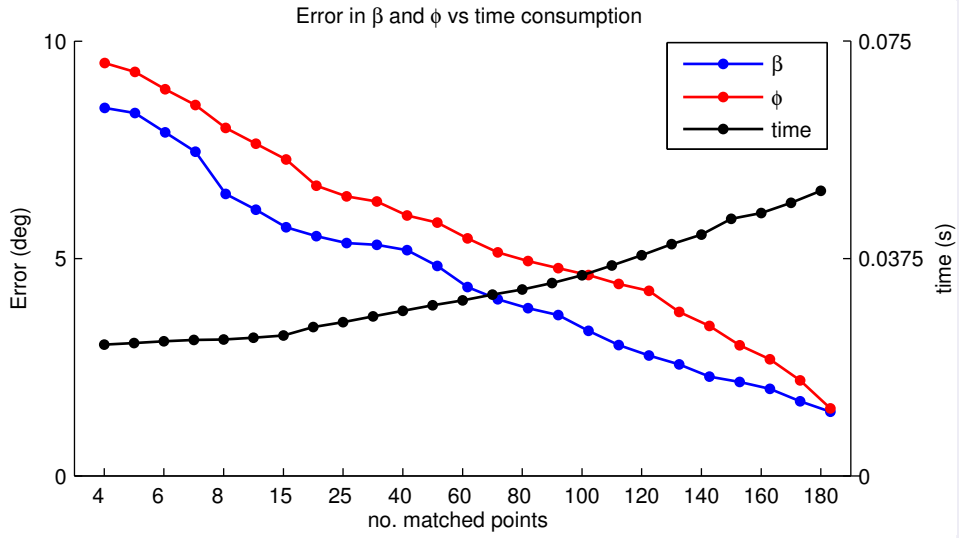
(b)

Figure 2.32: Scheme 3: SVD solver with n-subset inputs selected by combinational permutation, and histogram voting. Evolution of the error in β and ϕ (deg) against the number of matched points. The bins represent different subdivisions for the number of matched points detected. The frequency is presented as a % out of the total.

only valid for certain applications. In a platform with real-time requirements, the best trade-off solution between accuracy and computational cost is given by the Scheme 1, which is paradoxically the simplest. The Scheme 2 and Scheme 3 are capable to compute relative angles with an error close to the tenth of a degree, but at an expensive time cost over a second.



(a)



(b)

Figure 2.33: Scheme 1: Time consumption and error. Figure 2.33(a) shows the time consumed by the SVD, the matching process and the total time consumption. Figure 2.33(b) shows the error in β and ϕ against the total time consumption.

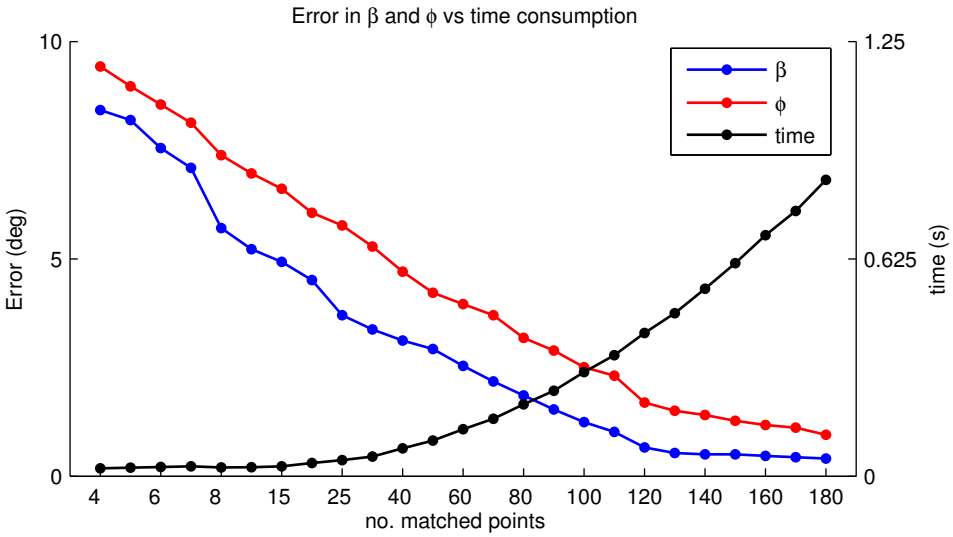


Figure 2.34: Scheme 2: Error in β and ϕ against the total time consumption.

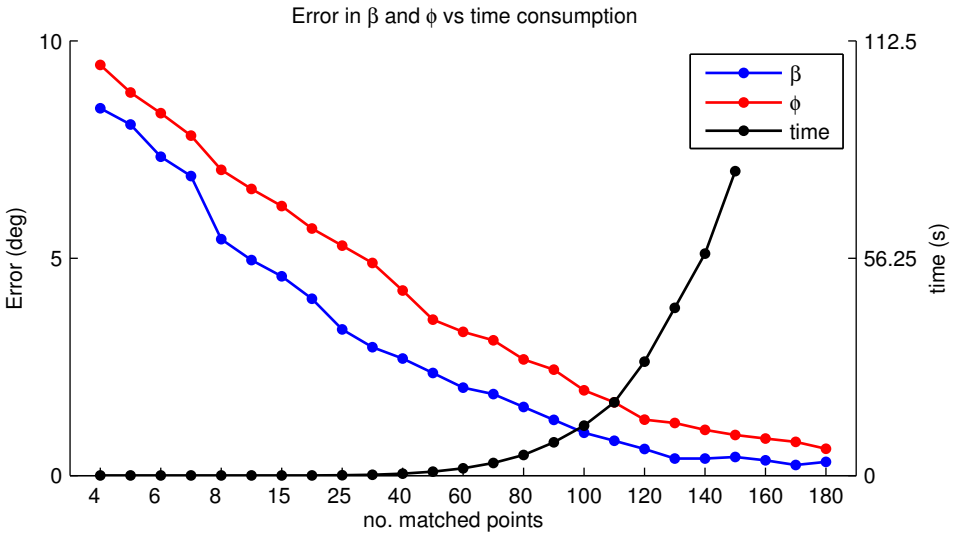


Figure 2.35: Scheme 3: Error in β and ϕ against the total time consumption

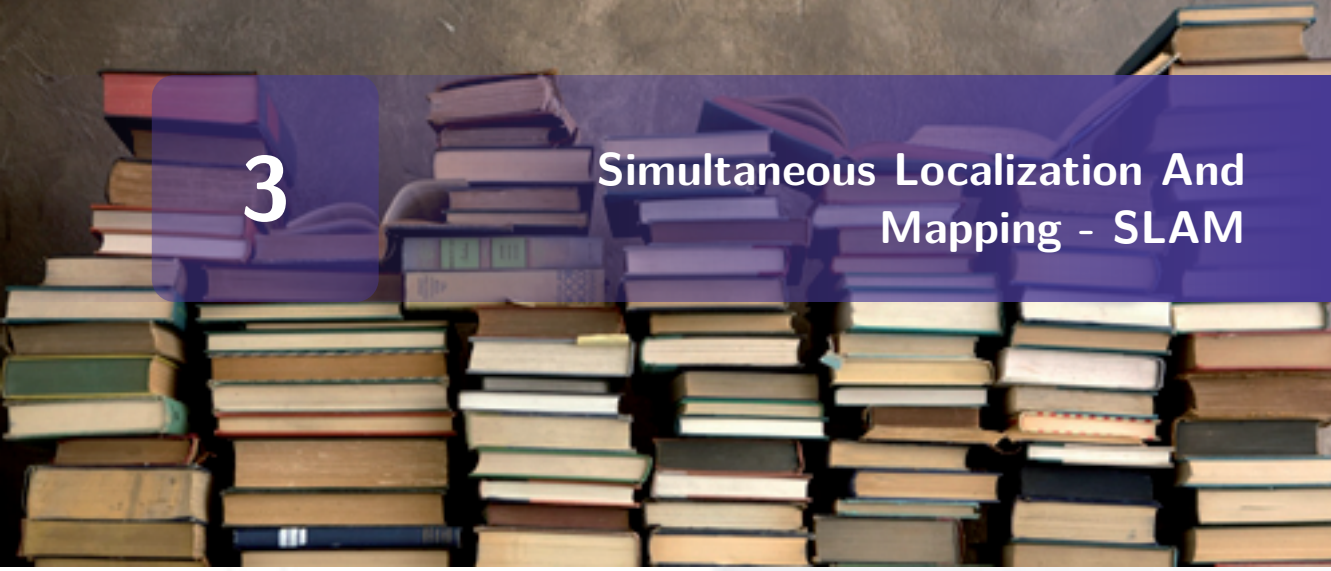
2.4 Conclusions

In this chapter we have provided an outline of the essentials for the catadioptric sensors, paying special attention to the omnidirectional visual sensor. In this context, we have described in detail the fundamentals of the projection model of our omnidirectional system, which represents the key mechanism for gathering visual data within the field of this thesis.

During the exposition of these fundamentals, we have dealt with different tasks that had to be implemented in the code framework of this thesis. The embedding of an omnidirectional calibration is one of them. Another development carried out was the definition of the motion transformation model for the extraction of a relative translation and rotation between two poses of the robot. This contribution is of paramount importance for its later application to the observation model when dealing with SLAM tasks. We have implemented it thanks to certain epipolar geometry considerations. However, a previous stage required that this epipolar geometry was precisely defined and adapted to our omnidirectional system, since the regular case is normally addressed for planar camera models. Furthermore, this contribution also allows to improve the observation model for our SLAM problem, since it may reduce the search for corresponding points in the second image. We have also included further details about the implementation of this motion transformation model.

Finally, the last subsections in this chapter have been aimed at the practical evaluation of the initial proposals and contributions. We have conducted experiments for a visual odometry platform, since generally, it only makes use of the motion transformation measurements. These results confirmed the reliability of the measurements in order to operate with real system applications. Moreover, we have defined a benchmark configuration which allows to assess the performance of the approach in terms of accuracy and computational requirements. We can conclude that this approach provides results that are usable by real-time applications, since they do not need an excessively powerful system to produce a good trade-off solution between performance and accuracy.

These results establish the starting point for the research and development of our observation model within the problem of SLAM. Further information about this line of research is given in the next chapters.



3 Simultaneous Localization And Mapping - SLAM

Navigating in unknown environments inherently couples the task of building a map and the computation of the relative robot's location referred to the map. This statement is the first and most straightforward interpretation inferred from the acronym of SLAM (Simultaneous Localization And Mapping). As it has been primarily detailed in Chapter 1, in this field of mobile robotics there is a crucial paradigm which the SLAM process has to deal with. After the former developments on SLAM [124], the last decade has witnessed a tremendous advance in obtaining a solution for the SLAM problem with different propitious implementations. Besides the possible designs for the sensor system and the representation of the environment, these implementations rely heavily on the kind of algorithm to provide a trustworthy backend core for any SLAM system application. In this sense, different SLAM algorithms have been extensively used, which can be principally distinguished by the sort of algorithm, such as online algorithms [25, 20, 94] and offline algorithms [53, 40].

Historically, the major research has concentrated on improving the computational efficiency [32, 123, 57, 29], assuring at the same time consistency and accuracy for the estimation of the map and vehicle pose [31, 80, 16, 56, 147, 28]. However, nowadays the tendency of research has deviated towards issues with regard to non-linearities [66, 70], data association [122, 119, 6, 25] and landmark recognition [78, 109, 26, 105]. All of which are crucial to outline the new milestones for polishing the theoretical and practical implementation of a robust SLAM model.

This chapter intends to provide with a general overview of the basis of a SLAM problem. Then its analytic fundamentals are object of in-depth analysis so as to provide the reader with sufficient background on the main algorithms and methods which are

later needed to implement and develop the new contributions proposed in the framework of this thesis. Consequently, we have designed the structure of this chapter to fulfill the main theoretical aspects involved in this thesis as follows:

- First, we introduce the fundamentals of the problem of SLAM. We focus on its probabilistic nature, initially defined through Bayesian considerations and materialized by the integration of probabilistic methods in the field of robotics and artificial intelligence sciences.
- Next we concentrate on the basis of several algorithm-specific methods we selected in this thesis to develop and implement new contributions to the framework of SLAM. In particular, we introduce the essentials of the Extended Kalman Filter (EKF) and the Stochastic Gradient Descent (SGD).
- Then we present an overview to Gaussian Processes (GP), as they are later needed in order to produce some contributions to keep the uncertainty of the system bounded.
- In consequence with the previous point, we provide a brief introduction to information-based theory, as it is required to work on the uncertainty considerations.

Therefore this chapter constitutes the theoretical framework which sustains all the later development of contributions and implementations carried out in this thesis. In consequence, it will be implicitly referenced in all the work, contributions and publications presented along this thesis document.

3.1 SLAM Definitions

Generally, a preliminary approach to the essentials of the SLAM problem can be depicted by the schematic presented in Figure 3.1. A mobile robot is expected to move through a certain environment while it acquires relative observations of a number of unknown landmarks by means of a specific sensor. This permits that the simultaneous estimation of the robot and landmark locations is accomplished. Please notice that the true locations are never available either measured at any time. Finally, observations are considered relative between the real path followed by the robot and the landmark locations. Assuming the scheme stated above, the following bullet points list the different variables as function of the time instant, t.

- Index t: Temporal instant.
- x_t: State vector that describes the location and orientation of the robot.
- u_t: Control vector that drives the robot to the next state at time t. It is applied between consecutive time steps, that is t and $t+1$.
- l_i: Vector that describes the location of the $i-th$ landmark, whose true location can be assumed as invariant.

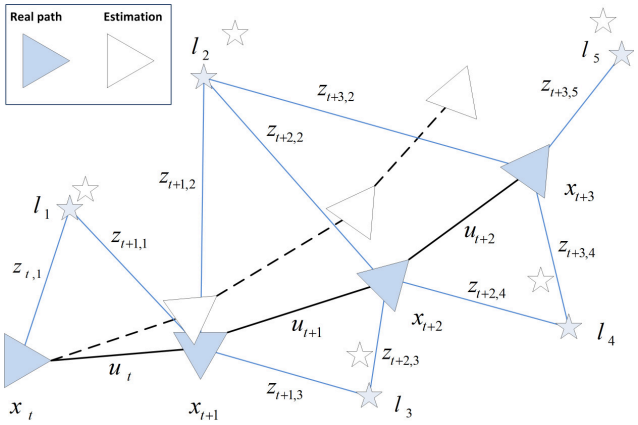


Figure 3.1: The colored items represent the real position of both, the path followed by a vehicle, denoted by its state vector x_t, and the set of discovered landmarks as l_i. The same variables are estimated by the SLAM algorithm and represented with blank items. The observation measurement between the vehicle and the landmarks are expressed by $z_{t,i}$, while the control input which drives the vehicle from consecutive states is indicated by u_t. Note that the true locations are never known or measured directly. Observations are made between true vehicle and landmark locations.

- $z_{t,i}$: Observation vector computed at the current robot's location, at time t. Index i expresses the observation to the $i-th$ landmark. Whether there are multiple landmark observations at time t or the specific landmark is not relevant, the observation will be simply written as z_t.

In order to keep record of the previous variables, the next subsets are defined:

- $X_{0:t}=\{x_0, x_1, \cdots, x_t\}=\{X_{0:t-1}, x_t\}$ $\Rightarrow$ History of vehicle location.
- $U_{0:t}=\{u_0, u_1, \cdots, u_t\}=\{U_{0:t-1}, u_t\}$ $\Rightarrow$ History of control inputs.
- $l_{0:t}=\{l_0, l_1, \cdots, l_k\}$ $\Rightarrow$ Set of landmarks.
- $Z_{0:t}=\{z_0, z_1, \cdots, z_t\}=\{Z_{0:t-1}, z_t\}$ $\Rightarrow$ Set of observations.

3.1.1 Bayesian Considerations

The Bayesian approach is crucial when dealing with variables associated with random noise. Bayesian filtering is one of the most accepted solutions in this sense, since it provides the fundamentals for the later state estimations of dynamics models [4, 130, 49, 38, 97, 110]. Based on Bayes theory, the probability density distributions of such variables has to be introduced so as to build a state vector that accounts for all the statistical information.

According to Bayes theory, the problem can be extended to a general case where probabilistic formulation leads to:

$$P = (x_t, l|Z_{0:t}, U_{0:t}, X_0), \tag{3.1}$$

being $l = l_{0:t}$. Equation (3.1) represents the SLAM problem expressed as a probability distribution, which is successively computed at every time t. Given the set of control inputs and recorded observations, this probability distribution characterizes the joint posterior density for the vehicle's state vector and the landmark locations. It considers probability up to time t, but it also includes the initial state of the vehicle.

Regularly, a recursive solution is desirable. The starting point is established by an initial guess (global or local) as an estimation for the distribution, that is $P(x_{t-1}, l|Z_{0:t-1}, U_{0:t-1})$. Here Bayes theorem is utilized to compute the joint posterior, following an observation z_t and a control input u_t. Notice that it is important to define a state transition model and an observation model to bear in mind the effect of the control input and observation respectively.

The observation model determines the probability of performing an observation z_t by assuming the vehicle and landmark locations as known. It may be stated as:

$$P(z_t|x_t, l) \tag{3.2}$$

It is feasible to assume that observations are conditionally independent given the map and the current vehicle state already defined. Thus the motion model for the vehicle can be addressed in terms of the probability distribution associated with the state transitions:

$$P(z_t|x_t, l)P(x_t|x_{t-1}, u_t) \tag{3.3}$$

That is a Markov assumption by which the corresponding state transition considers the next state x_t being only dependent on the immediate preceding state x_{t-1} and the control u_t. Conversely, it may be considered independent of both the map and observations. Therefore, the SLAM algorithm is expected to be deployed in a recurrent two-stage process, which is triggered by observations: prediction and update. Figure 3.2 represents a graph that depicts the Markov model for the SLAM problem. Nonetheless, the discussion stated above can be simplified and the conditioning on historical variables in (3.1) eventually dropped. Now, the required joint posterior on map and vehicle location can be rewritten as $P(x_t, l|z_t)$, or even $P(x_t, l)$ at certain contexts. The observation model $P(z_t|x_t, l)$ explicitly expresses the dependence of observations on both the vehicle and landmark locations. As a result, the resulting joint posterior cannot be easily disconnected in the known form:

$$P(x_t, l|z_t) \neq P(x_t|z_t)P(l|z_t) \tag{3.4}$$

3.2 Extended Kalman Filter - EKF

The Extended Kalman filter is an extensively employed representation in the form of a state-space model with additive gaussian noise, which also integrates Bayesian considerations to deal with such noise [92, 72].

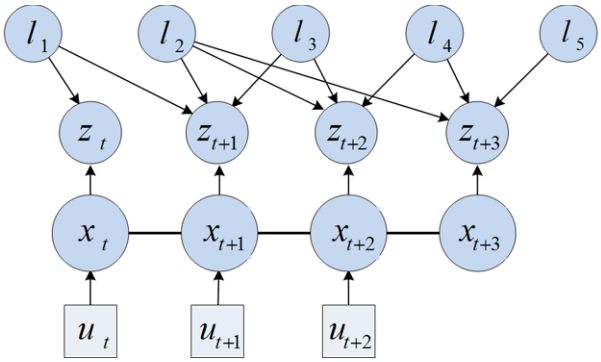


Figure 3.2: Markov model for the SLAM problem, where the observation measurements are assumed conditionally independent. x_t and u_t represent the state vector and control inputs respectively, meanwhile l_i and z_t are the landmark locations and their pertinent observation measurements. Note that z_t at each t comprises the whole set of observation measurements to all the visible landmarks.

The solution for the SLAM problem was first introduced by Smith et al. in [124, 123], who utilized the EKF [30] as the basic kernel for the system. Most of the research on this technique concentrated on approaches which relied on artificial landmarks [27, 67, 24, 58, 20]. Although in general, any sort of landmarks in the environment may be considered. As per the sensor to get observation measurements of these landmarks, the most commonly used was laser [44, 140, 143, 12, 81]. The EKF is able to process jointly the observation measurements for the landmarks and the control inputs in order to retrieve a map of the environment and an estimation for the robot's pose.

An actual problematic of using this filter comes tied to computational requirements. It exists a critical dependence on the number of landmarks, namely N, and the speed of convergence, expressed by $O(N^2)$. In addition, non-linearities introduced into the system compromise gravely the filter's behaviour, which commonly diverges under non-linear conditions. In particular, these conditions imply that achieving a precise data association becomes tremendously challenging, since the identification of the proper landmark is very sensitive to such undesired situations.

The EKF concentrates its operation on the linearization of the probability distributions of its variables. Hence the filter presents serious drawbacks to deal with non-linear systems that reveal non-gaussian noise nature. Unfortunately, this situation corresponds to most of the physical systems in actual science. The linearization adopted by the filter for those systems comes at a cost: a downside on the performance, due to the oversimplification of the distributions involved. Then it is not an occasional fact that its reliability is affected, the final estimation impaired and the ultimate convergence considerably compromised. Nevertheless, despite these last issues, a precise verification of the appropriateness of the system to be run by an EFK solver, ensures an optimum operation of the filter with excellent online estimates. With this purpose, the nature of the input variables and the dynamics of the system has to be studied in depth.

3.2.1 Notation

Mathematically, the principle of this filter lays on the estimation of an augmented state vector which is constantly updated in real time. In the framework presented above, the variables to estimate conform the map itself. So that the state vector may be reformulated in order to include both robot's estimated poses and landmark's. Accordingly, the probabilistic notation for the definitions exposed in Section 3.1 can be transformed to introduce the new augmented state vector for a dynamic system, in the following form:

$$\bar{x}(t) = [x_v, x_{l_1}, x_{l_2}, \cdots, x_{l_i}]^T \tag{3.5}$$

which mixes notations so as to express both, position of the vehicle x_v and landmarks x_{l_i}.

Maintaining the focus on the general probabilistic case introduced above in (3.1), a slight modification in the structure of the equations (3.1) and (3.2) has to be devised, so that the following expressions reveal a linearization for the state transition function, and also consider external inputs such as the control input and noise:

$$P(x_t|x_{t-1}, u_t) \Leftrightarrow x(t) = f(x_{t-1}, u_t) + v_t \tag{3.6}$$

$$P(z_t|x_t, l_i) \Leftrightarrow z(t) = h(x_t, l_i) + w_t \tag{3.7}$$

where f is the function that produces the transition between the previous position of the robot x_{t-1} and the current control input u_t. Similarly, h exposes the relation between the previous position and the position of the i-landmark.

Once the state vector has already been defined in (3.5) and fused through probability notation by the two equations above, the state transition between $\bar{x}(t)$ and $\bar{x}(t+1)$ is:

$$\bar{x}(t+1) = f[\bar{x}(t), u(t+1)] + v(t+1) \tag{3.8}$$

where again f synthesizes the information pertinent to the transition between states and the control vector $u(t+1)$, which normally represents the movement generated by the odometer of the wheels of the robot. Then $v(t+1)$ acts as the gaussian noise introduced in the system, being additive, with zero mean and uncorrelated nature.

Equivalently, a linear relation may be defined so as to couple the observation measurement $z_i(t)$ with the current state vector:

$$z_i(t) = h[\bar{x}(t), l_i] + w_i(t) \tag{3.9}$$

being h the geometric encoding relation between $\bar{x}(t)$, $z_i(t)$ and the observed landmark l_i. Here, $w_i(t)$ represents the random noise generated by the sensors, which is also gaussian, with zero mean and uncorrelated nature. Its covariance is expressed by $R(t)$.

Then, the filter's procedure has to be divided into three indispensable stages which must be well differentiated: State prediction, observation measurement and update. The following subsections provide with a brief introduction to all of them.

3.2.2 State Prediction

Firstly, meanwhile there is no movement in the system between states, a prediction of the state, $\hat{x}(t)$, is carried out, and based on it, also a prediction for the observation measurement, $\hat{z}_i(t)$, may be proposed in the following terms:

$$\hat{x}(t+1|t) = f[\hat{x}(t|t), u(t)] \tag{3.10}$$

$$\hat{z}_i(t+1|t) = h[\hat{x}(t+1|t), l_i] \tag{3.11}$$

$$P(t+1|t) = \frac{\partial f(t|t)}{\partial x} P(t|t) \frac{\partial f(t|t)}{\partial x}^T + Q(t) \tag{3.12}$$

where $P(t|t)$ and $P(t+1|t)$ are the covariance matrices which reflect the increase in the uncertainty of the estimation at instants t and $t+1$ respectively. $Q(t)$ represents the covariance matrix for the noise added by $u(t)$, which specifies the source of noise for the transition. Note that both $Q(t)$ and f are dependent on $u(t)$, which indicates the movement of the robot. $\frac{\partial f(t|t)}{\partial x}$ is the jacobian matrix of f at the estimated state.

3.2.3 Observation Measurement

The second stage performs the real observation $z_i(t)$ at the current instant t, of a specific landmark i of the map. Now the concept of innovation, denoted as $v_i(t)$, has to be introduced in order to explain the deviation between the prior prediction $\hat{z}_i(t)$ and the current measurement $z_i(t)$:

$$v_i(t+1) = z_i(t+1) - \hat{z}_i(t+1|t) \tag{3.13}$$

$$S_i(t+1) = \frac{\partial h(t|t)}{\partial x} P(t+1|t) \frac{\partial h(t|t)}{\partial x}^T (t) + R_i(t+1) \tag{3.14}$$

where $S_i(t+1)$ represents the innovation's covariance that contains the uncertainty in $v_i(t+1)$, as the amount by which the real observation measurement deviates from the prediction. $\frac{\partial h(t|t)}{\partial x}$ is the jacobian matrix of h evaluated at the corresponding state and landmark.

3.2.4 Update

Finally, the third stage takes into account the refinement of the estimation obtained during the prediction, seen as an updating step. Once a successful observation is performed, the state estimation becomes more precise and the uncertainty P decreases. The value of the innovation is significantly relevant in the computation of the final solution provided by the filter. This solution estimation at instant $t+1$, is finally obtained as:

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + K_i(t+1) v_i(t+1) \tag{3.15}$$

$$P(t+1|t+1) = P(t+1|t) - K_i(t+1) S_i(t+1) K_i^T(t+1) \tag{3.16}$$

where in this case $K_i(t+1)$ plays a role of weighting, and corresponds with the gain of the EKF. It is calculated in the following manner:

$$K_i(t+1) = P(t+1|t) \frac{\partial h(t|t)}{\partial x}^T (t) S_i^{-1}(t+1) \tag{3.17}$$

It is worth mentioning that the matrices referred to as the noise's covariance $Q(t)$ and $R(t)$ have to be initialized. $Q(t)$ is established by means of the noise parameters which characterize the odometer of the wheels of the vehicle. Conversely, $R(t)$ is determined by experimental accuracy thresholds which are associated with the visual sensor. The odometry $u(t)$ is required as an initial seed for the prediction generation, together with the previous state, as deduced from (3.10). The uncertainty matrix of the map, $P(t)$, contemplates the noise introduced by the odometry in the form presented in (3.12), and the noise introduced by the visual sensor when carrying out an observation measurement, as detailed in (3.14) and (3.16).

Assuming a linear model for movement and observation usually leads to an EKF estimate that shows a monotonically increasing sequence in its convergence, as long as landmarks are introduced in the map. In the end, its accuracy is determined by the uncertainty associated with the initial state.

On the other hand, non-linear models pose a significant problem which may involve inevitable and often severe inconsistency. Convergence and consistency can be only assured in the linear case. The non-linear case is nonetheless the most common in reality, and thus the case by which most of the physic's phenomenon are closely tied to. This last case is the main consideration taken into account above when the jacobian expressions for $f(t)$ and $h(t)$ were introduced in their linearized forms.

As for the computational side, there is a high dependency on the number of landmarks. Each step requires an update for the whole set of landmark's covariances and estimations. This means that a great computational effort is needed when the environment is large, being the dependency $O(N^2)$ with the number of landmarks N.

3.2.5 Matrix Notation

The equations presented above are more commonly found in matrix notation. Obviously, matrix terms aid in the software programming and computation tasks for real applications. In particular, the matrix for the uncertainty P, defined in (3.12), is truly important. The corresponding P matrix handled in the software development side presents the following structure:

$$P = \begin{pmatrix} P_{x_v x_v} & P_{x_v x_{l_1}} & \cdots & P_{x_v x_{l_i}} \\ P_{x_{l_1} x_v} & P_{x_{l_1} x_{l_1}} & \cdots & P_{x_{l_1} x_{l_i}} \\ \vdots & \vdots & & \vdots \\ P_{x_{l_i} x_v} & P_{x_{l_i} x_{l_1}} & \cdots & P_{x_{l_i} x_{l_i}} \end{pmatrix} \tag{3.18}$$

and transferring the terms into the elements of the matrix:

$$P(t+1|t) = \begin{pmatrix} \frac{\partial f(t|t)}{\partial x} P_{x_v x_v}(t|t) \frac{\partial f(t|t)}{\partial x}^T(t) + Q(t) & \cdots & \frac{\partial f(t|t)}{\partial x} P_{x_v x_{l_i}}(t|t) \frac{\partial f(t|t)}{\partial x}^T \\ P_{x_{l_1} x_v}(t|t) \frac{\partial f(t|t)}{\partial x}^T(t) & \cdots & P_{x_{l_1} x_{l_i}}(t|t) \\ \vdots & & \vdots \\ P_{x_{l_i} x_v}(t|t) \frac{\partial f(t|t)}{\partial x}^T(t) & \cdots & P_{x_{l_i} x_{l_i}}(t|t) \end{pmatrix} \tag{3.19}$$

3.2.5.1 Initializing Landmarks

Another important aspect to keep in mind when dealing with matrix notation is the initialization of new discovered landmarks. The new state vector form is evidently trivial since it is: $x(t)_{new} = [x_v, x_{l_1}, \cdots, x_{l_i+1}]^T$. By contrast P has to initialize the uncertainty of a new landmark with the current value of the robot uncertainty, that is the uncertainty of x_v:

$$P_{new} = \begin{pmatrix} P_{x_v x_v} & P_{x_v x_{l_1}} & \cdots & P_{x_v x_v} \frac{\partial x_{l_{i+1}}}{\partial x_v}^T \\ P_{x_{l_1} x_v} & P_{x_{l_1} x_{l_1}} & \cdots & P_{x_{l_1} x_v} \frac{\partial x_{l_{i+1}}}{\partial x_v}^T \\ \vdots & \vdots & & \vdots \\ \frac{\partial x_{l_{i+1}}}{\partial x_v} P_{x_v x_v} & \frac{\partial x_{l_{i+1}}}{\partial x_v} P_{x_v x_{l_1}} & \cdots & \frac{\partial x_{l_{i+1}}}{\partial x_v} P_{x_v x_v} \frac{\partial x_{l_{i+1}}}{\partial x_v} + \frac{\partial x_{l_{i+1}}}{\partial h} R \frac{\partial x_{l_{i+1}}}{\partial h}^T \end{pmatrix} \tag{3.20}$$

3.3 Stochastic Gradient Descent - SGD

This algorithm is commonly sustained by a graph-oriented map which is supported by the classical statistics concept of a sum-minimization problem. It may be seen as well as a least squares problem where a maximum-likelihood estimator with independent observations has to be conducted [9, 77, 11]. All these aspects come up with certain benefits that reveal a major strength in order to contribute to the mitigation of non-linear effects in a SLAM system [136].

3.3.1 Notation

In contrast to EKF, the SGD is an offline algorithm which tends to produce better results when dealing with non-linearities effects caused by the observation measurements [8, 53, 52, 104]. The EKF turns to be very susceptible to these effects and it often shows serious difficulties to maintain the convergence of the estimation. Nonetheless, this advantage of the SGD comes at an obvious cost of computation, and not to mention that the estimation is obtained offline.

In order to provide a brief approach to the background of the SGD, a light introduction on the structure of its state vector has to be devised similarly to the EKF's section above. However, when dealing with an offline algorithm allows to handle the kind of map differently. Now, it can be seen as a set of nodes defining the poses already traversed by the robot and the landmarks initialized into the map. Generalizing, a set of edges corresponds to the relationships between nodes. These relationships may come either from the odometry of the robot or from the observation measurements. Figure 3.3 represents this general schema where each edge is marked with different error terms. For instance, the edge between the two initial poses at t and $t+1$, represents the error which consists of the difference between the odometry prediction, $g(u_{t+1}, x_t)$, and the real distance, weighted by the noise of the odometry Q by means of its inverse form Q^{-1}. Likewise, nodes where there are observation measurements available, are connected by edges with the error between this observation measurement, z_t, and the

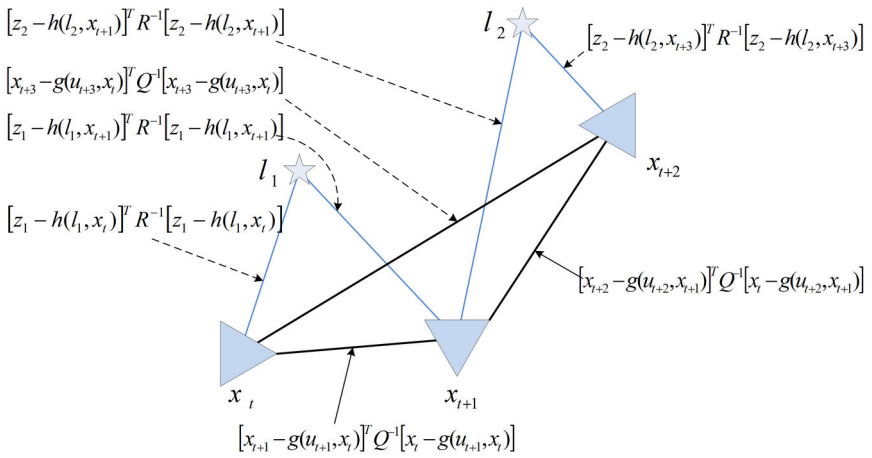


Figure 3.3: This diagram represents a general approach for offline graph methods such as SGD. A set of nodes are included to define both robot's poses and landmarks'. Each node introduces an error term which is determined by the error between the odometry prediction, g, and the distance between nodes, or similarly by the error between the observation measurement to a landmark, z_t, and the prediction based on the state h.

prediction based on the state, determined by $h(l, x)$. In this case, they are weighted analogously by the inverse of the matrix of noise introduced by the sensor, R^{-1}. Finally, the solver algorithm intends to minimize the sum of all these error terms, as the main objective of a Maximum Likelihood Estimation (MLE) method [61]. It is worth noting that the nodes for the robot's pose are referred to as x_t so that nomenclature with EKF is maintained. Nevertheless, SGD is an offline algorithm that presents the potential capability to include any possible relationship between nodes regardless the time. This fact enhances the accuracy of final estimates since high number of constraints are taken into account. Here t may be also devised as a linear index to denote nodes in the graph.

Adapting the former schema to a more practical situation, leads to define a state vector x_t which encodes this representation through a set of variables expressed in the following manner:

$$x_t = \left[(x_0, y_0, \theta_0), (x_1, y_1, \theta_1) \dots \quad (x_n, y_n, \theta_n)\right] \tag{3.21}$$

being now (x_n, y_n, θ_n) the 2D coordinates and bearing in a general reference system for each pose (namely nodes as introduced above). The general idea presented in Figure 3.3, can be specifically extrapolated to the SGD. Hence in the same line, the complementary subset of edges represents the relationships between nodes, by means of either distance measurements generated by the odometry or observations measurements provided by the on board sensors. Here, the nomenclature commonly refers to the measurements as constraints and it denotes them as δ_{ji}, where j indicates the observed node, seen from node i. The general objective stated by methods based on standard

SGD approaches [104, 53, 8] is to minimize the error likelihood expressed as:

$$P_{ji}(x) \propto \eta \exp\left(-\frac{1}{2}(f_{ji}(x) - \delta_{ji})^T \Omega_{ji} (f_{ji}(x) - \delta_{ji})\right) \tag{3.22}$$

being $f_{ji}(x)$ a function dependent on the state, (here the state is expressed as x_t due to encoding considerations) and both nodes j and i. The difference between $f_{ji}(x)$ and δ_{ji} expresses the error deviation between nodes. Such error term is weighted by the information matrix:

$$\Omega_{ji} = \Sigma_{ji}^{-1} \tag{3.23}$$

where Σ_{ji}^{-1} is the associated covariance matrix, which considers the uncertainty of the measurements. The assumption of logarithmic notation in (3.22) leads to:

$$F_{ji}(x) \propto (f_{ji}(x) - \delta_{ji})^T \Omega_{ji} (f_{ji}(x) - \delta_{ji}) \tag{3.24}$$

$$= r_{ji}(x)^T \Omega_{ji} r_{ji}(x) \tag{3.25}$$

being $r_{ji}(x)$ the error determined by $f_{ji}(x)$-$\delta_{ji}(x)$, which shows its condition of residue. Finally, the global problem seeks the minimization of the objective function which represents the accumulated error:

$$F(x) = \sum_{\langle j,i \rangle \in G} F_{ji}(x) = \sum_{\langle j,i \rangle \in G} r_{ji}(x)^T \Omega_{ji} r_{ji}(x) \tag{3.26}$$

$$\tag{3.27}$$

where $G = \{\langle j_1, i_1 \rangle, \langle j_2, i_2 \rangle \ldots\}$ defines the subset of particular constraints conforming the map, either pertaining to odometry or observation measurements. Then, the optimal problem forces to find $x^* = argmin_r[F(x)]$.

3.3.2 Estimation

The SGD algorithm implements an iterative procedure to reach a valid estimation for the SLAM problem. The basis of a SGD method lays on the minimization of (3.26) through derivative optimization techniques such as mean square estimators, so that the estimated state vector is obtained as:

$$x_{t+1} = x_t + \Delta x \tag{3.28}$$

where Δs expresses a certain update with respect to x_t, term which is sequentially generated by means of the constraint optimization procedure. It is worth noting that in a general case, this update is calculated independently at each step by using only a simple constraint, that is to say $\Delta s_n = f(\delta_{ji})$. The general expression for the transition between x_t and x_{t+1} has the following form:

$$x_{t+1} = x_t + \lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji} \tag{3.29}$$

- λ is a learning factor to re-scale the term $H^{-1} J_{ji}^T \Omega_{ji} r_{ji}$. Normally, λ follows a decreasing criteria such as $\lambda = 1/n$, where n is the iteration step. This strategy pretends to achieve a final estimation by using higher values of λ at first steps, and presuming that lower values of λ will be useful in preventing from oscillations around the final solution.

- $J_{ji}(x)$ is the Jacobian of $f_{ji}(x)$ with respect to x_t, that is $J_{ji} = \frac{\partial f_{ji}}{\partial s}$. It translates the error deviation into a spacial variation.
- H is the Hessian matrix, calculated as $J^T \Omega J$, and it shapes the error function through a preconditioning matrix to scale the variations of J_{ji}:

$$H \approx \sum_{\langle i,j \rangle} J_{ji} \Omega_{ji} J_{ji}^T \tag{3.30}$$

- Ω_{ji} is the information matrix associated to a constraint. $\Omega_{ji} = \Sigma_{ji}^{-1}$, being Σ_{ji} the covariance matrix corresponding with the observation constraints δ_{ji}.

This procedure updates the estimation by computing the rectification introduced by each constraint at each iteration step respectively. Despite the fact that the learning factor reduces the weight by which each constraint updates the estimation, at certain scenarios the procedure may cause the estimation to diverge due to undesired oscillations which are interweaved with the stochastic nature of the constraints' selection.

3.4 Gaussian Processes - GP

In the scope of this work, we have concentrated on Bayesian approaches. Gaussian Processes [111] have been traditionally considered as some other kind of these approaches. At first instance, they appeared as an alternative formula to neural networks. Focusing on our framework, we can specify up to the extent of considering GP as a regression technique [146] in order to take the most of data modeling within the field of SLAM. The observation model in a SLAM problem can be assumed as a dataset prior, for which the GP generates a posterior distribution with certain set of weights and hyperparameters rather than a simple induction of an estimation over concrete points.

GP priors outperform neural networks when it comes to analytical treatment, since the last ones cannot be tackled analytically at least in the lowest level of a Bayesian hierarchical model [111]. All together with the work proposed in [100], disseminated the use of these priors to higher complexity scenarios, which have been habitually addressed with other methods such as neural networks or decision trees [71, 18, 96, 144, 68]. The results confirm the positive outcomes of GP in comparison with these other methods.

3.4.1 General Notation

The main purpose is to provide with a brief introduction to the main notation of GPs, which will be later used as a promising tool for determining uncertainty on the matching detection process, and consequently for bounding the uncertainty on the estimated map.

Continuing with the introductory line exposed in the previous section, we can extend the conceptual meaning of a GP to be seen as a predictor that centralizes on priors over functions rather than on computation in the parameter space. Thus GPs are

expected to define a distribution over functions. Then these distributions are updated under the light of the specific training dataset of consideration.

A GP may also be seen as set of random variables, which is fully determined by its mean function $m(x)$ and covariance function $k(x, x_0)$. These parameters are assumed to be structured in the form of a vector and a matrix, respectively. Hence, the distribution is over vectors, meanwhile the GPs itself are over functions. This leads to a formal translation into the following expression:

$$f(x) \sim \mathcal{GP}(m, k) \tag{3.31}$$

From now on, a redefinition of terms has to be introduced so as to differentiate between GP and the former distribution. Thus the former m and k variables shall be renamed as μ and σ^2 for the latter, which represent the mean and covariance values equivalently. Now a random vector from this distribution can be generated. Its coordinates are represented by the function values $f(x)$ for the corresponding inputs $x's$:

$$f(x') \sim \mathcal{GP}(\mu, \sigma^2) \tag{3.32}$$

This last restructuring allows the GP to be employed as a prior for Bayesian inference. Though the prior is not dependent on the training data, but indicates some properties of the functions. As a result, it is necessary to derive the updating process for this prior. Moreover, computing the posterior has also to be devised in order to make predictions for unseen test cases. The following expressions represent the final formulation for the posterior distribution of a specific dataset:

$$f(x) \sim \mathcal{GP}[m(x), k(x, x')] \tag{3.33}$$
$$f(x') \sim \mathcal{N}(\mu, \sigma^2) \tag{3.34}$$
$$\mu = E(f' \mid x, y, x') = k(x', x)[k(x, x) + \sigma_n^2 I]^{-1} y \tag{3.35}$$
$$\sigma = k(x', x') - k(x', x)[k(x, x) + \sigma_n^2 I]^{-1} k(x, x') \tag{3.36}$$

where, x and x' can be seen as the training and test (query) input vectors, respectively, and finally the target data y. $f'(x)$ indicates the output values at the test points.

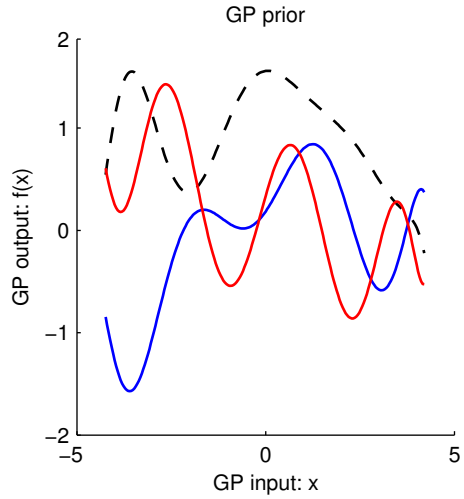
The above expressions manifest the statistical inference that is used to learn dependencies between points in a dataset [40, 41] rather than defining an explicit relationship between inputs and outputs through a function.

3.4.2 Training

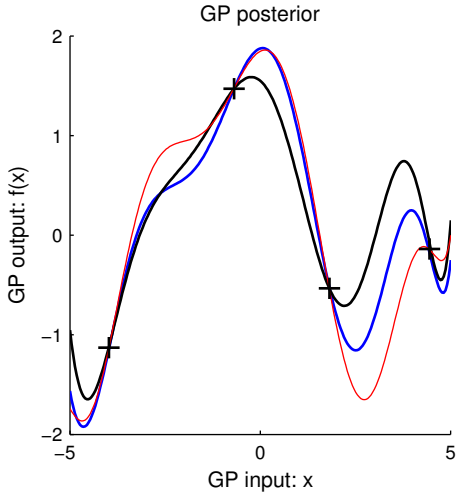
This stage is necessary when there is not enough prior information regarding the dataset in order to determine the prior mean and covariance functions. A lack of knowledge on detailed prior data is the most frequently expected situation in the typical practical case. In order for the GP to succeed, different mean and covariance functions should be tested in the training stage of the GP model. In this sense, the mean and covariance functions are parameterized by means of hyperparameters.

The intent of this assumption is to weight imprecise prior data in a light manner. A simple example is that which shows a known dependency of the function on an n-polynomial order, but without any detail about the polynomial. That's the reason why the inferences over all the hyperparameters is desired. To that end, the probability of the data given the hyperparameters has to be computed. Then logarithmic notation is introduced so as to define a marginal likelihood, which accounts for these hyperparameters. Finally the values of the hyperparameters which optimizes the marginal likelihood are retrieved by means of partial derivatives.

However, the behaviour of this likelihood may differ from a parametric model. Firstly, it is very likely for the model to fit the training data extremely well. Setting a null noise level leads to a mean predictive function which perfectly follows the training points. Note that in Figure 3.4(a), three functions are shown from a random GP prior. The dotted line represents the y values and the colored lines represent a large number of evaluated points. Meanwhile, Figure 3.4(b) presents random functions from a posterior. These are actually priors conditioned on several observations without considering noise, and therefore they match and fit perfectly the data at test points.



(a)



(b)

Figure 3.4: Figure 3.4(a) plots three functions at prior GP randoms. The dotted line shows generated y values, the blue and red lines represent larger set of evaluated points. Figure 3.4(b) plots the three random functions corresponding to the GP posterior. That is the prior conditioned on the four indicated observations with crosses, which are free from noise.

3.5 Information Theory

This section follows the line established in the previous section about GPs. Within the same context of machine learning now we extend it to information theory-related content. Being more specific, the combination of GP and information theory will aid in the modeling of a distribution for the sensor data. This fact is of crucial importance in order to introduce latter contributions on the uncertainty bounding. Therefore this is the main motivation to provide a slight approach to its fundamentals.

3.5.1 Entropy

The entropy is a concept settled in information theory to assess the expected value of the information contained in a certain event within a system. In other words, the information produced from a data distribution. It represents a useful tool that performs a model for the uncertainty of the system itself. Its behaviour follows a simple procedure: the less probability an event has to occur, the more information it provides when it occurs.

This concept has a highly relevant field of applicability within a SLAM system. Uncertainty becomes one of the most important focus to concentrate on when dealing with convergence and consistency. It is known that uncertainty effects usually arise under non-linear conditions, which is the case of study in the framework of this thesis. For this reason, information theory notation has to be defined, in order to count on a helpful tool to assess these effects.

Generally, the entropy is expressed as a negative logarithm of the probability distribution so as to define the information term. This information term jointly with the probability distribution of the events, returns a variable with an average that represents the amount of information, also known as entropy, generated in the distribution stated by [121].

$$H(X) = \mathrm{E}[\mathrm{I}(X)] = \mathrm{E}[-\ln(\mathrm{P}(X))]. \tag{3.37}$$

where E is the expected value operator, and I is the information content of a random variable $X \in (x_1, ..., x_n)$. P represents the probability function.

Similarly for a finite sample:

$$H(X) = \sum_i \mathrm{P}(x_i)\,\mathrm{I}(x_i) = -\sum_i \mathrm{P}(x_i)\log_b \mathrm{P}(x_i) \tag{3.38}$$

3.5.2 Information Gain

In this same context, the information gain, also known as Kullback-Leible divergence [74, 75], plays an important role for determining the uncertainty within a SLAM system. More specifically, the expected value of the information gain is also known as the mutual information of a system. And technically, the definition for the expected

information gain maintains that it is a change in the information entropy, H, from a prior state to the following as:

$$IG(A,B) = H(A) - H(A|B) \tag{3.39}$$

where $H(A|B)$ is the conditional entropy. In accordance to this, the conditional entropy shall be defined between two events A and B, taking values a_i and b_j respectively as:

$$H(A|B) = \sum_{i,j} p(a_i, b_j) \log \frac{p(a_j)}{p(a_i, b_j)} \tag{3.40}$$

where $p(a_i, b_j)$ is the probability that A equals a_i and B equals b_j. This term is essential to asses uncertainty within a random variable, given that a certain value for such variable has been previously known.

One of the most straightforward examples of application of information theory to the SLAM framework is the Extended Information Filter (EIF). This filter is a conversion of the EKF which seeks to characterize the system from an information point of view [134, 65, 142]. Analytically, the EKF and the EIF might be seen as the same, since the conversion applies to the following parameters as follows:

- $\sum = \sigma^{-1} \Rightarrow \sigma = \sum^{-1}$
- $\mu = \sigma^{-1}\epsilon \Rightarrow \epsilon = \sum^{-1} \mu$

As a general convention, the usual notation for the covariance matrix and the mean vector for EIF are translated to canonical notation.

Some other considerations have to do with efficiency. Although the EIF is analytically equivalent to the EKF, it shows a more efficient update with respect to the EKF, but at cost of a slower prediction [15]. The inverse covariance matrix simplifies the procedures to work with, since it is usually sparser than the covariance matrix [141].

3.6 Conclusions

In this chapter we have addressed an overall overview to the theoretical backbone sustaining the operation of a SLAM system. In this sense, within the framework of this thesis, we have concentrated on the analytics of the main algorithms and methods involved in the proposal and implementation of new contributions. As a result, we have approached to the background of the Bayesian nature of the SLAM problem. Subsequently, we have detailed the essentials of its kernel algorithms such as EKF and SGD. It is worth noticing the main difference that exists between such techniques. We have concentrated at first instance on the EKF, due to the fact that it provides a valid framework for online SLAM approaches. Conversely, the SGD entails an offline technique, which offers certain benefits against system instability and non-linear effects, such as those rather likely to appear on the EKF side. And finally, we have also provided

with a brief introduction to regression techniques and information theory such as GP and information-based aspects, as they are greatly profitable in this thesis to come up with contributions that enhance the uncertainty bounds of our SLAM system.

As a summary, this chapter represents the basic frame structure to describe and sustain all the theoretical framework for the later development of the approaches proposed in this thesis and all the specific implementations associated with the publications presented. In consequence, it will be necessary and repeatedly referenced along this document in order to list and detail all the theoretical side of the research conducted under the context of this thesis.

4 EKF-based SLAM Contributions

The purpose of this chapter is to present all the details regarding the major contributions made in this thesis in terms of the implementation of a specific SLAM approach. More specifically, our proposal consists of an EKF-based visual SLAM approach that we have intentionally customized to embed our omnidirectional camera system. This implementation aims to fit in the field of application of visual SLAM with single cameras. As a result, a new map model is proposed. To date, most of the work done in the visual SLAM framework has been supported by EKF developments which have dealt with the estimation of a set of 3D visual landmarks, such as [27, 26, 25, 20], where those landmarks are actively discovered with a monocular camera, or with stereo systems [43, 44, 105], or even throughout artificial environments, for both planar and catadioptric camera sensors in [67, 24] and [58] respectively. Generally, the utilization of EKF implies that computation efforts might increase considerably in large scenarios due to the continuos re-estimation. Thus this issue is extended to the dimension of the map and to the complexity of the entire process, which finally becomes critical.

Conversely, in this thesis we suggest a different representation of the environment in order to simplify the computation of the map and to provide a more compact representation of the environment. Particularly, the map is sustained by a reduced set of omnidirectional images, denoted as views, which are acquired at certain poses of the environment. The information gathered by these views allows to encode it for large environments, and at the same time they ease the observation process by which the pose of the robot is retrieved. Moreover, we embed in this EKF-based visual SLAM approach the concept proposed in Chapter 2 for matching feature points based on the adaption of the epipolar geometry to the scope of omnidirectional images. This contribution is enhanced by the integration of the current uncertainty at every EKF iteration

step. Its basis relies on the generation of a gaussian distribution which propagates the current error and the uncertainty to the the matching process. As a result, this consideration produces an improved matching procedure. It certainly becomes more robust and presents a trustworthy capability for mitigating the troublesome issue of finding correspondences in a non-linear system affected by noise.

As a summary, we can list the following features and contributions that are going to be illustrated throughout this chapter:

- Proposal of a new representation of the environment: map model based on a reduced set of omnidirectional views.
- Essentials of the map building process: exploiting the potential of our omnidirectional system. The design of the observation model takes the most of the definitions proposed in Chapter 2.
- Enhancement of the observation model by means of a redesigned matching process, through the propagation of the uncertainty of the system.
- Experimental results: validating the appropriateness and suitability of this approach to deal with real data in a real scenario and application.

4.1 Map Building

Here we address the main purpose of a visual SLAM scheme, which basically entails the retrieval of a feasible representation of the environment explored by the robot, as well as the position of this vehicle. In this approach, the map of the environment is defined by a set of omnidirectional images acquired from different poses of the robot along the environment, denoted as views. These views do not express information about any physical landmarks as it might have traditionally expected in the field of vision-based SLAM. By contrast, a view n consists of a single omnidirectional image captured at a certain pose of the robot $x_{l_n} = (\mathrm{x}_l, \mathrm{y}_l, \theta_l)_n$ and a set of interest points extracted from that image. Such arrangement, allows us to exploit the capability of an omnidirectional image to gather a large amount of information in a simple image, due to its large field of view. Thus, an important reduction is achieved in terms of the number of variables to estimate the solution.

The position of the mobile robot at a certain time, t, is denoted as:

$$x_v = (\mathrm{x}_t, \mathrm{y}_t, \theta_t)^T \tag{4.1}$$

Each view n is constituted by the pose where it was acquired, with $n \in [1, \ldots, N]$, being N the total number of views constituting the final map. Then the view is represented by its pose as:

$$x_{l_n} = (\mathrm{x}_l, \mathrm{y}_l, \theta_l)_n^T \tag{4.2}$$

together with its uncertainty P_{l_n} and a set of m feature points p_{n_m}, expressed in image coordinates. Each point is associated with a visual descriptor d_m.

Therefore, according to (3.5) these are the variables which compose the augmented state vector:

$$x = \begin{bmatrix} x_v & x_{l_1} & x_{l_2} & \cdots & x_{l_N} \end{bmatrix}^T \tag{4.3}$$

where $x_v = (\mathrm{x}_t, \mathrm{y}_t, \theta_t)^T$ is the pose of the moving vehicle and $x_{l_N} = (\mathrm{x}_{l_N}, \mathrm{y}_{l_N}, \theta_{l_N})$ is the pose of the last view N that exists in the map. Please note that the index for the number of views is n, but we can refer to it as N, in case that we need to indicate that the entire map is conformed by a total number of N views. Therefore, the state vector definition sustains the new map representation, which basically consists of the current pose of the robot $x_v = (\mathrm{x}_v, \mathrm{y}_v, \theta_v)^T$ at each t and the location of the set of views, which can be indexed as $x_{l_n} = (\mathrm{x}_{l_n}, \mathrm{y}_{l_n}, \theta_{l_n})_n^T$.

As for the EKF-based implementation, we have dealt with the adaption of a new omnidirectional observation model which relies on a new map building process in terms of view initialization but also in terms of data association.

In this sense, the map building task is depicted in Figure 4.1 by a real example of operation which clarifies the explanation and simplifies its comprehension. It can be observed how the robot starts exploring the environment at the origin point A, where it captures an omnidirectional image I_A, stored in the map as a view with pose x_{l_A}. This view is representative of the relevant visual information around this position. Now I_A is assumed to be the first part of the map. Then the robot moves towards the first office room. As long as there is not any major obstruction, the robot extracts corresponding points between I_A and the omnidirectional image at its current pose, x_v. This fact makes the robot able to localize itself. However, once the robot enters the office room, the appearance of the images varies significantly and consequently less matches are likely to be found on I_A. At this point, there is some uncertainty to extract a reliable localization. As a result, the robot needs to consciously initialize a new view named I_B at the current robot's position x_{l_B}. Otherwise, the situation might come to the extend that there is not any correspondence extracted, and then the robot would be only driven by the control inputs and the internal odometer, being unable to localize itself by means of the information provided by the feature matching procedure.

Afterwords, the view x_{l_B} aids the robot in localizing itself inside the office room. Finally, the robot concludes the exploration of the environment with an accumulated map defined by views I_C, I_D, I_E. The number of views initialized in the map directly depends on the sort of environment and its visual appearance. Figure 4.1 also provides a synthesis of the localization procedure. A comparison between I_A and I_E is presented, where corresponding points and the motion transformation given by the relative angles between the pose of the images are indicated.

4.1.1 View Initialization

Once the basis of the map building process has been introduced, the view initialization stage requires a more detailed explanation. Inclusion of new views in the map have to come at a trade-off solution. Obviously, the more views stored in the map the

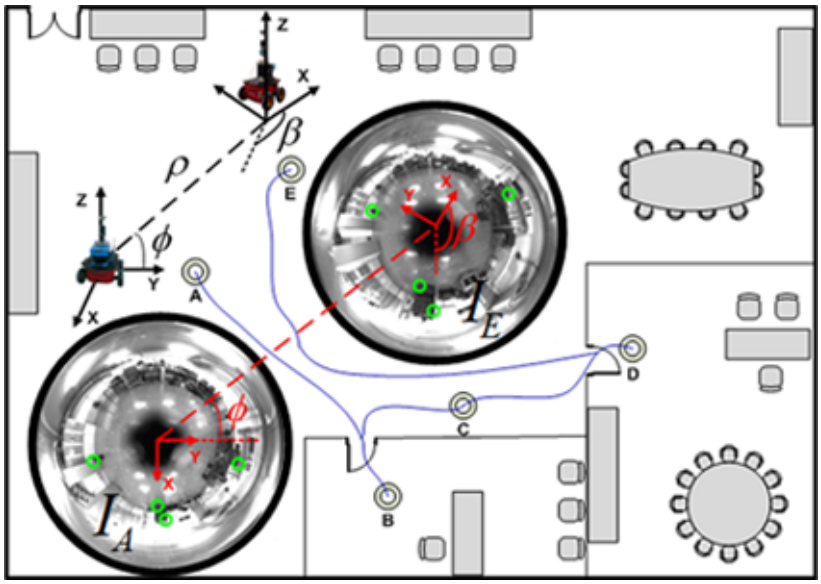


Figure 4.1: Map building process. First view in the map, I_A, is initialized at the origin A, namely pose x_{l_A}. While the robot traverses the environment, correspondences may be found between I_A and the current image captured at the current robot's pose x_v, so that the robot can extract its location. In case there is not any correspondence found, a new view is initialized using the current image, for instance I_B at point B, namely pose x_{l_B}. The procedure finalizes when the entire environment is represented.

more accurate estimation. On the other hand, computation and memory requirements may generate an overdimensioned approach which fails to be scalable and feasible in real applications. The main concern of this work is to limit the view initialization in parallel with the uncertainty on the estimation. This idea pursues that high values of uncertainty, which typically make the approach to diverge, can trigger the initialization of a new view in the map. Thus new observation measurements can be performed and consequently the uncertainty bounded.

The first strategy tackles this aspect by assessing a pseudo-appearance ratio. More precisely, a new view is initialized in the map whenever the number of corresponding feature points between images do not surpass a certain threshold. Our first approach to this idea [135] relies on this relative measurement between images so as to define an initialization ratio. This ratio was experimentally defined as:

$$A = k\frac{c}{p_1 + p_2} \tag{4.4}$$

being p_1 and p_2 the feature points detected on each image and c the correspondences between them. The value of k was aimed at weighting the measurement according to the visual appearance of each particular scenario. Note that this aims to define a threshold which assesses the similarity of the visual appearance between two views, but with special relevance on the number of matches. Please note that we referred to it

as a pseudo-appearance measurement, and similarity ratio, since it does not use any appearance-based technique such as [37]. It can be definitely seen as a strategy only based on feature point information.

As mentioned above, the ratio A represents a measure of similarity and it is the factor which eases the robot to decide whether to initialize a new view in the map. High values of A mean that the current robot's image is similar enough to some view in the map, so is not necessary to initialize a new one. On the contrary, if A drops below a certain threshold, the similarity is supposed to be low and also the robot's capability to localize itself feasibly. Moreover, the uncertainty expected at this point should be harmfully high for the system to maintain convergence. Therefore, the most straightforward solution is to acquire a new view in the map.

4.1.2 Observation Model

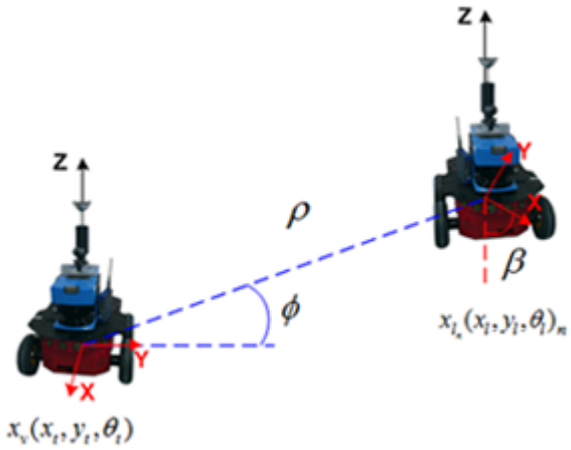
In accordance with the view-based representation presented above, the next stage to formulate is a new observation model. Thanks to the versatility of omnidirectional images, we can apply epipolar constraints [60] which allows to determine a motion transformation, and finally at last instance, to extract an observation measurement. This enables the development of a procedure that determines the motion transformation between two poses, as we presented in Section 2.3.1. This concept can be noticed in Figure 4.2. For further detail, a quick look at Chapter 2 provides complete comprehension about the formal procedure implemented in this work in order to retrieve the motion transformation between views. This procedure exploits the visual relation between views under the epipolar geometry context. In fact, these poses represent the positions where the robot acquired two images. To that effect, only two images with a set of corresponding points between them are required to obtain the transformation. As a result, the observation measurement may be expressed as:

$$z_t = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{\mathrm{y}_{l_n} - \mathrm{y}_t}{\mathrm{x}_{l_n} - \mathrm{x}_t}\right) - \theta_t \\ \theta_{l_n} - \theta_t \end{pmatrix} \tag{4.5}$$

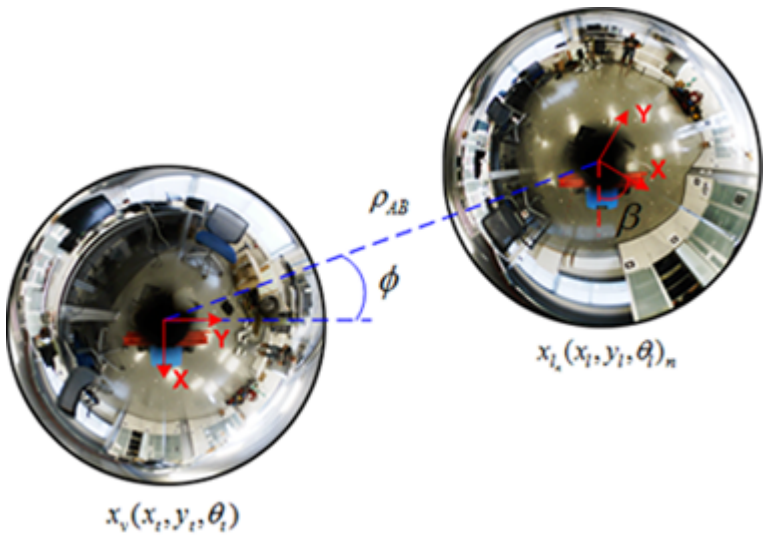
where ϕ and β are the relative angles which express the bearing and orientation at which the view n is observed from the current robot's pose x_v. Please note that the structure of the view n was presented as $x_{l_n} = (\mathrm{x}_l, \mathrm{y}_l, \theta_l)^T_n$, whereas the pose of the robot is given as $x_v = (\mathrm{x}_t, \mathrm{y}_t, \theta_t)^T$. Figure 4.2 graphically exposes the meaning of these measurements (ϕ, β) on the image frame. The motion transformation that this observation model provides us can be noticed in Figure 4.2(a), expressed in the spatial reference system of the robot, whereas Figure 4.2(b) represents the same transformation expressed on the image reference system.

4.1.3 Data Association

The data association problem is posed in the following manner: given a set of observations $z_t = [z_{t_1}, ..., z_{t_B}]$ at each t, the views which generate each observation have to be discerned. In the approach presented here, the data association process is tackled



(a)



(b)

Figure 4.2: Observation model variables: Figure 4.2(a) represents the motion transformation between the pose of the robot x_v and a certain view x_{l_n}. Similarly, Figure 4.2(b) depicts the same transformation represented on the image frame of the two views acquired at x_v and x_{l_n}. The relative angles of the transformation are indicated as ϕ, β and the unknown scale factor ρ. Corresponding points between images are shown by green circles.

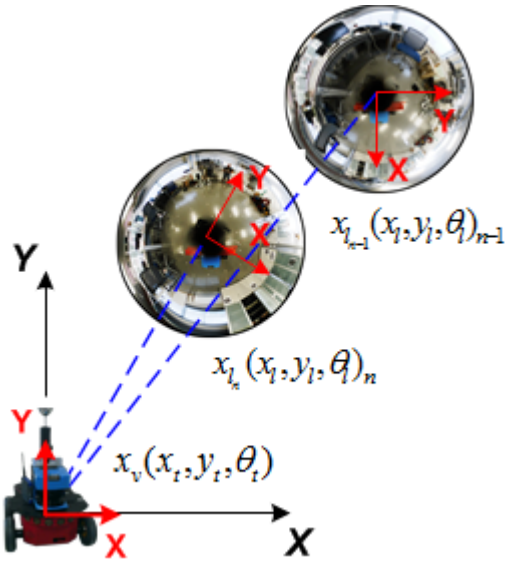


Figure 4.3: Multiple data association with low parallax.

through the computation of the similarity ratio A, described by (4.4). However, before computing A between views, we need to select a subset of candidate views from the map, based on the euclidean distance between the pose of the current view acquired by the robot, that is, at the current robot's pose, and the position of each candidate. Then this metric can be posed as $D_n = \sqrt{(x_v - x_{l_n})^T (x_v - x_{l_n})}$, where the notation corresponds with (4.1) and (4.2).

Once D_n has been defined, we can establish a maximum observation range at which the robot is capable of performing proper observation measurements. More distinctly, it represents the maximum distance at which any view can be observed at each t. Consequently, this maximum range also determines the enablement of the comparison between views in the map and the current view at the robot's pose. Hence it also implies the same assumption for the feature points detection, matching process and in general, motion transformation computation.

Therefore, after extracting the set of candidates, we can now extract corresponding points between the image acquired at the current pose of the robot and the rest of the candidate views. This allows to set up a new stage at which the similarity ratio A (4.4) is evaluated. In this sense, the view which provides the maximum similarity ratio A is eventually chosen as the data association. The view with maximum A reveals the highest similarity with the current image. However, if none of the candidate views provide a value for A higher than a predefined threshold, this will mean that the appearance of the current image of the robot differs substantially from the set of candidate views, which ultimately does not encode nor represents the visual information of the area where the robot currently moves around. Thus it will be necessary to initialize a new view into the map at the current robot's position.

Please note that multiple data association can be performed by simply selecting a desired number of candidates which surpass certain value of A. We have not dismissed this option, however, some inconveniences may arise. One of the most usual handicaps we dealt with is the low parallax error. Figure 4.3 exposes such situation, where two candidates for the data association could be both assumed as valid, however the low parallax is very likely to introduce uncertainty and inconsistency in the latter stage of motion transformation computation. It is worth noticing that our observation model is entirely angular-based, so we believe it is recommended to avoid this issue by establishing a threshold angle which prevents from low parallax.

It is also necessary to highlight that the data association is vital within the SLAM problem. It has a high relevance on the convergence of the system. This task may become troublesome in the presence of considerable high non-linearities in the observation. Also under certain circumstances as when there is at short distance between landmarks and it provokes difficulties to distinguish them. Some work concentrated on this issue [89], [82]. Conversely, here we define an efficient visual approach based on the novel representation of the map sustained by views.

As a synthesis of this section, we depict the entire data association and the view initialization subprocesses with a summarized pseudocode in the terms expressed by Algorithm 2. It must be noticed that this algorithm returns the view candidate with highest value of A, however we can easily modify it to return more than one view. The modification simply implies that these views surpass a certain value of A, which will be specific for the environment. As a result we could perform several observation measurements to more than one view at the same time step within the SLAM system. To conclude with this synthesis, Figure 4.4 presents a diagram with the integration of the data association and the view initialization within the entire visual SLAM approach implemented in this work.

4.1.4 Enhanced Matching

Matching exact feature points between images is crucial for retrieving a reliable motion transformation, and consequently, a consistent observation model. In this sense, as we have mentioned above, we formerly designed in Chapter 2 the schema for computing the motion transformation between two poses of the robot by means of certain views acquired at those poses. Now we move forward to propose a fused implementation. The goal is to take advantage of the epipolar constraints and to introduce several uncertainty considerations which help us improve the matching process. This enhanced model allows to reinforce the matching of feature points since it delimits the search for correspondences. We achieve this by computing the expected epipolar lines with the proper uncertainty deviations. This has been materialized thanks to the potential advantages provided by the EKF to predict the next state. Nevertheless the key point of this implementation is determined by the consideration of the uncertainty at every EKF step, that is at $t + 1$. As a result, the most important outcome of this idea is a more robust matching process which prevents from false imparity but it also relaxes the search for points in the second image in terms of computation.

Algorithm 2 Data Association algorithm

Require: Inputs

$x_{l_n} \in x(t)\ \forall\ n$, where $x(t)$=[(x_v, y_v, θ_v), $(x_{l_1}, y_{l_1}, \theta_{l_1})$, ..., $(x_{l_N}, y_{l_N}, \theta_{l_N})$]

Candidates: Set of candidate views accomplishing the requirements of visual appearance.

Dassoc: Views accomplishing data association (maximum similarity ratio A).

d_{max}: Maximum distance at which views are observed.

p_1: feature points at $x(t)$.

for i=1:N **do**

  $D_i = \sqrt{(x_v - x_{l_i})^T (x_v - x_{l_i})}$

  if D_i<d_{max} **then**

    New candidate to the subset:

    $Candidates$=[$Candidate_1$, $Candidate_2$, ..., $(x_{l_i}, y_{l_i}, \theta_{l_i})$]

  end if

end for

for j=1:length($Candidates$) **do**

  Extracting feature points p_2 of view candidate $Candidate_j$

  if A_j=$k\ \frac{c}{p_1+p_2} = max$ **then**

    $Dassoc$=[$Candidate_j$]

  end if

end for

return *Dassoc*

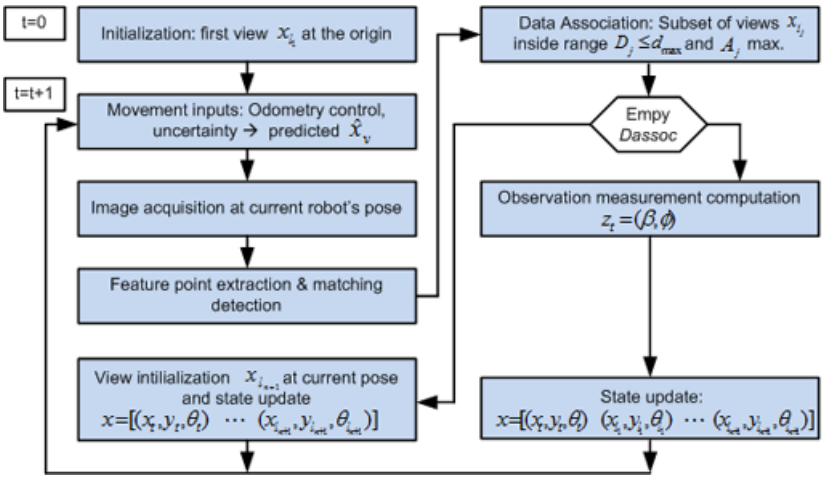


Figure 4.4: Block diagram of the visual-based EKF approach.

The procedure for computing the motion transformation entails the notation $p = [x, y, z]^T$ and $p' = [x', y', z']^T$ for the same point detected from two different camera points of view. That is p and p' are matched points at two different views. Then, the epipolar condition we use to state the relationship between both 3D points p and p' seen from different views is:

$$p'^T E p = 0 \tag{4.6}$$

where the matrix E is the essential matrix and it can be computed from a set of corresponding points in two images.

$$E = \begin{bmatrix} 0 & 0 & \sin(\phi) \\ 0 & 0 & -\cos(\phi) \\ \sin(\beta - \phi) & \cos(\beta - \phi) & 0 \end{bmatrix} \tag{4.7}$$

being ϕ and β the relative angles that determine the planar motion transformation (4.5), as it may be observed either in Figure 4.1 or Figure 4.2.

The fundamentals for the enhanced matching proposal rely on the information provided by the EKF. Its prediction stage presented in (3.2) aids to devise a realistic search for valid corresponding points between images. In an idealistic case, the epipolar constraint defined in (4.6) should equal a fixed threshold, very close to zero, which it only implies that the epipolar curve defined between images might present a little static deviation. However, in our approach we consider the propagation of uncertainties in the map into (4.6) by introducing a dynamic threshold. This implies a more realistic SLAM approach, since this threshold depends on the existing error on the map, which dynamically varies at each step of the SLAM algorithm, so that it eventually defines the current uncertainty in the system.

Notice that the avoidance of false correspondences has been studied extensively so as to mitigate bad effects on the final estimation for the SLAM problem. For instance, techniques such as RANSAC [19] and histogram voting [108] have been widely used, and mainly applied to visual odometry approaches [116]. These examples are focused on the epipolar constraint (4.6), and they eventually reveal good results in the achievement of false positive rejection. All of them can be labeled under a context of visual odometry, with consecutive images are close enough to disregard high errors in the pose from where the images are taken. They finally conclude with good results since under these circumstances the epipolar constraint is highly likely to be satisfied [117].

Contrarily to the last examples, concentrating on the framework of our SLAM problem, the accumulative uncertainties are substantially higher, either in the pose of the robot or in the pose of the views which compose the map. This fact requires to define a reliable strategy to accomplish with a correct data association. Thus we bring into focus the information provided by the predicted state vector extracted from the EKF, by which we are able to obtain a predicted observation measurement $\hat{z}_t$, with the same structure stated in (4.5). Then it is also necessary to consider the current

map uncertainties so as to deal with a realistic search for valid corresponding points between images. We can propagate the map uncertainties in accordance with (4.6) by introducing a dynamic threshold δ. Here we consider δ dependent on the existing error on the map, which dynamically varies at each step of the SLAM algorithm. Since this error is correlated with the error on $\hat{z}_t$, we rename δ as $\delta(\hat{z}_t)$ to express such dependency. In addition, it has to be noted that (4.7) is defined up to a scale factor, which is another reason to keep $\delta(\hat{z}_t)$ as a dynamic value. Therefore, given two corresponding points between images, they must satisfy:

$$p'^T \hat{E} p < \delta(\hat{z}_t) \tag{4.8}$$

This approach not only mitigates the undesired harmful effects associated with false positives, but also simplifies the search for corresponding points between images as it restricts the area where correspondences are expected. The procedure is depicted in Figure 4.5, where a detected point in 3D is assumed, $P(x, y, z)$, and represented in the first image reference system by a normalized vector $\vec{p_1}$ due to the unknown scale. To deal with this scale ambiguity, we suggest to introduce a point distribution to generate a set of multi-scale points $\lambda_i \vec{p_1}$, being representative for the lack of scale in $\vec{p_1}$. This distribution considers a valid range for λ_i according to the predicted $\hat{\rho}$. Please note that the error of the current estimation of the map has to be propagated along the procedure. To that end, we look back into the Kalman filter theory, where the innovation is defined as the difference between the predicted $\hat{z}_t$ and the real z_t observation measurement as stated in (3.13), and the covariance of the innovation in (3.14). Hence $S_i(t+1)$ presents the following structure:

$$S_i(t+1) = \begin{bmatrix} {\sigma_\phi}^2 & \sigma_{\phi\beta} \\ \sigma_{\beta\phi} & {\sigma_\beta}^2 \end{bmatrix} \tag{4.9}$$

If we extend the notation to transform terms into predicted terms, the predicted $\hat{E}$ can be decomposed in a rotation $\hat{R}$ and a translation $\hat{T}$. Next we make use of these transformation tools in order to transform the distribution $\lambda_i \vec{p_1}$ into the second image reference system, obtaining $\vec{q_i}'$. The introduction of (4.9) allows to propagate the error, and thus it redefines a transformation between images through the normal distributions $\hat{R} \sim \mathrm{N}(\hat{\beta}, \sigma_\beta)$ and $\hat{T} \sim \mathrm{N}(\hat{\phi}, \sigma_\phi)$. Therefore $\vec{q_i}'$ is a gaussian distribution correlated with the current map uncertainty. Once obtained $\vec{q_i}'$, they are projected into the image plane of the second image, seen as circled points in Figure 4.5. This projection of the normal multi-scale distribution determines the predicted area. This area is drawn in blue on the second omnidirectional image. In other words, the epipolar curve defined in Chapter 2 becomes an elliptical area due to the consideration of the uncertainty. This area establishes the specific image pixels where correspondences for $\vec{p_1}$ must be searched for. The shape of this area depends on the error of the prediction, which is directly correlated with the current uncertainty of the current map estimation. Dashed lines represent the possible candidate points located inside the predicted area. Hence the problem of matching is simplified. Now it consists of searching for the correct corresponding points for $\vec{p_i}$ amongst those candidates inside a restricted area, instead of a global search along the whole image.

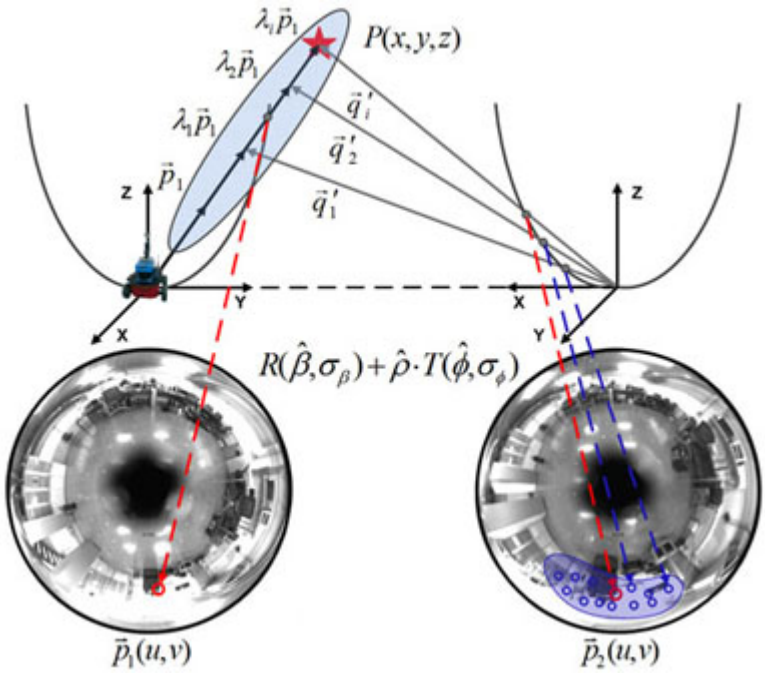


Figure 4.5: Given a detected point $\vec{p_1}$ in the first image reference system, a point distribution is generated to obtain a set of multi-scale points $\lambda_i\vec{p_1}$. By using the EKF prediction, they can be transformed into $\vec{q_i}'$ on the second image reference system by means of epipolar geometry with a rotation $R\sim\mathrm{N}(\hat{\beta},\sigma_\beta)$, translation $T\sim\mathrm{N}(\hat{\phi},\sigma_\phi)$ and scale factor $\hat{\rho}$. Finally, $\vec{q_i'}$ are projected into the image plane to determine a restricted area where correspondences have to be found. The circled points represent the projection of the normal point distribution for the multi-scale points that determine this area.

Figure 4.6 shows the transformation suffered by the epipolar curve on the the image plane, which now reveals an elliptical shape, as a consequence of the intersection generated by an epipolar plane with the hyperbolic mirror. Notice that due to the propagation of the error, now the epipolar plane varies its position within a range determined by the gaussian distribution of $\vec{p_i}$ and $\vec{q_i}'$. As a result the intersection with the mirror produces an elliptical area.

Finally, Figure 4.7 presents a diagram with all the stages of the enhanced matching model. Note that this model is embedded within the feature and matching process shown in Figure 4.4.

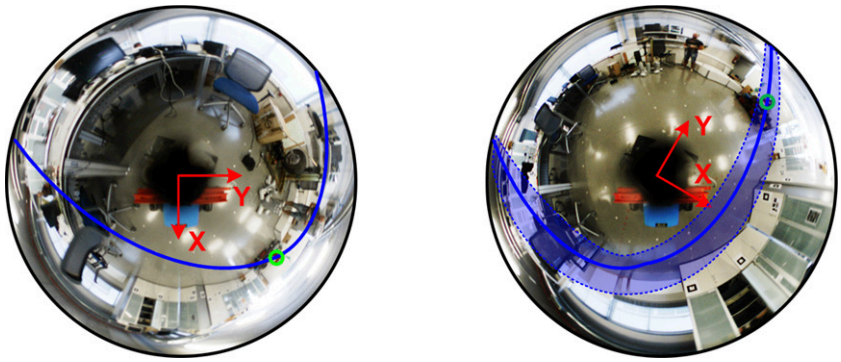


Figure 4.6: Transformation of the epipolar curve into an elliptical area as a consequence of the propagation of the current uncertainty of the map estimation. A point in the first image lies on the epipolar line. In the second image it also lies on the epipolar line, which is inside the elliptical area predicted by means of the uncertainty propagation.

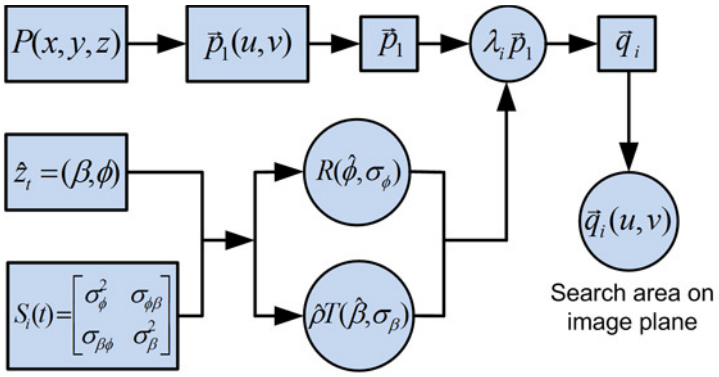


Figure 4.7: Block diagram of the enhanced matching model.

4.2 Results

In order to confirm the suitability and effectiveness of the approach exposed in this chapter, we present a series of real data experiments conducted with the Pioneer P3-AT robot equipped with the camera DFK-41BF02 and the hyperbolic mirror $Eizoh$ Wide 70 as the basis of the catadioptric system. The visual SLAM approach is run in the backend by an EKF-based algorithm which is solely sustained by the visual information gathered through the omnidirectional views. The entire configuration, physics and specifications of this equipment have been presented in Section 2.2.3 of Chapter 2.

Besides, in order to obtain a reference for comparison we use a SICK LMS range finder to generate a ground truth [127, 51], which provides a resolution of 1m in position. As for the odometry, this has been acquired by the odometer of the P3-AT robot. Its parametrization has been presented by Algorithm 1 in Section 2.2.3.

We present three different experimental datasets:

- Simulation dataset. Here we intend to evaluate the appropriateness of the approach. Simulated environments imply a first assessment step to confirm the theoretical proposals and hypothesis.
- Real dataset. With this experiments we verify the validity of the approach to provide reliable results in real data scenarios. We present results for the estimated map and for the pose of the robot, as well as the evolution of the error at every time instant. We varied the value of different parameters in order to study the dependencies on the compactness of the representation with the dimension of the estimated map.
- Finally we quantify the efficiency and accuracy of the final estimation provided by this approach.

4.2.1 Simulation Dataset

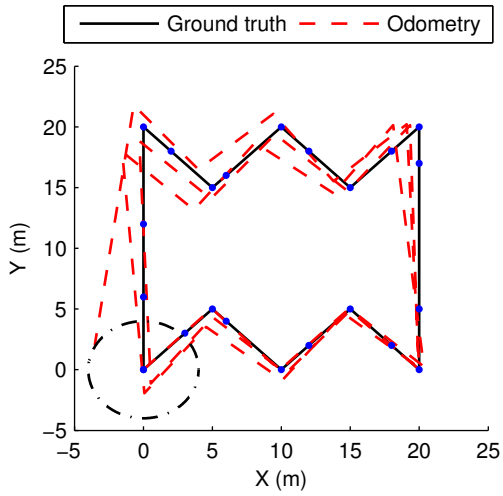
First of all we would like to highlight the importance of assuring the convergence of an EKF-based SLAM algorithm when a new observation model is introduced. This aspect is not of trivial assertion due to the fact that the EKF tends to be gravely affected by noise and non-linearities' effects, such as those introduced by an omnidirectional sensor. For this reason, ensuring convergence is of paramount significance in this approach. To that purpose, we present preliminary results of two simulated scenarios, as detailed in Figure 4.8 and Figure 4.9 respectively.

The first scenario merely consists of a random trajectory traversed by the robot, which runs the implementation proposed in this work in order to correct the initial input of the odometry. Such odometry is obtained with the same model described in Section 2.2.3. In Figure 4.8(a), the continuous line represents this random trajectory, whereas the odometry has been plotted with dashed line. A set of views have been placed randomly along the trajectory and shown with blue dots. Please note that

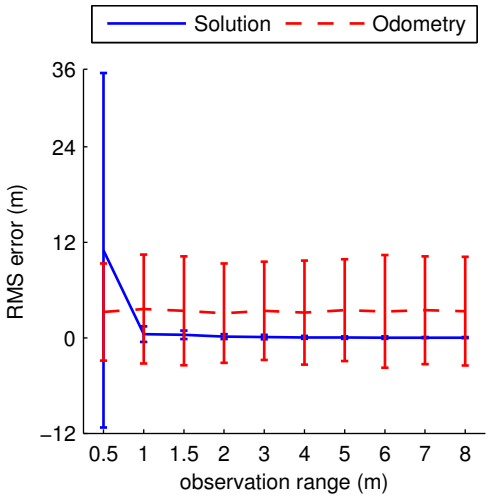
the arrangement of the views depends on the randomization of the similarity ratio A, described in (4.4). Every time the robot moves, it assesses a random value of A and initializes a new view whenever A<0.3. As for the observation measurements to these views, $z_t(\phi, \beta)$, its generation is also randomized with an added gaussian noise of $\sigma_\phi = \sigma_\beta = 0.2rad$. The dashed circle represents the maximum range at which the robot is able to perform observation measurements to the views in the map. These two variables, the observation range and A, represent the basic parameters to tune and modify in order to randomize the experiment.

Figure 4.9(a) presents another simulated environment that emulates a typical indoor environment, where the computation of the observations is restricted by certain obstructions and obstacles such as walls. Again, we vary the observation range of the robot so as to analyze the relevance of the number of views observed.

The results for the RMS error with the observation range are presented in both scenarios by Figure 4.8(b) and Figure 4.9(b) respectively. Standard deviation and 2σ values are included, since the experiments were repeated 100 times with random datasets. The continuous line shows the mean evolution of the RMS error for the proposed approach, meanwhile the dashed line shows the same evolution in the odometry. Note that the only observation range for which the solution diverges is 0.5 m, since it is too short to observe any views in such scenario. Despite this exception, the rest of observation range values produce acceptable results in terms of error. It is worth paying attention to the following figures' output: the higher observation range, the more number of views observed. Thus the more accurate observation measurement, the more accurate estimation. Nevertheless, an excessive number of views does not necessarily imply a significant relevance on the accuracy of the solution, as the error tends to stabilize. In the following subsections we present an analysis that assesses the relevance of the number of views with the accuracy and the efficiency of the approach in terms of time. Overall, we can confirm the preliminary suitability of this approach to converge to a proper estimation with a reduced error that reveals to outperform the odometry estimation.

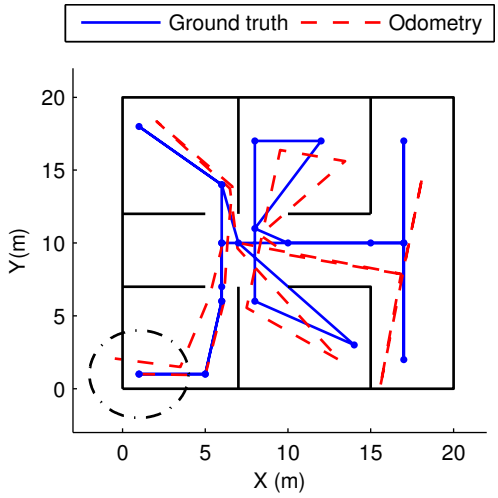


(a)

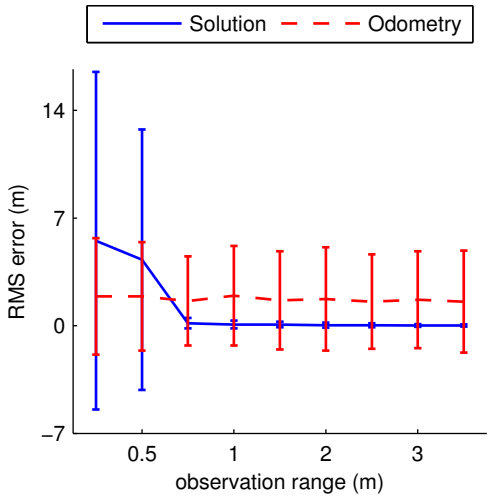


(b)

Figure 4.8: Results obtained in the first simulated scenario over 100 repetitions. Figure 4.8(a) shows the ground truth in continuous line and the odometry in dashed line. The location of the views that conform the final map is indicated by blue dots and the observation range by a dash-dotted circle. Figure 4.8(b) represents the variation of the RMS error on the estimation against the observation range of the robot. The continuous line represents the mean error on the estimation and the dashed line the mean error on the odometry.



(a)



(b)

Figure 4.9: Results obtained in the second simulated scenario over 100 repetitions. Figure 4.9(a) shows the ground truth in continuous line and the odometry in dashed line. The location of the views that conform the final map is indicated by blue dots and the observation range by a dash-dotted circle. Figure 4.9(b) represents the variation of the RMS error on the estimation against the observation range of the robot. The continuous line represents the mean error on the estimation and the dashed line the mean error on the odometry.

Dataset characteristics				
Dataset	**No. images**	**Distance**	**Figures**	**Mockup**
Dataset 3	267	26.7 m	Figures: 4.13, 4.14, 4.15	Figure: 4.10
Dataset 4	416	41.6 m	Figures: 4.16	Figure: 4.11
Dataset 5	1085	108.5 m	Figures: 4.17, 4.18	Figure: 4.12

Table 4.1: Dataset characteristics

4.2.2 Real Dataset

4.2.2.1 Validation

First of all we intend to assess the validity of this approach and its contributions to work with real data. According to this purpose, we compute the final estimated map, the position of the views that conform such map and the final pose of the robot along the traversed path.

As for the kind of environments, here we concentrate on the experiments conducted at indoor environments. The specific scenarios consist of clear spaces with corridors and large rooms, in either office-like or laboratory-like spaces. Table 4.1 synthesizes the main characteristics of the scenarios in such environments, where the different datasets were acquired. Note that for each dataset we include references to the corresponding figures in this document. These figures show the final results for such scenarios. Besides, each dataset is also associated with a mockup in top view that synthesizes the layout of the real scenario.

As a general norm for the interpretation of the following figures, the legends have to be read as follows:

- Ground truth: reliable estimation for comparison matters. It is represented by a continuous dark line in the figures. It is computed from the raw input data provided by the SICK-LMS laser boarded on the P3-AT robot. This data is processed by means of a gmapping technic [127, 51].
- Odometry: representation of the raw input data provided by the odometer of the robot. It is represented by a dashed red line in the figures. Note that this data is modeled by the parametrization exposed in Algorithm 1 in Section 2.2.3. Some experiments present noise terms that have been overweighted with this parametrization so as to test the behaviour of the SLAM approach under worse noise-condition scenarios.
- Solution: estimation of the trajectory of the robot and the arrangement of the views of the map, which are indicated with crosses at their location. The estimated solution for the trajectory is computed by the visual SLAM approach presented in this chapter. It is represented by a dash-dotted blue line.
- Uncertainty: extracted from the covariance matrix $P(t)$, defined in (3.12) and (3.18). It is represented in two manners:

  - As a set of ellipses on the position of the views of the map, indicated with crosses.
  - As the convergence interval for each experiment, denoted as 2σ and represented by a continuos dark line in the error figures. This value determines the confidence interval where the estimation of an EKF-based SLAM model is expected to converge. It expresses the uncertainty at each time step in the system.

- Error: computed at every pose of the robot as the difference value between the ground truth and the estimated solution. It is represented by a dashed red line for the odometry and by a dash-dotted blue line for the estimated solution. The error is divided into the X, Y and θ over the y-axis. The x-axis represents the map step, being the iteration time of the system.

This series of experiments are principally focused on the map building process that has been previously introduced in Section 4.1. For all the stated scenarios, the robot starts navigating the environment by capturing an initial view in the map at the origin. As long as the vehicle moves, it computes its localization thanks to the observation model. Note that this process entails that the robot simultaneously assesses the value of the similarity or initialization ratio A (4.4), so as initialize new views in the map according to the changes in the visual appearance of the environment. Similarly, A has also to be evaluated in order to obtain the appropriate data association.

Dataset 3

The following figures represent in detail the final estimated map for the Dataset 3 after the map building process finalizes. In particular, Figure 4.13, Figure 4.14 and Figure 4.15 represent the estimated map and the error on the estimation. Figure 4.10 synthesizes the layout of this real scenario in top view.

Dataset 4

Likewise, Figure 4.16 shows the same results for the Dataset 4. In the same manner, Figure 4.11 provides further detail on the layout of this real scenario.

Dataset 5

Ultimately, Figure 4.17 presents the results corresponding with the Dataset 5. Figure 4.12 depicts the layout of this last real scenario.

Generally, in terms of error, we observe that the calculated trajectory provides satisfactory estimations in all scenarios. In addition, the evolution of the error demonstrates the convergence of the estimation at all time steps. This fact confirms the suitability of the solution, not only at the end of the experiment, but also at any time instant.

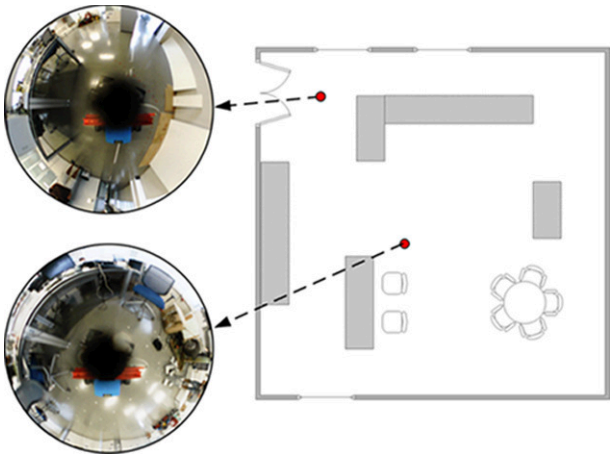


Figure 4.10: Mockup for the Dataset 3. Two views are indicated.

In particular, for the first scenario, Dataset 3, three variants of the final map estimation are presented by Figure 4.13, Figure 4.14 and Figure 4.15 respectively. The intention is to establish a benchmark that allows to test the behaviour of the approach when dealing with different number of views in the map. These three maps have been estimated over the same scenario presented in Figure 4.10. The main difference consists of a variation on the value of A within the map building process. The specific values are A_1=0.02, A_2=0.05 and A_3=0.1, for Figure 4.13, Figure 4.14 and Figure 4.15 respectively. This implies that the system is prone to acquire more views in the map whenever the values of A are higher and there is not any view within the maximum observation range of the robot. At first sight, it can be proved that the higher number of views in the map the better results in terms of error and accuracy.

Note that each scenario presents an arrangement of views which is fully dependent on the visual appearance of the environment, as recently mentioned above. As introduced in Section 4.1.1 and Section 4.1.3, the data association, and consequently the initialization of views, are determined by the initialization ratio A (4.4), since this performs the evaluation of the appearance of the environment. According to this, each scenario produces a different number of views with a different arrangement.

As a preliminary overview, these results reinforce the validity of the approach to deal with real data. In the same manner, convergence is ensured as well as feasible and accurate results for real applications. Secondly, another extracted outcome is that the more number of views in the map, the more number of observation measurements are likely to succeed, and consequently the more accurate estimation is obtained. Next subsection presents a further study on performance, based on the context established by these results.

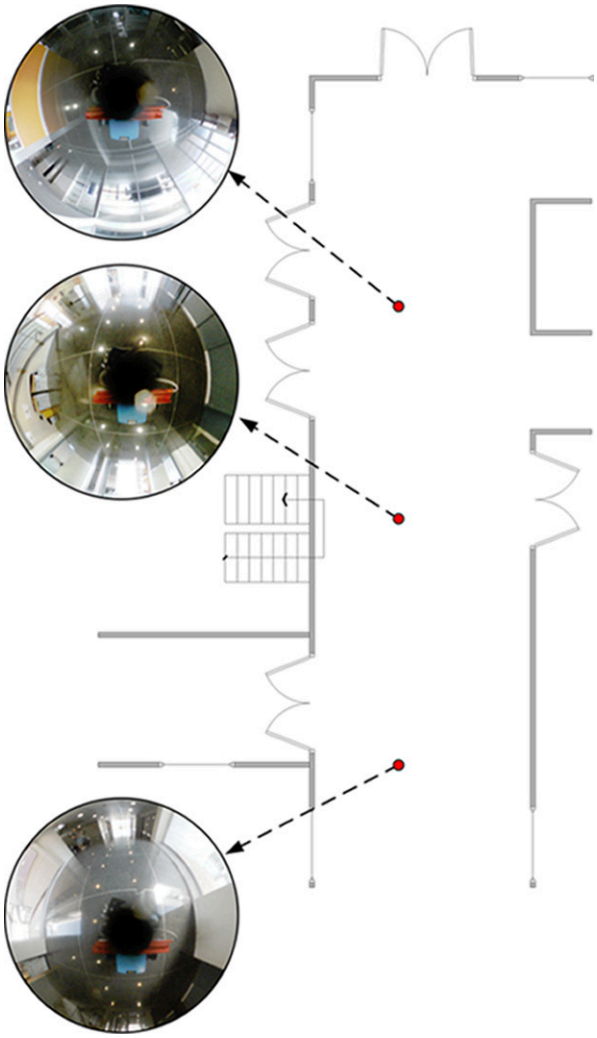


Figure 4.11: Mockup for the Dataset 4. Three views are indicated.

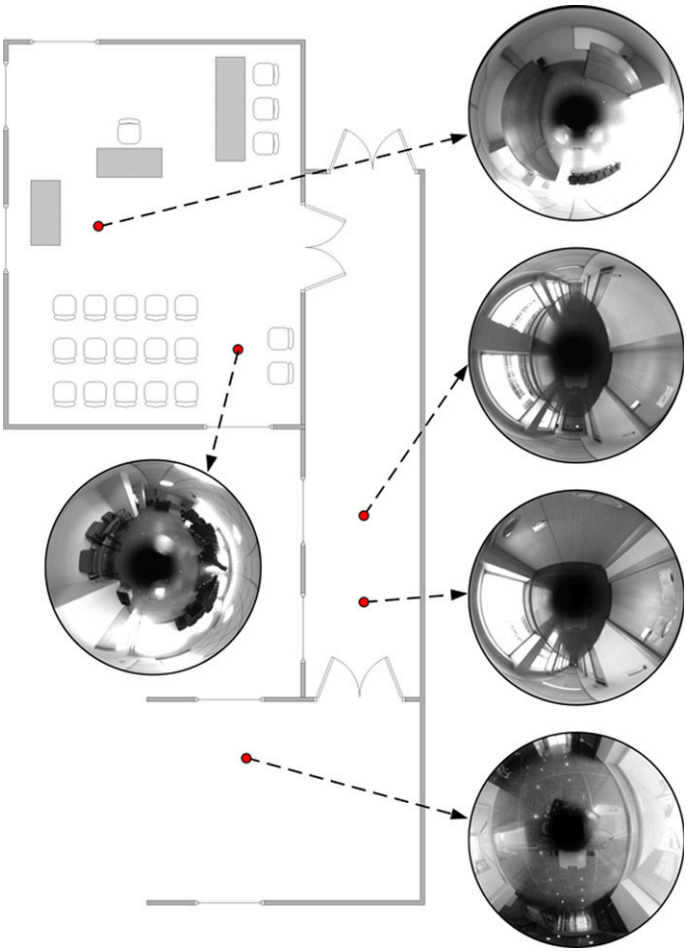
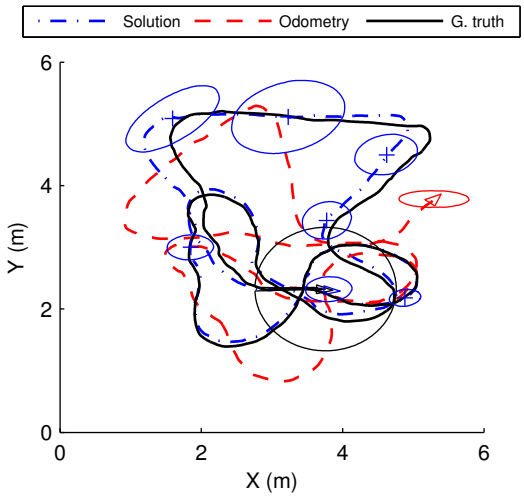
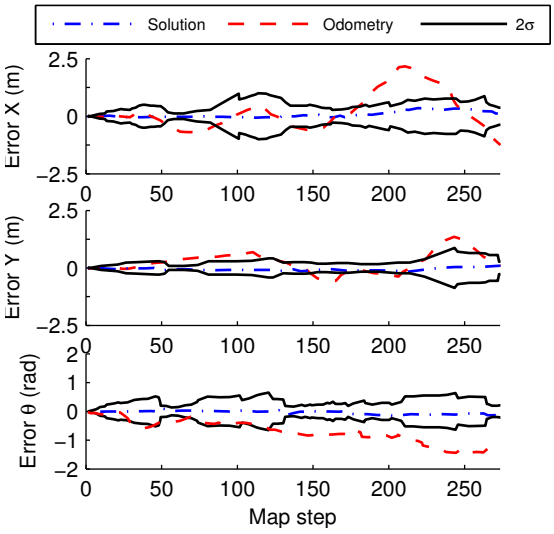


Figure 4.12: Mockup for the Dataset 5. Five views are indicated.

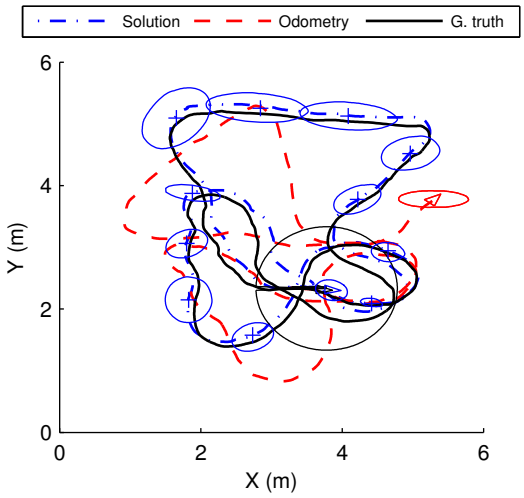


(a)

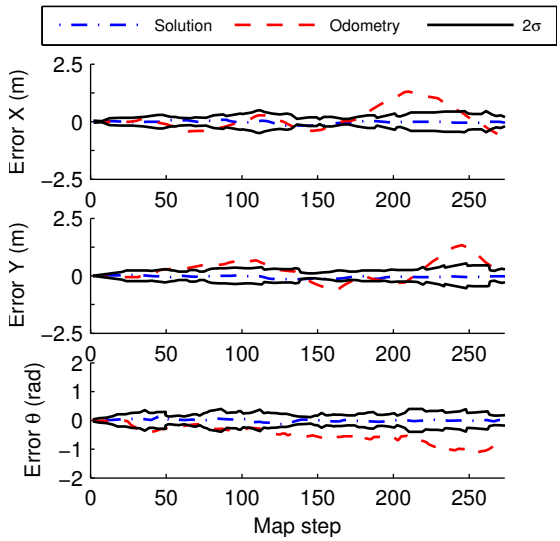


(b)

Figure 4.13: Results obtained in the Dataset 3 (Figure 4.10) for a final map constituted by N=7 views with A=0.02. Figure 4.13(a) presents the estimated solution in dash-dotted line, the odometry in dashed line and the ground truth in continuous line. The location of the views is indicated by crosses and their uncertainty by error ellipses. Figure 4.13(b) represents the error at each step in X, Y and θ within convergence intervals of 2σ.

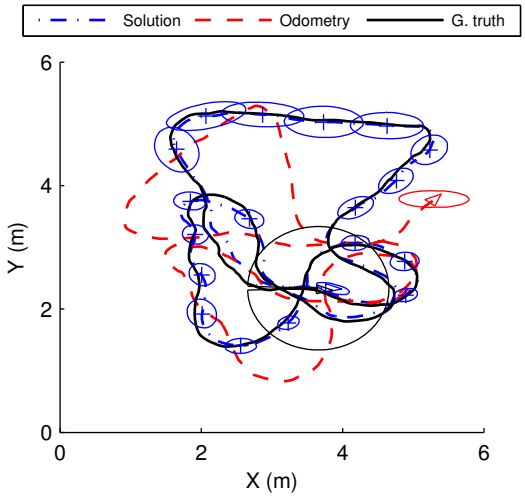


(a)

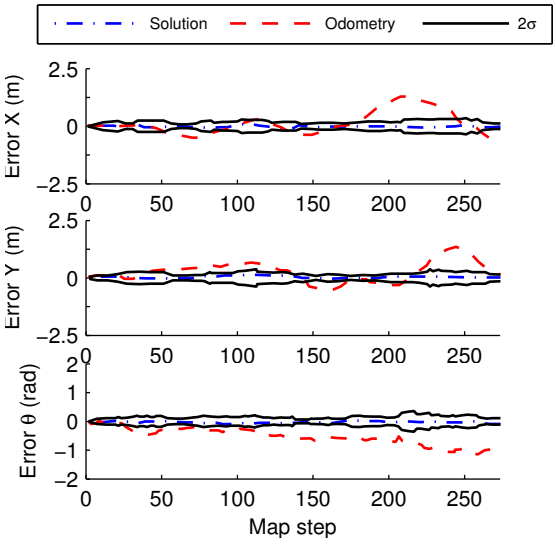


(b)

Figure 4.14: Results obtained in the Dataset 3 (Figure 4.10) for a final map constituted by N=12 views with A=0.05. Figure 4.14(a) presents the estimated solution in dash-dotted line, the odometry in dashed line and the ground truth in continuous line. The location of the views is indicated by crosses and their uncertainty by error ellipses. Figure 4.14(b) represents the error at each step in X, Y and θ within convergence intervals of 2σ.

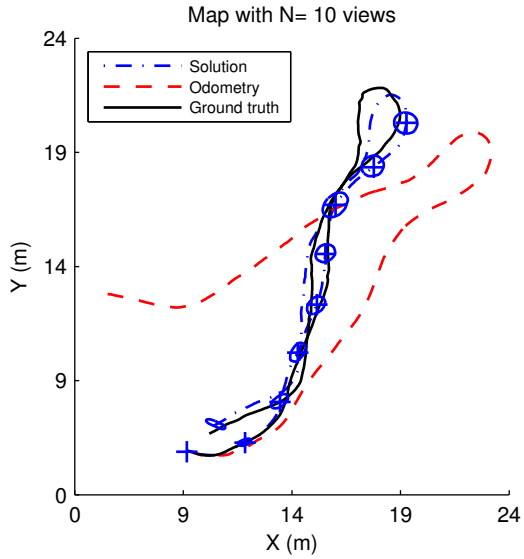


(a)

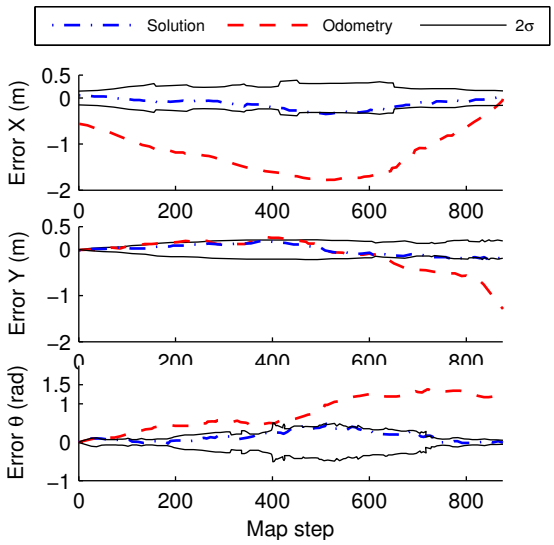


(b)

Figure 4.15: Results obtained in the Dataset 3 (Figure 4.10) for a final map constituted by N=19 views with A=0.1. Figure 4.15(a) presents the estimated solution in dash-dotted line, the odometry in dashed line and the ground truth in continuous line. The location of the views is indicated by crosses and their uncertainty by error ellipses. Figure 4.15(b) represents the error at each step in X, Y and θ within convergence intervals of 2σ.



(a)



(b)

Figure 4.16: Results obtained in the Dataset 4 (Figure 4.11) for a final map constituted by N=10 views with A=0.04. Figure 4.16(a) presents the estimated solution in dash-dotted line, the odometry in dashed line and the ground truth in continuous line. The location of the views is indicated by crosses and their uncertainty by error ellipses. Figure 4.16(b) represents the error at each step in X, Y and θ within convergence intervals of 2σ.

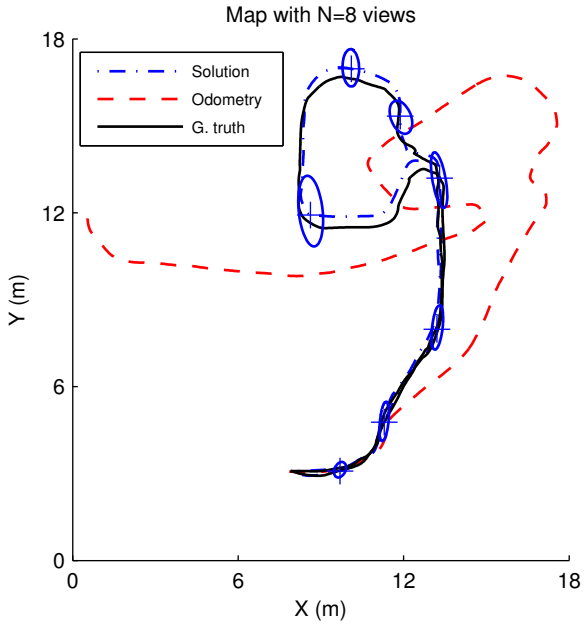
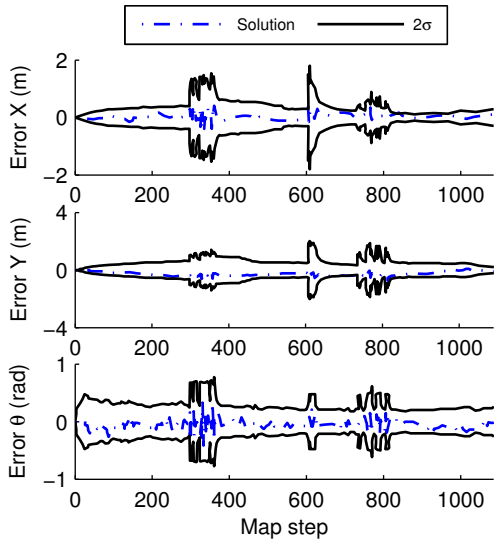
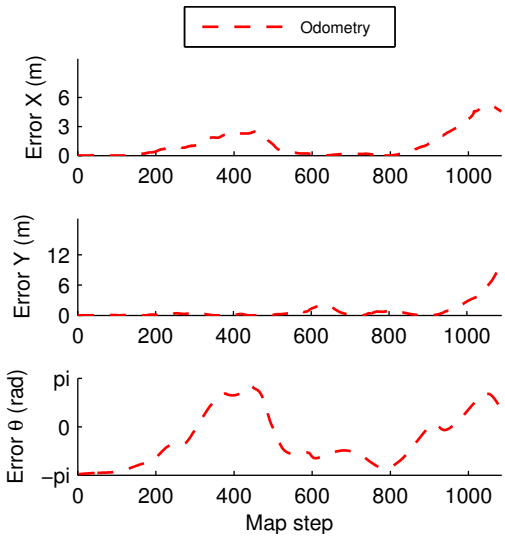


Figure 4.17: Results obtained in the Dataset 5 (Figure 4.12) for a final map constituted by N=8 views with A=0.02. The estimated solution is presented in dash-dotted line, the odometry in dashed line and the ground truth in continuous line. The location of the views is indicated by crosses and their uncertainty by error ellipses.



(a)



(b)

Figure 4.18: Error results obtained in the Dataset 5. Figure 4.18(a) represents the error of the estimation at each step in X, Y and θ within convergence intervals of 2σ. Likewise Figure 4.18(b) represents the error of the odometry at each step.

4.2.2.2 Performance Analysis

Once the suitability of the approach and the convergence of the estimation have been ensured, then it is necessary to establish certain tests that are aimed at the performance analysis of this approach. In this context we have assessed the relevance of certain specific variables and their interdependencies through the study of the following aspects:

- Dimension of the estimated map in terms of number of views.
- Time consumption to obtain the estimation of the solution.
- RMS error.

Under these conditions for the analysis benchmark, Figure 4.19 presents results of the time consumption required by this approach to compute the estimation of the map and the pose of the robot at a certain instant t. These results intend to analyze the time dependency with the number of views that are observed at each t. Note that Figure 4.13(a), Figure 4.14(a) and Figure 4.15(a) already exposed the different results when the final estimation consisted of a different number of views N. Focusing on the time study, Figure 4.19(a) divides the total time consumption into observation time and processing time, plotted with blue and green lines respectively. This separation implies the following contributions to the final time computed:

- Processing time: consists of the overall time taken by the system in order to deal with data management, such as memory access, data association, and to compute the final estimation, after all the observation measurements are performed.
- Observation time: consists of the overall time taken by the system in order to extract observation measurements to all the views observed at a time t. That is basically the time invested in computing the motion transformation to all the views observed, as detailed in Section 2.3.1.

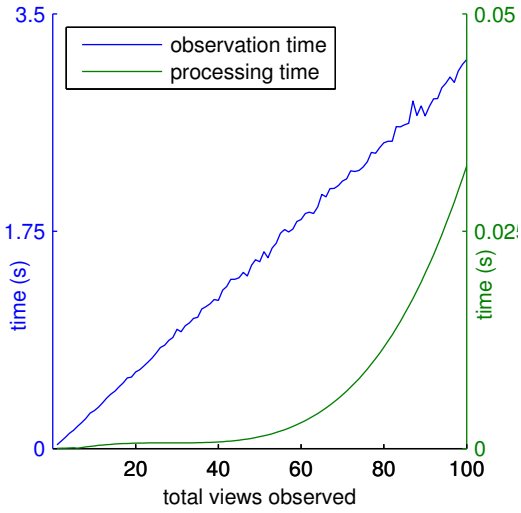
Such differentiation evidences the exponential growth with the number of views observed, as the expected behaviour in an EKF-based algorithm. Note that the complexity should fit an order $O(N^2)$, being N the dimension of the map as defined in Section 4.1. This fact can be noticed in Figure 4.19(a) by inspection of the green line that represents the processing time with its time scale on the right-side y-axis, whereas the observation time is represented by a blue line with its time scale on the left-side y-axis. The other aspect to point out is the huge growth in the observation time in comparison to the processing time, when the number of views observed increases. It can be proved by simply observing the gap in the order of their time scales. For instance, with 80 views observed, the observation time is $\sim$0.75 s, whereas the processing time is $\sim$0.01 s. This reaches such an extent that the observation time produces a masking effect on the processing time, which turns to be irrelevant when the total number of views observed is high. That is the main reason why both times have been plotted separately.

This experiment has been repeated over 300 times, thus Figure 4.19(b) presents the standard deviation for the computation of the observation time, as well as its mean value. The results shown by this figure prove that the time results are considerably stable within a small range of variation of $\Delta\sigma$ =2 ms. Such variation is due to the fact that each computation depends on each observation measurement, and so does the matched points extracted, their robustness and the amount of them, as it was deduced in Section 2.3.1.

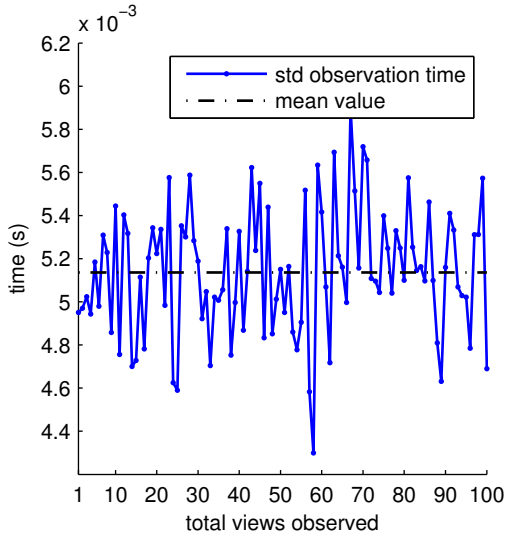
These results reveal that exists an expensive cost on the observation model when the number of views increases. This aspect may compromise the effectiveness of this approach to deal with real-time applications. For this reason, we also assess the RMS error with the time consumption. Figure 4.20 presents this kind of results. Again, a subdivision has been carried out in order to differentiate the contribution made by the observation time, denoted in Figure 4.20(a), and by the processing time, denoted in Figure 4.20(b). Note that the right-side y-axes represent the time scale and the left-side y-axes the RMS error scale. Again, Figure 4.21 presents the standard deviation on the RMS error values along the 300 repetitions of the experiment. It has to be noticed that in this case there exists a higher variation on the RMS error when the number of views observed is low. The lower number of views observed, the more likelihood for a less accurate solution in average terms.

Moreover, the results also show a certain dependency with those presented in Section 2.3.3. Obviously, the time required by the observation model depends on the number of matched points detected. However, in comparison to those results in former sections, here we have extended the experimental benchmark to assess the number of views, since we are dealing with a map building task in SLAM. For that reason, it is important to highlight that when referring to analysis on the map dimensions, the concept of number of matched points is not entirely equivalent to the concept of number of views. Even though they are closely related, the dimension of the map is expressed by a number of views N. Then, the number of views observed is portion out of the total N.

Now we can extract further outcomes. Although the approach is liable to generate a considerably high overload in terms of observation time with the number of views, it also confirms that a reduced set of views can provide a highly reliable estimation in an acceptable time to be run on real-time applications. We can observe that for simultaneous observations up to 10 views observed, the RMS error in the estimation is lower than 0.4 m, with a total time consumption below 40 ms. This also confirms the compactness of the representation of the map, which is capable of encoding the most relevant information of an environment in a map composed by a reduced set of views.

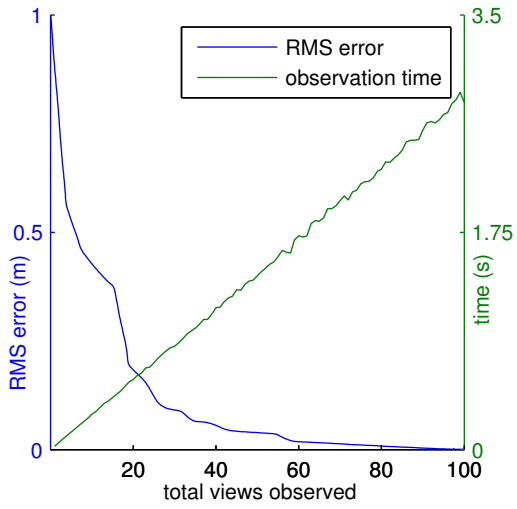


(a)

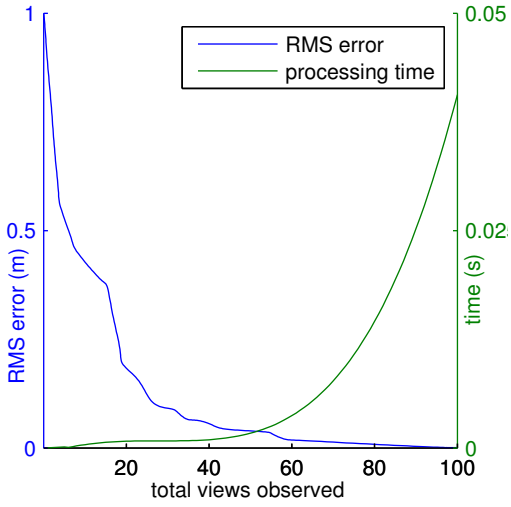


(b)

Figure 4.19: Time consumption against number of views observed. Figure 4.19(a) presents the total computation time divided into: observation time (blue, left-side y-axis) and processing time (green, right-side y-axis). Figure 4.19(b) represents with continuous line the standard deviation in the observation time along the 300 repetitions of the experiment. The mean value is drawn with dash-dotted line.



(a)



(b)

Figure 4.20: RMS error (blue, left-side y-axes) and time consumption (green, right-side y-axes) against number of views observed. Figures 4.20(a) and 4.20(b) present separately the observation time and the processing time against the number of views observed, respectively. The times values and the RMS error are drawn with colored continuous line whereas the mean value for the RMS error is drawn with dash-dotted line.

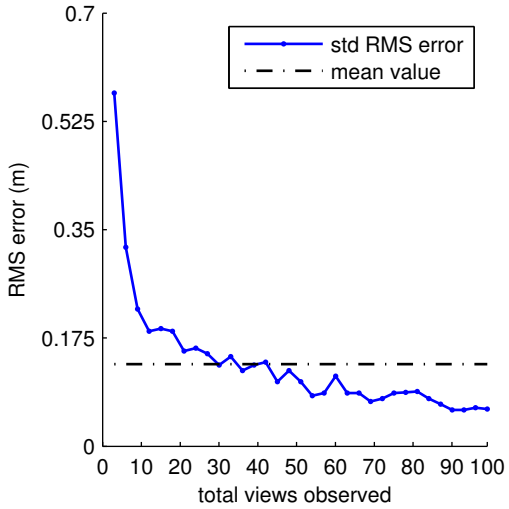


Figure 4.21: Standard deviation for the RMS error in Figure 4.20.

4.3 Conclusions

In this section we have presented a visual EKF-based approach to the SLAM problem using a single omnidirectional camera as a visual sensor. We have proposed a different representation of the environment in terms of the estimated map, which now consists of a reduced set of omnidirectional views. With that purpose, we have detailed the essentials of the approach in terms of map building and its associated substages: view initialization, observation model and data association. The observation model poses a challenge as we intended to only estimate the pose and orientation of a set of omnidirectional images that represent part of the map. The motion transformation scheme defined in Chapter 2 sustains the observation process, by which a set of feature points associated with each omnidirectional image allow the robot to compute its localization in the surroundings. As a result, the reduction of elements in the map, namely views in this approach, permits to provide a compact representation of the environment.

The initialization stage also implies a significant improvement for the assessment of a better view initialization according to the visual appearance of the environment. Whenever the visual appearance differs from the current pose of the robot, the system triggers the procedure to acquire a new view that encodes the surroundings around the new area which the robot currently explores.

Another contribution is an enhanced matching scheme to deal with the problem of finding robust correspondences between images. We exploit the benefits of the EKF state prediction in order to integrate the current uncertainty of the estimation into the matching process.

Finally, the visual inspection of the results section provides several evidences that permit to evaluate this approach but also to draw further conclusions about performance. The study of the dimension of the estimation reveals some aspects: the larger number of N views composing the map provide the more accurate results in terms of error, since more views are observed and hence more observation measurements are correctly computed. However, the computational cost produces an exponential increase with N. The evolution of the RMS error has also been tested.

Overall, these results suggest that a trade-off solution has to be reached, since generally, SLAM systems are real-time oriented, being the time a limiting factor. Despite this fact, the approach presented here maintains accurate results even when using a reduced set of views, which is an important benefit to consider under circumstances of limited computational resources. On the other extent, maps with excessive number of views do not necessarily imply better results. This is a consequence of the results shown in Section 2.3.3, as the limit factor for the accuracy is established by the quality of the matched points and the robustness of the motion transformation within the observation model. This analysis also confirms the compactness of the new representation of the map based on views. In comparison to traditional EKF-based approaches, we propose a map that produces an estimation with a huge reduction on the dimensions,

in terms of views, which stills provides results that prove enough accuracy to operate at real-time scenarios.

Summarizing, the experimental results provide a satisfactory validation for this approach to work appropriately in a real scenario under real-time requirements. Nonetheless, in this context, during the research and development stages of this implementation, different drawbacks arose. The main weakness of EKF-based schemes lies on the inconsistency under circumstances of high uncertainty, due to the presence of non-linear effects. This reason made us define further lines of investigation. Thus next chapters present some of the actions taken and further contributions in relation to this last reflection.

5 SGD-based SLAM Contributions

This chapter intends to present the design and implementation of a new contribution to the framework of this thesis, that is, the visual SLAM problem. In particular, we propose a variant of a SGD solver, adapted to the combination of omnidirectional images with the map representation already introduced in Chapter 4. In the field of mobile robots applications, SGD techniques have rarely been evaluated with information gathered by visual sensors. In this work we define a SGD algorithm for our SLAM system which profits the beneficial characteristics of a single omnidirectional camera.

The obtention of a feasible map of the environment poses a complex challenge, since the presence of noise arises as a major problem which may gravely affect the estimated solution. Consequently, our SLAM algorithm has to cope with this issue but also with the data association problem. In this sense, some of the outputs we can extract from Chapter 4 confirm that the EKF is highly sensitive to non-linear visual observation models, as the omnidirectional. Conversely, the SGD emerges in this work as an offline alternative to minimize the non-linear effects which deteriorate and compromise the convergence of traditional estimators.

Generally, EKF methods are usually liable to become troublesome when dealing with external errors and jeopardize the final behaviour of the system, since they find difficulties to maintain the convergence of the estimation. The main reasons are the linearization of the movement and the observation model accomplished by this filter, especially under such circumstances of non-linearities. This situation normally appears in the presence of gaussian noise introduced by the observation measurement, fact that usually causes injurious data association problems [89]. A visual observation model, as in the case of the omnidirectional model, is susceptible to introduce non-linearities and

thus it is responsible for those kind of errors. On the contrary, an offline algorithm, such as SGD [8, 151], provides more robustness to face this issue. Similarly, parallel approaches [148, 128] confirm parallel its stability under non-linear contexts. Hence, we rely on the SGD algorithm to outperform the main EKF's drawbacks in terms of instability, despite the fact that SGD is an offline method.

The development of this SGD implementation meets with the requirements of the nature of the omnidirectional sensor, as well as with the associated observation model. Thus we modify the standard SGD version to adapt it to the omnidirectional geometry. Besides, the angular unscaled observation measurement needs to be considered. This upgraded SGD approach intends to minimize the non-linear effects which impair and compromise the convergence of traditional estimators. Nevertheless, undesired oscillations may occur due to the stochastic nature of the constraints' selection. For this reason, an optimization process is also suggested. In contrast to former SGD approaches, which only process one constraint independently, here we define a strategy for simultaneous processing of several constraints to overcome these issues.

Traditionally, the better known standard SGD applications [104, 53, 52] use a different geometric reference, and consider data range observations in a cartesian measurement system. Instead, we have to deal with a different map and observation model based on an omnidirectional geometry. Thus it is necessary to establish a comparative benchmark to assess the feasibility of the results obtained under different conditions. Therefore, we analyze the behaviour of the standard SGD, the EKF and this new SGD proposal, all applied to our view-based SLAM approach. Estimation accuracy, robustness, convergence and performance are the most important terms to evaluate.

Finally, we can synthesize the main contributions regarding this SGD implementation through the structure of this chapter in the following terms:

- Proposal of a modified SGD solver algorithm, adapted to the omnidirectional geometry of our view-based SLAM approach.
- Presentation of the design specifications: state equations and differential equations for the observation measurement.
- Contribution to improve the performance of the SGD by processing simultaneously several constraints into the system, in contrast to the standard SGD.
- Robustness against non-linear effects in contrast to traditional solvers, such as the EKF.
- Efficiency and accuracy comparison experiments with the standard SGD and the EKF, in both simulated and real data scenarios.

5.1 Proposed SGD

This section provides further details about the design and implementation of the proposed SGD algorithm achieved in this thesis. The main features and specifications of this contribution are addressed here.

In Chapter 3 we provided a brief introduction to the theory fundamentals of the standard SGD. Then, continuing under the same notation context, at first instance, we need to consider a redefinition of the standard state vector presented in (3.21) as $x_t = [(x_0, y_0, \theta_0), (x_1, y_1, \theta_1) \dots (x_n, y_n, \theta_n)]$, which will be now treated as a set of incremental variables. Please note that (x_n, y_n, θ_n) encodes the 2D coordinates and bearing in a general reference system for each pose (namely nodes). Contrary to the incremental representation, this standard global encoding (3.21) has the main drawback of not being capable to update more than one node and its adjacents per constraint. This aspect has led us to assume a general agreement in the use of the incremental representation, now defining the state incrementally encoded as:

$$x_t^{inc} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (dx_1, dy_1, d\theta_1) \\ \vdots \\ (dx_n, dy_n, d\theta_n) \end{bmatrix} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (x_1 - x_0, y_1 - y_0, \theta_1 - \theta_0) \\ (x_2 - x_1, y_2 - y_1, \theta_2 - \theta_1) \\ \vdots \\ (x_n - x_{n-1}, y_n - y_{n-1}, \theta_n - \theta_{n-1}) \end{bmatrix} \tag{5.1}$$

where $(dx_i, dy_i, d\theta_i)$ encode the variation between consecutive poses in coordinates of the global reference system. Please notice that, according to the formulation defined in (4.1) and (3.21), x_v and each x_{l_n} would correspond with certain poses $\in [(x_0, y_0, \theta_0), (x_1, y_1, \theta_1) ... (x_n, y_n, \theta_n)]$. Next, the relation between the global pose x_t^{glob} and incremental pose x_t^{inc} is:

$$x_t^{glob} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_{i-1} \\ 0 & 1 & 0 & y_{i-1} \\ 0 & 0 & 1 & \theta_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dx_i \\ dy_i \\ d\theta_i \\ 1 \end{bmatrix} \tag{5.2}$$

Now, the state vector is differentially encoded and each single update has influence on the whole map reestimation. Nonetheless, this incremental encoding might cause the appearance of some non-linearities in J_{ji}. However, despite this fact, the possibility to update every pose from a single constraint is a valuable advantage to take the most of. Therefore, due to this fact, Δx (3.28) weights all poses.

It is important to keep in mind that in this approach we are dealing with a visual observation given by an omnidirectional camera. This fact made us adjust and redesign the set of equations defined in Chapter 3 for the standard SGD. Now the nature of the constraints are not only metrical like odometry's constraints, but also angular. Therefore, the omnidirectional measurements and the incremental representation require the reformulation of several terms involved in the estimation. Following, we detail all the proposed modifications to the terms of the standard SGD. The complete structure for each derivative equation is detailed in the following subsection.

- The first modification is referred to $f_{ji}(x)$ (3.22), which expresses the observation as function of the state x_t, and nodes j and i. $f_{ji}(x)$ has to be differentiated between odometry and visual observation constraints:

$$f_{ji}^{odo}(x) = \begin{pmatrix} dx_j \\ dy_j \\ d\theta_j \end{pmatrix} + \begin{pmatrix} dx_{j-1} \\ dy_{j-1} \\ d\theta_{j-1} \end{pmatrix} + \cdots + \begin{pmatrix} dx_i \\ dy_i \\ d\theta_i \end{pmatrix} \tag{5.3}$$

where $(dx_i, dy_i, d\theta_i)$ has been defined in (5.1) as the variation between consecutive poses for node i, whereas $(dx_j, dy_j, d\theta_j)$ represents the same variation for node j. And for the case of the visual observation constraint:

$$f_{ji}^{visual}(x) = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{bmatrix} \arctan\left(\frac{\mathrm{dy}_j - \mathrm{dy}_i}{\mathrm{dx}_j - \mathrm{dx}_i}\right) - d\theta_i \\ d\theta_j - d\theta_i \end{bmatrix} \tag{5.4}$$

where β and ϕ are directly computed from the observation measurements, which express the motion transformation relation between two omnidirectional images, as detailed in Chapter 4.

- Then a second modification is necessary in order to recalculate $J_{ji} = \frac{\partial f_{ji}}{\partial x}$, according to the previous reformulation of $f_{ji}(x)$. It has to be noticed the importance of considering the value of each node's index, being either $j > i$ or $j < i$, since the structure of the derivatives differ considerably. Furthermore, as seen above, the dimensions of $f_{ji}(x)$ are different, fact which has also to be taken into consideration in order to resize appropriately the rest of the terms involved in the SGD algorithm.

$$J_{ji} = \frac{\partial f_{ji}(x)}{\partial x} = \left[\frac{\partial f_{ji}(\phi)}{\partial x}, \frac{\partial f_{ji}(\beta)}{\partial x}\right] \tag{5.5}$$

- Lastly, we suggest that the estimation of the new state x_{t+1} reflects the usage of several constraints at the same time, in contrast to the standard SGD model. We seek more relevance of constraints' weight when searching for the optimal minimum estimation. Obviously, computing more than one constraint at each step leads to a certain overload. Contrarily, in this approach, we reduce the expensive estimation of H. In a general case for the standard SGD, H is computed at every single iteration step. In opposition to this, we only compute H once for each subset of constraints introduced simultaneously into the system. Hence we drastically reduce the number of times that H is computed. This succeeds in performing a more efficient scheme which compensates possible time overloads.

According to such modifications, instead of operating in a one-constraint iteration scheme, such as the standard SGD, this SGD proposal operates as follows:

- At every iteration step t, the robot filters all the poses within its maximum observation range in order to extract $f_{ji}(x)$ and the corresponding constraints

δ_{ji}, from the current pose, (namely node j), to the poses under visual range, (namely node i). Note that each of these poses might be either composed by an odometry constraint or by a visual constraint too (in case that there is a view stored in such pose of the map).

- These poses are grouped in subsets c_q. The number of subsets, b, is arbitrarily selected due to experimental testing. We have proved that b provides satisfactory results when it is set to generate c_q with 5 to 10 constraints.

- Once b is selected, the total number of constraints under the visual observation range are uniformly randomized and divided into each c_q, which contain the same number of constraints (5-10).

- Two loops are implemented: a primary loop which minimizes $F(x)$ and a secondary loop, with length b, which processes all the constraints divided into subsets c_q. That is the approach to input several constraints into the same primary iteration.

An example of operation would be a certain pose from which 50 other poses are observed, so that we can select a total number of subsets of b=10, each one (c_q) containing 5 constraints uniformly distributed.

Once depicted the operation of this proposal, Algorithm 3 summarizes the procedure:

Algorithm 3 Proposed SGD algorithm

Require: $\delta_{ji} \in C \; \forall \; j,i$, where $C = [c_1, c_2, \ldots, c_q, \ldots c_b]$ and c_b= $\{\delta_{11}$,δ_{12},$\ldots\}$
  Each c_q represents different subsets of constraints δ_{ji} simultaneously processed by the robot.
  t: iteration step
  ϵ: threshold for $F(x)$
  while $F(x) > \epsilon$ **do**
    $t = t + 1$
    for k=1:b **do**
      Extract all δ_{ji} in c_q randomly
      Compute the following terms:
      $f_{ji}(x) = [f_{ji}^{odo}(x), f_{ji}^{visual}(x)]$, J_{ji}, Ω_{ji}, and r_{ji}
      Δx_q=$\lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji}$
      $x_q = x_{q-1} + \Delta x_q$
    end for
    $x_t = x_q + x_{t-1}$
  end while
  return $x_t = [(x_0, y_0, \theta_0), (dx_1, dy_1, d\theta_1), \ldots, (dx_n, dy_n, d\theta_n)]$

J_{ji}			
	$\frac{\partial f_{ji}(\phi)}{\partial dx_k}$	$\frac{\partial f_{ji}(\phi)}{\partial dy_k}$	$\frac{\partial f_{ji}(\phi)}{\partial d\theta_k}$
$j > i \ / \ k > i$			
$\frac{\partial f_{ji}(\phi)}{\partial x}$	$-\frac{\sum_{k=i+1}^{j} dy_k}{q}$	$\frac{\sum_{k=i+1}^{j} dx_k}{q}$	0
$\frac{\partial f_{ji}(\beta)}{\partial x}$	0	0	1
$j > i \ / \ k < i$			
$\frac{\partial f_{ji}(\phi)}{\partial x}$	0	0	-1
$\frac{\partial f_{ji}(\beta)}{\partial x}$	0	0	0
$j < i \ / \ k > i$			
$\frac{\partial f_{ji}(\phi)}{\partial x}$	$-\frac{\sum_{k=i+1}^{j} dy_k}{q}$	$\frac{\sum_{k=i+1}^{j} dx_k}{q}$	-1
$\frac{\partial f_{ji}(\beta)}{\partial x}$	0	0	-1
$j < i \ / \ k < i$			
$\frac{\partial f_{ji}(\phi)}{\partial x}$	0	0	-1
$\frac{\partial f_{ji}(\beta)}{\partial x}$	0	0	0

Table 5.1: Equations for J_{ji}.

5.1.1 Equations

Here we append the whole structure for each derivative equation associated with the redesign of the omnidirectional observation model to fit the SGD specifications. In particular, Table 5.1 contains the redesigned structure of J_{ji} (5.5) when $f_{ji}^{visual}(x)$ (5.4) is considered.

5.2 Results

In order to validate the appropriateness of the contributions to the SGD algorithm, we present a series of experiments obtained with both simulated and real data. We intend to demonstrate the suitability and reliability of this proposed SGD approach to support real applications. In addition to this, we establish a comparison framework to evaluate its performance and efficiency versus a standard SGD, but also versus an EKF estimator.

5.2.1 Simulation Results

Firstly, we seek to ensure the convergence of the proposed SGD approach. This is crucial when a new solver is introduced into a SLAM system. Besides this, another consideration requires evaluation: the performance of the new method when dealing with a visual observation model, which is a common source of non-linearities. With such purposes, we present preliminary results in two simulated scenarios, as detailed in Figure 5.1 and Figure 5.2.

The first scenario presented in Figure 5.1(a) consists of a random simulated environment of 20×20 m, where the robot traverses 300 m approximately. The real path followed by the robot is shown with continuous line, the odometry is represented with dashed line, whereas the estimated solution is shown with dash-dotted line. A set of views have been randomly placed along the trajectory. Again, the arrangement of these views is controlled by the similarity ratio A (4.4), so as to ensure a realistic placement of each view. Every time the robot moves, it compares a random value of A and initializes a new view whenever A <0.2. As for the observation measurements to the views, $z_t(\phi, \beta)$, its generation is also randomized with an added gaussian noise of $\sigma_\phi = \sigma_\beta$ =0.2 rad. A grid of circles represents the possible poses where the robot might move to and gather a new view. The number of iterations for the SGD to get a valid estimation is 25. As it can be observed in Figure 5.1(a), the final estimation follows the tendency of the real path. Figure 5.1(b) shows the decreasing evolution of the accumulated error probability, $P_{ji}(x)$ (3.22), expressed in logarithmic terms as $F(x)$ (3.26), against the number of iterations. At first sight, these results confirm the validity of this new approach to work with omnidirectional observations.

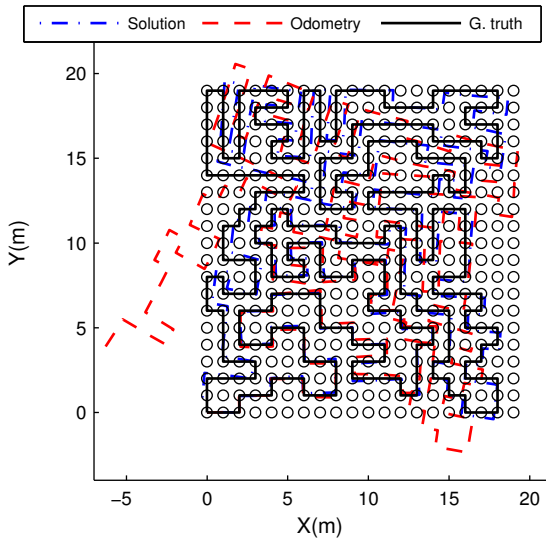
5.2.1.1 Comparing Results

Figure 5.2 presents a second simulated scenario. Now the purpose is to extend the validity of the approach when dealing with an office-like environment, since it is desirable to emulate a more realistic situation with obstructions, obstacles, etc. In addition, we present a comparison between our approach and the standard SGD method.

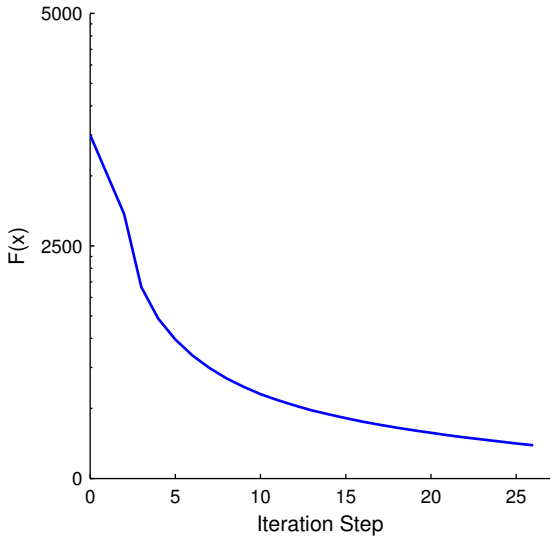
Figure 5.2(a) describes an environment with dimensions 20×50 m, where the robot moves through. The continuous line represents the real path followed by the robot, the dash-dotted line shows the odometry, whereas the estimated solution is shown by a dashed line. Again, the algorithm is able to estimate a rather reliable solution whose topology follows the real path. Contrarily, the error of the odometry grows out of bounds.

Next, as a first approach to comparison results, Figure 5.2(b) presents the evolution of the accumulated error, $F(x)$, for both our SGD approach and the standard SGD algorithm, in the same scenario presented in 5.2(a). Please keep in mind the main difference between the standard SGD and our proposal. The standard only introduces one constraint per iteration, in contrast to our proposal, which processes several simultaneously, as depicted in Algorithm 3.

Here we not only confirm the validity of our proposal, but also its improved capability to speed-up the convergence to a proper estimation, thus involving a better efficiency. In this particular case, it is worth mentioning that our approach requires approximately less than 6 times the computational effort of a standard SGD to reach an optimum value.

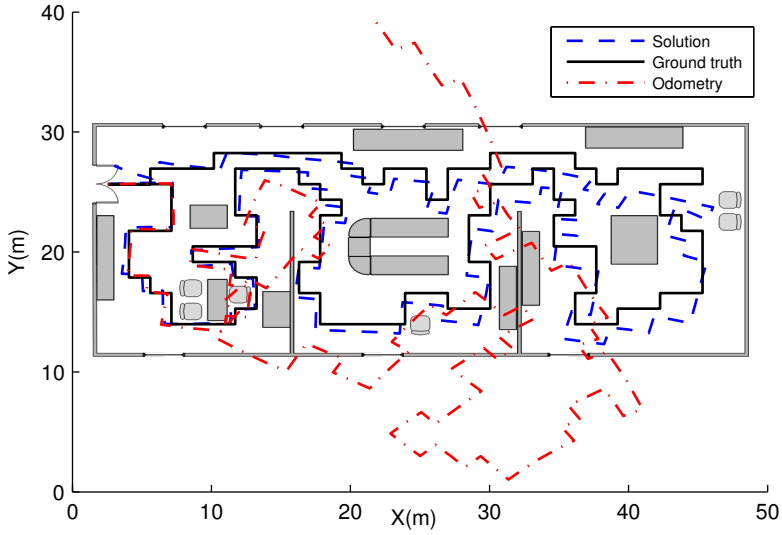


(a)

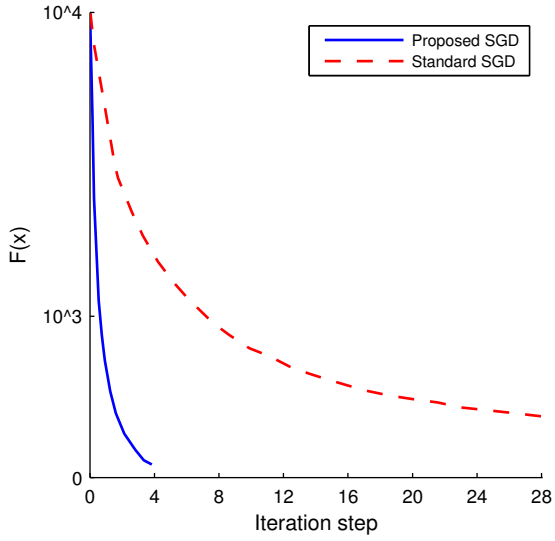


(b)

Figure 5.1: Figure 5.1(a) presents the estimated trajectory obtained with the proposed SGD approach in an environment of 20x20 m. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution. Figure 5.1(b) shows the accumulated error probability $F(x)$ along the number of iterations.



(a)



(b)

Figure 5.2: Figure 5.2(a) shows SLAM results in an office-like environment of 20×50m. Real path in continuous line, odometry in dash-dotted line and the estimated solution in dashed line. Figure 5.2(b) compares the accumulated error probability $F(x)$ of the presented approach (continuous line), and the $F(x)$ of the standard SGD (dashed line).

5.2.2 Real Dataset

Once presented the simulation results for validation, here we carry out an experimental set with real data. We seek confirmation of suitability and reliability of the approach for a realistic application such navigation. Furthermore, we also show extended comparisons with the standard SGD and the EKF algorithms.

Dataset 6

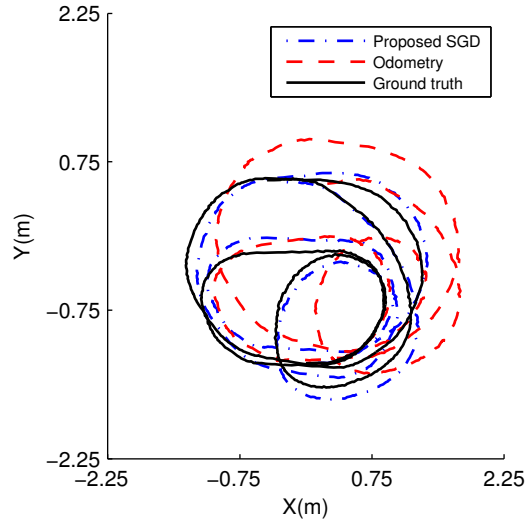
This dataset consists of an experimental set aimed at analyzing the behaviour of the approach when dealing with one of the most adverse situations, that is to say, when the robot constantly turns around, as shown in Figure 5.3(a). The real path is shown with continuous line, the odometry with dashed line and the estimated solution with dash-dotted line. This situation is seen as one of the worsts case scenarios, since it introduces a huge noise into the input associated with the odometry. Nevertheless, it should be noted that the estimation converges to a proper solution, whereas the odometry estimation differs considerably. Figure 5.3(b) shows the decreasing tendency of the accumulated error probability, $F(x)$, along the number of iterations, for both our approach and the standard SGD.

Having tested the validity of the previous experiments, the improved efficiency of our approach can be now confirmed in terms of speed of convergence, compared to the standard SGD method. Examining Figure 5.3(b), it can be seen that this approach reaches optimum values for $F(x)$ in less time than the standard SGD. The main advantage in terms of efficiency is therefore shown.

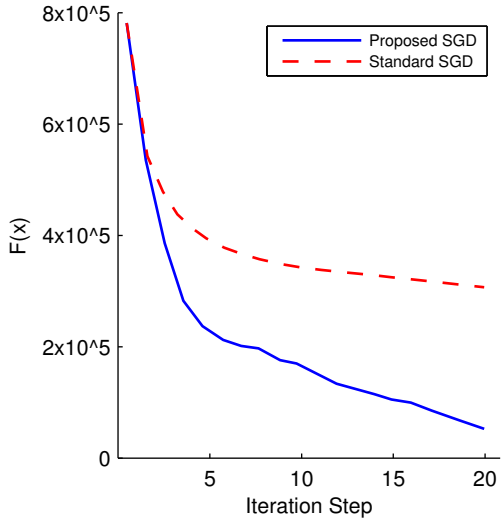
Dataset 7

This dataset presents an experiment that aims to support and reaffirm the beneficial results presented above. In this case we conducted an experiment in a large environment. Here, the robot moves through a real office of 20×50 m. There are obstacles and obstructions such as doors, walls and office furniture. As seen in Figure 5.4 the robot explores the whole environment describing a trajectory of approximately 280 m. Moreover, maps with different number of views N have been constructed to study its relevance on the estimation of the solution. Figures 5.4(a) and 5.4(b) show different results when the map is conformed by N=5 and $N = 30$ respectively. The real path is drawn with continuous line, the odometry with dash-dotted line and the estimated solution with dashed line. Some real views have been indicated.

Figure 5.5 shows the accumulated error probability, $F(x)$, for both experiments, expressing it with continuous line for $N = 5$ and with dashed line for $N = 30$. In addition, to demonstrate the improved efficiency of the method, we compare the values of $F(x)$ provided by this approach, in blue, with the obtained by the standard SGD, in red. According to the specific topology of the environment, it is confirmed that the larger number of views N, the more accurate estimation, since the robot is able to observe more views. Thus the rectification of the estimation is ensured by a higher



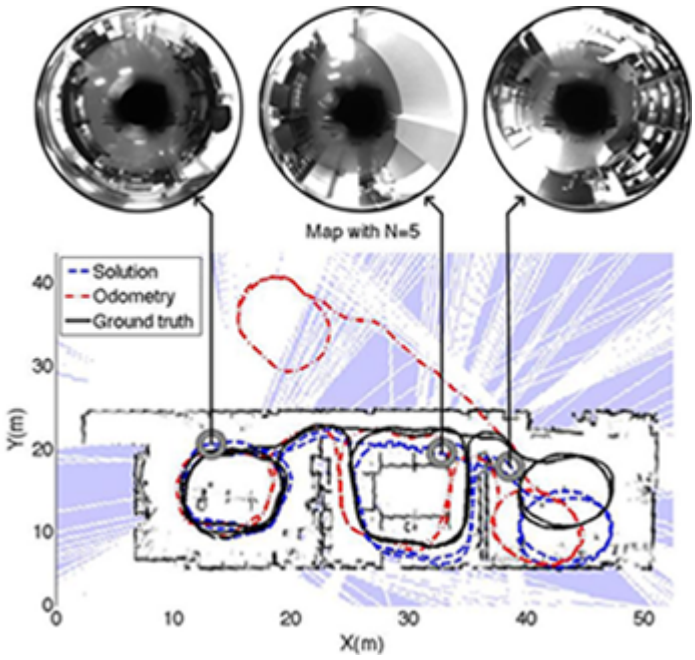
(a)



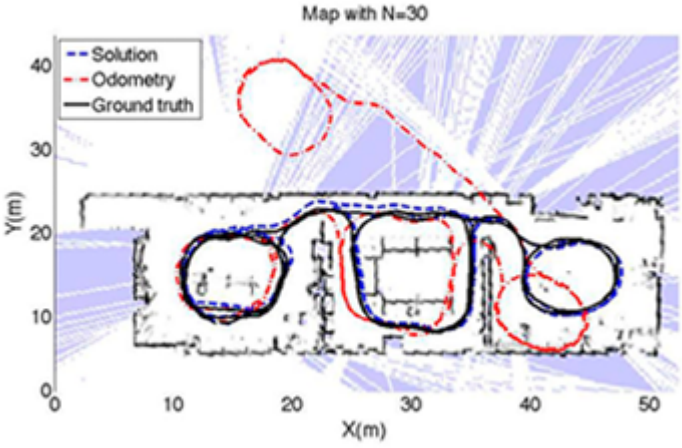
(b)

Figure 5.3: Figure 5.3(a) shows SLAM results in a real office environment. The continuous line shows the real path, the dashed line the odometry and the dash-dotted line the estimated solution. Figure 5.3(b) shows the accumulated error probability $F(x)$ along the number of iterations for our approach and the standard SGD respectively.

number of constraints. Moreover, our approach still reveals the main favorable features compared to the standard SGD, regardless of the value of N. As proven in the previous experiment, the faster speed of convergence is proved by observing Figure 5.5, where lower optimum values for $F(x)$ are confirmed in considerable less time. This fact shows the greater efficiency of this proposal compared to former SGD techniques.



(a)



(b)

Figure 5.4: Figures 5.4(a) and 5.4(a) show SLAM results in a real office environment, with N=5 and $N = 30$ views observed respectively. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution.

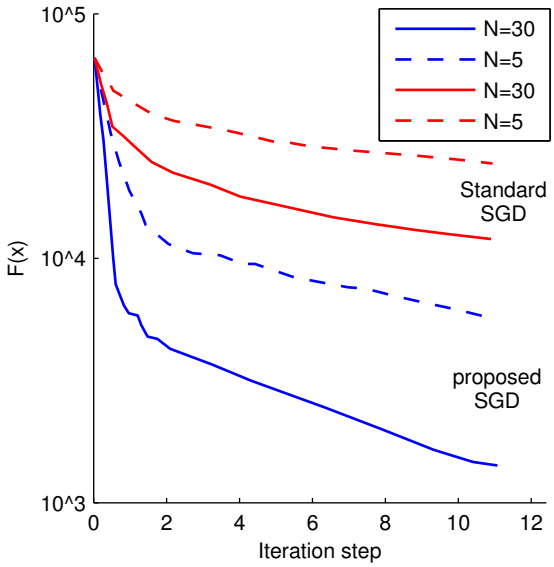


Figure 5.5: Accumulated error probability $F(x)$. Results obtained for the map shown in Figure 5.4(a) with N=5 views, are compared using dashed lines: the dashed blue line represents the proposed approach while the dashed red line represents the standard SGD. Results obtained for the map shown in Figure 5.4(b) with N=30 views, are compared using continuous lines: the continuous blue line represents the proposed approach whereas the continuous red line represents the standard SGD.

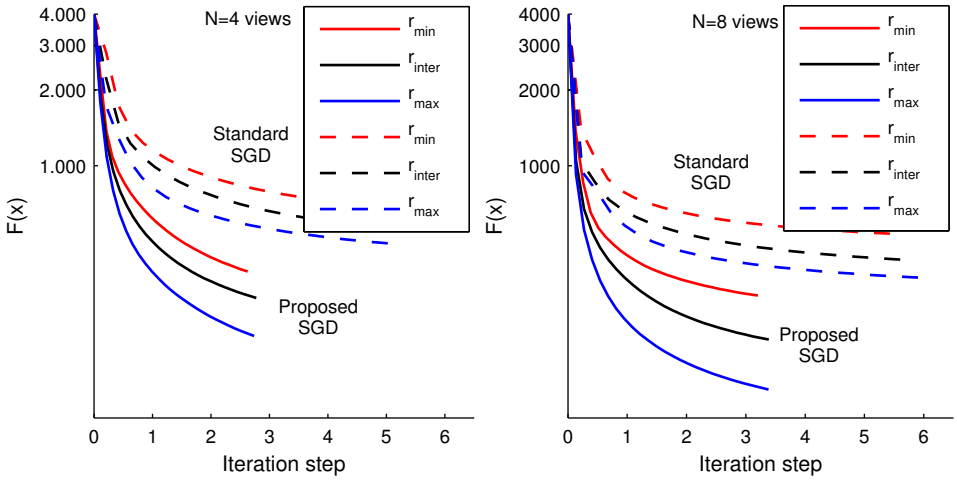
5.2.2.1 Comparison Results

In this section we intend to extend the previous results in terms of comparison, under a real data context.

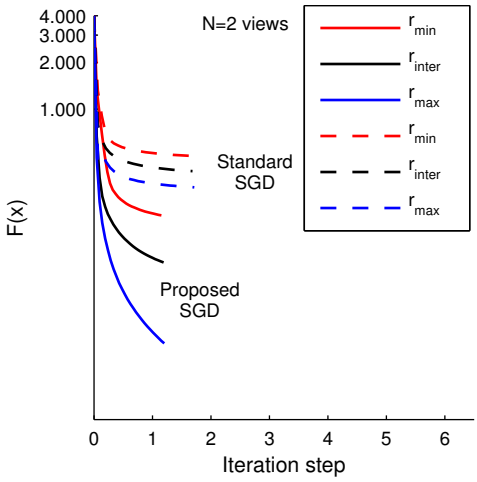
Proposed SGD vs standard SGD

The following experiments have been conducted in order to compare our approach with the standard SGD in terms of efficiency. We pursue the evaluation of our strategy to introduce several constraints simultaneously into the SGD algorithm. The main goal is to improve the speed by which the method iteratively optimizes until a final estimation is reached. In this sense, we have performed a SLAM experiment, where the robot traverses 50 m through a given environment. The same experiment has been repeated 200 times using the same series of odometry inputs, in order to provide mean values which express consistent results. The two approaches, ours and the standard SGD algorithm, have been compared. We have set three experiments where the number of views N that conform the map differ. The observation range r of the robot has also been varied. Figure 5.6 presents results for the accumulated error probability $F(x)$, being the objective function which the SGD algorithm seeks to minimize. Figure 5.6 compares the solution obtained by our approach, drawn with continuous line, and the solution obtained with the standard SGD algorithm, drawn with dashed line. Figures 5.6(c), 5.6(a) and 5.6(b) represent $F(x)$ when the robot observes N=2, N=4 and N=8 views, respectively.

In terms of efficiency, it may be proved that the solution provided by our approach outperforms the solution given by a standard SGD at every case, since the decreasing slope of $F(x)$ is clearly steeper. Hence a faster convergence, and thus a more efficient method is demonstrated. This is the main advantage achieved by means of combining several constraints simultaneously at each iteration step, instead of using only one as a standard SGD does. It is also notable the relevance of the observation range of the vehicle r. As seen in Figures 5.6(c), 5.6(a) and 5.6(b), longer values of r provide a better convergence to the detriment of shorter r, since more views are observed. However, when the robot is able to observe a high number of views, a trade-off solution should be found, since more computation effort is needed in order to process more visual constraints.



(a) (b)



(c)

Figure 5.6: Figures 5.6(c), 5.6(a) and 5.6(b) show the accumulated error probability $F(x)$ in a SLAM experiment, when the map is composed by $N = 2$, $N = 4$ and $N = 8$ views respectively. The continuous lines show the results provided by the proposed solution whereas the dashed lines show results provided by the standard SGD solution. Different lengths for the observation range are defined: r_{min}, r_{inter}, r_{max}.

Proposed SGD vs EKF

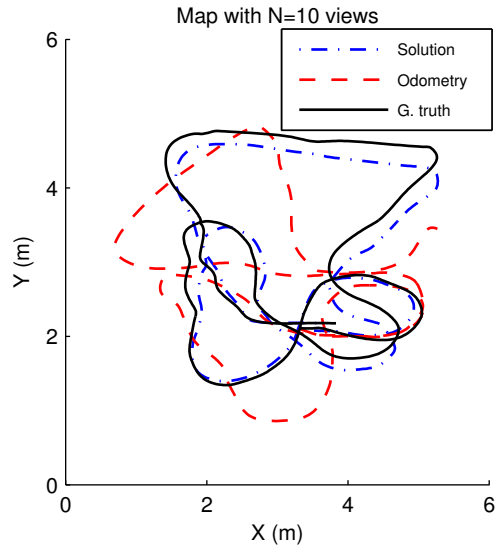
The following results intend to establish a comparison between the two major contributions made in this thesis in terms of SLAM algorithms, that is, the proposed visual-based EKF approach introduced in Chapter 4, and this proposal of SGD adapted to omnidirectional observations. We aim to provide a comparison under different circumstances in terms of noise so as to prove the robustness of the SGD under non-linear effects. In this subsection we first test the behaviour of the SGD and the EKF when working under an idealistic environment, where low non-linear effects are considered. Next subsection will consider a worse scenario, which is definitely the purpose of this contribution.

We need to refer to the experiments shown in Chapter 4 with the Dataset 3, since we use the same dataset as input for the proposed SGD algorithm. We compare both methods by testing their accuracy and robustness on the estimation. Moreover we establish another comparison benchmark to assess the behaviour of both, the EKF and the SGD approaches when data association errors arise.

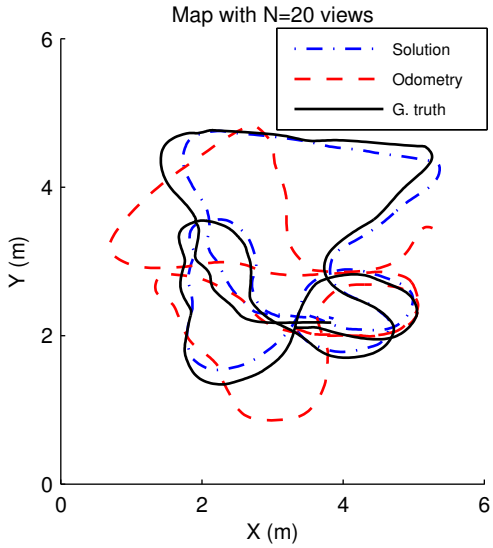
The main characteristics of the Dataset 3 can be observed in Table 4.1. Here we run the same experiment with our SGD estimator. Figure 5.7(a) and 5.7(b) represent two computed maps with N=10 and N=20 views respectively. The key point in the manner to proceed with respect to EKF is that SGD processes the observations offline.

Next, Figure 5.8 presents general results to establish a comparison between both methods, where the RMS error along the path is represented against the number of views N. The continuous line shows the RMS error for EKF while the dashed line shows the SGD's. Here it is crucial to remark that we are dealing with an idealistic situation, where low non-linear effects are considered. This is the main reason why we observe that the accuracy and time results for the EKF outperforms SGD's in this case. Hence it can be confirmed that faster speed of convergence is assured by EKF.

Nonetheless, as mentioned above, this experiment has dealt with a desirable situation where non-linear errors, if any, were low enough so that the EKF response was able to ensure convergence. For this reason, the following experiment will show the results obtained when the visual information is damaged and corrupted by significative noise errors. As a result, we will assess the robustness of both methods, thus obtaining a fair comparison.

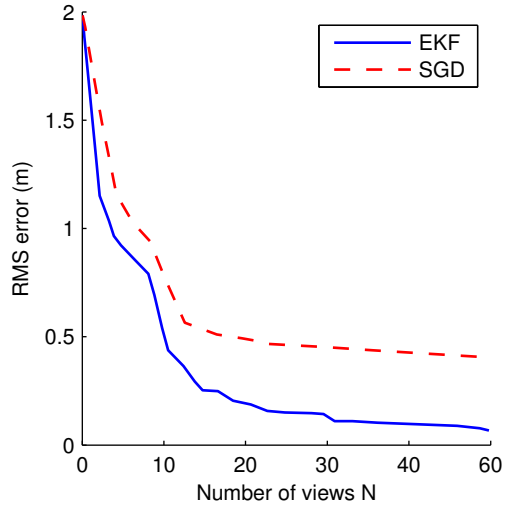


(a)

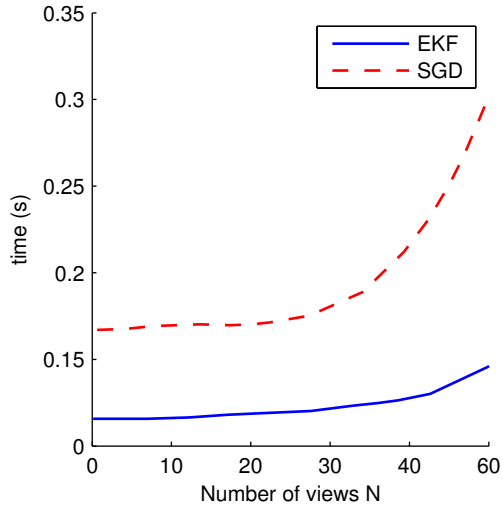


(b)

Figure 5.7: Figures 5.7(a) and 5.7(b), present results of SLAM using a SGD algorithm with real data. These map representations are formed by N=10 and N=20 respectively. The dash-dotted line represents the solution obtained with the SGD approach, the continuous line represents the ground truth whereas the odometry is drawn with dashed line.



(a)



(b)

Figure 5.8: Comparison results between SGD and EKF in a low non-linear noise scenario. Figure 5.8(a) presents RMS error against number of views N. Figure 5.8(b) presents time consumption against number of views N. The continuous line shows values for the solution provided by EKF, meanwhile the dashed line shows the error for the solution obtained with SGD.

Proposed SGD vs EKF under non-linear conditions

Now we intend to compare the behaviour of both methods in a more realistic situation, that is, when these methods are expected to suffer from non-linear error effects introduced by the observation measurements. This is one of the main contributions of our SGD proposal to the robustness of the SLAM methods, in contrast to traditional methods such as the EKF. As seen in 5.8(a), the EKF provides better speed of convergence and accuracy. Despite this fact, we aim to demonstrate that this SGD implementation becomes a more robust and stable method to work under worse case scenarios, such as those severely affected by non-linear noise effects.

Consequently, in this scenario we consider data association errors. We have conducted the same real experiments but forcing a highly relevant presence of non-gaussian errors. To that end, we have modeled a random generator scheme which introduces wrong data associations at each iteration step. Now the robot computes the observation measurements for the entire set of views which is able to observe, but it fails to associate the observation measurement with a corresponding view at a certain probability, meaning that a percentage out of the total data association is wrong. This fact implies that those observation measurement corresponding with wrong data associations will be wrong too.

Figure 5.9(a) and Figure 5.9(b) describe the RMS error tendency of both methods under such non-linear circumstances, which provoke the data association to fail at a given probability. The experiment has been repeated 200 times in order to retrieve consistent and coherent mean values. Again, the environment has been represented with different values of N in order to show differences. The results provided by the EKF in Figure 5.9(a) reveal that the resultant RMS error grows out of bounds when the probability of data association error is apparently low. This fact, contrarily to results shown in Figure 5.8, demonstrates the low reliability of the EKF when it has to deal with non-linearities and thus non-gaussian errors. Despite the fact that maps with more views provide a larger number of observation measurements to enable the rectification of the estimation (different colored lines), the error continuously increases (y-axis). These results prove that once the solution diverges, the EKF is unable to recover it, despite the fact that N can be higher. Consequently, the difficulties experienced by the EKF to keep the convergence of the estimation are evidenced.

Contrary to the EKF's results, and according to Figure 5.9(b), the SGD provides a lower RMS error under the same conditions. Moreover, it ensures convergence, as the RMS's tendency only increases slightly with the errors on the data association. It is worth noting the importance of selecting a suitable value for λ, so that new updates to x_{t+1} do not lead the estimation to diverge when there is evidence of errors. In this case, the SGD proves its capability to rectify the solution even in presence of non-linearities and thus non-gaussian errors. Therefore, and by contrast to the EKF, for the SGD, the more N views in the map, the more observations gathered, and thus the better results provided.

Finally, we can confirm the contribution of our SGD to provide a desirable robust and stable solution when dealing with environments of non-linear nature. As

for which method to use and its appropriateness, whether a EKF or a SGD, a trade-off solution must be agreed, depending on the requirements of the particular application, so as to ensure a balance between robustness against noisy terms (SGD) and speed of computation (EKF).

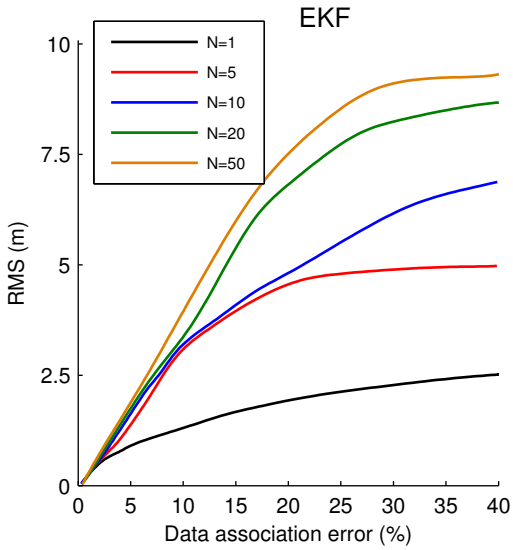
5.3 Conclusions

In this chapter we have presented an approach to the visual SLAM problem by introducing a SGD algorithm adapted to omnidirectional observations. The assumption of SGD has been aimed at reducing instabilities and harmful effects which compromise the convergence of the most extended SLAM algorithms, such as the EKF, which is especially sensitive to these effects. These erroneous circumstances are mainly consequences of the visual nature of the observation, which is non-linear, and particularly intensified on omnidirectional images. To that end, we have modified the standard SGD algorithm in order to integrate our unscaled observation model. Our proposed SGD model becomes more efficient, due to the design of a new strategy that exploits the information provided by several constraints simultaneously into the same SGD iteration, in contrast to the standard SGD algorithm.

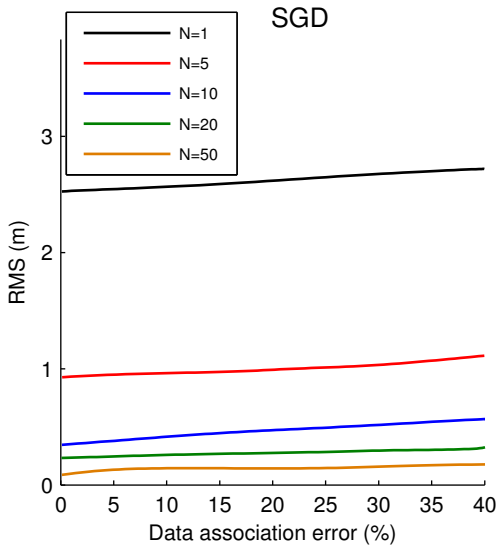
In order to confirm the validity and reliability of these contributions we have presented SLAM results with simulated and real data. We have also set a comparison framework to compare the proposed SGD method with the standard SGD algorithm and the EKF. The main issue to analyze has been the influence of non-linear errors, which are a clear indicator of added noise by the visual sensor's measurements, especially associated with our omnidirectional observation model.

Bearing in mind the results presented in this chapter, a key aspect to highlight about the SGD is the confirmation of its reliability to produce robust and stable solution which prevents the system from diverging. This is crucial when dealing with realistic environments under non-linear conditions. Despite the fact that the general SGD's performance in an idealistic situation is lower than the EKF's, the results obtained in presence of non-linear noise effects have evidenced the robustness of the SGD to provide a reliable situation. On the other hand, the EKF is highly sensitive to these kind of errors due to the linearization of the variables of the filter.

Therefore it has been proved that the effectiveness of each method depends on the assumed conditions. As a summary, we can conclude that if we intended to assure a SLAM approach to achieve the avoidance of the effects of non-linearities and non-gaussian errors, we would select a SGD method rather than the EKF. Nevertheless, in case of dealing with a more desirable situation, such as a low-noise environment, that would indicate that an EKF method would be more appropriated in order to succeed in providing a more precise solution with a higher rate of convergence.



(a)



(b)

Figure 5.9: Figures 5.9(a) and 5.9(b) presents the RMS error (m) against the probability of data association error (%) for EKF and SGD respectively. Error for maps with different number of views N are indicated.

6 Information-based SLAM Contributions

This chapter follows the line established in the previous chapters of this document, where new contributions to the field of visual SLAM have been presented. In this sense, we aim to propose new improvements to the main approach presented in Chapter 4, where we introduced a new map representation which benefits from the use of omnidirectional images, as an EKF view-based SLAM model. We have repeatedly exposed along this document that the presence of non-linear effects becomes one of the major risks for the convergence of an EKF-based SLAM system. Despite the fact that we have already proposed a different kind of solver algorithm in the previous chapter, here we seek to specifically contribute on the reinforcement of our EKF-based visual SLAM approach.

Particularly, our omnidirectional observation model induces a great part of such non-linear errors, thus becoming a potential source of uncertainty. In order to deal with this issue we propose a novel mechanism for the view initialization process which accounts for information gain and losses more efficiently. Please note that despite the fact that our EKF-based approach possesses a strategy to assess the similarity of the environment, as stated in (4.4), this is empirically suited at a prior stage and then particularly tuned for certain scenarios. This fact suggested us to seek a more reliable and general mechanism. Thus we come up with a contribution which confers a main outcome on the reduction of the map uncertainty. Therefore it achieves a higher consistency for the final estimation. Its basis relies on a Gaussian Process (GP) implementation to infer an information distribution model from the sensor data. This model aids in the representation of the probability of existence of feature points, and it also produces a specific representation of the visual information content, which is ultimately employed so as to define the new view initialization scheme, aimed at the

uncertainty reduction. In particular, the robot will initialize a new view whenever there is a high change in the inferred information distribution from the sensor data. In other words, whenever there are enough and relevant changes in the visual appearance of the environment.

In Chapter 3 we introduced the essentials of GPs as regression technique, together with information theory in order to support certain information-based aspects. Both are greatly profitable to enhance the uncertainty bounds of our SLAM approach. Within this context, the applications of non-parametric methods, such as GPs, have recently proven great enhancements on the mapping tasks for autonomous navigation. Continuous frontier maps are obtained by optimizing the process parameters, which reveal important uncertainty reduction [40, 41]. According to this, we propose the training of a GP as a tool to establish a bounded uncertainty scheme for our approach. By adopting such technique, we pursue a positive impact on the uncertainty, which we intend to minimize. As a result, the harmful effects that are likely to appear under high uncertainty conditions, such as errors induced by non-linearities and consequently instabilities and convergence difficulties, are mitigated. As a consequence, a more robust and consistent map and trajectory are obtained for the visual SLAM problem.

Summarizing, the fundamental aspects and contributions of this chapter may be listed as follows:

- A new view initialization mechanism for the map building process within our EKF view-based SLAM approach presented in Chapter 4.
- This strategy accounts for information gain and losses more efficiently.
- Probabilistic representation of features points and learning their correlations through Gaussian Processes regression.
- Bounding the uncertainty to the mitigation of harmful effects induced by non-linearities in the framework of EKF-based visual SLAM.

6.1 Sensor Data Distribution

GP has been introduced in this work in order to establish a sensor data distribution, which can be mapped into a global reference system. As already commented, GPs entail a non-parametric Bayesian regression method, which statistically infer the dependencies between points in a data set [111], in contrast to conventional functions which analytically relate inputs and outputs. As stated in (3.36), a GP can be denoted as $f(x)$, constituted by its mean, $m(x)$, covariance $k(x, x')$, and the training and test input vectors, x and x' respectively.

Having presented the fundamentals and the formulation of the GPs, then we are able to devise a model to represent our sensor data information distribution. The inference procedure through a GP takes the visual information gathered from the environment in the form of feature points detected on the image frame. Focusing on

the map building process described in Chapter 4, while the robot navigates, a certain observation measurement is performed at time t. Then, the feature points on the image corresponding with the current robot's pose are considered as our training data set, x_i, for the GP. The test points x_i' are determined by sampling uniformly the space defined in a global reference system. Finally, the GP returns the mean values μ_i and variances σ_i^2 inferred for these test points, as stated above. The most straightforward outcome of the GP's output is the probability of existence of a feature point at the locations specified by these test points.

There are several steps involved in the construction of the sensor data information distribution:

1. The feature points, $p_n(u, v)$, are locally processed on the camera reference system.
2. Then, $p_n(u, v)$ are back-projected into a global reference system, as $P(x, y, z)$, by means of the calibration parameters of the sensor [118].
3. Next, they become the input to the GP, which returns the probability distribution.
4. Ultimately, when new points are extracted from images acquired at new poses, the distribution is fused into the general information reference system.

At certain instant during the exploration tasks, we can expect relevant variations on the visual appearance along the environment. This fact implies the detection of new feature points which produce substantial changes on the information distribution representation. This poses a crucial point to be analyzed so as to assess the uncertainty variations. Hence our first intention was to apply this advantage to optimize the matching process, since feature points may be dealt with probability measurements as target (even combined with visual descriptors). However, this promising idea was refused due to reasons such as:

- Expensive computation resources to apply GP regression over images with large number of feature points.
- Lack of scale. The matching is carried out on the image plane (up to a scale factor). However, GP regression intends to return probability on the XY plane of the 2D general reference system.

As a consequence, we redirected our work to the implementation of GP regression exclusively aimed at obtaining a bounded uncertainty scheme for the map representation.

Figure 6.1 shows a real data example for this sensor data distribution. The GP produces such distribution in terms of probability of existence, which is associated to a bunch of feature points. In particular, it represents the 2D spatial coordinates of

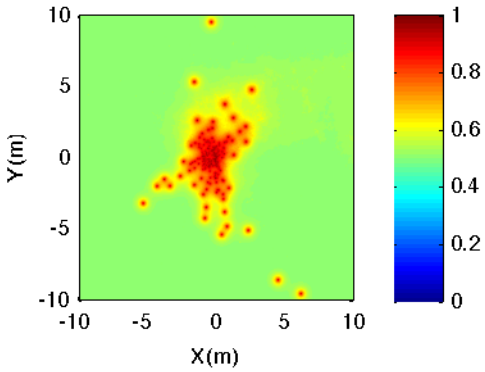


Figure 6.1: Sensor data information distribution: probability of existence of feature points on the 2D reference system.

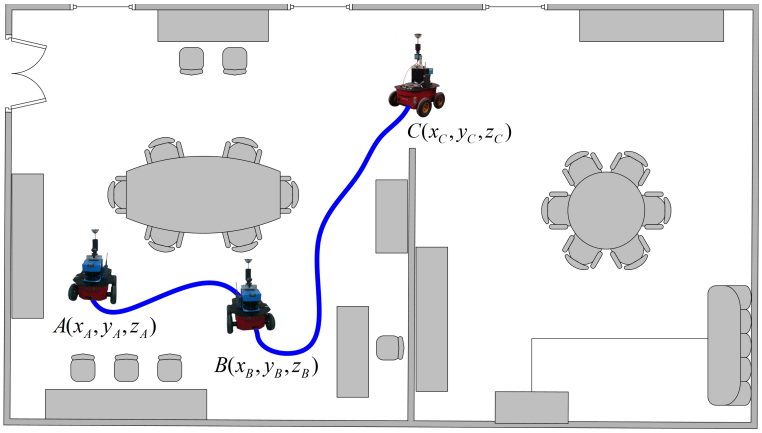


Figure 6.2: Map building process. The robot explores the environment while simultaneously initializes image views in the map at poses A, B and C.

a certain scenario. After GP regression, μ_i and variances σ_i^2 are inferred for the test points. This permits to represent the probability distribution, where each feature point may be identified by its probability of existence, expressed in the normalized range [0-1], as indicated in the legend.

In addition to the last example, Figure 6.2 presents another explicative illustration where the robot explores the environment, whilst the visual information varies along the trajectory. This causes that new feature points are detected and so the GP output varies too. It can be seen that poses $A(x_A, y_A, z_A)$ and $B(x_B, y_B, z_B)$ are relatively close enough so that the scene should be quite similar, and thus many feature points are matched between images, since they remain invariant to these poses. Contrarily, when the robot approaches the second room, the visual appearance of the environ-

ment is very likely to change substantially. In consequence, at pose $C(x_C, y_C, z_C)$, new feature points are detected with respect to images at poses A and B.

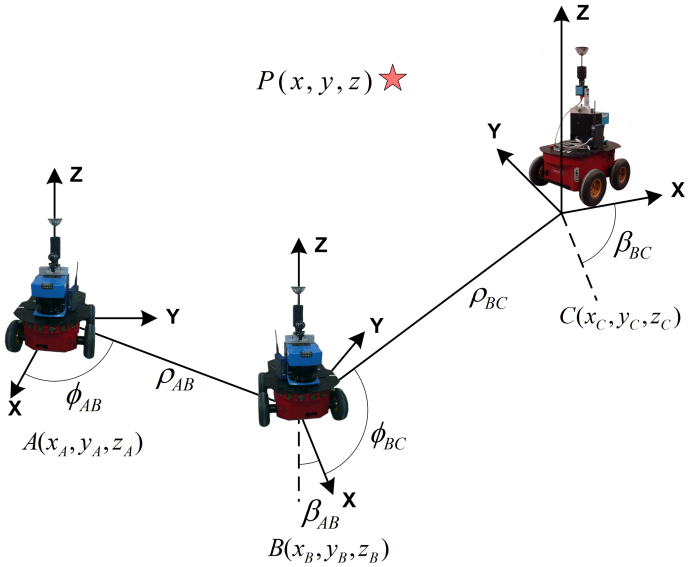
A further description to the last example is provided in Figure 6.3. Figure 6.3(a) represents the motion transformation between poses A, B and C, while Figure 6.3(b) shows the images acquired at these same poses. The feature points are projected on the image plane and indicated with crosses. The green crosses mark the matched points between images and the blue crosses the new feature points detected. These new points evidence the variation on the visual appearance of the environment at pose C. Figure 6.4 illustrates this last fact as a variation on the information distribution on the GP framework. Figure 6.4(a), Figure 6.4(b) and Figure 6.4(c) represent the probability of existence of feature points at poses A, B and C respectively. Thus the evolution of the sensor data information distribution along these poses can be noticed. Please also note that a noticeable variation appears between poses B and C. By contrast, between A and B the visual information has remained similar. Therefore there are overlapped areas with high probability between Figure 6.4(a) and Figure 6.4(b), which mean that some feature points have been repeatedly detected from these poses.

6.1.1 Uncertainty Reduction

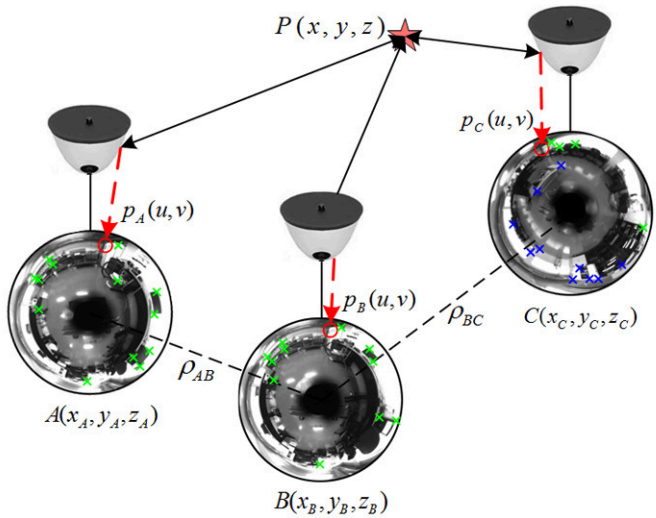
Once established the information distribution of our sensor data, we can exploit this tool in order to maintain the uncertainty bounded. In the previous example, the key idea to highlight is that, the visual information at the current's robot pose and at a new pose, is more likely to overlap when these poses are close. This implies that a larger number of feature points are observed and matched from these poses, and so the probability of existence is definitely high. Thus the information distribution remains mostly unvaried. By contrast, the information varies considerably when the robot discovers unknown areas, and then the corresponding points decrease dramatically, due to the fact that the visual appearance differs considerably between images.

According to this, we can propose an efficient map building process in terms of uncertainty. We seek to analyze these variations of visual information in order to decide the initialization of new views in the map. Hence we intend that every new view encodes the most relevant visual changes in the environment, according to their visual informative characteristics. So that the main contribution expected is the reduction of the total uncertainty of the estimated map.

Once said that, the objective is to propose a metric which accounts for these effects. So the arrangement of new views will be efficiently accomplished by means of the definition of a new initialization ratio, formerly named similarity ratio and presented in (4.4). In order to define the new ratio, we adopt the tool known as Kullback-Leibler divergence (KL) [74], already presented in (3.39) and (3.40), which is also sustained by the concept of entropy (3.38). KL is commonly known as Information Gain within the probability theory field, since it expresses the mutual information of a system, that is, the change in the information entropy from a prior state to the following. Thus its purpose is to evaluate the fluctuation expected in the entropy when a new sample set is introduced to a certain distribution.

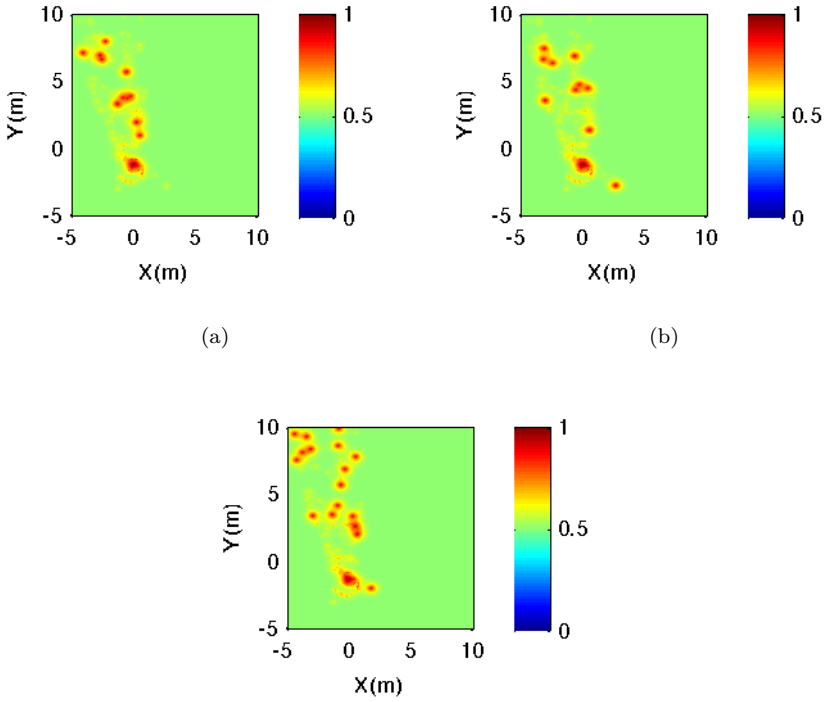


(a)



(b)

Figure 6.3: Detailed description of example presented in Figure 6.2: Figure 6.3(a) represents the motion transformation between poses A, B and C. Figure 6.3(b) shows the images acquired at A, B and C, where the projection of $P(x, y, z)$ on every image is indicated as $p_A(u, v)$, $p_B(u, v)$ and $p_C(u, v)$ respectively. Feature points matched between images are plotted with green crosses whereas the new feature points are plotted with blue crosses.



(a) (b)

(c)

Figure 6.4: Evolution of the sensor data information distribution along poses A, B and C, as described in the example presented in Figure 6.2: Figure 6.4(a), Figure 6.4(b) and Figure 6.4(c) correspond to A, B and C respectively. This sequence expresses the variation on the probability of existence of feature points on the 2D reference system.

In this context, we use the entropy to measure the uncertainty associated to the feature points given by our GP in terms of probability of existence. The KL divergence represents the change of entropy between the information distribution of the current feature points, observed until pose at t, F_1, and the new inferred feature points in the next pose at $t+1$, F_2, from new images. In other words, the higher value of KL divergence means that the newly introduced feature points are less similar, due to a considerable different visual appearance. Thus there is a higher amount of new visual information discovered by the robot. Consequently, the uncertainty on the estimated map will increase too. The structure to evaluate the KL divergence is:

$$H(F_1) = -\sum_i F_1(i) \log F_1(i) \tag{6.1}$$

$$KL(F_1 \parallel F_2) = H(F_1, F_2) - H(F_1) = \sum_{i=1}^{N} F_1(i) \log \frac{F_1(i)}{F_2(i)} \tag{6.2}$$

where $H(F_1)$ is the entropy of the information distribution of the current feature point set at time t, with the index $i \in [1, \ldots, N]$, being N the total number of feature points. $F_1(i)$ represents the probability of existence of the current i-feature points at time t and so does $F_2(i)$ with new points added at time $t+1$ from a new image. The relevance of their information contribution is directly proportional to $1/\sigma_i^2$.

The strategy to initialize a new view seeks to define an upper bound for the uncertainty, so as to get an efficient map in this sense. To that aim, since we keep the information distribution of the points referred to a global system, we consider the KL value in its accumulative format. Then, we accumulate measurements, given by the addition of new visual information at new poses. Finally, we can define the new initialization ratio as:

$$\gamma = \sum_t KL(P_t \parallel P_t + 1) \tag{6.3}$$

where P_t refers to the data information distribution obtained up to time t and $P_t + 1$ refers to the new data information, which is fused into the global reference system at time $t+1$. Note that both express probability of existence of feature points.

Establishing different thresholds for γ leads us to obtain different view initializations and thus different map estimations. Obviously, the associated uncertainty also fluctuates differently depending on the placement of the views. Whenever γ exceeds a certain threshold, a new view is initialized. A more detailed explanation with real results is presented in the next section.

In the end, with this approach, the final estimation benefits from this idea since any new view is initialized at an optimum pose in terms of uncertainty. The arrangement of new views now assures that the uncertainty of the estimation is bounded. This proposal reinforces the value that comes along with our view-based approach: major information changes on the environment are encoded by new views in the map. Figure 6.5 presents the implementation of this new contribution, embedded in the EKF-based approach presented in Chapter 4. The following section presents real data results to confirm the benefits of this approach.

6.2 Results

We have performed two different sets of real data experiments in an office-like environment in order to examine the behaviour of this proposal in terms of its associated uncertainty. We also provide different map solutions obtained with this enhanced EKF-based visual SLAM approach. All the set of results presented here are also compared with our former SLAM approach, detailed in Chapter 4, which does not use GP nor data information distribution in order to initialize views.

6.2.1 Initialization Ratio and Sampling Resolution

The first experimental set intends to evaluate under different conditions the new mechanism of view initialization. The first parameter to study is the threshold value for the

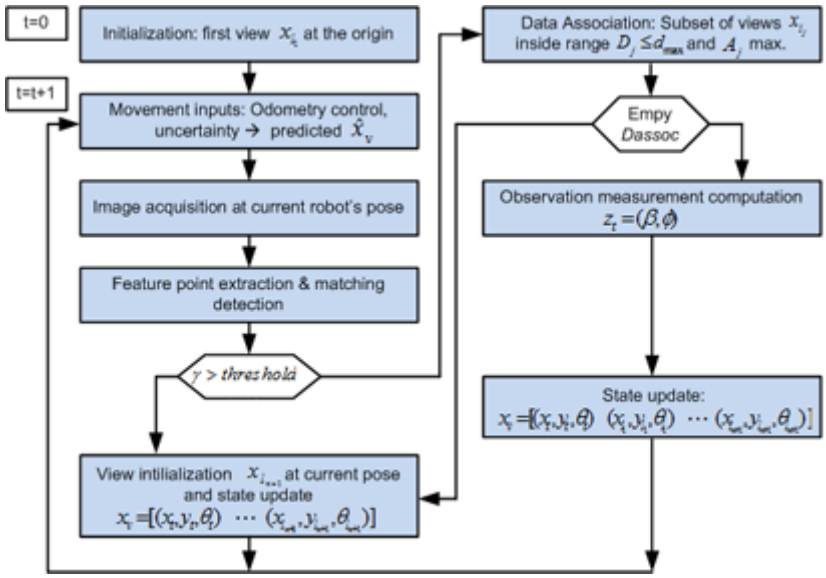


Figure 6.5: Block diagram summary for the EKF-based visual SLAM approach, with GP regression and Information-based view initialization for the uncertainty reduction.

new initialization ratio γ (6.3). Intuitively, the higher threshold for γ, the less views are initialized in the map. Thus implies that huge changes on the visual environment are encoded by less views, without inducing any new initialization. Please note that, from now on, uncertainty values have been computed as:

$$\sigma^2_{experiment} = trace[P(t) \cdot P(t)^T] \tag{6.4}$$

where $P(t)$ is the current covariance matrix at time t.

A real experiment has been conducted in a scenario with dimension 25×25m. Figure 6.6 presents the current uncertainty along the path followed by the robot at each time step. It can be confirmed that low values of the initialization threshold aid in the reduction of uncertainty. It is worth noting that the results provided by this proposal outperform the uncertainty associated with the former visual SLAM approach at every case. The main reason for this to be a more feasible mechanism is that we account for information gains and losses rather than the amount of feature points matched, as in the former initialization ratio (4.4). Figure 6.7 shows the mean value of the uncertainty accumulated over the total map, at each time step. Obviously, the shape and the evolution is quite similar, however the map uncertainty is very likely to be higher than the poses', since it computes the mean values of the entire set of uncertainties associated with all of the variables of the map, up to time t. That it is to say, the current uncertainty of the set of views stored in the map, and the trajectory traversed by the robot.

The results obtained with this approach confirm a better performance with regards to the uncertainty. This means that the view initialization strategy accomplishes

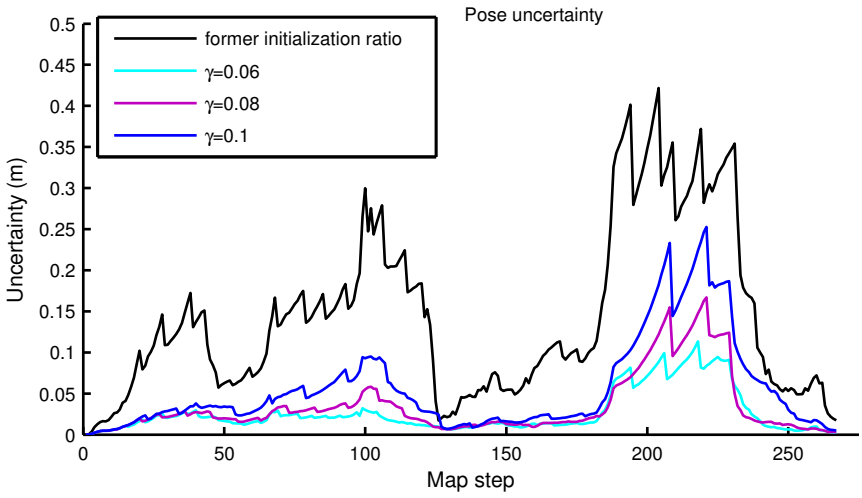


Figure 6.6: Evolution of the uncertainty along the robot's path. Different threshold values for γ are shown and compared to the uncertainty obtained with the former initialization ratio (4.4), employed in Chapter 4.

with the proposed scheme for obtaining a bounded uncertainty. Nonetheless, it is worth mentioning that low uncertainty values imply larger number of views in the map, and obviously a higher computational cost. Hence a trade-off solution is needed, which usually depends on the specific application.

Secondly, it is necessary to state the same analysis but aiming at the accuracy. To that purpose, we extract values of RMS error. Figure 6.8 plots RMS values associated with the different initialization ratios γ tested. Once again, the obtained error with this contribution is lower at any case, in contrast to the former approach.

Finally, another parameter which has a considerable importance on the efficiency of this approach is the sampling size for the test points selection. The global reference system is sampled uniformly by means of these test points x'. Then, the data information distribution inferred by the GP have a specific resolution which is directly linked with this sampling size, which is determined by x'. Now, Figure 6.9 represents the RMS error when the sampling size is varied. It can be observed that higher resolutions ensure better results since the probability areas are more precisely determined. However, a high resolution inference from the GP becomes very expensive in terms of computation. It is worth noting that the dimension of the grid is up to scale, according to the scale factor of the map.

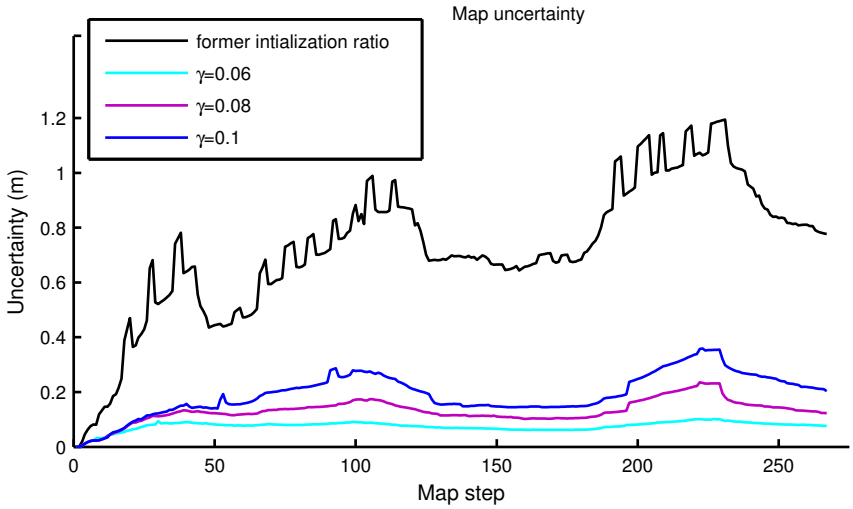


Figure 6.7: Evolution of the mean uncertainty accumulated on the total map. Different threshold values for γ are shown and compared to the uncertainty obtained with the initialization ratio (4.4) employed in the former SLAM approach.

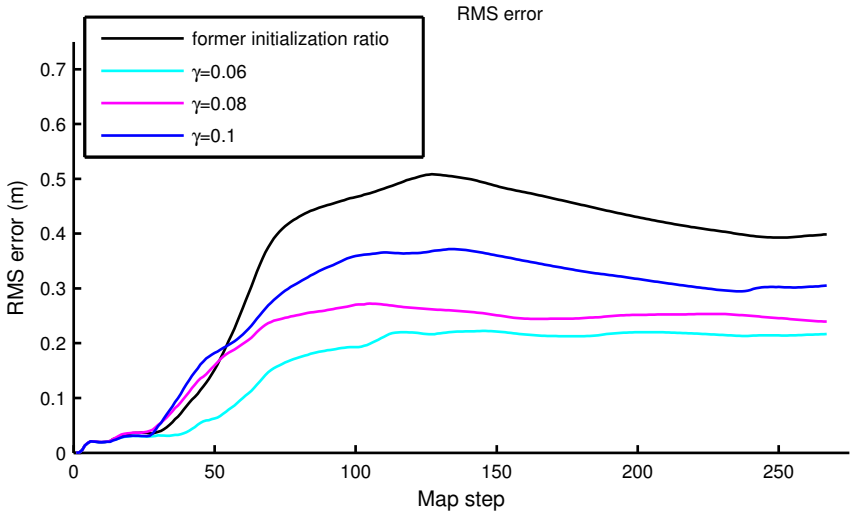


Figure 6.8: RMS error for different initialization ratios γ. The RMS value obtained with the former SLAM approach has been also plotted for comparison.

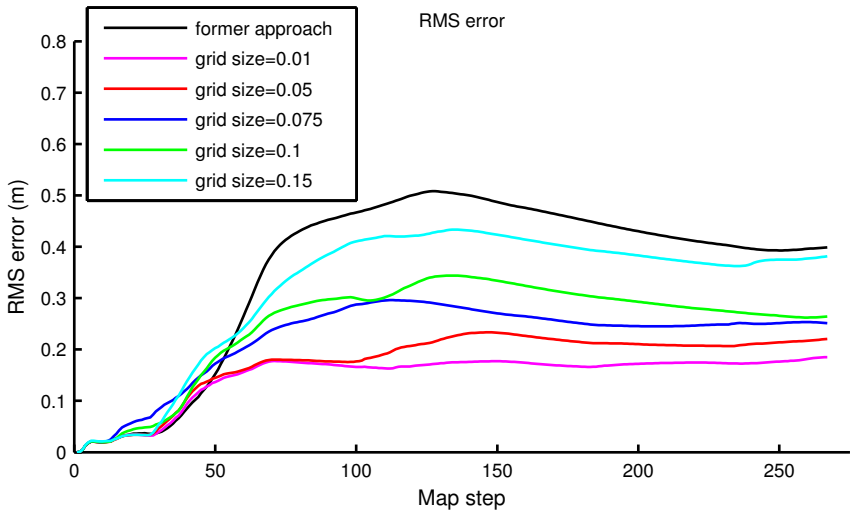
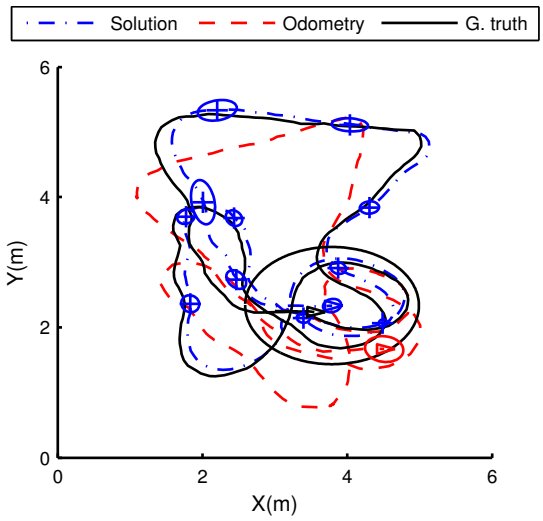


Figure 6.9: RMS error for different grid size resolutions. The grid size resolutions are expressed up to the scale factor of the current map. The RMS value obtained with the former SLAM approach has been also plotted for comparison.

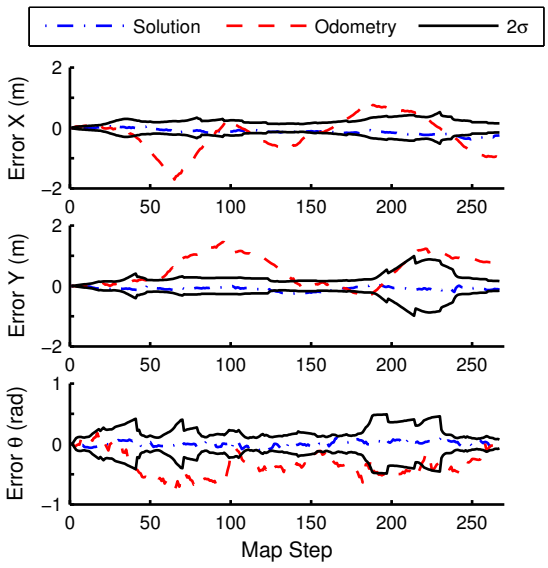
6.2.2 Map Building with Uncertainty Reduction

After having assessed the behaviour of γ and the RMS obtained, we can assume that a trade-off threshold must be set from a study, such as the one just presented. Hence now we can carry out a complete SLAM experiment. Figure 6.10 shows the final map and path estimation for an office-like environment. Figure 6.11 presents a different solution for the same scenario, where a different initialization ratio is considered. In order to compare and prove the benefits of this proposal, Figure 6.12 presents results obtained with our former EKF-based SLAM approach. Inspecting Figure 6.10(a) and Figure 6.11(a) confirms that lower thresholds for the initialization ratios ensure a more robust solution with a larger number of views in the map, but obviously at a higher computational cost. Figure 6.10(b) and Figure 6.11(b) show the behaviour of the error along the path. Both estimations confirm their improvements in comparison with the former approach, as seen in Figure 6.12(a) and Figure 6.12(b). An important reduction in terms of uncertainty is achieved.

Finally, the method has been used in a larger scenario with the aim of testing its robustness and feasibility in large environments. Figure 6.13 provides the details of this scenario, which corresponds to an indoor trajectory of 180 m. The areas where the robot navigates consist of office-like rooms, laboratories, corridors and open spaces. The main challenge is to deal with the big changes on the visual appearance between rooms, but also with the lighting changes on the images. Some omnidirectional images are also presented, as well as the real path followed by the robot. Figure 6.14 provides results for this scenario presented in Figure 6.13, when using the proposed approach. Again, the estimated path and map reveal their accuracy and similarity to the real path, but also its reduced uncertainty. Figure 6.15 illustrates the evolution of the pose and map uncertainty respectively.

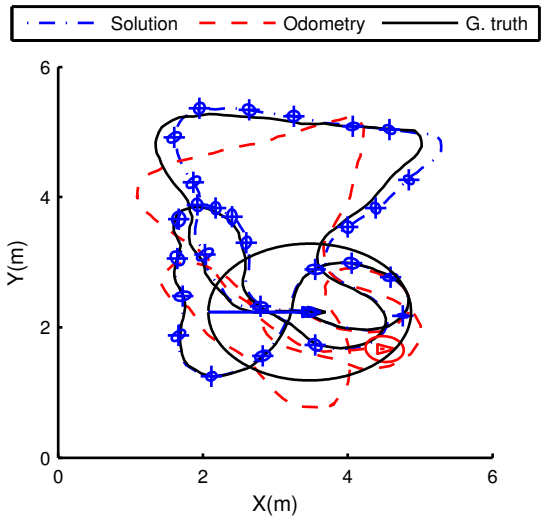


(a)

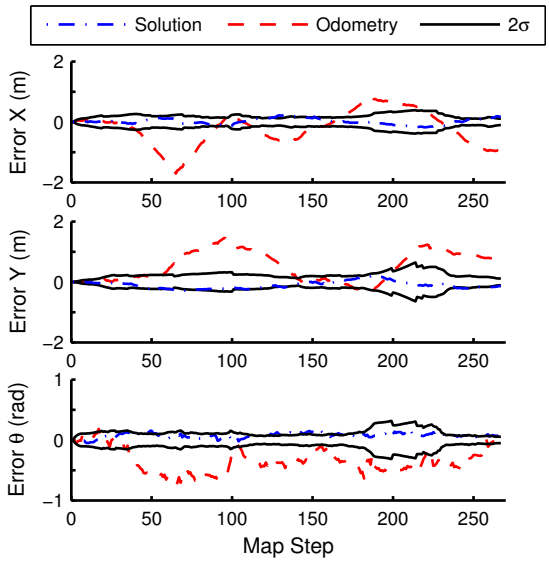


(b)

Figure 6.10: Figure 6.10(a) presents real data results obtained with uncertainty reduction in the EKF-based SLAM approach. The map of the environment is formed by N=12 views. The position of the views is presented with error ellipses. Figure 6.10(b) shows the estimation and the odometry error in X, Y and θ at each time step.

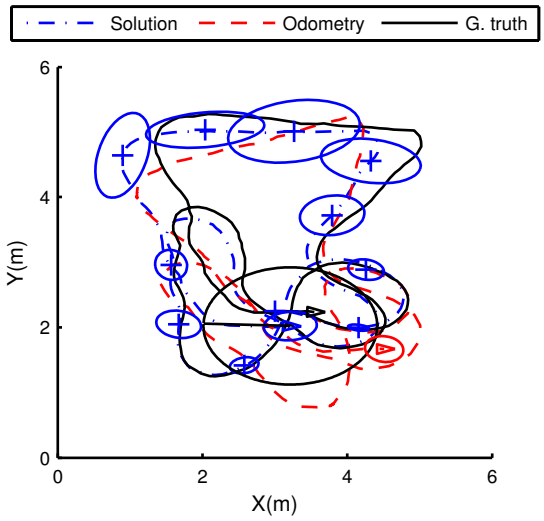


(a)

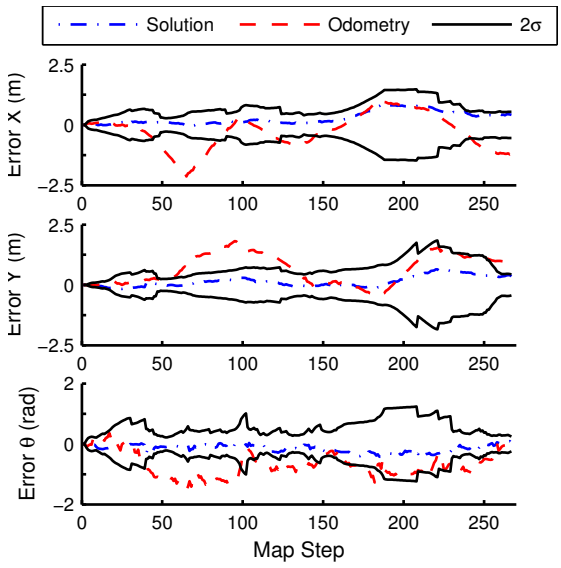


(b)

Figure 6.11: Figure 6.11(a) presents real data results obtained with uncertainty reduction in the EKF-based SLAM approach. The map of the environment is formed by N=28 views. The position of the views is presented with error ellipses. Figure 6.11(b) shows the estimation and the odometry error in X, Y and θ at each time step.



(a)



(b)

Figure 6.12: Figure 6.12(a) presents real data results obtained with the former EKF-based SLAM approach, detailed in Chapter 4. The map of the environment is formed by N=11 views. The position of the views is presented with error ellipses. Figure 6.12(b) shows the estimation and the odometry error in X, Y and θ at each time step.

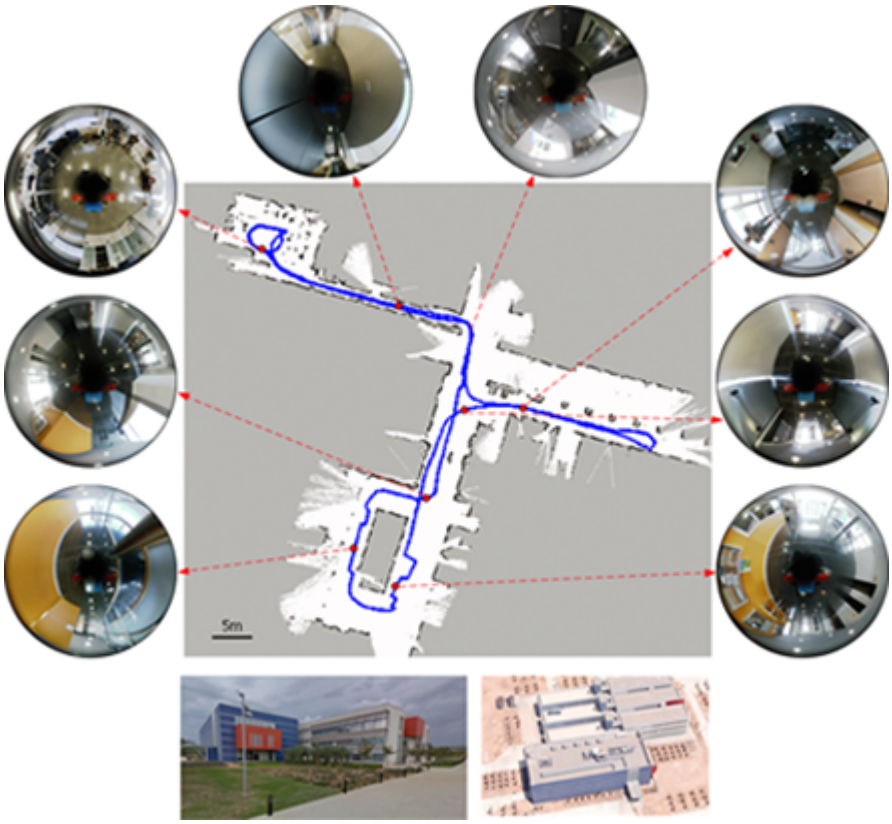
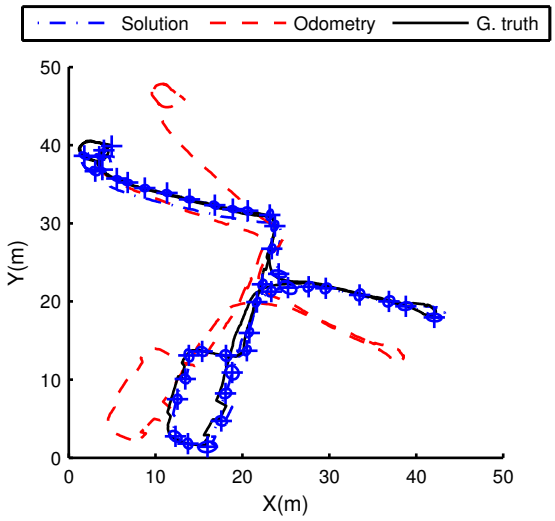
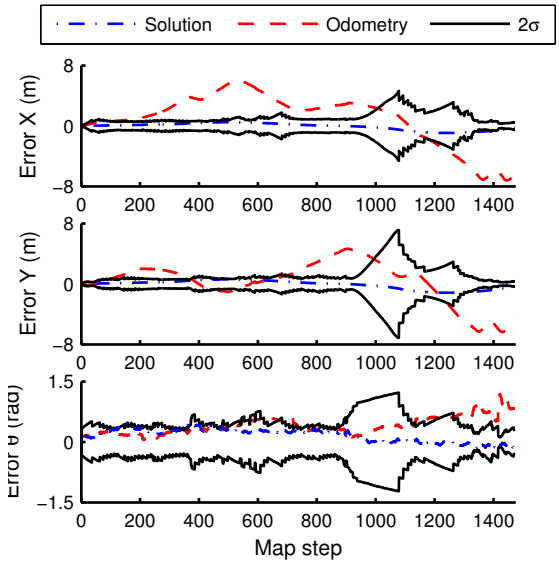


Figure 6.13: Main details of the large scenario where the last dataset was acquired. The layout of the building, real path followed by the robot and some omnidirectional views of different areas are indicated.



(a)



(b)

Figure 6.14: Real data results obtained with uncertainty reduction in the EKF-based SLAM approach for a large scenario presented in Figure 6.13. The map of the environment is formed by N=41 views. The position of the views is presented with error ellipses. Figure 6.14(b) shows the estimation and the odometry error in X, Y and θ at each time step.

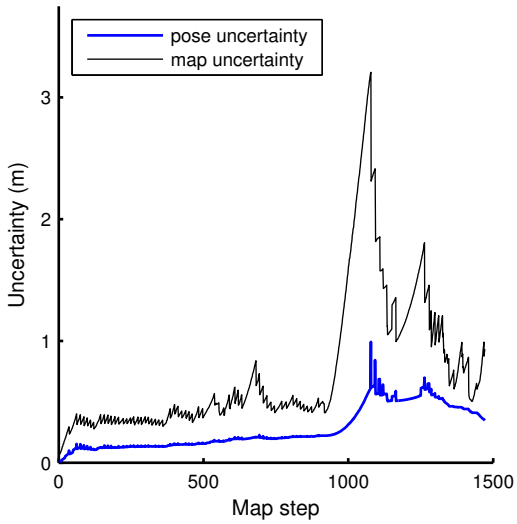


Figure 6.15: Evolution of the pose and map uncertainty for the large scenario presented in presented in Figure 6.13.

6.3 Conclusions

In this chapter we have presented a new contribution consisting of a mechanism for the view initialization within the map building process of our EKF-based visual SLAM approach supported by omnidirectional views. This contribution has proven a more feasible strategy which accounts for information gain and losses so that the harmful effects suffered by visual SLAM approaches are mitigated. Particularly, we tackled the non-linearities and undesired effects induced in the observation and movement, which jeopardize the convergence of traditional EKF-based SLAM approaches. The strategy achieves the uncertainty reduction to deal with these issues.

In this implementation we have focused on encoding information gain and losses to define the proposed mechanism to improve the view initialization stage. This new contribution has been achieved by means of the a data information distribution inferred with a Gaussian Process. This distribution represents a probability model for the existence of feature points, and it is exploited from an informative point of view. Thus an Information Gain method has been finally introduced to come up with the desired initialization process, which confirms its capability to bound the uncertainty and to efficiently initialize new views in the map. The results presented have proven the validity of this proposal and the expected benefits with regard to uncertainty reduction. Thus implies a more robust and consistent map and trajectory estimation. Similarly, these results demonstrate the effectiveness of this approach to set limits to the error. In order to reinforce the value of these results and the contributions made in this chapter, we have also compared them with the results obtained by our former EKF-based SLAM approach, presented in Chapter 4, which employs a more empirical initialization mechanism.

7 Conclusions and Future Work

Having detailed in the previous chapters all the work conducted under the framework of this thesis, now this chapter summarizes the main contributions that can be extracted from this research. In consequence, some possible future work can be also listed.

7.1 Contributions

Nowadays, research on mobile robotics has concentrated on several challenges of paramount importance for this field. Building visual maps is crucial in order to provide the robot with a balanced capability between autonomy, perception, adaptability and decision-making. Such task to obtain a feasible map estimation implies a real and complex challenge, with incremental and simultaneous nature: the SLAM (Simultaneous Localization And Mapping) problem. This thesis establishes its motivation under this framework. Accordingly, the main objectives were aimed at the development of a visual SLAM solution which exploits the benefits of an omnidirectional camera and the feature point information, extracted from the corresponding images. According to these objectives, different advances and contributions were derived, as presented separately in each chapter of this document. The general target points were divided into: new map representations, non-linearities mitigation, and uncertainty reduction. This section includes a synthesis with the most relevant contributions and achievements accomplished during the research period:

Chapter 2

- Adoption of the epipolar constraint to the geometry of our omnidirectional camera. This first accomplishment allows to design a motion transformation model.

- Development of the motion transformation model, solely based on omnidirectional images. It aids to define a simple angular observation model which is able to extract the localization of the robot by means of a pair of images, corresponding to physical poses of the robot.
- Performance and accuracy results that validate the motion transformation model. Besides they allow to provide with a reliable visual odometry approach, as a visual feed-forward input for a real time application.

Chapter 4

- Compact map representation to encode the environment with a reduced set of omnidirectional views, contrarily to traditional approaches which accumulate and re-estimate large amounts of visual landmarks. This represents the core of our view-based SLAM proposal.
- Enhanced observation model thanks to the motion transformation contribution, but also achieved with an improved matching process which accounts for the current uncertainty of the system.
- Map building design based on the information provided by the omnidirectional views and adapted to this geometry: view initialization and data association redesign.
- Acquisition of real datasets as a consistent background for testing and ensuring validity of the contributions made in this work.

Chapter 5

- Alternative offline core algorithm implementation. A modified SGD solver is adapted to the omnidirectional reference system. In contrast to former solvers, this contribution reinforces the robustness against non-linearities.
- Simultaneous processing of several constraints at the same time step. This improves the convergence and robustness of the SGD estimation, in contrast to standard SGD approaches.
- Comparison results to demonstrate the improvements under non-linear noise conditions.

Chapter 6

- View initialization mechanism based on an Information-based scheme that accounts for information gain and losses within the SLAM approach. It is sustained by Bayesian techniques such as GP in order to obtain a probabilistic distribution of our sensor data. Information theory complements this contribution.
- Bounded uncertainty system, in consequence with the previous point. The robust view initialization assures a limited uncertainty which prevents the system from diverging. Thus harmful noise effects are mitigated.

7.2 Future Work

Following, we describe possible future research lines which emerge as a consequence of the contributions and results presented in this work:

- Further study on this map approach when it is driven by different sort of algorithms. A comparison benchmark would be necessary in order to assess possible benefits from the use of particle filters, non-linear solvers and maximum likelihood optimizers.
- Analysis on the visual feature detectors and descriptors when they operate embedded in a final SLAM application. The conclusions extracted would help in the definition of a fused method which optimizes the final path and map estimation.
- Extension of the motion transformation in a 6 degrees of freedom (6 DOG) context. In general, these kind of models considerably increase the complexity of the problem. Nonetheless, extending our motion transformation to 3D would represent a simple and powerful tool to come up with a reliable and robust 6 DOG movement model.
- According to the last point, 3D visual map estimations would be the next line to work on. A 6 DOG movement model permits to devise a 3D representation of the environment. This task implies that the robot is enabled with a proper acquisition platform.
- Map building at large outdoor scenarios. The emergence of drones represent a great choice to easily acquire real outdoor datasets. GPS and IMU data provide a feasible input in order to be combined with the visual information of an omnidirectional camera.

Appendix: Set of Publications

The major implementations and contributions made in this thesis are supported by a set of publications in journals ranked in the JCR Science Edition. The following journal papers support the work conducted in this document:

Journal Paper 1

Creación de un modelo visual del entorno basado en imágenes omnidireccionales. [46]
A. Gil, D. Valiente, O. Reinoso, J.M. Marín
Revista Iberoamericana de Automática e Informática Industrial, RIAI. Vol 9. 2012
ISSN: 1697-7912. Ed. Elsevier.
JCR-SCI Impact Factor: 0.475, Quartile **Q4**.

Journal Paper 2

A modified stochastic gradient descent algorithm for view-based SLAM using omnidirectional images. [136]
D. Valiente, A. Gil, L. Fernández, O. Reinoso
Information Sciences. Vol 279. 2014
ISSN: 0020-0255. Ed. Elsevier.
JCR-SCI Impact Factor: 3.364, Quartile **Q1**.

Journal Paper 3

A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images. [135]
D. Valiente, A. Gil, L. Fernández, O. Reinoso
Robotics and Autonomous Systems. Vol 62. 2014
ISSN: 0921-8890. Ed. Elsevier.
JCR-SCI Impact Factor: 1.618, Quartile **Q2**.

Reprints of the publications are appended below:

SciVerse ScienceDirect
Disponible en www.sciencedirect.com

Revista Iberoamericana de Automática e Informática industrial 00 (2012) 1–12

www.elsevier.es/RIAI

Creación de un modelo visual del entorno basado en imágenes omnidireccionales

A. Gil*, D. Valiente, O. Reinoso, J.M. Marín

Universidad Miguel Hernández de Elche.
Avda. de la Universidad s/n, Ed. Quorum V
03202 Elche (Alicante), España

Resumen

En este artículo abordamos el problema de la construcción del mapa visual de un entorno mediante un robot móvil, ubicándose, por tanto, en el ámbito del problema de SLAM (*Simultaneous Localization and Mapping*). La solución presentada en este artículo se fundamenta en el uso de un conjunto de imágenes para representar el entorno. De esta manera, la estimación del mapa se plantea como el cálculo de la posición y orientación de un conjunto de vistas omnidireccionales obtenidas del entorno. La idea desarrollada se separa de la concepción habitual de mapa visual, en la que el entorno está representado mediante un conjunto de puntos tridimensionales definidos respecto de un sistema de referencia. En el caso presentado, se considera que el robot está equipado con un único sensor visual omnidireccional que permite detectar un conjunto de puntos de interés de las imágenes que, a continuación, son representados mediante un descriptor visual. El proceso de construcción del mapa se puede resumir de la siguiente manera: cuando el robot se mueve por el entorno captura imágenes omnidireccionales y extrae un conjunto de puntos de interés de cada una de ellas. A continuación, se buscan correspondencias entre la imagen capturada y el resto de imágenes omnidireccionales existentes en el mapa. Si el número de correspondencias encontradas entre la imagen actual y alguna de las imágenes del mapa es suficiente, se calcula una transformación consistente en una rotación y translación. A partir de estas medidas podemos deducir la localización del robot con respecto a las imágenes almacenadas en el mapa. Se presentan resultados obtenidos en un entorno simulado que validan la idea presentada. Además, se han obtenido resultados experimentales utilizando datos reales que demuestran la validez de la solución presentada.

Palabras Clave: SLAM, robótica móvil, visión omnidireccional

1. Introducción

La construcción de mapas es uno de los problemas fundamentales en el área de la Robótica Móvil, ya que una gran cantidad de aplicaciones se fundamentan en la existencia de un mapa (Aracil et al., 2008). Para construir el mapa, el robot debe desplazarse por el entorno mientras adquiere información de él. Frecuentemente, la información de la que dispone el robot para realizar el mapa consiste en un conjunto de lecturas de odometría y un conjunto de poses. La naturaleza acumulativa del error existente en la odometría implica un problema de localización, en consecuencia, se genera el problema de construir un mapa mientras, simultáneamente, el robot se localiza dentro de él. Por tanto, el conjunto de algoritmos desarrollados para esta tarea se agrupan bajo las siglas de SLAM (*Simultaneous Localization and Mapping*).

En la literatura existen un gran número de trabajos que proponen la utilización de cámaras para la construcción de mapas. Estas soluciones se denominan generalmente SLAM visual. A su vez, en este grupo podemos encontrar diversas alternativas que se diferencian en aspectos como el tipo de cámara utilizada, ya sea una única cámara, un par estéreo o una única cámara omnidireccional. Otro factor diferenciador es el tipo de información visual extraída de las imágenes. También se clasifican según extraigan *landmarks visuales* de las imágenes o utilicen un descriptor de apariencia global de las imágenes. En el caso de descriptores visuales, se encuentra un gran número de soluciones basadas en descriptores SIFT (Lowe, 2004) y SURF (Bay et al., 2006) en el ámbito del SLAM visual. La representación utilizada para definir el mapa: en este caso, se encuentran métodos que representan la posición 3D de un conjunto de *landmarks* visuales (Civera et al., 2008; Andrew J. Davison et al., 2004), o bien métodos que estiman un subconjunto de las poses del robot (Paya et al., 2009). El algoritmo de SLAM

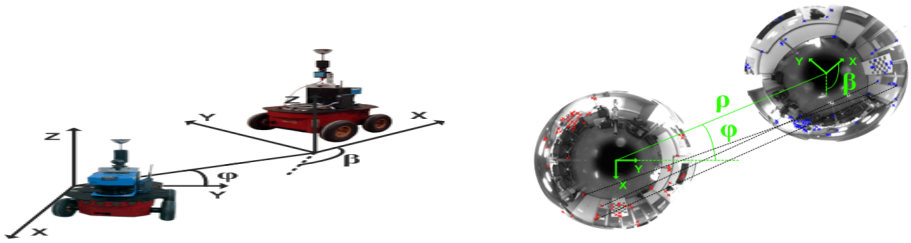
*Autor en correspondencia.
Correos electrónicos: arturo.gil@umh.es (A. Gil), dvaliente@umh.es (D. Valiente), o.reinoso@umh.es (O. Reinoso), jmarin@umh.es (J.M. Marín)
URL: arvc.umh.es (A. Gil)

utilizado: principalmente se encuentran soluciones basadas en el filtro de Kalman, filtros de partículas. Por ejemplo, en (Gil et al., 2006) se utiliza un par estéreo de cámaras calibradas para obtener medidas relativas de distancia a un conjunto de marcas visuales. El mapa está definido por un conjunto de marcas visuales, estando cada una acompañada de un descriptor visual basado en la transformada SIFT (Lowe, 2004). Se emplea un algoritmo basado en un filtro de partículas *Rao-Blackwell* para estimar el mapa y el camino seguido por el robot (Montemerlo et al., 2002). Una solución diferente la encontramos en (Civera et al., 2008), donde se utiliza una única cámara para construir un mapa tridimensional del entorno, constituido por un conjunto de puntos de interés extraídos con el detector de esquinas de Harris (Harris and Stephens, 1988) y descritos por una subventana de niveles de gris. Se capturan imágenes con gran frecuencia mientras la cámara es movida a mano. Cada uno de los puntos 3D detectados se representa mediante un vector adimensional y una escala. La posición 3D de los puntos se estima con bastante precisión en base a un filtro EKF al observar las marcas visuales desde puntos de vista separados por una línea base suficiente. Debido a que una única cámara no nos permite obtener observaciones de la distancia hasta los puntos detectados, la inicialización de la posición tridimensional de las *landmarks* plantea un problema. Este hecho inspiró una parametrización inversa de la profundidad para representar los puntos en el filtro de Kalman (Civera et al., 2008). Según (Andrew J. Davison et al., 2004) los resultados de SLAM visual utilizando una única cámara son mejores cuando se utiliza una óptica con un gran ángulo de visión, hecho que sugiere la utilización de una cámara omnidireccional para la creación del mapa, ya que, en este caso el ángulo de visión horizontal es máximo. Sin embargo, el empleo de cámaras omnidireccionales en aplicaciones de SLAM visual no es demasiado frecuente. Por ejemplo, (Joly and Rives, 2010) estiman con una única cámara omnidireccional y una variación del *Information Filter* estando cada punto modelado mediante una parametrización inversa de la profundidad (Civera et al., 2008). En (Jae-Hean and Myung Jin, 2003) dos cámaras omnidireccionales se combinan para obtener un sensor estéreo con un gran ángulo de visión. Las medidas de distancia obtenidas se integran en un filtro EKF para construir el mapa. En (Scaramuzza et al., 2009) se presenta un método para extraer el movimiento relativo entre dos imágenes omnidireccionales. En este caso, los resultados no se emplean para construir un mapa sino para estimar una odometría visual.

En el caso presentado aquí, consideramos el caso en el que un único robot explora el entorno. El robot está equipado con una única cámara omnidireccional, según se muestra en la Figura 1(a). Cuando el robot se mueve por el entorno, captura imágenes omnidireccionales y extrae un conjunto de puntos de interés de ellas. A continuación, busca correspondencias con el resto de imágenes omnidireccionales existentes en el mapa. Si se encuentra un número suficiente de correspondencias entre las imágenes, se calcula una rotación y translación (salvo un factor de escala) entre ambas imágenes (Scaramuzza et al., 2009). Estas medidas se integrarán en un filtro de Kalman extendido (EKF) para deducir la localización del robot en el mapa, así como la posición del robot cuando capturó cada una de las imágenes. El cálculo de la rotación y translación se detalla en el apartado 3. En la Figura 1(b) se presentan dos imágenes omnidireccionales donde se han indicado un conjunto de correspondencias. El cálculo de transformación consiste en la obtención de los ángulos (ϕ, β) indicados a partir de las correspondencias de puntos entre ambas imágenes, quedando el factor de escala en la transformación ρ indeterminado. El proceso de cálculo y la integración de las medidas obtenidas entre imágenes se presenta en el apartado 2.

En la mayoría de los trabajos de SLAM visual que se encuentran en la literatura, el mapa se representa mediante un conjunto de puntos tridimensionales que representan elementos del entorno (Gil et al., 2006, 2010; Civera et al., 2008; Davison and Murray, 2002; Ballesta et al., 2010). Típicamente, estos puntos son obtenidos mediante un algoritmo de detección de puntos de interés como, por ejemplo, Harris (Civera et al., 2008) y suelen acompañarse de un descriptor visual más o menos invariante de la apariencia visual del punto. Al conjunto del punto y del descriptor se le denomina *visual landmark* en la literatura anglosajona y se traduce al castellano como marca visual. Independientemente del algoritmo de SLAM utilizado, en los trabajos mencionados el proceso de cálculo del mapa implica la estimación de la posición de cada una de las marcas del mapa. En contraposición con este tipo de mapa, en este artículo exponemos una concepción del mapa diferente. El mapa está formado por la posición y orientación de un conjunto de vistas del entorno. Cada vista se define como la posición y orientación de la cámara cuando esta capturó la imagen en el entorno, junto con un conjunto de puntos de interés y descriptores visuales. El proceso de cálculo plantea la estimación de la posición y orientación de todas las vistas del mapa. La construcción del mapa se resume a continuación: supóngase que el robot parte desde el origen del sistema global de referencia. En ese instante, captura una *vista* inicial. Mientras el robot se mueve en las cercanías de esta vista inicial captura imágenes y encuentra puntos correspondientes entre la imagen actual y la vista inicial, calculando una rotación y translación y localizándose respecto de la vista inicial. Cuando el robot se aleja de la vista inicial, no será capaz de encontrar puntos correspondientes. En este momento iniciará una nueva vista en el mapa. Esta nueva vista permitirá la localización del robot en su cercanía.

La solución presentada en este artículo presenta algunas ventajas si la comparamos con otras soluciones de SLAM visual previas. La ventaja principal radica en la compacidad de la representación del entorno. Soluciones como (Andrew J. Davison et al., 2004; Civera et al., 2008) estiman la posición de las *landmarks* visuales, así como la posición y orientación de la cámara, utilizando 6 variables para tal fin, con lo que el vector de estado del problema de SLAM crece rápidamente con el número de *landmarks* almacenanadas en el mapa. Este hecho plantea un problema para la mayoría de algoritmos de SLAM, haciendo que los tiempos de cálculo aumenten de forma cuadrática con el número de *landmarks* en el mapa. En la solución presentada en este artículo, únicamente se estima la posición de un reducido conjunto de vistas. Cada vista encapsula información de un área del entorno en forma de un conjunto de puntos de interés. Según se demostrará mediante experimentos reales y en



(a) (b)

Figura 1: La Figura 1(a) muestra la configuración del sensor usado en los experimentos. La Figura 1(b) presenta dos imágenes omnidireccionales reales, con varias correspondencias indicadas.

simulación, esta representación del mapa es más eficiente y requiere de un menor coste computacional para su cálculo, aún así permite una localización precisa del robot.

A pesar de que el coste computacional sea un factor restrictivo, en el apartado 3 detallamos un algoritmo que puede ser utilizado para calcular la observación con una frecuencia alta y permite la realización de SLAM en tiempo real. En este caso, el cálculo de la transformación entre dos imágenes depende sólo del número de correspondencias encontradas en las imágenes y no del número de vistas existentes en el mapa, con lo que es un tiempo constante en cada iteración del filtro.

Durante los experimentos se han utilizado las características SURF para la detección y descripción de los puntos. La solución presentada no está limitada al uso de este detector y descriptor visual. El fundamento del uso de este descriptor se basa en un estudio anterior sobre detectores y descriptores visuales en su aplicación al SLAM visual (Gil et al., 2010; Ballesta et al., 2010), donde las características SURF presentaron muy buenas cualidades para esta tarea.

Se presentarán un conjunto de resultados obtenidos en simulación y utilizando datos reales que permiten demostrar la validez de la solución de SLAM visual presentada.

El resto del artículo se organiza de la siguiente manera. Primero, el apartado 2 se describe el proceso de SLAM. A continuación, se describe el algoritmo usado para estimar la transformación entre imágenes omnidireccionales en el apartado 3. Seguidamente se aborda el problema de la asociación de datos en el apartado 4. El apartado 5 presenta los principales resultados experimentales. Finalmente, las conclusiones más relevantes se exponen en el apartado 6.

2. Construcción de mapas (SLAM)

En el método de SLAM (*Simultaneous Localization and Mapping*, o construcción de mapas y localización simultánea) que se presenta aquí, cada imagen omnidireccional integrada en el mapa se denominará *vista*, para diferenciarla del concepto de *landmark* visual utilizado comúnmente en este ámbito. Es importante recalcar que una *landmark* visual corresponde a un punto físico en el entorno, como, por ejemplo, una esquina sobre una pared. Sin embargo, una *vista* representa la información visual obtenida desde una pose en particular del entorno. En consecuencia representamos una *vista* mediante una pose donde se capturó la imagen en el entorno, acompañada de la posición bi-dimensional de los puntos detectados en dicha imagen junto con sus descriptores visuales. El mapa estará integrado por un número finito de vistas capturadas por el robot desde poses diferentes. Cuanto mayor es el número de vistas utilizadas, más completa será la representación del mapa, pero mayor el número de variables a estimar. Según se demostrará en la parte experimental, un conjunto reducido de vistas permite modelar la mayoría de entornos, ya que cada vista permite al robot localizarse en un área cercana. En nuestro caso, según se indicará en el apartado 2.3, se incluyen nuevas vistas cuando la apariencia global de la imagen actual capturada por el robot difiere en gran medida de la apariencia de cualquiera de las vistas existentes en el mapa.

Consideramos que esta representación del entorno se puede emplear para la estimación de un mapa mediante algoritmos de SLAM diferentes, bien métodos *online* como EKF, FastSLAM o bien *offline*, como, por ejemplo, *Stochastic Gradient Descent* (Grisetti et al., 2007). En este artículo presentamos como ejemplo la estimación del mapa mediante un filtro EKF y probamos que se pueden obtener resultados correctos con datos reales.

Igualmente, esta representación del mapa y el modelo de observación pueden ser utilizados para la creación de un mapa basado en vistas capturadas mediante una única cámara estándar. La razón fundamental que justifica el empleo de una cámara omnidireccional es la habilidad de adquirir una visión global del entorno con una única imagen.

2.1. Representación y Estimación del Mapa mediante EKF

A continuación definimos con precisión la representación utilizada para la estimación del mapa del entorno mediante un filtro EKF. La pose del vehículo en el instante t se indicará como $x_v = (x_v, y_v, \theta_v)^T$. Cada vista i está representada por su pose $x_{l_i} = (x_{l_i}, y_{l_i}, \theta_{l_i})^T$, su incertidumbre P_{l_i} y un conjunto de M puntos de interés p_j expresados en coordenadas de imagen. Cada punto de interés esta asociado con un descriptor visual d_j, $j = 1, \ldots, M$. En total, consideramos que en el instante t existen N vistas x_{l_i} incluidas en el mapa, por tanto $i = 1, \ldots, N$.

En la Figura 2 se ilustra este tipo de mapa, donde se indica la posición de un conjunto de vistas. Por ejemplo, la vista A se capturó desde la pose particular $x_{l_A} = (x_{l_A}, y_{l_A}, \theta_{l_A})^T$ en el mapa y tiene un conjunto de M puntos de interés asociados. En el caso presentado, la vista A permite la localización del robot en sus inmediaciones. La vista B se utiliza para modelar una de las estancias y permitirá la localización del robot en sus cercanías. Finalmente, las vistas C, D y E modelan el resto de estancias del entorno. Lógicamente, se deberá establecer un método para la inicialización de las vistas cuando el robot no sea capaz de establecer correspondencias con ninguna de las vistas existentes en el mapa, o bien cuando las vistas existentes no le permitan localizarse con exactitud. En el apartado 2.3 se presenta un método sencillo para realizar esta tarea.

Para la estimación del mapa y de la posición del vehículo en el instante t, definimos un vector de estado para el filtro EKF como:

$$\bar{x}(t) = [x_v, x_{l_1}, x_{l_2}, \cdots, x_{l_N}]^T \quad (1)$$

donde N es el número de vistas que existen en el mapa, x_v la pose del robot y x_{l_i} la pose de la vista i.

La relación entre el estado en el instante $t+1$ y el estado actual es la siguiente:

$$\bar{x}(t+1) = F(t)\bar{x}(t) + u(t+1) + v(t+1) \quad (2)$$

donde $F(t)$ contiene la información relativa a la transición entre estados, $u(t+1)$ es el vector de control del movimiento que genera la odometría del robot y $v(t+1)$ es el ruido que se añade al sistema, el cual es de tipo gaussiano y con correlación nula.

Del mismo modo puede definirse una relación lineal entre la observación realizada por el sistema sensorial en un instante t de una vista i, $z_i(t)$, con la variable de estado.

$$z_i(t) = H_i(t)\bar{x}(t) + w_i(t) \quad (3)$$

donde $H_i(t)$ representa la relación entre $\bar{x}(t)$ y $z_i(t)$, y $w_i(t)$ es el ruido aleatorio que se genera en el proceso, el cual es gaussiano y de covarianza $R(t)$.

A continuación hay que diferenciar las tres etapas fundamentales del procedimiento de filtrado. En primer lugar se lleva a cabo una predicción del estado a estimar $\hat{x}(t)$, y en base a ésta se obtiene la predicción de la observación $\hat{z}_i(t)$:

$$\hat{x}(t+1|t) = F(t)\hat{x}(t|t) + u(t) \quad (4)$$

$$\hat{z}_i(t+1|t) = H_i(t)\hat{x}(t+1|t) \quad (5)$$

$$P(t+1|t) = F(t)P(t|t)F^T(t) + Q(t) \quad (6)$$

donde $P(t|t)$ y $P(t+1|t)$ son matrices de covarianza que representan la incertidumbre de la estimación en t y $t+1$ respectivamente.

En la segunda etapa se realiza la observación $z_i(t)$ de una determinada vista i del mapa, cuya asociación de datos se asume correcta, y mediante la cual se puede definir el concepto de innovación, como la variación entre la estimación a priori y la medida de observación:

$$v_i(t+1) = z_i(t+1) - \hat{z}_i(t+1|t) \quad (7)$$

$$S_i(t+1) = H_i(t)P(t+1|t)H_i^T(t) + R_i(t+1) \quad (8)$$

donde $S_i(t+1)$ representa la covarianza de la innovación.

Finalmente, en la tercera etapa se actualiza la estimación obtenida en la primera etapa según el valor de la innovación obtenida en la segunda etapa, obteniendo así la solución al filtro para el instante $t+1$:

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + K_i(t+1)v_i(t+1) \quad (9)$$

$$P(t+1|t+1) = P(t+1|t) - K_i(t+1)S_i(t+1)K_i^T(t+1) \quad (10)$$

donde en este caso $K_i(t+1)$ se corresponde con la ganancia del filtro EKF, obteniéndose del siguiente modo:

$$K_i(t+1) = P(t+1|t)H_i^T(t)S_i^{-1}(t+1) \quad (11)$$

Para el caso que nos ocupa, inicializamos las matrices de covarianza de ruido $Q(t)$ y $R(t)$ que introducen la odometría y el modelo de observación respectivamente. La primera de ellas se establece en base a los parámetros de ruido conocidos que genera la odometría del robot, y la segunda se determina en base a medidas experimentales, tal y como se detalla en el apartado 5. La odometría $u(t)$, se emplea para la obtención de la predicción, conjuntamente con el estado anterior, tal y como se deduce de la ecuación 4. La matriz de incertidumbre del mapa estimado, $P(t)$, tiene en cuenta el ruido de la odometría según la ecuación 6, y el ruido introducido por el sensor visual a la hora de realizar una medida de observación, como puede comprobarse en las ecuaciones 8 y 10. En particular, el modelo propuesto de observación $z_i(t)$ se detalla a continuación.

2.2. Modelo de Observación

El modelo de observación nos permite obtener una información relativa para la estimación indirecta de la pose del robot y de las vistas. En lo siguiente se asume que el robot se encuentra en una posición en el entorno y captura una imagen omnidireccional I_i. A continuación, suponemos que hemos sido capaces de encontrar un conjunto de puntos correspondientes entre I_i y una de las vistas almacenadas en el mapa I_{l_i}. Según se describirá en el apartado 3, obtenemos una observación $z_i(t)$:

$$z_i(t) = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{pmatrix} atan(\frac{y_{l_i} - y_v}{x_{l_i} - x_v}) - \theta_v \\ \theta_{l_i} - \theta_v \end{pmatrix} \quad (12)$$

donde el ángulo ϕ es la orientación con la que la vista i es observada desde el sistema de referencia móvil asociado al robot y β es la orientación relativa entre ambas imágenes. La vista i está representada por $x_{l_i} = (x_{l_i}, y_{l_i}, \theta_{l_i})$, mientras que la pose del robot está descrita por $x_v = (x_v, y_v, \theta_v)$. Ambas medidas (ϕ, β) se presentan en la Figura 1(a).

2.3. Inicialización de Nuevas Vistas

Según se dijo, es necesario proporcionar un método para incluir nuevas vistas en el mapa cuando estas sean necesarias. En nuestro caso, se incluye una nueva vista en el mapa cuando la apariencia de la imagen actual es muy diferente de cualquiera

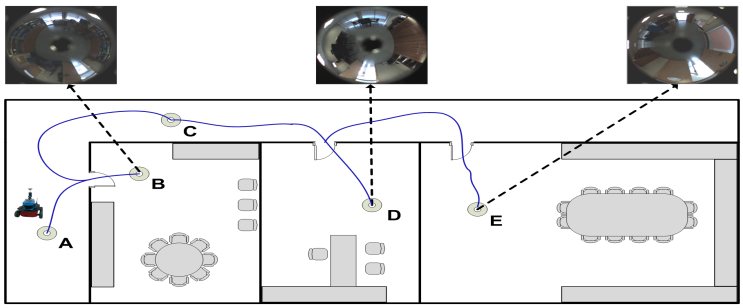


Figura 2: La figura presenta la idea básica para la construcción del mapa. El robot comienza la exploración en el punto A y almacena una vista I_A en el origen. A continuación se mueve. Cuando no se encuentran correspondencias entre la imagen actual e I_A, una nueva vista es creada en la posición actual del robot, B. El proceso continúa hasta que el entorno queda completamente representado.

de las vistas almacenadas en el mapa. El cálculo de la apariencia global se aproxima mediante la siguiente ratio:

$$R = \frac{2K}{n_A + n_B} \tag{13}$$

que calcula el grado de similitud entre las vistas A y B, siendo K el número total de correspondencias puntuales entre A y B, mientras que n_A y n_B son el número de puntos detectados en las imágenes A y B respectivamente.

El robot decidirá incluir una nueva vista en el mapa cuando la ratio R cae por debajo de un valor predefinido. Así pues, la inicialización de las vistas depende de un único factor de similitud R. Si se selecciona un valor de R alto, se incluirá un gran número de nuevas vistas en el mapa, aumentando la precisión con la que podremos localizar al robot pero incrementando el coste computacional necesario para calcular el mapa. En el caso opuesto, si se selecciona un valor de R bajo, el número de vistas será menor, reduciéndose el tiempo de cómputo pero también reduciendo la precisión con la que podemos localizar al robot en el mapa.

En la inicialización de cada vista, la pose x_{l_i} y la incertidumbre asociada se obtienen de la estimación de x_v en el instante actual t y de su submatriz de covarianza asociada, ya que en el instante t la posición de la vista y la posición del robot coinciden.

3. Cómputo de la Transformación entre Imágenes Omnidireccionales

En este apartado proponemos un método para calcular la transformación entre dos imágenes omnidireccionales. La transformación se puede calcular salvo un factor de escala y está representada mediante los ángulos (β, ϕ), según se indicó en el apartado 2.2. Estos ángulos representan la posición relativa del robot a una de las vistas del mapa y permiten su localización. Para su obtención deben detectarse puntos característicos en ambas imágenes y encontrar sus correspondencias aplicando la condición de epipolaridad. Los esquemas tradicionales, tales como (Kawanishi et al., 2008; Nister, 2003; Stewenius et al., 2006) resuelven el caso general con 6 GDL, mientras que en nuestro caso, asumiendo que el movimiento del robot se restringe a un plano, podemos limitar el cálculo a 4 variables de la matriz esencial, reduciendo de este modo el coste computacional.

3.1. Detección de Puntos Significativos y Correspondencias

Durante las pruebas experimentales se han empleado las características SURF (Bay et al., 2006) con el fin de obtener puntos de interés y correspondencias entre imágenes. Según el estudio presentado en (Gil et al., 2010; Ballesta et al., 2010), el detector y descriptor SURF obtuvo excelentes resultados en términos de robustez de los puntos detectados y de invarianza del descriptor al compararse con otros métodos empleados en el ámbito de SLAM visual. La extracción de puntos de interés y su descripción se realizan a partir de una imagen panorámica, si bien una vez obtenidos los puntos, se trabaja con sus coordenadas en la esfera unidad sobre el sistema de referencia original. Para ello, como primer paso, se transforma la imagen omnidireccional capturada con la cámara a una vista panorámica y se extraen un conjunto de puntos característicos. A continuación, para cada uno de estos puntos se calcula un descriptor SURF. Según se comprueba experimentalmente, ante un movimiento de la cámara, la variación en la apariencia local de los puntos (y, por tanto, la variación en el descriptor) es menor en la imagen panorámica que en la omnidireccional. En la Figura 3.1 se presenta un ejemplo de imagen omnidireccional capturada por el sistema catadióptrico y su transformación a imagen panorámica. De esta manera se consigue aumentar el número de correspondencias válidas entre imágenes. Hay que destacar, que finalmente los puntos detectados en la imagen panorámica se reproyectan sobre la esfera unidad en las coordenadas de la vista original, es decir sobre la vista omnidireccional, y se almacenan junto con los descriptores calculados. El cambio a vista

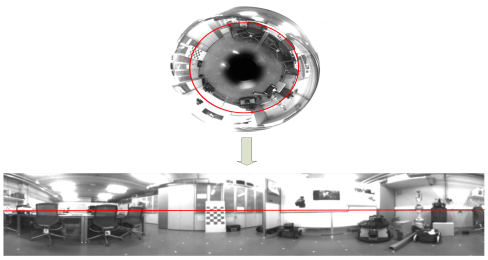


Figura 3: La figura muestra una imagen omnidireccional y su vista panorámica correspondiente. En la figura, la línea dibujada representa la posición de los mismos puntos en ambas imágenes. Una vez detectados puntos de interés en la vista panorámica, se realiza la reconversión de estos puntos sobre la esfera unidad en el sistema de referencia original.

panorámica se realiza únicamente con la intención de incrementar el número de puntos detectados, puesto que este modelo ha sido planteado para obtener transformaciones entre imágenes omnidireccionales.

3.2. *Cómputo de la Transformación*

Una vez detectados los puntos SURF en cada una de las vistas y suponiendo conocidas un conjunto de correspondencias entre imágenes, ha de establecerse un proceso para calcular los ángulos relativos β y ϕ.

3.2.1. *Geometría Epipolar*

La condición de epipolaridad establece la relación entre dos puntos 3D observados desde diferentes vistas. Se puede expresar como:

$$\rho p'^T E p = 0 \tag{14}$$

donde la matriz E recibe el nombre de matriz esencial. El mismo punto detectado en dos imágenes se expresa como $p = [x, y, z]^T$ en el sistema de referencia fijo de la primera cámara y $p' = [x', y', z']^T$ en el de la segunda (considerado móvil). La matriz esencial E representa una rotación R y una traslación T (salvo un factor de escala ρ) entre los sistemas de referencia de dos imágenes, con $E = R \cdot T_x$. Por tanto los ángulos deseados (β, ϕ), pueden ser obtenidos a partir de los elementos de E. Debe señalarse que la Geometría Epipolar puede ser usada en imágenes omnidireccionales ya que reproyectamos el sistema 2D del plano imagen a 3D mediante el modelado del espejo hiperbólico de la cámara, a partir de una calibración previa (Scaramuzza et al., 2006). A causa de la ambigüedad en la profundidad, denotamos $\vec{p}$ and $\vec{p'}$ en 3D, como los vectores unitarios que indican la dirección de los puntos en los dos sistemas de referencia, ya que la posición 3D no puede ser totalmente definida con una única vista de la escena. De otra manera: el método presentado permite calcular la matriz E salvo un factor de escala ρ, el cual a efectos prácticos se elige de manera arbitraria para la resolución del problema. Aun así, en los experimentos con datos reales que se presentan, la escala real del

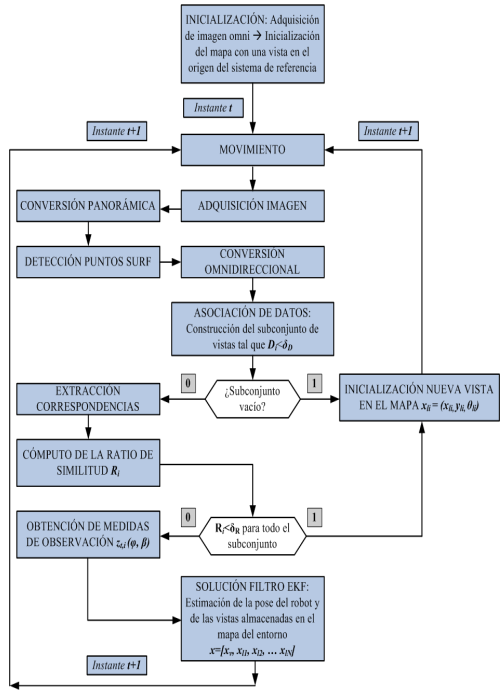


Figura 4: Diagrama de bloques representativo del modelo de SLAM propuesto.

mapa puede determinarse con bastante precisión a partir de las medidas de odometría del robot, resolviendo de este modo la indeterminación del factor de escala ρ.

Con el fin de obtener β y ϕ, hemos considerado (Hartley and Zisserman, 2004), donde se sugiere el empleo de la matriz de proyección P, la cual también define la transformación entre imágenes. Se ha adoptado este método por su simplicidad a la hora de calcular las cuatro posibles soluciones del problema. Al estimar una rotación y una traslación sobre un movimiento en el plano XY, sólo son necesarias $K = 4$ corresponcencias para resolver el problema, ya que la matriz E tiene la siguiente forma:

$$E = \begin{bmatrix} 0 & 0 & e_{13} \\ 0 & 0 & e_{23} \\ e_{31} & e_{32} & 0 \end{bmatrix} \tag{15}$$

En cambio, calculamos E con un mayor número de puntos a fin de obtener soluciones fiables en presencia de ruido y falsas correspondencias. Además, empleamos un algoritmo RANSAC (Nistér, 2005) para filtrar posibles correspondencias erróneas.

Para el cálculo de E, primero aplicamos la condición de epipolaridad $\vec{p'}^T \cdot E \cdot \vec{p} = 0$ sobre K puntos, y resolvemos la ecuación resultante $D \cdot \vec{e} = 0$. Donde, D es una matriz de coeficientes que se obtiene como resultado de aplicar la ecuación (14) a K

puntos, y $\vec{e} = [e_{13}\ e_{23}\ e_{31}\ e_{32}]$. A continuación descomponemos E mediante SVD:

$$[U|S|V] = SVD(E), \tag{16}$$

que permite calcular:

$$R_1 = [UV^TW],\ R_2 = [UV^TW^T],\ T = [UZU^T] \tag{17}$$

siendo W y Z matrices auxiliares (Hartley and Zisserman, 2004) y las posibles rotaciones (R_1, R_2) y traslaciones $(T_{1x}, -T_{1x})$. Para obtener las cuatro posibles P-matrices, computamos:

$$P_1 = [R_1|T_{1x}],\ P_2 = [R_1| - T_{1x}], \tag{18}$$

$$P_3 = [R_2|T_{1x}],\ P_4 = [R_2| - T_{1x}], \tag{19}$$

En nuestro caso, las matrices de proyección tienen la forma:

$$P_i = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & \rho\cos(\phi) \\ \sin(\beta) & \cos(\beta) & 0 & \rho\cos(\phi) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

Nótese que β, ϕ y ρ pueden tomar diferentes valores que cumplan la condición de epipolaridad (14) debido a la indeterminación del factor de escala ρ. Esto plantea un proceso de selección de una de las cuatro posibles soluciones descritas en (18) como la correcta. En nuestro caso, hemos utilizado una solución basada en mínimos cuadrados, según se detalla en (Bunschoten and Krose, 2003). Dicho proceso se detalla a continuación:

3.2.2. Selección de la Solución

La selección de la solución correcta debe llevarse a cabo mediante un procedimiento inverso. Multiplicamos $\vec{p'}$ por la inversa de cada una de las cuatro matrices de proyección posibles P_i, obteniendo así cuatro estimaciones de $\vec{p}$. Se asume como solución correcta aquella que genera la estimación con menor desviación respecto de $\vec{p}$. Por último, β y ϕ son directamente recuperados a partir de los elementos de P definidos en (20).

4. Asociación de datos

El problema de la asociación de datos reviste gran importancia en el caso general de SLAM visual basado en *landmarks*. Dicho problema se puede enunciar de la siguiente manera: dado un conjunto de observaciones $z_i(t) = \{z_{i,1}, ..., z_{i,B}\}$ obtenidas en el instante t, se deberá decidir cuáles de las *landmarks* del mapa generaron dichas observaciones. Así pues, el resultado del proceso de asociación de datos es un vector de índices $H = \{j_1, ..., j_B\}$ donde cada uno de los índices $j_i \in [1, N+1]$ denota una de las *landmarks* del mapa, siendo N el número total de *landmarks* en el mapa. Si la observación $z_i(t)$ no está asociada a ninguna de las *landmarks* del mapa, se inicializará una nueva con índice $N + 1$. Este proceso de asociación de datos es crucial en SLAM. El caso del SLAM visual basado en *landmarks* visuales es particularmente complejo, ya que en el mapa pueden existir un gran número de marcas visuales y la apariencia de los puntos correspondientes puede variar considerablemente. Por ejemplo, en (Gil et al., 2006) se utiliza una distancia de Mahalanobis para encontrar un conjunto de candidatos entre las *landmarks* del mapa. A continuación, se elige la correspondencia en función de la similitud entre los descriptores visuales. Otras soluciones más elaboradas para hallar la asociación de datos, como la presentada en (Neira and Tardós, 2001) son de difícil aplicación, debido al coste computacional que implican.

En el caso presentado aquí, basado en el uso de *vistas* omnidireccionales para la construcción del mapa, la asociación de datos se puede abordar de una manera diferente. Consideremos que en un instante t el robot captura una imagen omnidireccional I_i. Asumamos que, en ese instante t existen N vistas en el mapa $I_1, I_2, \ldots, I_N$. Primero, se seleccionan un conjunto de vistas cercanas a la vista actual, según las poses de las mismas. Esta selección se realiza en base a la distancia Euclídea:

$$D_i = \sqrt{(x_v - x_{l_i})^T \cdot (x_v - x_{l_i})} \tag{21}$$

La vista i se incluye en el conjunto de candidatos si $D_i < \delta_D$, donde δ_D es una distancia elegida experimentalmente. Valores altos de δ_D precisan un mayor número de candidatos para realizar la búsqueda, con lo que se incrementa el coste computacional.

A continuación, se busca un conjunto de puntos correspondientes entre la imagen I_i y cada uno de las imágenes en el grupo de candidatos $\{I_1, I_2, \ldots, I_J\}$. La búsqueda de puntos correspondientes se realiza teniendo en cuenta la restricción epipolar (14). Finalmente, en base al número de correspondencias encontradas, se calcula la ratio R (13) y se obtiene una observación $z_i(t) = (\phi, \beta)$ entre la vista I_i y la vista I_k ($k \in [1, J]$) si la ratio R supera un determinado valor fijado experimentalmente.

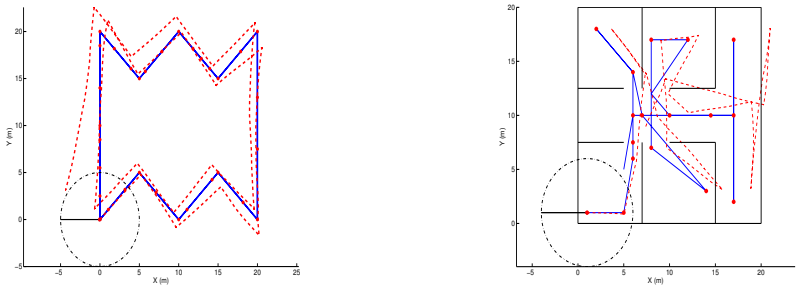
De esta manera, cuando la ratio R es alta, existen un gran número de correspondencias correctas entre la vista actual I_i y la vista candidato, con lo que la observación $z_i(t) = (\phi, \beta)$ será precisa. Si la ratio R es baja, el número de correspondencias entre las imágenes es reducido, con lo que la observación $z_i(t)$ con gran probabilidad, será incorrecta. De esta manera, en base al factor R podemos decidir la asociación de datos y la inicialización de nuevas vistas.

5. Resultados

Los resultados experimentales se agrupan en dos apartados diferentes. Primero, en el apartado 5.1 presentamos los resultados obtenidos en simulación que permiten validar el esquema de SLAM aquí propuesto. Seguidamente, en el apartado 5.2 mostramos resultados experimentales reales.

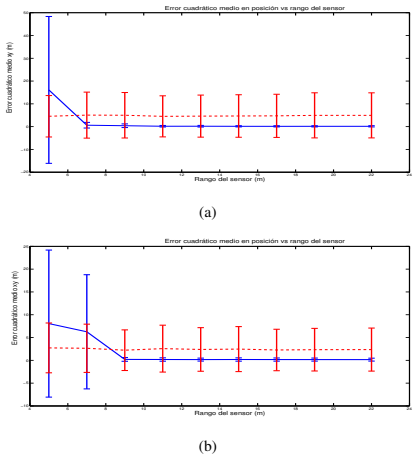
5.1. SLAM: Resultados en Simulación

Hemos realizado una serie de experimentos en simulación que permiten validar el concepto general de vista y su estimación mediante un algoritmo de SLAM basado en el filtro EKF. Nótese la importancia de asegurar la convergencia de un algoritmo de SLAM basado en EKF, con el modelo de observación presentado en la ecuación 12, ya que el modelo de observación se debe linealizar para incluirlo en el filtro de Kalman. Los experimentos en simulación se han realizado en dos escenarios



(a) (b)

Figura 5: La Figura 5(a) representa el escenario simulado 1. La localización de las distintas vistas en el mapa se representa con puntos. La Figura 5(b) representa el escenario simulado 2.



(a)

(b)

Figura 6: La Figura 6(a) presenta los resultados obtenidos en el escenario simulado 1. La Figura 6(b) presenta los resultados obtenidos en el escenario simulado 2.

virtuales con características diferentes. La Figura 5(a) muestra el escenario simulado 1, donde se simula un entorno en el que no existen obstáculos que impidan la visibilidad sobre las diferentes vistas del mapa. Por otra parte, la Figura 5(b) representa el escenario simulado 2, el cual emula un entorno típico de interior donde existen obstáculos, tales como paredes, que limitan la visibilidad sobre las vistas del mapa. Se asume que el robot puede calcular una observación $z_i(t)$ con alguna de las vistas del mapa cuando podemos trazar una línea recta entre la pose del robot y la posición de la vista, siempre que estén dentro del radio de observación δ_D. En las Figuras, 5(a) y 5(b) se presenta además con línea continua el camino real seguido por el robot, mientras que con línea de puntos se muestran las lecturas de odometría. Un conjunto de vistas han sido aleatoriamente dispuestas a lo largo de las trayectorias y se muestran con puntos. Nótese que, según se indicó en el apartado 2.3, el emplazamiento de las vistas depende de la similitud entre las imágenes y de la ratio R elegida. En los dos escenarios, la simulación genera una variación aleatoria de R, por tanto se está simulando el procedimiento de inicialización de vistas en el mapa por parte del robot a medida que va descubriendo el entorno. Esta simulación de la disposición de vistas emula el comportamiento normal de los experimentos reales, ya que la variación de R se ha escogido para tal efecto. Puesto que la intención principal de estos primeros experimentos es la validación de la convergencia del algoritmo de SLAM, las imágenes asociadas a las vistas son omitidas. De este modo las observaciones $z_i(t)$ que realiza el robot también son simuladas con una covarianza obtenida experimentalmente de $\sigma_\phi = 0,1 = \sigma_\beta = 0,1rad$. El radio de observación del robot δ_D se representa mediante un círculo discontinuo centrado en la pose real del robot.

A continuación presentamos los resultados obtenidos en simulación con el escenario simulado 1. En ambos casos, el robot comienza el proceso de SLAM en el origen y realiza dos vueltas a lo largo de la trayectoria indicada. Las observaciones obtenidas por el robot han sido simuladas según el modelo presentado en la ecuación 12 con un ruido gaussiano simulado mediante una matriz de covarianza $R(t) = diag(\sigma_\phi^2 = 0,1^2 rad^2, \sigma_\beta^2 = 0,1^2 rad^2)$. Hemos llevado a cabo una serie de experimentos donde se varía el radio de observación del robot δ_D. Los resultados se presentan en la Figura 6(a) y 6(b), donde se muestra el error RMS en la trayectoria frente al radio de observación. Dicho error, representa la desviación cuadrática media tanto de la estimación según el filtro EKF (línea contínua), como de la odometría (línea a trazos) comparada con el camino real. El experimento se ha repetido 50 veces, generando aleatoriamente 50 series diferentes de odometría. En las Figuras 6(a) y 6(b) se observa el error RMS medio de los distintos experimentos, así como intervalos de 2σ. Según se puede observar en la Figura 6(a), cuando el radio de observación está por debajo de 6

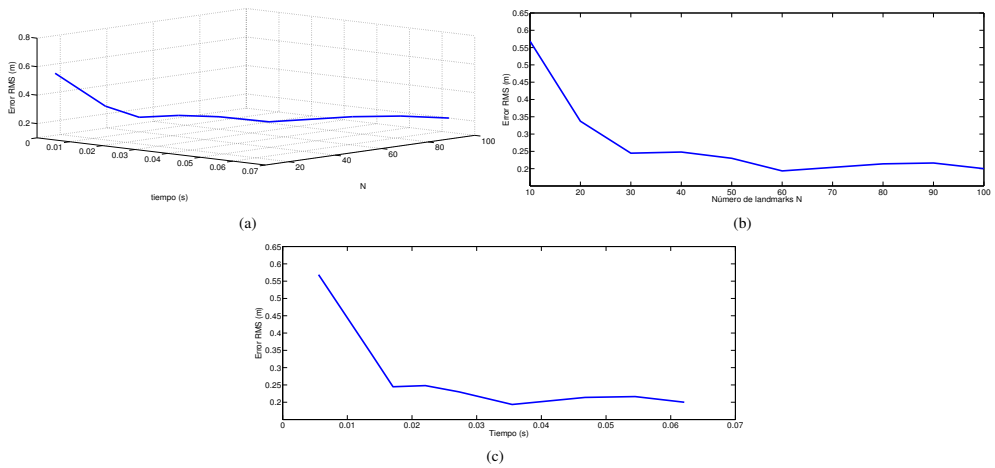


Figura 7: La Figura 7(a) representa el error RMS cometido en el camino frente al número de *landmarks N* y el tiempo de cómputo medio por iteración del algoritmo de SLAM. La Figura 7(b) representa el error RMS frente al número de *landmarks N* mientras que la Figura 7(c) muestra el error RMS frente al tiempo de cómputo medio por cada iteración del algoritmo de SLAM.

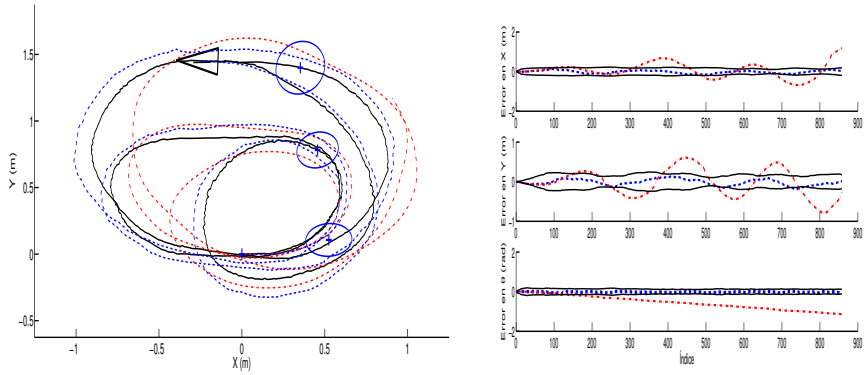
m la incertidumbre en la posición es alta, denotando que el filtro no ha sido capaz de filtrar el ruido en la odometría. Esto es debido a que la vistas están emplazadas a distancias mayores, y el robot no ha sido capaz de obtener suficientes observaciones. Para valores de radio superiores a 6 m el error RMS disminuye demostrando la convergencia del filtro. Un resultado similar se presenta en la Figura 6(b), la cual se corresponde con el escenario simulado 2. En este caso se obtienen resultados satisfactorios con valores de δ_D por encima de $9m$. La diferencia existente entre ambos resultados se puede explicar debido a la existencia de elementos en el entorno que limitan la visibilidad entre las vistas, dificultando así la obtención de observaciones.

Es necesario recalcar que los resultados obtenidos dependen fuertemente del emplazamiento y número de vistas. Si se sitúan más vistas en el entorno se consigue un cálculo más preciso tanto del mapa como de la trayectoria, a cambio de un mayor coste computacional. Con esta idea en mente se realizaron un conjunto de simulaciones en las que se mantuvo constante el radio de observación δ_D y se varió el número N de vistas en el mapa mientras se media el tiempo necesario para realizar el experimento. Cada simulación en el escenario 1 se repitió 50 veces, obteniendo valores medios del error RMS cometido en la estimación de la trayectoria del robot. En la Figura 7 representamos el error en la trayectoria estimada en función del número de vistas incluidas en el mapa y el tiempo necesario para calcular cada iteración. Típicamente, una aplicación de SLAM debe ser capaz de funcionar a tiempo real, por tanto, debe existir un compromiso entre la precisión del mapa y el tiempo de cálculo. Dada la capacidad de computación del robot, la Figura 7(a) nos permite determinar el número máximo N de vistas para poder procesar las observaciones a tiempo real y prever la precisión con la que podemos estimar la trayectoria del robot. Las Figuras 7(b) nos permiten observar cómo varía el error RMS en función de las vistas del mapa. Obsérvese como el error tiende hacia un límite mínimo conforme N tiende a infinito. Por otra parte, en la Figura 7(c) se puede observar cómo aumenta el tiempo necesario de cómputo en función del error RMS deseado en el camino. Se puede comprobar cómo, el error no se corresponde de forma lineal con el tiempo de cálculo necesario.

5.2. *SLAM: resultados con Datos Reales*

En este apartado presentamos resultados que validan el esquema de SLAM propuesto mediante imágenes reales capturadas en un entorno interior. Los datos experimentales se obtuvieron con un robot Pioneer P3-AT equipado con una cámara firewire con una resolución 1280x960 píxeles y un espejo hiperbólico. El eje óptico de la cámara está instalado aproximadamente perpendicular al plano del suelo como se describe en la Figura 1(a). Como consecuencia, una rotación del robot se corresponde con una rotación de la imagen respecto al eje óptico de la cámara. Durante las pruebas, se capturaron imágenes omnidireccionales cada vez que el robot avanzó más de $0,05m$ o giró más de $0,05rad$. Igualmente, se almacenaron datos de distancia de un sensor SICK LMS y se obtuvo un mapa y un camino con el algoritmo descrito en (Stachniss et al., 2004) que se ayuda de la gran precisión de las medidas del sensor láser. Durante los experimentos el camino calculado a partir de datos de láser se utiliza únicamente para compararlos con los resultados de SLAM visual obtenidos mediante la cámara omnidireccional.

El robot es guiado a través del entorno mientras captura imágenes omnidireccionales y datos de distancia láser a lo largo de la trayectoria. De nuevo, para poder comparar resultados, hacemos uso de un algoritmo de SLAM basado en distancias



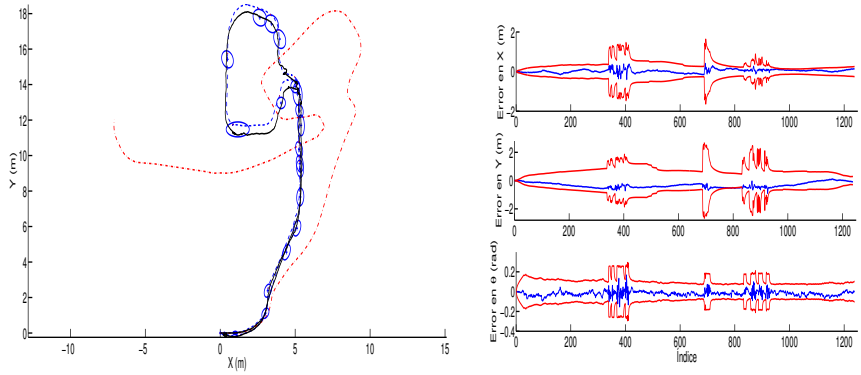
(a) (b)

Figura 8: La Figura 8(a) presenta los resultados de SLAM con datos reales, para trayectoria real (punteada), estimación (continua) y odometría (trazos). La posición de las vistas se presenta con elipses de error. La Figura 8(b) presenta el error en cada paso temporal en X, Y y θ de la estimación (punteada) y la odometría (trazos).

láser, descrito en (Stachniss et al., 2004), para definir la trayectoria real. El robot comienza inicializando una vista a partir de la adquisición de una imagen omnidireccional en el origen. A continuación se mueve a lo largo de la trayectoria mientras continúa adquiriendo imágenes. Instantes después se inicializa una nueva vista. Mientras se calcula el mapa, se realiza una comparación entre la imagen actual y el resto de vistas del mapa, obteniendo un conjunto de correspondencias. Al mismo tiempo, la ratio de similitud (13) es evaluada, y cuando ésta cae por debajo de $\delta_R = 0{,}5$, se crea una nueva vista y se inicializa con la posición actual del robot. Finalmente el robot recorre la trayectoria mostrada en la Figura 10, donde mostramos con puntos el resto de posiciones en las que el robot decide inicializar una nueva imagen. La línea punteada muestra la trayectoria real, la línea continua muestra la estimación del EKF, mientras que la línea a trazos muestra la odometría. Cabe señalar que el robot continúa el movimiento dentro de la misma estancia siendo capaz de realizar observaciones de las vistas inicializadas anteriormente. En nuestro caso el umbral δ_R fue determinado experimentalmente con el objetivo de generar un número reducido de vistas y representar el entorno de un modo más compacto. Si se eligiese un valor más bajo de δ_R, menos imágenes serían inicializadas en el mapa. Por el contrario si se eligiese un valor superior, el mapa resultante almacenaría un mayor número de vistas. Puede observarse en la Figura 8(a) como una vez la cuarta vista es inicializada no es necesario inicializar ninguna otra, obteniendo así una representación más compacta. En la Figura 8(b) comparamos la trayectoria estimada con la trayectoria real y con la odometría. Hay que señalar que este error tiene la misma escala que la solución del mapa estimado y no ha sido normalizado. Presentamos el error en la estimación de la trayectoria (línea punteada) junto a los intervalos 2σ y al error en la odometría (línea a trazos).

La Figura 9 presenta otro experimento. En este caso, el robot explora una habitación, recorre un pasillo, entra en una habitación diferente y vuelve al punto de origen. La distancia total recorrida es de $45m$. La Figura 9(a) presenta la trayecoria real (punteada), la odometría (a trazos) y la estimación (continua). La localización de las vistas y su incertidumbre asociada se indica mediante puntos y elipses de error. En la Figura 9(b) presentamos el error en la posición para cada paso temporal con intervalos de 2σ. Puede observarse cómo el error presenta varias oscilaciones a lo largo del recorrido, lo cual se debe a momentos en los que el robot realiza giros muy pronunciados para entrar y salir de la habitación, así como para rodearla. Además, en estos instantes aparecen elementos obstructores y por tanto disminuye más si cabe la capacidad para visualizar vistas y obtener medidas de observación precisas. Este hecho puede comprobarse en la Figura 9(a), donde se observa cómo la disposición de las vistas del mapa en dichos instantes lleva asociada una mayor incertidumbre. Pese a todo sigue quedando de manifiesto que el filtro es capaz de mantener la convergencia en todo momento. Una vez que el robot vuelve a visualizar vistas almacenadas anteriormente, se comprueba que estos intervalos momentáneos de mayor incertidumbre se reducen, obteniendo un error para la estimación en torno a una decena de centímetros respecto al camino real.

Por último, la Figura 10 presenta otro experimento llevado a cabo en un entorno de dimensiones $32x45m$, donde en la Figura 10(a) se representa la trayecoria real (continua), la odometría (a trazos) y la estimación (punteada). En este caso la distancia recorrida es mayor que en los casos anteriores y la complejidad del problema también aumenta debido a la presencia de un mayor número de obstrucciones. La localización de las vistas y su incertidumbre asociada se indica mediante puntos y elipses de error. En la Figura 10(b) presentamos el error en la posición para cada paso temporal con intervalos de 2σ. Hay que destacar que los valores de error obtenidos para este caso son ligeramente mayores a causa de la complejidad del entorno. En general, los elementos obstructores hacen que existan muchos instantes



(a) (b)

Figura 9: La Figura 9(a) presenta los resultados de SLAM con datos reales, para trayectoria real (punteada), estimación (continua) y odometría (trazos). La posición de las vistas se presenta con elipses de error. La Figura 9(b) presenta el error en cada paso temporal en X, Y y θ de la estimación (punteada) con intervalos de 2σ.

en los cuales las observaciones se llevan a cabo con dificultad. Por ello la incertidumbre que se genera es mayor. Sin embargo, pese a esta situación desfavorable, el filtro EKF logra resolver el problema manteniendo la convergencia en todo momento, y asegurando un error dentro de los límites esperados.

6. Conclusiones

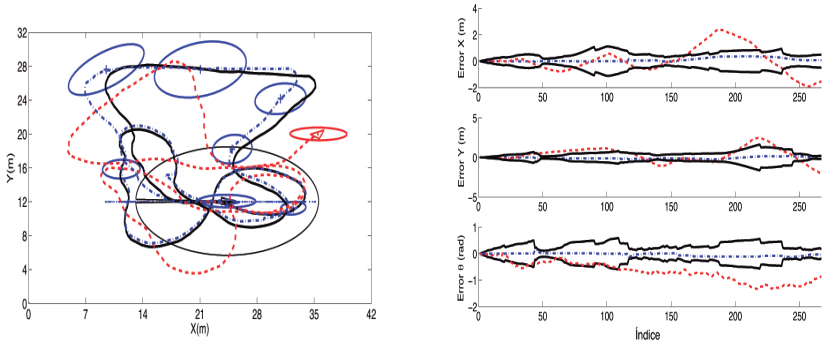
Hemos presentado un modelo para la resolución del problema de (SLAM) empleando imágenes omnidireccionales. Proponemos una representación del entorno que se aleja del concepto de mapa visual tradicional en el campo de SLAM visual. Habitualmente, el SLAM visual plantea la estimación de la posición 3D de un conjunto de marcas visuales y sus descriptores. En contraposición a este modelo, en este trabajo simplificamos el problema a la estimación de la posición y orientación de un conjunto reducido de imágenes omnidireccionales. Cada imagen omnidireccional, renombrada como *vista*, tiene asociado un conjunto de puntos de interés y sus descriptores visuales que describen el entorno de una forma compacta. Cada una de las imágenes permite representar un área del entorno, haciendo posible la localización del robot en las inmediaciones de cada una de ellas. La aportación fundamental se basa en la posibilidad de extraer una transformación entre dos imágenes omnidireccionales en las que existe un conjunto de correspondencias puntuales. Dicha transformación, formada por una rotación y una traslación (salvo un factor de escala), nos permite proponer un nuevo modelo de observación y resolver el problema de SLAM con un algoritmo basado en el EKF. Presentamos resultados obtenidos en entornos simulados que validan el esquema de SLAM en diferentes condiciones. Además, mostramos la validez de la propuesta con experimentos reales realizados con un robot móvil real. Las pruebas experimentales realizadas han demostrado que se puede modelar un entorno mediante un número reducido de imágenes omnidireccionales, lo que da lugar a un problema con un menor número de variables a estimar. Al mismo tiempo, los resultados demuestran que este modelo permite obtener un buen resultado en términos de localización del robot, así como un mapa mucho más compacto que el obtenido con un mapa visual tradicional.

English Summary

Construction of a visual model of the environment based on omnidirectional images

Abstract

This paper deals with the problem of Simultaneous Localization and Mapping (SLAM). The solution presented is based on the utilisation of a set of images to represent the environment. In this way, the estimation of the map considers the computation of the position and orientation of a set of omnidirectional views captured from the environment. The proposed idea sets apart from the usual representation of a visual map, in which the environment is represented by a set of three dimensional points in a common reference system. Each of these points is commonly denoted as visual landmark. In the case presented here, the robot is equipped by a single omnidirectional visual sensor that allows to extract a number of interest points in the images, each one described by a visual descriptor. The map building process can be summed up in the following way: as the robot traverses the environment, it captures omnidirectional images and extracts a set of interest points from each one. Next, a set of correspondences is found between the current image and the rest of omnidirectional images existing in the map. When the number of correspondences found is enough, a transformation is computed, consisting of a rotation and a translation (up to an unknown scale factor). In the paper we show a method that allows to build a map while localizing the robot using these kind of observations. We present results obtained in a simula-



(a) (b)

Figura 10: La Figura 10(a) presenta los resultados de SLAM con datos reales, para trayectoria real (continua), estimación (punteada) y odometría (trazos). La posición de las vistas se presenta con elipses de error. La Figura 10(b) presenta el error en cada paso temporal en X, Y y θ de la estimación (punteada) y la odeometría (trazos) con intervalos de 2σ.

ted environment that validate the proposed idea. In addition, we present experimental results using real data that prove the suitability of the solution.

Keywords:

SLAM, mobile robotics, omnidirectional vision

Agradecimientos

Este trabajo se ha llevado a cabo gracias en parte al Ministerio de Ciencia e Innovación a través del proyecto DPI2010-15308, con título "Exploración integrada de entornos mediante robots cooperativos para la creación de mapas 3D visuales y topológicos que puedan ser usados en navegación con 6 grados de libertad"

Referencias

Andrew J. Davison, A. J., Gonzalez Cid, Y., Kita, N., 2004. Improving data association in vision-based SLAM. In: Proc. of IFAC/EURON. Lisboa, Portugal.

Aracil, R., Balaguer, C., Armada, M., 2008. Robots de servicio. RIAI (Revista Iberoamericana de Automática e Informática Industrial) 5(2), 6–13.

Ballesta, M., Gil, A., Reinoso, O., Úbeda, D., 2010. Análisis de detectores y descriptores de características visuales en slam en entornos interiores y exteriores. RIAI (Revista Iberoamericana de Automática e Informática Industrial) 7(2), 68–80.

Bay, H., Tuytelaars, T., Van Gool, L., 2006. SURF: Speeded up robust features. In: Proc. of the ECCV. Graz, Austria.

Bunschoten, R., Krose, B., 2003. Visual odometry from an omnidirectional vision system. In: Proc. of the ICRA.

Civera, J., Davison, A. J., Martínez Montiel, J. M., 2008. Inverse depth parametrization for monocular slam. IEEE Trans. on Robotics.

Davison, A. J., Murray, D. W., 2002. Simultaneous localisation and map-building using active vision. IEEE Trans. on PAMI.

Gil, A., Martinez-Mozos, O., Ballesta, M., Reinoso, O., 2010. A comparative evaluation of interest point detectors and local descriptors for visual slam. Machine Vision and Applications.

Gil, A., Reinoso, O., Martínez-Mozos, O., Stachniss, C., Burgard, W., 2006. Improving data association in vision-based SLAM. In: Proc. of the IROS. Beijing, China.

Grisetti, G., Stachniss, C., Grzonka, S., Burgard, W., 2007. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In: Proc. of RSS. Atlanta, Georgia.

Harris, C. G., Stephens, M., 1988. A combined corner and edge detector. In: Proc. of Alvey Vision Conference. Manchester, UK.

Hartley, R., Zisserman, A., 2004. Multiple View Geometry in Computer Vision. Cambridge University Press.

Jae-Hean, K., Myung Jin, C., 2003. Slam with omni-directional stereo vision sensor. In: Proc. of the IROS. Las Vegas (Nevada).

Joly, C., Rives, P., 2010. Bearing-only SAM using a minimal inverse depth parametrization. In: Proc. of ICINCO. Funchal, Madeira (Portugal).

Kawanishi, R., Yamashita, A., Kaneko, T., 2008. Construction of 3D environment model from an omni-directional image sequence. In: Proc. of the Asia International Symposium on Mechatronics 2008. Sapporo, Japan.

Lowe, D., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., 2002. Fastslam: a factored solution to the simultaneous localization and mapping problem. In: Proc. of the 18th national conference on Artificial Intelligence. Edmonton, Canada.

Neira, J., Tardós, J. D., 2001. Data association in stochastic mapping using the joint compatibility test. IEEE Trans. on Robotics and Automation.

Nister, D., 2003. An efficient solution to the five-point relative pose problem. In: Proc. of the IEEE CVPR. Madison, USA.

Nistér, D., 2005. Preemptive RANSAC for live structure and motion estimation. Machine Vision and Applications.

Paya, L., Fernández, L., Reinoso, O., Gil, A., Úbeda, D., 2009. Appearance-based dense maps creation: Comparison of compression techniques with panoramic images. In: 6th International Conference on Informatics in Control, Automation and Robotics ICINCO. Milan (Italy), pp. 250–255.

Scaramuzza, D., Fraundorfer, F., Siegwart, R., 2009. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In: Proc. of the ICRA. Kobe, Japan.

Scaramuzza, D., Martinelli, A., Siegwart, R., 2006. A toolbox for easily calibrating omnidirectional cameras. In: Proc. of the IROS. Beijing, China.

Stachniss, C., Grisetti, G., Haehnel, D., Burgard, W., 2004. Improved Rao-Blackwellized mapping by adaptive sampling and active loop-closure. In: Proc. of the SOAVE. Ilmenau, Germany.

Stewenius, H., Engels, C., Nister, D., 2006. Recent developments on direct relative orientation. ISPRS Journal of Photogrammetry and Remote Sensing.

Information Sciences 279 (2014) 326–337

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A modified stochastic gradient descent algorithm for view-based SLAM using omnidirectional images

David Valiente *, Arturo Gil, Lorenzo Fernández, Óscar Reinoso

Miguel Hernández University, System Engineering Department, 03202 Elche, Spain

ARTICLE INFO

Article history:
Received 25 March 2013
Received in revised form 25 March 2014
Accepted 29 March 2014
Available online 12 April 2014

Keywords:
Mobile robotics
Visual SLAM
Omnidirectional images
SGD

ABSTRACT

This paper describes an approach to the problem of Simultaneous Localization and Mapping (SLAM) based on Stochastic Gradient Descent (SGD) and using omnidirectional images. In the field of mobile robot applications, SGD techniques have never been evaluated with information gathered by visual sensors. This work proposes a SGD algorithm within a SLAM system which makes use of the beneficial characteristics of a single omnidirectional camera. The nature of the sensor has led to a modified version of the standard SGD to adapt it to omnidirectional geometry. Besides, the angular unscaled observation measurement needs to be considered. This upgraded SGD approach minimizes the non-linear effects which impair and compromise the convergence of traditional estimators. Moreover, we suggest a strategy to improve the convergence speed of the SLAM solution, which inputs several constraints in the SGD algorithm simultaneously, in contrast to former SGD approaches, which process only constraint independently. In particular, we focus on an efficient map model, established by a reduced set of image views. We present a series of experiments obtained with both simulated and real data. We validate the new SGD approach, compare the efficiency versus a standard SGD and demonstrate the suitability and the reliability of the approach to support real applications.

1. Introduction

In the field of mobile robot applications, the problem of SLAM is a crucial factor, due to the need for a complete representation of the environment, especially for navigation purposes. The objective of building a map entails considerable complexity, since the map has to be built incrementally, while, the localization of the robot inside it needs to be calculated simultaneously. Generating a reliable and coherent map is even more challenging and laborious when sensor data is affected by noise, and this directly impairs the simultaneous estimation of the map and the path followed by the robot.

To date, SLAM approaches have been differentiated according to several factors, such as the way to estimate the representation of the map, the main algorithm for computing a solution and the kind of sensor to extract information from the environment. For instance, several map representations were obtained thanks to the extensive use of laser data range and sonar [8]. In this area, maps were principally generated following two representation models [16,11], corresponding, respectively, to 2D occupancy grid maps based on raw laser and 2D landmark-based maps focused on the extraction of features, described from laser data measurements.

* Corresponding author. Tel.: +34 96 665 9005; fax: +34 96 665 8979.
E-mail addresses: dvaliente@umh.es (D. Valiente), arturo.gil@umh.es (A. Gil), l.fernandez@umh.es (L. Fernández), o.reinoso@umh.es (Ó. Reinoso).

http://dx.doi.org/10.1016/j.ins.2014.03.122
0020-0255/

More recently, the tendency has turned to the use of visual information by means of digital cameras. Many applications benefit from the use of these sensors, whose characteristics outperform previous sensors such as lasers in terms of the amount of usable information from the environment for building the map. For instance, the approaches that use two calibrated cameras, known as stereo-pairs, in order to extract a set of 3D visual landmarks determined by a visual description [5]. Other approaches simply exploit a single camera to estimate 3D visual landmarks [2,10]. They initialize the coordinates of each 3D landmark relying on the inverse depth parametrization, since there exists a scale uncertainty about the distance to each landmark which cannot be directly retrieved with a single camera. Omnidirectional cameras have also been used alone [15], and some others have even arranged two omnidirectional images, in order to take the best advantage of the wider field of view provided by these cameras.

As important as the kind of sensor and the map representation is the estimation algorithm for a SLAM scheme. It defines the core of the system, as it is responsible for the ultimate solution. Most extensively used are online methods such as EKF [4], Rao-Blackwellized particle filters [11] and offline algorithms, such as, Stochastic Gradient Descent [7].

The combination of data sensors, map representation and the core of the algorithm therefore determines the final effectiveness of a SLAM which seeks reliability and suitability for realistic applications. Great efforts have been made in this field. For example, certain approaches [4,5,3,2,14] have concentrated on estimating of the position of a set of 3D visual landmarks in a main reference system, while, simultaneously, building the map. The main idea lies in the capability of an EKF filter to converge the estimation to an appropriate solution for the SLAM problem. In this same line of EKF usage, [18] has recently proposed a distinctive map representation consisting of a reduced set of image views, determined by their position and orientation in the environment. Such a technique establishes an estimation of a state vector which includes the map and the current localization of the robot at each timestep k. The estimation of the transition between states at k and $k+1$ considers the wheel's odometry as initial estimate, but also the observation measurements gathered by sensors.

Generally, EKF methods are troublesome in the presence of non-linear errors as they have difficulties in maintaining the convergence of the estimation. This situation normally appears in presence of Gaussian errors introduced by the observation measurement, which usually causes data association problems [12]. A visual observation model, such as the omnidirectional, is susceptible to introduce non-linearities and is thus responsible for this kind of errors. By contrast, an offline algorithm such as SGD [1] may deal with this issue caused by non-linearity effects. Similarly, in [20,19,17], parallel approaches are presented to maintain stability in non-linear contexts.

Regarding the basic goals of this study, we present a new visual SLAM approach based on omnidirectional images and sustained by a SGD solver algorithm which helps overcome the harmful effects caused by errors. To achieve this, and depending on the nature of the problem, different aspects have to be taken into consideration so that the research is conducted towards the achievement of new contributions and advantages compared to former applications based on the standard SGD algorithm [13,7,6,1]. Firstly, a map model has to be adapted to the omnidirectional observation. Along the same line, the standard SGD has to be redesigned to be able to work with the omnidirectional geometry of the images, but also considering the nature of the measurement, which lacks scale. This implies that the solution to the problem is not a trivial one. So, the difference between our approach, which uses a different geometrical environment, and all the previous SGD applications, which consider data range observations in a Cartesian measurement system, should be noted. Next, to improve the efficiency of the standard method, in terms of the convergence speed, we propose a modification in the estimation procedure. The traditional models mentioned above, usually process every odometry and observation measurement (denoted as constraints) independently at each iteration step. By contrast, with the aim of finding a valid solution quickly, we propose a strategy based on the simultaneous use of a certain set of information provided by our visual observation measurements. This proposal might appear to be liable to cause an increase in the required computational resources. Nevertheless, we have concentrated on preventing this by updating several stages of the SGD's iterative optimization so as to avoid possible harmful bottleneck handicaps. Therefore, the main expected contributions and advantages of this SLAM approach compared to traditional approaches might be synthesized as it follows:

- An efficient map model established by a reduced set of omnidirectional images.
- A modified SGD solver algorithm adapted to the omnidirectional geometry which is the basis of the proposed SLAM's observation model. Development of the new differential equations related to the observation measurements.
- Improved efficiency of the estimation thanks to the use of simultaneous constraint processing.

The structure of the paper has been divided as it follows: Section 2 depicts the SLAM problem within this framework. Then, Section 3 describes the general specifications of a SGD algorithm, concentrating on the standard SGD. Section 4 details the proposed modification of the standard SGD and the main contributions mentioned above. Next, Section 5 provides both simulated and real data experimental results to validate the model and test its reliability and expected benefits versus traditional methods. Finally, Section 6 analyzes the results to draw a general conclusion.

2. SLAM

A visual SLAM technique is expected to retrieve a feasible estimation of the position of the robot within a certain environment, which also has to be precisely determined by the estimation. In our approach, the map is composed of a set of

omnidirectional images obtained from different poses in the environment, denoted as views. These views do not represent any physical landmarks, as they will consist of an omnidirectional image captured at the pose $x_l = (x_l, y_l, \theta_l)$ and a set points of interest extracted from that image. Such an arrangement, allows us to exploit the capability of an omnidirectional image to gather a large amount of information in a simple image, due to its large field of view. Thus, an important reduction is achieved in terms of the number of variables for estimating the solution.

The pose of the mobile robot at time t will be denoted as $x_v = (x_v, y_v, \theta_v)^T$. Each view $i \in [1, \ldots, N]$ is constituted by its pose $x_{l_i} = (x_l, y_l, \theta_l)_i^T$, its uncertainty P_{l_i} and a set of M interest points p_j expressed in image coordinates. Each point is associated with a visual descriptor $d_j, j = 1, \ldots, M$.

Thus, the augmented state vector is defined as:

$$\bar{x} = \begin{bmatrix} x_v & x_{l_1} & x_{l_2} & \cdots & x_{l_N} \end{bmatrix}^T \tag{1}$$

where $x_v = (x_v, y_v, \theta_v)^T$ is the pose of the moving vehicle and:

$$x_{l_N} = (x_{l_N}, y_{l_N}, \theta_{l_N})$$

is the pose of the N-view that exists in the map.

2.1. Map building

The map building procedure is described by Fig. 1. The exploration task starts navigating the environment at the origin, denoted as A. At this time, the robot captures an omnidirectional image I_A, stored as a view with pose x_{l_A}. While the robot keeps moving towards the first office room, it is able to find correspondences between I_A and the current omnidirectional image, which makes it able to localize itself. Once the robot enters the office room, the appearance of the images varies significantly, so no matches are found between the current image and image I_A. In this case, the robot will initialize a new view named I_B at the current robot's position, which will be used for localization inside the office room. Finally, the robot completes the exploration of the environment as it traverses the different areas of the environment, while acquiring the rest of the necessary views I_C, I_D, I_E, to compose the final map. The number of views initiated in the map depends directly on the kind of environment and its visual appearance. In particular, in Fig. 1 it may be also perceived a synthesis of the localization procedure carried out by the robot, which translates the depicted comparison between I_A and I_E into a single-computation process.

2.2. Observation model

In accordance with the view-based representation recently presented, a new observation model has to be formulated. The versatility of omnidirectional images enables to apply epipolar constraints [9] to extract an observation measurement, which defines the motion transformation between two poses, as seen in Fig. 1. Actually, these poses represent the positions where the robot acquired two specific images. To that effect, only two images, with a set of corresponding points between them, are required to obtain the transformation. The observation measurement may be expressed as:

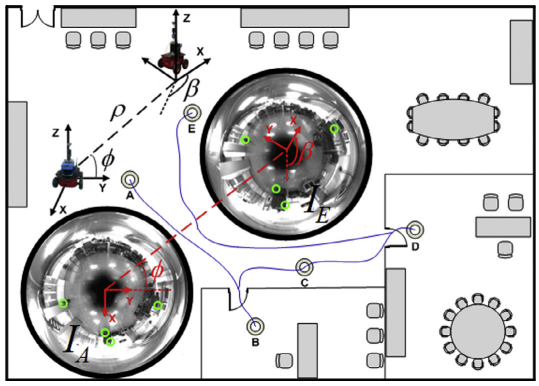


Fig. 1. Map building procedure. The robot starts the exploration at A by acquiring a view I_A. While the robot moves, correspondences are found between I_A and the current image captured at the current robot's pose. When no correspondences are found, the current image is stored as a new view of the map, for instance I_B at B. The procedure ends when the whole environment is represented.

$$z_t = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{pmatrix} \arctan\left(\frac{y_{l_N} - y_v}{x_{l_N} - x_v}\right) - \theta_v \\ \theta_{l_N} - \theta_v \end{pmatrix} \tag{2}$$

where the angle ϕ is the bearing at which the view N is observed and β is the relative orientation between the images. The view N is represented by $x_{l_N} = (x_{l_N}, y_{l_N}, \theta_{l_N})$, whereas the pose of the robot is described as $x_v = (x_v, y_v, \theta_v)$. Both measurements (ϕ, β) are represented in Fig. 1.

3. Standard SGD algorithm

3.1. Specifications

A graph-oriented map is composed by a set of nodes defining the poses traversed by the robot and the landmarks initialized into the map. The state vector s_t encodes this representation through a set of variables which are expressed in the following manner:

$$s_t = [\,(x_0, y_0, \theta_0), (x_1, y_1, \theta_1) \ldots \quad (x_n, y_n, \theta_n)\,] \tag{3}$$

where (x_n, y_n, θ_n) are the 2D coordinates and the bearing in a general reference system. A complementary subset of edges represents the relationships between nodes, by means of either distance measurements generated by the odometry or observations measurements provided by the on-board sensors. Both measurements are commonly known as constraints and denoted as δ_{ji}, where j indicates the observed node, seen from node i. The general objective stated by methods based on standard SGD approaches [13,7] is to minimize the error likelihood, expressed as:

$$P_{ji}(s) \propto \eta \exp\left(-\frac{1}{2}(f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji})\right) \tag{4}$$

being $f_{ji}(s)$ a function dependent on the state s_t and both nodes j and i. The difference between $f_{ji}(s)$ and δ_{ji} expresses the error deviation between nodes. Such error term is weighted by the information matrix:

$$\Omega_{ji} = \Sigma_{ji}^{-1} \tag{5}$$

where Σ_{ji}^{-1} is the associated covariance matrix, which considers the uncertainty of the measurements. The assumption of logarithmic notation in (4) leads to:

$$F_{ji}(s) \propto (f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji}) = r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{6}$$

being $r_{ji}(s)$ the error determined by $f_{ji}(s) - \delta_{ji}(s)$, which shows its condition of residue. Finally, the global problem seeks the minimization of the objective function which represents the accumulated error:

$$F(s) = \sum_{\langle j,i\rangle \in G} F_{ji}(s) = \sum_{\langle j,i\rangle \in G} r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{7}$$

where $G = \{\langle j_1, i_1\rangle, \langle j_2, i_2\rangle \ldots\}$ defines the subset of particular constraints that define the map, either odometry or observation measurements.

3.2. Estimation

Once the formulation of the problem has been presented, the Stochastich Gradient Descent algorithm must be detailed. The basic goal is to compute in an iterative manner a estimation to achieve a valid solution for the SLAM problem. The basis of a SGD method lies in minimizing Eq. (7) through derivative optimization techniques. The estimated state vector is obtained as:

$$s_{t+1} = s_t + \Delta s \tag{8}$$

where Δs expresses a certain update with respect to s_t, term which is sequentially generated by means of the constraint optimization procedure. It is worth noting that, in a general case, this update is calculated independently at each step by using only a simple constraint, that is to say $\Delta s_n = f(\delta_{ji})$. The general expression for the transition between s_t and s_{t+1} has the following form:

$$s_{t+1} = s_t + \lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji} \tag{9}$$

- λ is a learning factor to re-scale the term $H^{-1}J_{ji}^{T}\Omega_{ji}r_{ji}$. Normally, λ takes decreasing values following the criteria $\lambda = 1/n$, where n is the iteration step. This strategy is intended to reach the final solution quickly using large values of λ. When the solution moves close to the optimum, lower values of λ are used, thus preventing the estimation to oscillate around the final solution.
- H is the Hessian matrix, calculated as $J^{T}\Omega J$, and it represents the shape of the error function through a preconditioning matrix to scale the variations of J_{ji}. According to [6], H can be computed:

$$H \approx \sum_{\langle i,j \rangle} J_{ji}\Omega_{ji}J_{ji}^{T} \tag{10}$$

- J_{ji} is the Jacobian of $f_{ji}(s)$ with respect to s_t, $J_{ji} = \frac{\partial f_{ji}}{\partial s}$. It converts the error deviation into a spatial variation.
- Ω_{ji} is the information matrix associated to a constraint. $\Omega_{ji} = \Sigma_{ji}^{-1}$, being Σ_{ji} the covariance matrix corresponding to the observation constraints δ_{ji}.

This scheme updates the estimation by computing the rectification introduced by each constraint at each iteration step respectively. Despite the learning factor to reduce the weight by which each constraint updates the estimation, the procedure may lead to an inefficient method to reach a stable solution, as undesired oscillations may occur due to the stochastic nature of the constraint selection. For this reason, we propose an optimization process which takes into account several constraints in the same iteration. It might be thought that the same drawbacks could arise with the addition of some other inconveniences such as undesired time overloads, as a consequence of the simultaneous processing of several constraints in the same iteration. However, we have modified some calculations at specific stages of the algorithm in order to maintain the time requirements and even reduce them. As a result, we achieved improved convergence ratios in terms of speed. Further details will be provided in the next section.

4. Modified SGD

This section has been intended to explain the main advantages and contributions achieved in this study. The first assumption to consider is the redefined state vector s_t, which will be treated as a set of incremental variables. The pose incremental state is defined as:

$$s_t^{inc} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (dx_1, dy_1, d\theta_1) \\ \vdots \\ (dx_n, dy_n, d\theta_n) \end{bmatrix} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (x_1 - x_0, y_1 - y_0, \theta_1 - \theta_0) \\ (x_2 - x_1, y_2 - y_1, \theta_2 - \theta_1) \\ \vdots \\ (x_n - x_{n-1}, y_n - y_{n-1}, \theta_n - \theta_{n-1}) \end{bmatrix} \tag{11}$$

where $(dx_i, dy_i, d\theta_i)$ encode the variation between consecutive poses in coordinates of the global reference system. A global encoding has the main drawback of not being capable to update more than one node and its adjacent ones per constraint. Regarding a relative codification of the state, the problem of non-linearities in J_{ji} arises. By contrast, an incremental state vector allows a single constraint to generate a variation at every pose. In this context, Δs in Eq. (8) affects all poses because the state vector is differentially encoded.

Note that in this approach we are dealing with a visual observation given by an omnidirectional camera. This makes us to adapt the equations defined in the previous section to the case of omnidirectional geometry, as the nature of the constraints are not simply metrical like the odometry constraints. According to (2), given two nodes, the observation measurement allows us to determine a specific motion transformation between them up to a scale factor. Therefore, the omnidirectional measurements and the incremental representation require the reformulation of several terms involved in the estimation. Following this, we detail all the proposed modifications to the terms of the standard SGD, which must be necessarily redefined and recalculated. The complete structure for each derivative is detailed in Appendix A.

- The first modification is referred to $f_{ji}(s)$, differentiating between odometry and visual observation constraints:

$$f_{j,i}^{odo}(s) = \begin{pmatrix} dx_j \\ dy_j \\ d\theta_j \end{pmatrix} + \begin{pmatrix} dx_{j-1} \\ dy_{j-1} \\ d\theta_{j-1} \end{pmatrix} + \cdots + \begin{pmatrix} dx_i \\ dy_i \\ d\theta_i \end{pmatrix} \tag{12}$$

where $(dx_i, dy_i, d\theta_i)$ has been defined in (11). And for the case of the visual observation constraint:

$$f_{j,i}^{visual}(s) = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{bmatrix} \arctan\left(\frac{dy_j - dy_i}{dx_j - dx_i}\right) - d\theta_i \\ d\theta_j - d\theta_i \end{bmatrix} \tag{13}$$

where β and ϕ are directly computed from the observation measurement [18] model, which expresses the relationship of transformation between two omnidirectional images and the encoded pose of the robot in Eq. (11). Observing Fig. 1 may also help understand the definition of Eq. (13).

- The second modification considers the recalculation of $J_{ji} = \frac{\partial f_{ji}}{\partial s}$, according to the previous reformulation of $f_{ji}(s)$. The importance of considering the indexes of the corresponding nodes, either $j > i$ or $j < i$ must be noted, as the derivatives considerably change its form. Furthermore, as seen above, the dimensions of $f_{ji}(s)$ are different, something which also has to be taken into consideration, as the rest of the terms involved in the SGD algorithm have to be resized.

$$J_{j,i} = \frac{\partial f_{j,i}(s)}{\partial s} = \left[\frac{\partial f_{j,i}(\phi)}{\partial s}, \frac{\partial f_{j,i}(\beta)}{\partial s}\right] \tag{14}$$

- Finally, we suggest the estimation of the new state s_{t+1} by considering several constraints at the same time. We seek greater relevance of the weight of the constraints when searching for the optimal minimum estimation. Obviously, computing more than one constraint at each step leads to a certain overload. By contrast, with this approach, we reduce the expensive estimation of H. In a general case, H is computed whenever a single constraint is introduced, that is to say, as many times as there are constraints. In our case we compute H only once for each subset of constraints introduced simultaneously into the system. Consequently we obtain H in a more efficient manner, thus compensating for possible time overloads. The following example depicts the practical meaning of this concept.

Require: $\delta_{ji} \in C \; \forall \, j, i$, where $C = [c_1, c_2, \ldots, c_b]$ and $c_b = \{\delta_{11}, \delta_{12}, \ldots\}$
  Each c_b represents different subset of constraints δ_{ji} simultaneously processed by the robot.
  t: iteration step
  ϵ: threshold for $F(s)$
  while $F(s) > \epsilon$ **do**
    $t = t + 1$
    for $q = 1{:}b$ **do**
      Extract all δ_{ji} in c_q randomly
      Computing the following terms:
      $f_{ji}(s) = [f_{ji}^{odo}(s), f_{ji}^{visual}(s)]$, J_{ji}, H, Ω_{ji}, and r_{ji}
      $\Delta s_q = \lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji}$
      $s_q = s_{q-1} + \Delta s_q$
    end for
    $s_t = s_q + s_{t-1}$
  end while
  return $s_t = [(x_0, y_0, \theta_0), (dx_1, dy_1, d\theta_1), \ldots, (dx_n, dy_n, d\theta_n)]$

---

5. Results

We have carried out three different sets of experiments. Firstly, in Section 5.1 we show SLAM results obtained from simulated data to confirm the validity of the new SLAM approach supported by SGD. We add a comparison of results obtained by our approach with a standard SGD algorithm, like those used in applications like [13,7,6,1]. Finally, in Section 5.2 we present SLAM results using real data acquired by the robot, which have also been compared with a traditional SGD estimator. The equipment consists of a Pioneer P3-AT indoor robot with a firewire 1280 × 960 camera and a hyperbolic mirror. The optical axis of the camera is installed approximately perpendicular to the ground plane, as described in Fig. 1. Consequently, a rotation of the robot corresponds to a rotation of the image with respect to its central point. In addition, we used a SICK LMS range finder in order to compute a ground truth using the method presented in [16].

5.1. SLAM results with simulated data

Confirmation of the convergence of an SLAM algorithm is crucial when a new solver proposal, such as SGD, is introduced. Furthermore, other considerations require evaluation, since the performance of the new method has to deal with a visual observation model, which is a common source of non-linearities.

5.1.1. Experiment 1

Fig. 2(a) presents a random simulation environment of 20 × 20 m, where the robot traverses approximately 300 m. The real path followed by the robot is shown with a continuous line, the odometry is represented with a dash-dotted line, and the estimated solution is shown with a dashed line. A set of views have been placed randomly along the trajectory. The arrangement of these views is controlled by an appearance ratio between images, to assure a realistic placement of each view. A grid

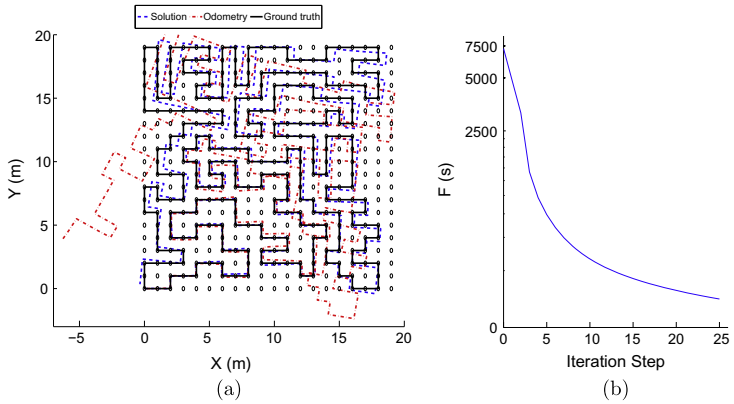


Fig. 2. (a) presents a map obtained by the proposed approach in an environment of 20 times 20 m. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution. (b) Shows the accumulated error probability $F(s)$ along the number of iterations.

of circles represents the possible poses where the robot might move to and gather a new view. The number of iterations of the SGD algorithm is 25. As it can be observed in Fig. 2(a), starting from a noisy odometry estimate, the final estimation has been rectified following the tendency of the real path. Fig. 2(b) shows the decreasing evolution of the accumulated error probability $P_{ji}(s)$ in (4), expressed in logarithmic terms, versus the number of iterations. The reliability of this new approach to work with omnidirectional observations can be confirmed, as it provides a proper solution.

5.1.2. Comparison of results

The following experiments have been conducted in order to compare our approach with the traditional standard SGD in terms of efficiency. We suggest a strategy to introduce several constraints simultaneously into the SGD algorithm. The main goal is to improve the speed by which the method iteratively optimizes until a final solution is achieved. In this sense, we have performed a SLAM experiment, where the robot traverses 50 m through a given environment. Again, the number of views in the map has been randomly placed, by following the same policy explained above. The same experiment has been repeated 200 times using the same series of odometry inputs, in order to provide mean values that express consistent results. The two approaches, ours and the standard SGD algorithm, have been compared. We have modified the number of views N which the robot is able to observe from each pose. The observation range r of the robot has also been varied. Fig. 3 presents results for the accumulated error probability, $F(s)$, being the objective function which the SGD algorithm seeks to minimize. Fig. 3 compares the solution obtained by our approach, drawn with a continuous line, and the solution obtained by the

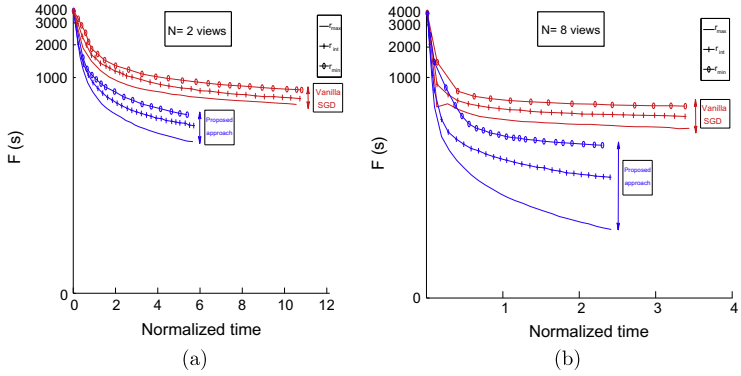


Fig. 3. (a) and (b) Show the accumulated error probability $F(s)$ versus time in a SLAM experiment, when the number of views observed by the robot is $N = 2$ and $N = 8$ respectively. The continuous lines show the results provided by the proposed solution, while the dashed lines show results provided by the standard SGD solution. Different lengths for the observation range are defined: $r_{min}, r_{int}, r_{max}$.

standard SGD algorithm, drawn with a dashed line. Fig. 3(a) and (b) represent $F(s)$ when the robot observes $N = 2$ and $N = 8$ views, respectively. As we are looking for a fair comparison, the x-axis, originally representing iteration steps, has been transformed into a normalized time variable to generate a trustworthy comparison between the two schemes. Please note that the time spent at each iteration step differs from one method to another due to their different convergence speeds. Therefore, in terms of efficiency, it can be shown that the solution provided by our approach outperforms the solution given by a standard SGD in every case, as the decreasing slope of $F(s)$ is clearly steeper. Hence a quicker convergence speed demonstrating a more

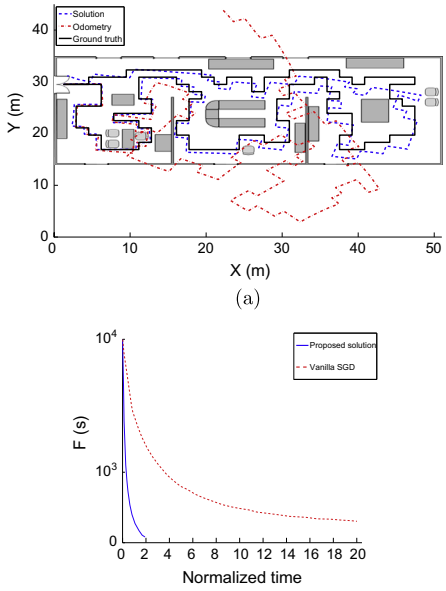


Fig. 4. (a) Shows the SLAM results in an office-like environment of 20 times 50 m. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution. (b) Compares the accumulated error probability $F(s)$ provided by the approach presented in a continuous line and the $F(s)$ provided by the standard SGD in a dashed line.

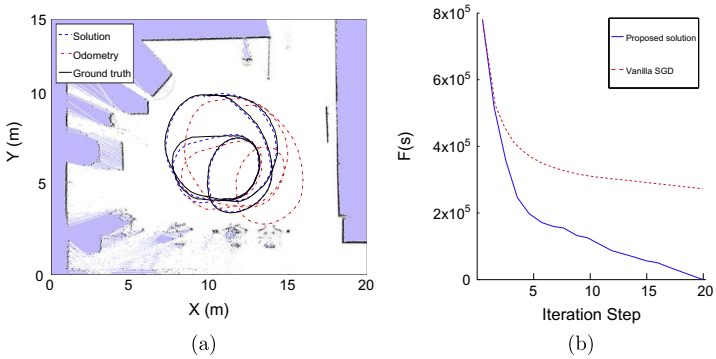


Fig. 5. (a) Shows SLAM results in a real office environment of 15 times 15 m. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution. (b) Shows the accumulated error probability $F(s)$ along the number of iterations for our approach and the standard SGD respectively.

efficient method. This is the main advantage achieved by means of combining several constraints simultaneously at each iteration step, instead of using only one as a traditional SGD used to. The relevance of the observation range of the vehicle r is also notable. As seen in Fig. 3(a) and (b), longer values of r provide a better convergence, compared to shorter r. As the omnidirectional observation is angular, and lacks scale, views seen by the robot at longer distances in the map allow the computation of a more feasible localization. In addition, when the robot is able to observe a higher number of views, the optimum value for $F(s)$ is evidently lower, and is reached quickly, as there are more constraints to compute.

5.1.3. Experiment 2

The purpose of this next experiment is to confirm the favorable results shown in Section 5.1, now dealing with an office-like environment, since it is desirable to emulate a more realistic situation, with obstructions, obstacles, etc. Fig. 4(a) describes the environment of 20 × 50 m which the robot moves through. The continuous line represents the real path followed by the robot, the dash-dotted line shows the odometry, whereas the estimated solution with our approach is shown by a dashed line. It may be noticed that in only 15 iterations of the algorithm the robot is able to estimate a quiet reliable solution, whose topology follows the real path. On the other hand, the odometry error grows out of bounds. Fig. 4(b) shows a

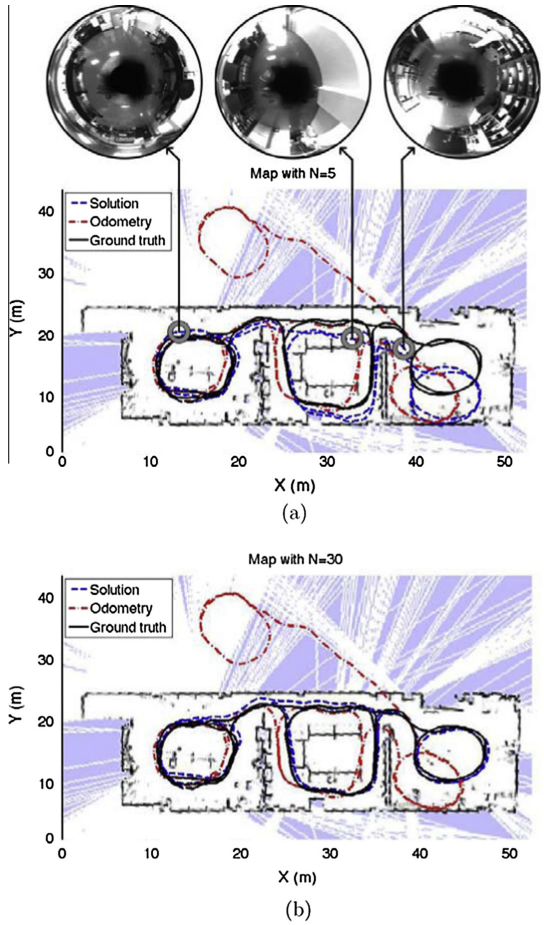


Fig. 6. (a) and (b) show SLAM results in a real office environment of 20 times 50 m, with $N = 5$ and $N = 30$ views observed respectively. The continuous line shows the real path, the dash-dotted line the odometry and the dashed line the estimated solution.

comparison of the evolution of the accumulated error $F(s)$ along the time for both our SGD approach and the standard SGD. Once again, the improved capability in quickly reaching a solution is shown, demonstrating better efficiency. In this particular case, it is worth mentioning that our approach requires a computational cost approximately six times lower than that for a standard SGD to reach an optimum value.

5.2. SLAM results with real data

Having presented the simulation results validating the proposed approach, we carried out a set of experiments with real data. We were seeking for confirmation of the suitability and reliability of the approach in a realistic application such as exploration tasks. Furthermore, we also show comparisons with the standard SGD.

5.2.1. Experiment 3

The first experiment analyzes the behavior of the approach when dealing with one of the most adverse situations, that is to say, when the robot constantly turns around as shown in Fig. 5(a). The real path is shown with a continuous line, the odometry with a dash-dotted line and the estimated solution with a dashed line. This case is seen as one of the worst, as the frequent turns introduce significant noise into the input associated with the odometry. Nevertheless, it should be noted that the estimation converges to a proper solution, which is practically overlapped with the real path, whereas the odometry estimation differs considerably. Fig. 5(b) shows the decreasing tendency of the accumulated error probability $F(s)$ along the number of iterations for both our approach and the standard SGD. Having tested the validity of the main benefits with the previous experiments, the improved efficiency of our approach can now be confirmed in terms of the speed of convergence compared to the standard SGD method. Examinating Fig. 5(b), it can be seen that this approach reaches optimum values for $F(s)$ in less time than the standard SGD. The main advantage in terms of efficiency is therefore shown.

5.2.2. Experiment 4

This last experiment aims to support the beneficial results presented above, which have been compared to traditional SGD approaches. In this case we conducted an experiment in a large environment. Here, the robot moves through a real office of 20×50 m. Again, there are obstacles and obstructions such as doors, walls and office furniture. As seen in Fig. 6 the robot explores the whole environment describing a trajectory of approximately $280m$. Moreover, maps with different number of views N have been constructed to study its relevance to the estimation of the solution. Fig. 6(a) and (b) show different results when the robot observes $N = 5$ and $N = 30$ respectively. The real path is drawn with a continuous line, odometry with a dash-dotted line and the estimated solution with a dashed line. Some real views stored in the map have been indicated. Fig. 7 shows the accumulated error probability $F(s)$ for both experiments, expressing it with continuous line for $N = 5$ views and with dashed line for $N = 30$ views. In addition, to demonstrate the improved efficiency of the method, we compare the values of $F(s)$ provided by this approach, in blue, versus that obtained by a standard SGD, in red. In accordance to the

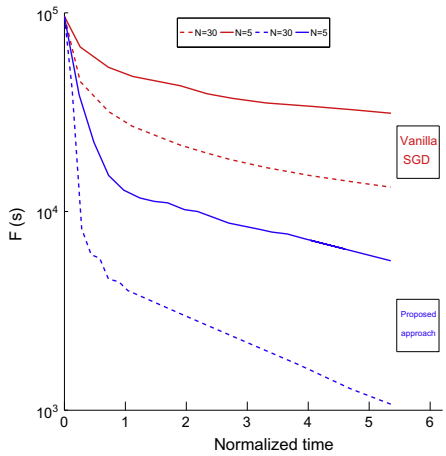


Fig. 7. Accumulated error probability $F(s)$ in a real SLAM experiment versus time. Results obtained for the map showed in Fig. 6(a) with $N = 5$ views, are compared using continuous lines: the continuous blue line represents the proposed approach while the continuous red line represents the standard SGD. Results obtained for the map shown in Fig. 6(b) with $N = 30$ views are compared using dashed lines: the dashed blue line represents the proposed approach whereas the dashed red line represents the standard SGD.

specific topology of the environment, it is confirmed that the larger the number of views N, the more accurate the estimation. Since the robot is able to observe more views, the rectification of the estimation is ensured by a higher number of constraints. Moreover, our approach still reveals the main favorable features compared to the standard SGD, regardless of the value of N. Along the same lines as the previous experiment, the faster convergence speed is proved by observing Fig. 7, where lower optimum values for $F(s)$ are confirmed in considerably less time. This fact shows the greater efficiency of this proposal compared to former SGD techniques.

6. Conclusions

This work has proposed an approach to the visual SLAM problem by introducing a SGD algorithm adapted to omnidirectional observations. The assumption of SGD has been aimed at reducing instabilities and harmful effects which compromise the convergence of the most extended SLAM algorithms such as EKFs. These erroneous effects are mainly consequences of the visual nature of the observation, which is non-linear, and particularly intensified on omnidirectional images. We present a visual SLAM approach which computes a map consisting of a reduced set of omnidirectional views. A single computation of two views allows us to easily retrieve a motion transformation between the poses where the robot captured the views. The standard SGD algorithm has been modified to integrate an unscaled observation model. We propose a more efficient SGD model, which suggests a new strategy designed to exploit the information provided by several constraints simultaneously into the same SGD iteration. We have presented SLAM results with simulated data which validate the combination of SGD with omnidirectional images proposed by this new approach. In addition, we have established a comparison between the results obtained by our approach and those obtained by a standard SGD algorithm. Finally, SLAM results with real data have been presented so as to demonstrate the suitability of the approach and also its efficiency compared to the traditional SGD algorithm.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation under the Project DPI2010-15308 and the FPU scholarship BES-2011-043482.

Appendix A. SGD equations adapted to omnidirectional observations

$$J_{j,i} = \frac{\partial \mathbf{f}_{j,i}(s)}{\partial s} = \left[\frac{\partial f_{j,i}(\phi)}{\partial s}, \frac{\partial f_{j,i}(\beta)}{\partial s}\right] \tag{A.1}$$

- $j > i$
  - $k > i$

$$\frac{\partial f_{j,i}(\phi)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = -\frac{\sum_{k=i+1}^{j} dy_k}{q} = a \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = \frac{\sum_{k=i+1}^{j} dx_k}{q} = b \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = 0 \end{cases} \tag{A.2}$$

$$\frac{\partial f_{j,i}(\beta)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = 1 \end{cases} \tag{A.3}$$

  - $k < i$

$$\frac{\partial f_{j,i}(\phi)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = -1 \end{cases} \tag{A.4}$$

$$\frac{\partial f_{j,i}(\beta)}{\partial s} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = 0 \end{cases} \tag{A.5}$$

$$J_{j,i} = \frac{\partial \mathbf{f}_{j,i}(s)}{\partial s} = \begin{cases} 0 & 0 & -1 \dots & a & b & 0 \dots \\ 0 & 0 & 0 \dots & 0 & 0 & 1 \dots \end{cases} \tag{A.6}$$

- $j < i$
  - $k > i$

$$\frac{\partial \mathbf{f}_{j,i}(\phi)}{\partial \mathbf{s}} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = -\frac{\sum_{k=i+1}^{j} dy_k}{q} = a \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = \frac{\sum_{k=i+1}^{j} dx_k}{q} = b \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = -1 \end{cases} \tag{A.7}$$

$$\frac{\partial \mathbf{f}_{j,i}(\beta)}{\partial \mathbf{s}} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = -1 \end{cases} \tag{A.8}$$

  - $k < i$

$$\frac{\partial \mathbf{f}_{j,i}(\phi)}{\partial \mathbf{s}} = \begin{cases} \frac{\partial f_{j,i}(\phi)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\phi)}{\partial d\theta_k} = -1 \end{cases} \tag{A.9}$$

$$\frac{\partial \mathbf{f}_{j,i}(\beta)}{\partial \mathbf{s}} = \begin{cases} \frac{\partial f_{j,i}(\beta)}{\partial dx_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial dy_k} = 0 \\ \frac{\partial f_{j,i}(\beta)}{\partial d\theta_k} = 0 \end{cases} \tag{A.10}$$

$$J_{j,i} = \frac{\partial \mathbf{f}_{j,i}(s)}{\partial \mathbf{s}} = \begin{bmatrix} 0 & 0 & -1 \dots & a & b & -1 \dots \\ 0 & 0 & 0 \dots & 0 & 0 & -1 \dots \end{bmatrix} \tag{A.11}$$

References

[1] C. Berger, Weak constraints network optimiser, in: Proceedings of the International Conference on Robotics and Automation (ICRA), Saint Paul, USA, 2012, pp. 1270–1277.
[2] J. Civera, A.J. Davison, J.M. Martínez Montiel, Inverse depth parametrization for monocular SLAM, IEEE Trans. Robotics 24 (2008) 932–945.
[3] A.J. Davison, Y. Gonzalez Cid, N. Kita, Real-time 3D SLAM with wide-angle vision, in: Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 2004, pp. 117–124.
[4] A.J. Davison, D.W. Murray, Simultaneous localization and map-building using active vision, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 24 (2002) 865–880.
[5] A. Gil, O. Reinoso, M. Ballesta, M. Juliá, L. Payá, Estimation of visual maps with a robot network equipped with vision sensors, Sensors 10 (2010) 5209–5232.
[6] G. Grisetti, C. Stachniss, W. Burgard, Non-linear constraint network optimization for efficient map learning, IEEE Trans. Intell. Transport. Syst. 10 (2009) 428–439.
[7] G. Grisetti, C. Stachniss, S. Grzonka, W. Burgard, A tree parameterization for efficiently computing maximum likelihood maps using gradient descent, in: Proceedings of the Robotics: Science and Systems (RSS), Atlanta, USA, 2007, pp. 1–8.
[8] S. Guadarrama, A. Ruiz-Mayor, Approximate robotic mapping from sonar data by modeling perceptions with antonyms, Inf. Sci. 180 (2010) 4164–4188.
[9] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2004.
[10] C. Joly, P. Rives, Bearing-only SAM using a minimal inverse depth parametrization, in: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, Funchal, Madeira, Portugal, 2010, pp. 281–288.
[11] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, Fast SLAM: a factored solution to the simultaneous localization and mapping problem, in: Proceedings of the 18th National Conference on Artificial Intelligence, Edmonton, Canada, 2002, pp. 593–598.
[12] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Trans. Robotics Automat. 17 (2001) 890–897.
[13] D. Olson, J. Leonard, Fast iterative alignment of pose graphs with poor initial estimates, in: S. Teller (Ed.), Proceedings of the International Conference on Robotics and Automation (ICRA), Orlando, USA, 2006, pp. 2262–2269.
[14] S. Park, S. Kim, M. Park, S.P. Kim, Vision-based global localization for mobile robots with hybrid maps of objects and spatial layouts, Inf. Sci. 179 (2009) 4174–4198.
[15] S.-E. Yu, D. Kim, Image-based homing navigation with landmark arrangement matching, Inf. Sci. 181 (2011) 3427–3442.
[16] C. Stachniss, G. Grisetti, D. Haehnel, W. Burgard, Improved Rao-blackwellized mapping by adaptive sampling and active loop-closure, in: Proceedings of the Workshop on Self-Organization of Adaptive behavior (SOAVE), Ilmenau, Germany, 2004, pp. 1–15.
[17] Y.T. Sun, C.-H. Wang, C.C. Chang, Switching T–S fuzzy model-based guaranteed cost control for two-wheeled mobile robots, Int. J. Innov. Comput. Inf. Control 8 (2012) 3015–3028.
[18] D. Valiente, A. Gil, L. Fernández, O. Reinoso, View-based maps using omnidirectional images, in: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, Rome, Italy, 2012, pp. 48–57.
[19] L. Wu, D.W.C. Ho, Fuzzy filter design for It stochastic systems with application to sensor fault detection, IEEE Trans. Fuzzy Syst. 17 (2009) 233–242.
[20] R. Yang, H. Gao, P. Shi, Delay-dependent robust H control for uncertain stochastic time-delay systems, Int. J. Robust Nonlinear Control 20 (2010) 1852–1865.

Robotics and Autonomous Systems 62 (2014) 108–119

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images

David Valiente *, Arturo Gil *, Lorenzo Fernández, Óscar Reinoso

Miguel Hernández University, Systems Engineering and Automation Department, 03202, Elche, Spain

HIGHLIGHTS

- Proposal to overcome the influence of the non-linear errors on traditional visual SLAM methods.
- We focus on a highly non-linear observation model: the omnidirectional.
- Comparison of traditional filters like EKF, versus SGD.
- Compact map representation, consisting of a reduced set of omnidirectional views.
- We compare accuracy, robustness against errors and speed of convergence.

ARTICLE INFO

Article history:
Received 4 July 2013
Received in revised form
6 November 2013
Accepted 22 November 2013
Available online 4 December 2013

Keywords:
Visual SLAM
SLAM algorithm
EKF
SGD
Omnidirectional images

ABSTRACT

The problem of Simultaneous Localization and Mapping (SLAM) is essential in mobile robotics. The obtention of a feasible map of the environment poses a complex challenge, since the presence of noise arises as a major problem which may gravely affect the estimated solution. Consequently, a SLAM algorithm has to cope with this issue but also with the data association problem. The Extended Kalman Filter (EKF) is one of the most traditionally implemented algorithms in visual SLAM. It linearizes the movement and the observation model to provide an effective online estimation. This solution is highly sensitive to non-linear observation models as it is the omnidirectional visual model. The Stochastic Gradient Descent (SGD) emerges in this work as an offline alternative to minimize the non-linear effects which deteriorate and compromise the convergence of traditional estimators. This paper compares both methods applied to the same approach: a navigation robot supported by an efficient map model, established by a reduced set of omnidirectional image views. We present a series of real data experiments to assess the behavior and effectiveness of both methods in terms of accuracy, robustness against errors and speed of convergence.

1. Introduction

The solution of the SLAM problem is vital for most applications in the field of mobile robotics, for example in navigation tasks. A reliable map representation of the environment has to be built dynamically, in an incremental manner, meanwhile the mobile vehicle requires an appropriated localization inside it, which has to be calculated simultaneously. This fact poses a challenge for the SLAM techniques, since this process involves a notable complexity. The appearance of noise arises as a severe problem, which highly aggravates the achievement of a valid estimation to the problem.

Different SLAM approaches may be classified according to aspects such as the representation of the map, the solver algorithm to compute a solution and the kind of sensor which gathers information of the environment. For instance, the utilization of a laser range sensor [1] has been extensively applied to the obtention of map representations. In this area, two kinds of map representations were principally generated: 2D occupancy grid maps [2] based on raw laser, and 2D landmark-based maps [3] focused on the extraction of features, which were described thanks to laser data measurements. An interesting comparison of both representations is provided in [4].

Nowadays, the emergence of visual sensors has made the tendency to turn into the utilization of digital cameras as the main sensor to gather information. A huge number of applications benefit from the use of these sensors, whose characteristics outperform preceding sensors such as laser, in the sense of the amount of available information. In contrast to laser data sensors, vision sensors provide a wide amount of information of the scene, being as well less expensive, lighter and more efficient in terms of consumption at the price of needing a computational cost to obtain profitable information to build the map. The extraction

* Corresponding authors. Tel.: +34 96 665 9005; fax: +34 96 665 8979.
E-mail addresses: dvaliente@umh.es (D. Valiente), arturo.gil@umh.es (A. Gil), l.fernandez@umh.es (L. Fernández), o.reinoso@umh.es (Ó. Reinoso).

0921-8890/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.robot.2013.11.009

of significant feature points has been a procedure widely used in order to encode the visual information. Diverse arrangements are commonly known by their configuration in reference to the number of cameras they consist of. For instance, approaches which utilize two calibrated cameras, known as stereo-pair, in order to extract a set of 3D visual landmarks determined by a visual description [5]. Other approaches simply exploit a single camera to estimate 3D visual landmarks [6,7]. They initialize the coordinates of each 3D landmark by relying on an inverse depth parametrization, since there exists a scale uncertainty on the distance to each landmark, which cannot be directly calculated by using only a single image. Omnidirectional cameras have also been used solely [8], and even some others have arranged two omnidirectional images [9], following the line stated by stereo-based, but pursuing the major advantage associated with the wider field of view provided by omnidirectional cameras.

The estimator algorithm for a SLAM scheme has to be considered as important as the kind of sensor and the map representation. It represents the core of the system, since it is responsible for the ultimate solution. Amongst the most widely used online methods deserving to be highlighted are the EKF [10] and the Rao-Blackwellized particle filters [3,11]. Regarding the offline algorithms, one of the most effective is SGD [12].

Therefore, the correct balance in the combination of data sensors, map representation and kind of algorithm, eventually determines the effectiveness of a SLAM approach which pursues reliability and suitability for realistic applications. Great efforts have been made in this field. For example, certain approaches [10,5,13,6,14] have concentrated on the estimation of the position of a set of 3D visual landmarks in a main reference system, while dealing with the obtention of the map simultaneously. Their principle of working lays on the capability of an EKF filter to converge the estimation to an appropriate solution for the SLAM problem. In [15], an EKF algorithm also supports an approach which proposes a distinctive map representation, consisting of a reduced set of image views. These views are determined by their position and orientation in the environment. Such technique establishes an estimation of a state vector which includes the map and the current localization of the robot at each timestep.

The methods based on EKF are generally liable to become troublesome when dealing with external errors. This issue is directly deduced from the linearization of variables carried out by the EKF. In this sense, such difficulties compromise the proper convergence of the estimation. This situation normally appears in presence of gaussian noise introduced by the observation measurement, fact that usually causes injurious data association problems [16]. A visual observation model as in the case of the omnidirectional model, is susceptible to introduce non-linearities and thus it is responsible for those kind of errors. On the contrary, an offline algorithm such as SGD [17] provides more robustness to face this issue. It is worth mentioning that the vanilla SGD approach has been modified in this work to deal with omnidirectional geometry as well as with the associated observation model. Traditionally, every odometry and observation measurements are processed in an independent manner. Nevertheless, with the aim of finding a valid solution quickly, we have designed a strategy based on the simultaneous usage of a certain set of observation measurements. This proposal might seem to be likely to cause an increase of the required computational resources. However, we have concentrated on the prevention of such effect by updating several stages of the SGD's iterative optimization. According to this, some amendments have been performed so as to accomplish the avoidance of possible harmful bottleneck handicaps.

Hence, the main goal of this paper is to provide with results which help analyze the behavior of both EKF and SGD applied to a view-based SLAM approach. As it can be inferred, the solution's convergence is not trivial with EKF, neither with SGD, especially when the nature of the observation measurement is up to a scale factor. The results extracted from the experiments are intended to assess the capability of both methods to maintain a feasible estimation under different conditions. Estimation accuracy, robustness and convergence of the estimation and speed of convergence will be the most important terms to evaluate.

The structure of the paper has been divided as it follows: Section 2 introduces the most important aspects of the visual SLAM approach proposed here. The EKF principles are detailed in Section 3. Then, Section 4 concentrates on the SGD's specifications. Next, Section 5 provides a series of experiments in order to extract real data results. Finally, Section 6 pursues the analysis of the results and the discussion.

2. SLAM

The main purpose of a visual SLAM scheme is to retrieve a reliable representation of the environment explored by the robot, as well as the position of this vehicle. In this approach, the map of the environment is defined by a set of omnidirectional images acquired from different poses of the robot along the environment, denoted as views. These views do not express information about any physical landmarks as it is traditionally in the field of vision-based SLAM. By contrast, a view consists of a single omnidirectional image captured at a certain pose of the robot $x_l = (\mathrm{x}_l, \mathrm{y}_l, \theta_l)$ and a set of interest points extracted from that image. In accordance with the large field of view provided by omnidirectional images, such arrangement allows us to exploit this capability to gather a large amount of information of the scene in a single image. Thus, a highly notable reduction in terms of number of variables to estimate the solution is achieved.

The position of the mobile robot is denoted as:

$$x_v = (\mathrm{x}_v, \mathrm{y}_v, \theta_v)^T. \tag{1}$$

Each view n with $n \in [1, \ldots, N]$ is constituted by its pose:

$$x_{l_n} = (\mathrm{x}_l, \mathrm{y}_l, \theta_l)_n^T \tag{2}$$

together with its uncertainty P_{l_n} and a set of M interest points p_j, expressed in image coordinates. Each point is associated with a visual descriptor $d_j, j = 1, \ldots, M$.

Therefore, these are the variables which compose the augmented state vector:

$$\bar{x} = \begin{bmatrix} x_v & x_{l_1} & x_{l_2} & \cdots & x_{l_N} \end{bmatrix}^T. \tag{3}$$

2.1. Map building

The process of map building may be clearly understood by inspecting an example in Fig. 1. It shows the exploration procedure carried out by a robot, which starts its navigation of the environment at the origin A. At this moment, capturing an omnidirectional image I_A is required to determine the first view of the map. This view is associated with the pose x_{I_A} and it encodes the relevant information of the local area around this pose. Then, the robot moves towards the first office room. Assuming that the robot does not find any major obstruction, it will be capable of extracting correspondences between I_A and the omnidirectional image referred to the pose where it currently moves through. This procedure makes it able to localize itself. Once the robot enters in the office room, the appearance of the images vary significantly, thus, no matches are found between the current image and image I_A. In this case, the robot will initialize a new view into the map I_B at the current robot position x_{l_B}. Now, this view will facilitate the localization of the vehicle inside this office room. Finally, the

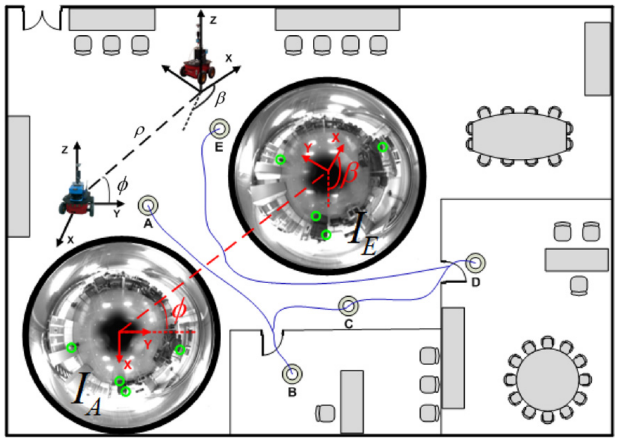


Fig. 1. Map building process. Origin is set at *A*, where a first view I_A is initiated into the map. While the robot traverses the environment, correspondences may be found between I_A and the current image captured at the current robot's pose. In case that no correspondences are found, a new view is initiated as the current image, for instance I_B at *B*. The procedure finalizes when the entire environment is represented.

robot concludes the exploration of the environment by successfully achieving a well-defined trajectory and a map representation of the different areas. As it may be seen, it has been necessary to acquire a set of views I_C, I_D, I_E to complete the final map. The size of the map in terms of the number of views initiated, directly depends on the specific appearance of the environment. Fig. 1 also depicts how the robot accomplishes the computation of its localization, by which it eventually obtains two relative angles thanks to the processing of the information provided by I_A and I_E.

The relative appearance between images is determined by a specific ratio, which it has been experimentally defined as:

$$A = k\frac{c}{p_1 + p_2} \tag{4}$$

where p_1 and p_2 are the interest points detected on each image and c are the corresponding points found between them. The value of k has also been experimentally determined according to the visual appearance of the environment. The ratio A represents a measure of similarity and it is the factor which ease the robot to decide whether to initialize a new view in the map. In particular, the robot will initialize a new view whenever the ratio A drops a certain threshold.

2.2. Data association

The data association problem is posed in the following way: given a set of observations $z_t = [z_{t_1}, \ldots, z_{t_B}]$ at each t, the views which generate each observation have to be discerned. In the approach presented here, the data association process is tackled through the computation of the appearance ratio A. First, we select a subset of candidate views from the map, based on the euclidean distance between the current pose of the robot and the position of each candidate, $D_n = \sqrt{(x_v - x_{l_n})^T (x_v - x_{l_n})}$. The maximum observation range of the robot is established as the maximum distance at which any view can be observed at each t. Then we extract corresponding points between the image acquired at the current pose of the robot and the rest of the candidate views. This allows to find the view which provides the maximum appearance ratio A, defined in (4), which will eventually be chosen as the data association. The view with maximum A reveals the highest similarity with the current image. However, if none of the candidate views provide a value for A higher than a predefined threshold, this will mean that the appearance of the current image of the robot differs substantially from the set of candidate views. Therefore it will be necessary to initialize a new view into the map at the current robot's position.

2.3. Observation model

In consequence with the view-based representation, the formulation of a new observation model is required. The intention is to retrieve a motion transformation between two poses. As observed in Fig. 1 a comparison involving two images provides a motion transformation between two poses. In fact these poses represent the positions where the robot acquired these two specific images. To that effect, only two images with a set of corresponding points between them are required to obtain the transformation. So that the observation measurement may be expressed as:

$$z_t = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{pmatrix} \arctan\left(\dfrac{y_{l_n} - y_v}{x_{l_n} - x_v}\right) - \theta_v \\ \theta_{l_n} - \theta_v \end{pmatrix} \tag{5}$$

where ϕ and β are the relative angles which express the bearing and orientation at which the view i is observed. Please notice that the structure of the view i follows (2), whereas the pose of the robot is described in (1). Both measurements (ϕ, β) are shown in Fig. 1. Please note that the feature point detector chosen is SURF [18] due to its success and robustness when working with omnidirectional images [19].

3. EKF

The EKF [20] is the first algorithm which has been considered in this work to be applied to the case of visual SLAM with the intention of generating a valid estimation for the problem.

The basis of this filter lays on the estimation of the augmented state vector which is constantly updated in real time. In this framework of a view-based representation, the variables to estimate are the map itself, consisting of views and their poses, and

the pose of the robot inside it. Hence the state vector defined in (3) can be adapted to introduce t:

$$\bar{x}(t) = [x_v, x_{l_1}, x_{l_2}, \cdots, x_{l_N}]^T. \quad (6)$$

Once the state vector is defined, the transformation relation between $\bar{x}(t)$ and $\bar{x}(t+1)$ is:

$$\bar{x}(t+1) = F(t)\bar{x}(t) + u(t+1) + v(t+1) \quad (7)$$

where $F(t)$ contains the information pertinent to the transition between states, $u(t+1)$ is the vector related to the movement generated by the odometry of the wheels of the robot, and $v(t+1)$ represents the noise introduced in the system, which has gaussian uncorrelated nature.

Similarly, a linear relation may be defined so as to connect the observation measurement $z_i(t)$ with the current state vector:

$$z_i(t) = H_i(t)\bar{x}(t) + w_i(t) \quad (8)$$

where $H_i(t)$ encodes the relation between $\bar{x}(t)$ and $z_i(t)$. Here, $w_i(t)$ represents the random noise generated by the sensors, which is gaussian and with covariance $R(t)$.

Then, the filter's procedure has to be divided into three fundamental stages well differentiated. Firstly, a prediction of the state $\hat{x}(t)$ is carried out, and based on it, a prediction for the observation measurement $\hat{z}_i(t)$ is also proposed in the following terms:

$$\hat{x}(t+1|t) = F(t)\hat{x}(t|t) + u(t) \quad (9)$$

$$\hat{z}_i(t+1|t) = H_i(t)\hat{x}(t+1|t) \quad (10)$$

$$P(t+1|t) = F(t)P(t|t)F^T(t) + Q(t) \quad (11)$$

where $P(t|t)$ and $P(t+1|t)$ are the covariance matrices which represent the uncertainty of the estimation at instants t and $t+1$ respectively.

The second stage performs the real observation $z_i(t)$ at the current instant t, of a specific view i of the map. Now the concept of innovation has to be introduced to explain the deviation between the prior prediction $\hat{z}_i(t)$ and the current measurement $z_i(t)$:

$$v_i(t+1) = z_i(t+1) - \hat{z}_i(t+1|t) \quad (12)$$

$$S_i(t+1) = H_i(t)P(t+1|t)H_i^T(t) + R_i(t+1) \quad (13)$$

where $S_i(t+1)$ represents the innovation's covariance.

Finally, the third stage takes into account the refinement of the estimation obtained during the first stage, seen as an updating step. The value of the innovation is significantly relevant in the computation of the final solution provided by the filter. This solution estimation at instant $t+1$, is finally obtained as:

$$\hat{x}(t+1|t+1) = \hat{x}(t+1|t) + K_i(t+1)v_i(t+1) \quad (14)$$

$$P(t+1|t+1) = P(t+1|t) - K_i(t+1)S_i(t+1)K_i^T(t+1) \quad (15)$$

where in this case $K_i(t+1)$ plays a role of weighting, and corresponds to the gain of the EKF. It is calculated in the following manner:

$$K_i(t+1) = P(t+1|t)H_i^T(t)S_i^{-1}(t+1). \quad (16)$$

It is worth mentioning that the matrices referred to the noise's covariance $Q(t)$ y $R(t)$ have to be initialized. $Q(t)$ is established by means of the noise parameters which characterize the odometry of the wheels of the vehicle. On the other hand, $R(t)$ is determined by experimental accuracy thresholds associated with the visual sensor. The odometry $u(t)$ is required as an initial seed for the prediction obtention, together with the previous state, as deduced from (9). The uncertainty matrix of the map, $P(t)$, considers the noise introduced by the odometry in the form presented in (11), and the noise introduced by the visual sensor when carrying out an observation measurement, as detailed in (13) and (15).

3.1. Correspondence of interest points

With the aim of obtaining a set of feasible correspondences between two views, some restrictions have to be taken into account. Considering the use of epipolar constraints is generally agreed to delimit the search for correspondences [21]. The same point detected in a first camera reference system, denoted as $p = [x, y, z]^T$, may be expressed as $p' = [x', y', z']^T$ in the second camera reference system. Then, the epipolar condition is used to state the relationship between both 3D points p and p' seen from different views.

$$p'^T E p = 0 \quad (17)$$

where the matrix E is the essential matrix and it can be computed from a set of corresponding points in two images.

$$E = \begin{bmatrix} 0 & 0 & \sin(\phi) \\ 0 & 0 & -\cos(\phi) \\ \sin(\beta-\phi) & \cos(\beta-\phi) & 0 \end{bmatrix} \quad (18)$$

being ϕ and β the relative angles that determine a planar motion transformation between two different views, as shown in Fig. 1 and (5).

The avoidance of false correspondences has been studied extensively so as to mitigate bad effects on the final estimation for the SLAM problem. Techniques such as RANSAC and Histogram voting have been widely used, and mainly applied to visual odometry approaches [21]. Together with the epipolar constraint (17), they reveal good results in the achievement of false positive rejection. In such context of visual odometry, consecutive images are close enough to disregard high errors in the pose from where images were taken, so that the epipolar constraint is highly likely to be satisfied. Nevertheless, concentrating on the framework of our SLAM problem, the accumulative uncertainties are substantially higher, either in the pose of the robot or in the pose of the views which compose the map. This fact requires to define a reliable strategy to accomplish with a correct data association. We rely on the information provided by the predicted state vector extracted from the Kalman filter, by which we are able to obtain a predicted observation measurement $\hat{z}_t$, as stated in (5). Then it is also necessary to consider the current map uncertainties so as to deal with a realistic search for valid corresponding points between images. The map uncertainties are propagated in accordance with (17) by introducing a dynamic threshold δ. In an idealistic case, the epipolar constraint may equal a fixed threshold, implying that the epipolar curve defined between images always presents a little static deviation. On the contrary, a realistic SLAM approach, should consider that this threshold depends on the existing error on the map, which dynamically varies at each step of the SLAM algorithm. Since this error is correlated with the error on $\hat{z}_t$, we rename δ as $\delta(\hat{z}_t)$. In addition, it has to be noted that (18) is defined up to a scale factor, which is another reason to keep $\delta(\hat{z}_t)$ as a variable value. Therefore, given two corresponding points between images, they must satisfy:

$$p'^T \hat{E} p < \delta(\hat{z}_t). \quad (19)$$

This approach not only mitigates the undesired harmful effects associated with false positives, but also simplifies the search for corresponding points between images as it restricts the area where correspondences are expected. The procedure is depicted in Fig. 2, where a detected point $P(x, y, z)$ is assumed, and it is represented in the first image reference system by a normalized vector $\vec{p_1}$ due to the unknown scale. To deal with this scale ambiguity, we suggest a point distribution to generate a set of multi-scale points $\lambda_i \vec{p_1}$, being

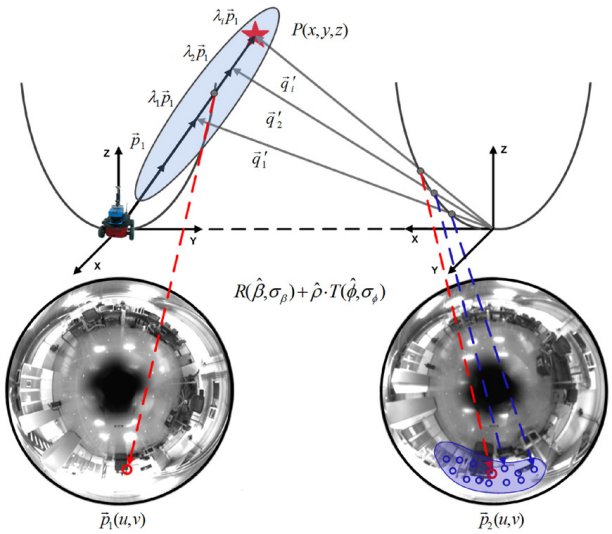


Fig. 2. Given a detected point $\vec{p}_1$ in the first image reference system, a point distribution is generated to obtain a set of multi-scale points $\lambda_i\vec{p}_1$. By using the Kalman prediction, they can be transformed into $\vec{q}_i'$ in the second image reference system by means of $R \sim \mathrm{N}(\hat{\beta}, \sigma_\beta)$, $T \sim \mathrm{N}(\hat{\phi}, \sigma_\phi)$ and $\hat{\rho}$. Finally $\vec{q}_i'$ are projected into the image plane to determine a restricted area where correspondences have to be found. Circled points represent the projection of the normal point distribution for the multi-scale points that determine this area.

representative for the lack of scale in $\vec{p}_1$. This distribution considers a valid range for λ_i according to the predicted $\hat{\rho}$. Please note that the error of the current estimation of the map has to be propagated along the procedure. To that end, we look back to the Kalman filter theory, where the innovation is defined as the difference between the predicted $\hat{z}_t$ and the real z_t observation measurement as stated in (12), and the covariance of the innovation defined in (13). So that $S_i(t+1)$ presents the following structure:

$$S_i(t+1) = \begin{bmatrix} \sigma_\phi{}^2 & \sigma_{\phi\beta} \\ \sigma_{\beta\phi} & \sigma_\beta{}^2 \end{bmatrix}. \tag{20}$$

As the predicted $\hat{E}$ can be decomposed in a rotation $\hat{R}$ and a translation $\hat{T}$, we can transform the distribution $\lambda_i\vec{p}_1$ into the second image reference system, obtaining $\vec{q}_i'$. The introduction of (20) allows to propagate the error, and thus it redefines a transformation between images through the normal distributions $R \sim \mathrm{N}(\hat{\beta}, \sigma_\beta)$ and $T \sim \mathrm{N}(\hat{\phi}, \sigma_\phi)$. Therefore $\vec{q}_i'$ is a gaussian distribution correlated with the current map uncertainty. Once obtained $\vec{q}_i'$, they are projected into the image plane of the second image, seen as circled points in Fig. 2. This projection of the normal multi-scale distribution determines the predicted area which is drawn with a continuous curve line on the omnidirectional image. This area establishes the specific image pixels where correspondences for $\vec{p}_1$ must be searched for. The shape of this area depends on the error of the prediction, which is directly correlated with the current uncertainty of the current map estimation. Dash lines represent the possible candidate points located inside the predicted area. Hence the problem of matching is simplified to the search for the correct corresponding points for $\vec{p}_i$ amongst those candidates inside a restricted area, instead of a global search along the whole image.

4. SGD

4.1. Structure

The SGD algorithm has been the second method considered in this work to be applied to the case of visual SLAM and it is responsible for generating a feasible estimation for the problem.

In this case, the problem is dealt with a graph-oriented map, which contains a set of nodes to define the poses traversed by the robot and the views initialized into the map. It is considered as a maximum-likelihood estimator, and it seeks a least squares minimization [22]. The state vector s_t encodes this representation through a set of variables which are expressed in the following manner:

$$s_t = \left[(x_0, y_0, \theta_0), (x_1, y_1, \theta_1) \cdots (x_n, y_n, \theta_n)\right] \tag{21}$$

being (x_n, y_n, θ_n) the 2D position and orientation of each node in a general reference system. Despite the fact that this kind of representation seems the most natural and intuitive, such global encoding has the main drawback of not being capable to update more than one node and its adjacents per constraint. This aspect has led to a general agreement in the use of the incremental representation:

$$s_t^{inc} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (dx_1, dy_1, d\theta_1) \\ \vdots \\ (dx_n, dy_n, d\theta_n) \end{bmatrix} \tag{22}$$

where $(dx_n, dy_n, d\theta_n)$ represents the deviation between two consecutive poses in the global reference system. According to the

formulation defined in (1) and (21), x_v and each x_{l_n} correspond with (x_0, y_0, θ_0), $(x_1, y_1, \theta_1) \cdots (x_n, y_n, \theta_n)$, and thus:

$$s_t^{inc} = \begin{bmatrix} (x_0, y_0, \theta_0) \\ (x_1 - x_0, y_1 - y_0, \theta_1 - \theta_0) \\ (x_2 - x_1, y_2 - y_1, \theta_2 - \theta_1) \\ \vdots \\ (x_n - x_{n-1}, y_n - y_{n-1}, \theta_n - \theta_{n-1}) \end{bmatrix}. \tag{23}$$

Now, the state vector is differentially encoded and each single update has influence on the whole map reestimation.

Regarding the observation measurements, a complementary subset of edges are introduced to relate nodes to each other. That is to say, they express the observation measurements between poses, either from odometry of the wheels or visual sensors. The nomenclature commonly refers to the observations as constraints, and it denotes them as δ_{ji}, where j indicates the observed node, seen from node i. The general objective stated by these kind of methods [23,12] is to minimize the error likelihood expressed as:

$$P_{ji}(s) \propto \eta \exp\left(-\frac{1}{2}(f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji})\right) \tag{24}$$

being $f_{ji}(s)$ a function dependent on the state s_t and both nodes j and i. The difference between $f_{ji}(s)$ and δ_{ji} expresses the error deviation between nodes, which in this case are views of the map and poses traversed by the robot. Such error term is weighted by the information matrix:

$$\Omega_{ji} = \Sigma_{ji}^{-1} \tag{25}$$

where Σ_{ji}^{-1} is the inverse covariance matrix responsible for the uncertainty of the observation measurements. After taking the logarithm we have:

$$F_{ji}(s) \propto (f_{ji}(s) - \delta_{ji})^T \Omega_{ji} (f_{ji}(s) - \delta_{ji}) \tag{26}$$

$$= e_{ji}(s)^T \Omega_{ji} e_{ji}(s) = r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{27}$$

being $e_{ji}(s)$ the error resultant from $f_{ji}(s)$–$\delta_{ji}(s)$, which is also named as $r_{ji}(s)$ to emphasize its condition of residue. Finally, the global problem seeks the minimization of the objective function which represents the accumulated error on the map:

$$F(s) = \sum_{\langle j,i \rangle \in G} F_{ji}(s) = \sum_{\langle j,i \rangle \in G} r_{ji}(s)^T \Omega_{ji} r_{ji}(s) \tag{28}$$

where $G = \{\langle j_1, i_1 \rangle, \langle j_2, i_2 \rangle \ldots\}$ defines the subset of particular constraint conforming the map, either pertaining to odometry or visual observation measurements.

4.2. Estimation

Once the formulation of the problem has been stated, the SGD algorithm develops an iterative process to reach a valid estimation for the SLAM problem. The basis of a SGD method lays on the minimization of (28) through derivative optimization techniques such as mean square estimators, so that the estimated state vector is obtained as:

$$s_{t+1} = s_t + \Delta s \tag{29}$$

where Δs updates s_t, by means of an adaptive constraint's optimization. It is worth noting that in a general case, this update is calculated independently at each step by using only a single constraint, that is to say $\Delta s = f(\delta_{ji})$. The general expression for the transition between s_t and s_{t+1} has the following form:

$$s_{t+1} = s_t + \lambda \cdot H^{-1} J_{ji}^T \Omega_{ji} r_{ji}. \tag{30}$$

- $J_{ji}(s)$ is the Jacobian of $f_{ji}(s)$ with respect to s_t. It translates the error deviation into a spacial variation.
- H is the Hessian matrix, calculated as $J^T \Omega J$, and it shapes the error function through a preconditioning matrix to scale the variations of J_{ji}:

$$H \approx \sum_{\langle i,j \rangle} J_{ji} \Omega_{ji} J_{ji}^T. \tag{31}$$

- Ω_{ji} is the information matrix associated with a constraint, and equals Σ_{ji}^{-1}.
- λ is a learning factor to re-scale the term $H^{-1} J_{ji}^T \Omega_{ji} r_{ji}$. Normally, λ follows a decreasing criteria such as $\lambda = 1/n$, where n is the iteration step. This strategy pretends to achieve a final estimation by using higher values of λ at first steps, and presuming that lower values of λ will be useful in preventing from oscillations around the final solution.

This method updates the estimation by computing the rectification introduced by each constraint at each iteration step respectively. Despite the fact that the learning factor reduces the weight by which each constraint updates the estimation, the procedure may be inefficient as it may lead to an unstable solution. Undesired oscillations may occur due to the stochastic nature of the constraints' selection. For this reason, we propose an optimization process which takes into account several constraints at the same iteration step. Such idea might cause undesired overloads of time. However, we also propose some amendments to avoid this effect, which succeed in maintaining the time requirements and even reduce them.

4.3. Adaption to omnidirectional images

Regarding the observation measurements provided by an omnidirectional camera, some assumptions have to be contemplated in the structure of the SGD algorithm.

Note that in this approach we are dealing with a visual observation given by an omnidirectional camera. This fact requires the adaption of the equations defined in the previous section, since the nature of the constraints are not only metrical like odometry's constraints. Following, we detail the terms related to the observation measurements, emphasizing on the visual observation, which has been redefined in consequence with (5):

- The first adaption was referred to $f_{ji}(s)$, differentiating between odometry and visual observation constraints:

$$f_{j,i}^{odo}(s) = \begin{pmatrix} dx_j \\ dy_j \\ d\theta_j \end{pmatrix} + \begin{pmatrix} dx_{j-1} \\ dy_{j-1} \\ d\theta_{j-1} \end{pmatrix} + \cdots + \begin{pmatrix} dx_i \\ dy_i \\ d\theta_i \end{pmatrix} \tag{32}$$

$$f_{j,i}^{visual}(s) = \begin{pmatrix} \phi \\ \beta \end{pmatrix} = \begin{bmatrix} \arctan\left(\dfrac{dy_j - dy_i}{dx_j - dx_i}\right) - d\theta_i \\ d\theta_j - d\theta_i \end{bmatrix} \tag{33}$$

  where ϕ and β express the relation between views and the pose codification (21), and are directly computed as defined in [15]. Visual inspection of Fig. 1 may ease to define (33).
- Then, it is necessary to recalculate $J_{ji}(s) = \frac{\partial f_{ji}(s)}{\partial s}$, accordingly with the previous reformulation. It has to be noticed the importance of considering the value of each node's index, being either $j > i$ or $j < i$, since the derivatives vary its form considerably. Furthermore, as seen above, the dimensions of $f_{ji}(s)$ are different, fact which has also to be considered in order to resize the rest of the terms involved in the SGD algorithm.

$$J_{ji}(s) = \frac{\partial f_{ji}(s)}{\partial s} = \left[\frac{\partial f_{ji}(\phi)}{\partial s}, \frac{\partial f_{ji}(\beta)}{\partial s}\right]. \tag{34}$$

- Lastly, we also propose that the estimation of the new state s_{t+1} reflects the usage of several constraints at the same time. We seek more relevance of constraints' weight when searching for the optimal minimum estimation. Obviously, computing more than one constraint at each step may cause a certain overload. Contrarily, in this approach we reduce the expensive estimation of H. In a general case, at every step, H is computed as many times as constraints exist in the map. In opposition with this, we only compute H once for each subset of constraints introduced simultaneously into the system at each step. Thus we dramatically reduce the number of times that H is calculated, so that we proceed in a more efficient manner which compensates possible time overloads.

5. Results

We have performed different real data experiments in an office environment. The equipment utilized in the experiments consisted of a Pioneer P3-AT indoor robot equipped with a firewire 1280 × 960 camera and a hyperbolic mirror to build the omnidirectional image. The optical axis of the camera is installed approximately perpendicular to the ground plane, as described in Fig. 3. As a result, a rotation of the robot corresponds to a rotation of the image with respect to its central point. In addition, we used a SICK LMS range finder in order to compute a ground truth by means of the method presented in [2]. The exposition of the results is structured as it follows: First in Section 5.1 we show SLAM results obtained with both methods EKF and SGD when the dimension of the map in terms of N views is variable. Then in Section 5.2, we also compare both methods by testing their accuracy and robustness on the estimation when data association errors arise. Finally, in Section 5.3 we present results with regard to the speed of convergence.

5.1. SLAM results with EKF and SGD

This experiment has been conducted in an indoor environment which corresponds to an office area of 42 × 32 m. The robot navigates this area while it acquires omnidirectional images and laser data along the trajectory. The laser data is an auxiliary reference to aid in generating a ground truth for fair comparison.

In the EKF's case, as mentioned above, the procedure of map building is accomplished in an incremental manner. Fig. 4 shows the results obtained in this experiment, where the robot starts the SLAM process by adding the first view of the map. Next, it keeps moving along the trajectory while capturing omnidirectional images. The image at the current robot pose is compared with the views stored in the map so as to extract some corresponding points that allow the robot to compute a relative measurement of its position, as explained in Section 2. The robot decides to initiate a new view whenever the relative appearance of the current image compared to the appearance of the map's views drops below a specific similarity threshold R. The ellipses indicate the uncertainty in the pose of each view and the robot. The dash-dotted line represents the solution obtained with the EKF approach, indicating with crosses the points along the trajectory where the robot decided to initiate new views in the map. The continuous line represents the ground truth whereas the odometry is drawn with dash line. The modification of R, leads to a variation of the size of the map in terms of N. As it can be observed in Fig. 4(a), a map for an environment of 42 × 32 m may be perfectly generated by a reduced set of $N = 5$ views, thus leading to a compact representation. However, the same environment may also be represented with a different number of views N as shown in Fig. 5(a). Figs. 4(b) and 5(b) compare the errors for the estimated trajectory, each one associated with the maps composed by $N = 5$ and $N = 20$ views respectively. Based on the ground truth comparison, the solution error is shown with dash-dotted line and the odometry's with dash line at every step of the trajectory. The validity of the solution is confirmed due to the accomplishment of the convergence requirements. It may be noticed that the solution error is inside the 2σ interval, drawn in continuous line, whereas the odometry error grows out of bounds. According to these results, it should be noticed that the higher values of N the lower the resultant error in the map.

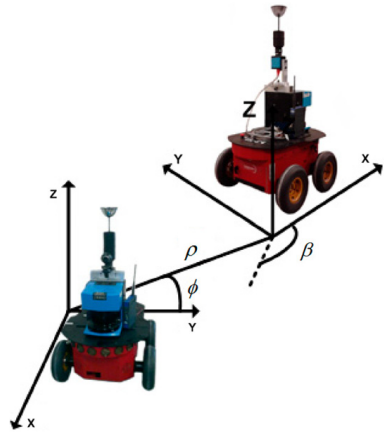


Fig. 3. Robot Pioneer P3-AT used in the experiments. Two poses are indicated with their corresponding relative angles which determine the motion transformation.

On the other hand, we run the same experiment with a SGD estimator. Fig. 6(a) and (b) represent the same two situations with $N = 5$ and $N = 20$ views previously performed. The placement of the views is exactly the same. The main difference in the manner to proceed with respect to EKF is that SGD processes the observations offline. Inspecting Figs. 4(a), 5(a), 6(a) and (b) reveals that EKF estimations are more accurate than the SGD estimations. To generalize, Fig. 7 establishes a fair comparison between both methods, where the RMS (Root Mean Square) error along the path is represented versus the number of views N. The continuous line shows the RMS error for SGD and the dash line shows the EKF's. The results of EKF outperforms in this case SGD's. However, this experiment has dealt with a desirable situation where non-linear errors, if any, were low enough so that the EKF response was able to ensure convergence. The following experiment will show the results obtained when the visual information is damaged and corrupted by significative noise errors.

5.2. Comparing accuracy

Now we intend to compare the behavior of both methods in a more realistic situation, that is to say, when they are expected to suffer non-linear errors introduced by the observation measurements and it consequently causes wrong data association errors. We have conducted the same real experiment shown above but assuming a highly relevant presence of non-gaussian errors. To that end, we have modeled a random generator scheme which introduces wrong data associations. At each estimation step, the robot computes the observation measurements for the entire set of views which is able to observe. However the robot fails to associate the observation measurement with its corresponding view at a

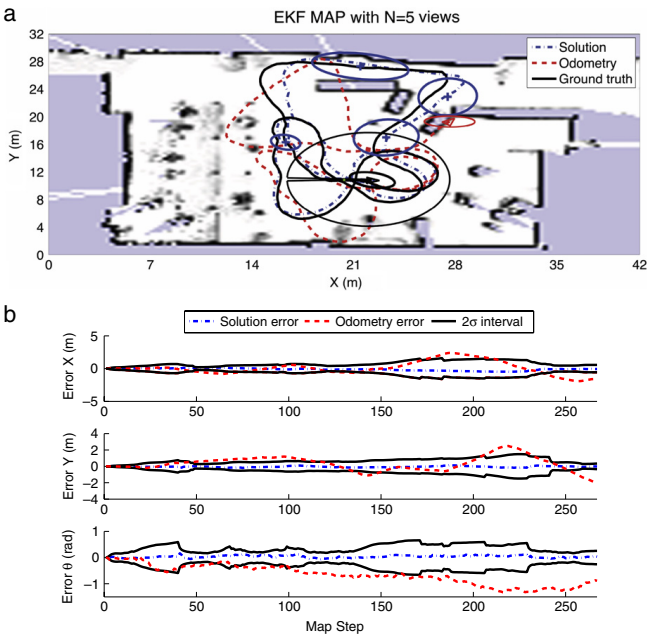


Fig. 4. (a) presents results of SLAM using an EKF algorithm with real data. The map representation of the environment is formed by $N = 5$ views. The position of the views is presented with error ellipses. (b) shows the solution and the odometry error in X, Y and θ at each time step.

certain probability, meaning that a percentage out of the total data association are wrong, and thus the observation measurement as well.

Fig. 8(a) and (b) describe the RMS error tendency of both methods, when data association fails with a given probability. The experiment has been repeated 200 times in order to retrieve consistent and coherent mean values. Again, the environment has been represented with different values of N in order to show differences. The results provided by EKF reveal that the resultant RMS error grows out of bounds when the probability of data association error is apparently low. This fact demonstrates the low reliability of the EKF when it has to deal with non-linearities and thus non-gaussian errors. Despite the fact that maps with more views provide a larger number of observation measurements to enable the rectification of the estimation, the error continuously increases. The results prove that once the solution diverges, the EKF is unable to recover it, despite the fact that N is higher. Consequently, the difficulties experienced by the EKF to keep the convergence of the estimation are evidenced.

Contrary to the EKF's results, and according to Fig. 8(b), the SGD provides a lower RMS error under the same conditions. Moreover, it ensures convergence, as the RMS's tendency only increases slightly. It is worth noting the importance of selecting a suitable value for λ, so that new updates to s_{t+1} do not lead the estimation to diverge when there is evidence of errors. In this case, the SGD proves its capability to rectify the solution even in presence of non-linearities and the consequent non-gaussian errors. Therefore, in the case of SGD, as it could be intuitively expected, the more N views in the map, the more observations gathered, and thus the better results provided.

5.3. Comparing speed of convergence

As it may be seen in the previous subsection, the SGD outperforms EKF in terms of robustness and accuracy when the system is considerably affected by non-gaussian errors. However, one should think about the speed of convergence of both methods. A compromising solution will have to be agreed so as to ensure a balance which provides robustness against the influence of noisy terms and speed of computation. With this experiment we would like to compare the speed ratios by which EKF and SGD compute a valid solution. Fig. 9 represents the time consumption to reach a valid solution versus the number of views N of the map. Since we look for a fair comparison, the y-axis, has been transformed into a normalized time variable which achieves a trustworthy comparison between both schemes. This adoption has been considered since the stochastic nature of the SGD method may lead each experiment to last a different number of iterations, and consequently a different time. Therefore the mean values for each iteration step have to be considered, so that the final estimation time can be obtained. Hence this normalization allows a fair and simpler comparison between methods.

In this sense, it may be proved that the solution provided by EKF outperforms the solution given by a basic SGD for each N-view map, since its gradient is definitely lower. However, it is also worthwhile to analyze these results together with the tendency of each corresponding RMS error. Fig. 10(a) and (b) show the normalized RMS error, versus the total time consumption to reach the final estimation. Now it can be clearly confirmed that quicker speed of convergence is assured by EKF.

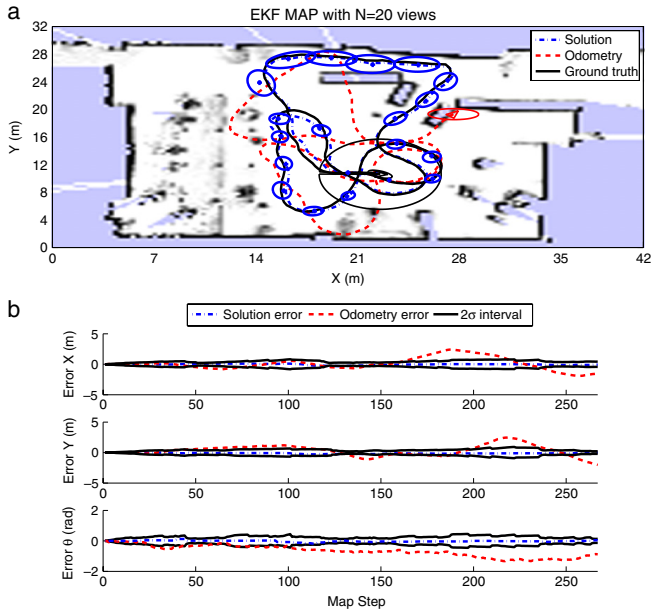


Fig. 5. (a) presents results of SLAM using an EKF algorithm with real data. The map representation of the environment is formed by $N = 20$ views. The position of the views is presented with error ellipses. (b) shows the solution and the odometry error in X, Y and θ at each time step.

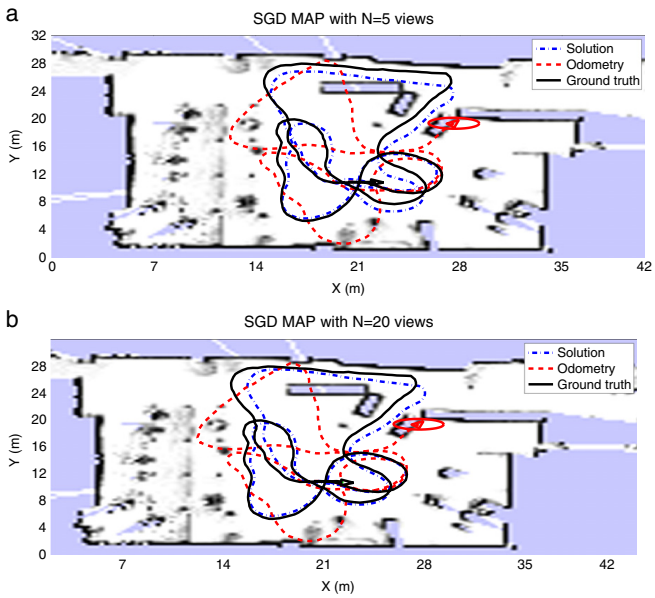


Fig. 6. (a) and (b), present results of SLAM using a SGD algorithm with real data. These map representations of the environment are formed by $N = 5$ and $N = 20$ respectively. The dash-dotted line represents the solution obtained with the SGD approach, the continuous line represents the ground truth whereas the odometry is drawn with dash line.

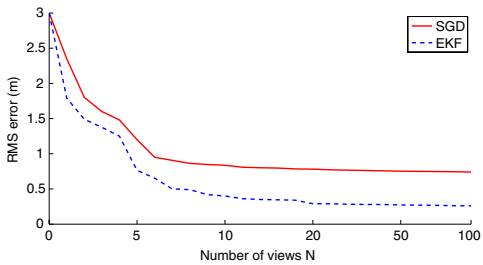


Fig. 7. RMS error (m) versus the number of views N of the map. The continuous line shows the error for the solution provided by SGD, meanwhile the dash line shows the error for the solution obtained with EKF.

6. Conclusions

We have presented a comparison between EKF and SGD algorithms, according to their provided solution to the Simultaneous Localization and Mapping (SLAM) approach. The main issue to analyze has been the influence of non-linear errors, which are a clear indicator of added noise by the visual sensor's measurements, especially associated with the omnidirectional observation model. We have presented a real data experimental set, which has considered different modifications so as to test the behavior of both methods under different conditions. The approach to the map representation relies on an efficient view-based map model, which is built by means of a reduced set of omnidirectional image views. Bearing in mind the results presented in this work, a key aspect to remark about EKF is definitely its capability to provide a suitable estimation in real time, thanks to its adequate speed of

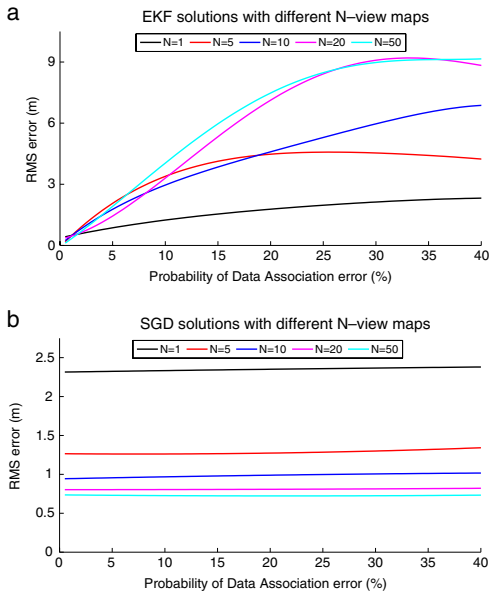


Fig. 8. (a) and (b) presents the RMS error (m) versus the probability of data association error (%) for EKF and SGD respectively. Errors for maps with different number of views N are indicated.

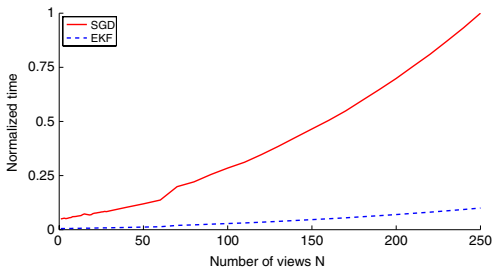


Fig. 9. Normalized time consumption versus number of views N of the map. The continuous line shows the time consumed by SGD, meanwhile the dash line shows the time consumed by EKF.

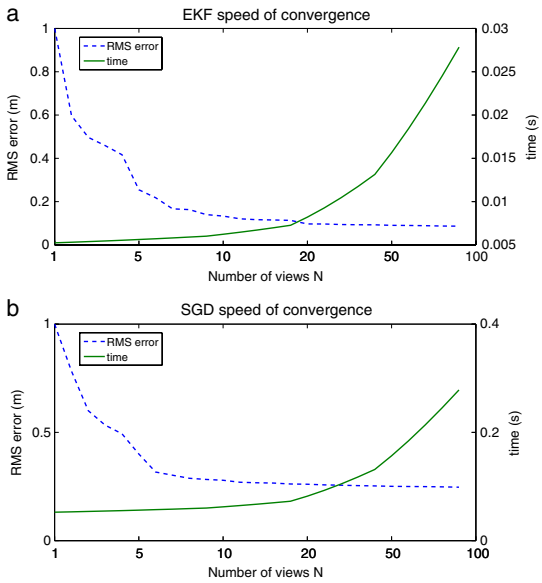


Fig. 10. (a) and (b) present the normalized RMS error (m), and time consumption (s) versus the number of views N of the map for EKF and SGD respectively. The dash lines show the RMS error, meanwhile the continuous lines show the time consumed by EKF and SGD respectively.

convergence. Moreover, other favorable aspect in case of an idealistic situation without clear evidence of non-linearities, is that EKF provides a more accurate estimation in contrast to SGD. On the other hand, contrary to EKF, the SGD has evidenced to be more reliable when a robust solution is required. Despite the fact that SGD's accuracy in an idealistic situation is lower than the EKF's, the results obtained in presence of non-linear noise effects, indicate that SGD provides a solid and stable solution which prevents the system from diverging. As it is well known, this is not accomplished by EKF, since is highly sensitive to errors due to the linearization of the variables of the filter. However, the SGD reveals a lower speed of convergence.

Therefore it has been proved that the effectiveness of each method depends on the assumed conditions. Assuring and approach to SLAM which achieves the avoidance of the effects of non-linearities and non-gaussian errors, would lead to select a SGD method. Nevertheless, in case of dealing with a more desirable situation, such as in a low-noise environment, would indicate that an EKF method would be more appropriated in order to succeed in providing a more precise solution with a higher rate of convergence.

Acknowledgments

This work has been supported by the Spanish government through the project DPI2010-15308, and the grant program FPI2011.

References

[1] Y. Chou, L. Jing-Sin, A robotic indoor 3D mapping system using a 2D laser range finder mounted on a rotating four-bar linkage of a mobile platform, Int. J. Adv. Robot. Syst. 10 (2013), http://dx.doi.org/10.5772/54655.

[2] C. Stachniss, G. Grisetti, D. Haehnel, W. Burgard, Improved Rao–Blackwellized mapping by adaptive sampling and active loop-closure, in: Proceedings of the Workshop on Self-Organization of Adaptive Behavior (SOAVE), Ilmenau, Germany, 2004, pp. 1–15.

[3] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: a factored solution to the simultaneous localization and mapping problem, in: Proceedings of the 18th National Conference on Artificial Intelligence, Edmonton, Canada, 2002, pp. 593–598.

[4] K. Wurm, C. Stachniss, G. Grisetti, Bridging the gap between feature- and grid-based SLAM, Robot. Auton. Syst. 58 (2010) 140–148.

[5] A. Gil, O. Reinoso, M. Ballesta, M. Juliá, L. Payá, Estimation of visual maps with a robot network equipped with vision sensors, Sensors 10 (2010) 5209–5232.

[6] J. Civera, A.J. Davison, J.M. Martínez Montiel, Inverse depth parametrization for monocular SLAM, IEEE Trans. Robot. 24 (2008) 932–945.

[7] C. Joly, P. Rives, Bearing-only SAM using a minimal inverse depth parametrization, in: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), Vol. 2, Funchal, Madeira, Portugal, 2010, pp. 281–288.

[8] S.-E. Yu, D. Kim, Image-based homing navigation with landmark arrangement matching, Inform. Sci. 181 (2011) 3427–3442.

[9] Y. Rasmussen, Y. Lu, M. Kocamaz, Integrating stereo structure for omnidirectional trail following, in: Proceedings of the International Conference on Intelligent Robots and Systems (IROS), San Francisco, USA, 2011, pp. 4084–4090.

[10] A.J. Davison, D.M. Murray, Simultaneous localisation and map-building using active vision, IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) 24 (2002) 865–880.

[11] F.A. Moreno, J.L. Blanco, J. Gonzalez, Stereo vision specific models for particle filter-based SLAM, Robot. Auton. Syst. 57 (2009) 955–970.

[12] G. Grisetti, C. Stachniss, S. Grzonka, W. Burgard, A tree parameterization for efficiently computing maximum likelihood maps using gradient descent, in: Proceedings of the Robotics: Science and Systems (RSS), Atlanta, USA, 2007, pp. 1–8.

[13] A.J. Davison, Y. Gonzalez Cid, N. Kita, Real-time 3D SLAM with wide-angle vision, in: Proceedings of the 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal, 2004, pp. 117–124.

[14] S. Park, S. Kim, M. Park, S.-K. Park, Vision-based global localization for mobile robots with hybrid maps of objects and spatial layouts, Inform. Sci. 179 (2009) 4174–4198.

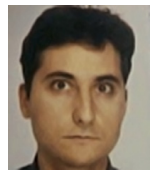
[15] D. Valiente, A. Gil, L. Fernández, O. Reinoso, View-based maps using omnidirectional images, in: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO), Vol. 2, Rome, Italy, 2012, pp. 48–57.

[16] J. Neira, J.D. Tardós, Data association in stochastic mapping using the joint compatibility test, IEEE Trans. Robot. Automat. 17 (2001) 890–897.
[17] C. Berger, Weak constraints network optimiser, in: Proceedings of the International Conference on Robotics and Automation (ICRA), Saint Paul, USA, 2012, pp. 1270–1277.
[18] H. Bay, T. Tuytelaars, L. Van Gool, Speeded up robust features (SURF), Comput. Vis. Image Underst. 110 (2008) 346–359.
[19] A.C. Murillo, J.J. Guerrero, C. Sagüés, SURF features for efficient robot localization with omnidirectional images, in: Proceedings of the International Conference on Robotics and Automation (ICRA), San Diego, USA, 2007, pp. 3901–3907.
[20] R.E. Kalman, R.S. Bucy, New results in linear filtering and prediction theory, J. Basic Eng. 83 (1961) 95–107.
[21] D. Scaramuzza, Performance evaluation of 1-point RANSAC visual odometry, J. Field Robot. 28 (2011) 792–811.
[22] L. Bottou, Stochastic Learning, in: Lecture Notes in Artificial Intelligence (LNAI), vol. 3176, Springer Verlag, Berlin, 2004.
[23] D. Olson, J. Leonard, S. Teller, Fast iterative optimization of pose graphs with poor initial estimates, in: Proceedings of the International Conference on Robotics and Automation (ICRA), Orlando, USA, 2006, pp. 2262–2269.

David Valiente received the M. Eng. degree in Telecommunications Engineering from the Miguel Hernández University (UMH), Elche, Spain, in 2009, receiving also the Best Academic Student award in Telecommunications Engineering by the UMH. From 2009 to 2011, he worked as a researcher at the Communications department and at the Systems Engineering and Automation department of the UMH. Since 2011 he has a research position as scholarship holder in the area of Systems Engineering and Automation of the UMH, receiving a grant (FPI) by the Spanish Government. His research interests are focused on mobile robots, omnidirectional vision, visual feature extraction and visual SLAM.

Arturo Gil received the M. Eng. degree in Industrial Engineering from Miguel Hernández University (UMH), Elche, Spain, in 2002, receiving also the Best Student Academic award in Industrial Engineering by the UMH. He obtained the Ph.D. degree in 2008, entitled: "Cooperative construction of visual maps by means of a robot team". Since 2003, he works as a lecturer and researcher at the UMH, teaching subjects related to Control and Computer Vision. His research interests are focused on mobile robotics, visual SLAM and cooperative robotics. He is currently working on techniques to build visual maps using teams of mobile robots.

Lorenzo Fernandez received the M. Eng. degree in Telecommunications Engineering from the Miguel Hernández University (UMH), Elche, Spain, in 2008. He joined the Systems Engineering and Automation department of the UMH in 2008, where he is involved in research projects. In 2010 he obtained a Ph.D. fellowship (VALi+d) from the Valencian Regional Government to enroll in a Ph.D program at the Systems and Automation department. His research interests are mobile robots, visual appearance and visual SLAM.



Óscar Reinoso received the M. Eng. degree from the Polytechnical University of Madrid (UPM), Madrid, Spain, in 1991. Later, he obtained the Ph.D. degree in 1996. He worked at Protos Desarrollo S.A. company in the development and research of artificial vision systems from 1994 to 1997. Since 1997, he works as a professor at Miguel Hernández University (UMH), teaching subjects related to Control, Robotics and Computer Vision. His research interests are in mobile robotics, climbing robots and visual inspection systems. He is member of CEA–IFAC and IEEE.

Bibliography

[1] LTD. Accowle Company. Accowle Omnidirectional Vision Sensor. 24

[2] Adept MobileRobots LLC. Robot Pioneer P3-AT. 33

[3] H. Andreasson, R. Triebel, and W. Burgard. Improving plane extraction from 3d data by fusing laser data and vision. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2656–2661, Aug 2005. 4

[4] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. 71

[5] S. Baker and S. K. Nayar. Theory of catadioptric image formation. In *6th International Conference on Computer Vision (ICCV)*, Bombay, India, 1998. 20

[6] M. Baum, B. Noack, and U. D. Hanebeck. Kalman filter-based slam with unknown data association using symmetric measurement equations. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI 2015*, pages 49–53, Sept 2015. 69

[7] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, Graz, Austria, 2006. 8, 50

[8] C. Berger. Weak constraints network optimiser. In *International Conference on Robotics and Automation (ICRA)*, pages 1270–1277, Saint Paul, USA, 2012. 77, 79, 124

[9] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999. 77

[10] S.L. Bogner. An introduction to panospheric imaging. In *IEEE International Conference on Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century.*, volume 4, pages 3099–3106 vol.4, 1995. 20

[11] Léon Bottou. Stochastic learning. In *Advanced Lectures on Machine Learning*, Lecture Notes in Artificial Intelligence, LNAI 3176, pages 146–168. Springer Verlag, Berlin, 2004. 77

[12] H. Bulata and M. Devy. Incremental construction of a landmark-based and topological model of indoor environments by a mobile robot. In *IEEE International Conference on Robotics and Automation. ICRA 1996*, volume 2, pages 1054–1060 vol.2, Apr 1996. 73

[13] R. Bunschoten and B. Krose. Visual odometry from an omnidirectional vision system. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 577–583, Taipei, Taiwan, Sept 2003. 45

[14] E.L.L. Cabral, J.C. de Souza, and M.C. Hunold. Omnidirectional stereo vision with a hyperbolic double lobed mirror. In *17th International Conference on Pattern Recognition. ICPR 2004*, volume 1, pages 1–9, 2004. 20

[15] C. Cadena and J. Neira. Slam in O(log n) with the combined kalman-information filter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2009*, pages 2069–2076, Oct 2009. 7, 85

[16] J. A. Castellanos, J. D. Tardos, and G. Schmidt. Building a global map of the environment of a mobile robot: the importance of correlations. In *IEEE International Conference on Robotics and Automation, 1997*, volume 2, pages 1053–1059 vol.2, Apr 1997. 69

[17] J. Choi, S. Ahn, and W. Chung. Robust sonar feature detection for the slam of mobile robot. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2005*, pages 3415–3420, Edmonton,Canada, Aug 2005. 19

[18] M. Choi, R. Sakthivel, and W. K. Chung. Neural network-aided extended kalman filter for slam problem. In *IEEE International Conference on Robotics and Automation, 2007*, pages 1686–1690, April 2007. 80

[19] C. Chou and C. Wang. 2-point ransac for scene image matching under large viewpoint changes. In *IEEE International Conference on Robotics and Automation. ICRA 2015*, pages 3646–3651, Seattle, Washington, U.S.A., May 2015. 96

[20] J. Civera, A. J. Davison, and J. M. Martínez Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 2008. 5, 6, 69, 73, 87

[21] D. Cobzas and Hong Zhang. Cylindrical panoramic image-based model for robot localization. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2001.*, volume 4, pages 1924–1930 vol.4, 2001. 30

[22] D.M. Cole and P.M. Newman. Using laser range data for 3d slam in outdoor environments. In *IEEE International Conference on Robotics and Automation. ICRA 2006*, pages 1556–1563, Orlando, Florida, U.S.A., May 2006. 3, 19

[23] P. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2004*, volume 4, pages 4007–4012 vol.4, Edmonton, Canada, Sept 2004. 49

[24] A. Costa, G. Kantor, and H. Choset. Bearing-only landmark initialization with unknown data association. In *IEEE International Conference on Robotics and Automation. ICRA 2004*, volume 2, pages 1764–1770, April 2004. 73, 87

[25] A. J. Davison, Y. Gonzalez Cid, and N. Kita. Real-time 3d slam with wide-angle vision. In *IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, Lisboa, Portugal, 2004. 6, 69, 87

[26] A. J. Davison and D. W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2002. 6, 7, 69, 87

[27] A. J. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2002. 5, 73, 87

[28] M. Deans and M. Hebertl. *Experimental Robotics VII*, chapter Experimental comparison of techniques for localization and mapping using a bearing-only sensor, pages 395–404. Springer, Berlin, Heidelberg, 2001. 69

[29] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (slam) problem. In *IEEE International Conference on Robotics and Automation. ICRA 2000*, volume 2, pages 1009–1014, 2000. 69

[30] G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, Jun 2001. 73

[31] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics Automation Magazine*, 13(2):99–110, June 2006. 69

[32] H. F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal on Robotics and Automation*, 4(1):23–31, 1988. 69

[33] E. Einhorna, C. Schrötera, and H.M. Grossa. Attention-driven monocular scene reconstruction for obstacle detection, robot navigation and map building. *Robotics and Autonomous Systems*, 59:296–309, 2011. 8

[34] LTD. Eizoh Company. Eizoh Omnidirectional Vision Sensor. 24

[35] A. Eliazar and R. Parr. Learning probabilistic motion models for mobile robots. In *International Conference on Machine Learning (ICML)*, page 32, Alberta, Canada, 2004. ACM Press. 36

[36] J. Engel, T. Schöps, and D.I Cremers. Lsd-slam: Large-scale direct monocular slam. In *ECCV 2014: 13th European Conference on Computer Vision, Zurich, Switzerland, September 6-12, 2014, Part II*, pages 834–849. Springer International Publishing, 2014. 4

[37] L. Fernández, A. Gil, L. Payá, and O. Reinoso. An evaluation of weighting methods for appearance-based monte carlo localization using omnidirectional images. In *IEEE International Conference on Robotics and Automation. ICRA 2010. Workshop on Omnidirectional Robot Vision*, Anchorage, Alaska, U.S.A., 2010. 6, 91

[38] V. Fox, J. Hightower, Lin Liao, D. Schulz, and G. Borriello. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, 2(3):24–33, July 2003. 71

[39] U. Frese, P. Larsson, and T. Duckett. A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics*, 21(2):196–207, April 2005. 6

[40] M. Ghaffari Jadidi, J. Valls Miró, R. Valencia, and J. Andrade-Cetto. Exploration on continuous gaussian process frontier maps. In *International Conference on Robotics and Automation. ICRA 2014*, pages 6077 – 6082, Hong Kong, China, May 31 2014-June 7 2014. 69, 81, 146

[41] M. Ghaffari Jadidi, J. Valls Miró, R. Valencia, J. Andrade-Cetto, and G. Dissanayake. Exploration in information distribution maps. In *RSS'13 Workshop on Robotic Exploration, Monitoring, and Information Content*, pages 1–8, Berlin, Germany, 24-28 June 2013. 81, 146

[42] A. Gil, O. Martínez-Mozos, M. Ballesta, and O. Reinoso. A comparative evaluation of interest point detectors and local descriptors for visual SLAM. *Machine Vision and Applications*, 2010. 8, 32, 50

[43] A. Gil, O. Reinoso, M. Ballesta, M. Juliá, and L. Payá. Estimation of visual maps with a robot network equipped with vision sensors. *Sensors*, 10:5209–5232, 2010. 5, 6, 87

[44] A. Gil, O. Reinoso, O. Martínez-Mozos, C. Stachniss, and W. Burgard. Improving data association in vision-based SLAM. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2006*, Beijing, China, 2006. 8, 73, 87

[45] A. Gil, D. Valiente, L. Fernández, and J. Marin. Building visual maps with a single omnidirectional camera. In *International Conference on Informatics in Control, Automation and Robotics ICINCO 2011*, volume 2, pages 145–154, Noordwijkerhout, The Netherlands, 28-31 July 2012. 49

[46] A. Gil, D. Valiente, O. Reinoso, and J.M. Marín. Creación de un modelo visual del entorno basado en imágenes omnidireccionales. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 9(4):441 – 452, 2012. iii, xv, 10, 169

[47] The Imaging Source Europe GmbH. Color CCD camera DFK 21BF04. 24

[48] The Imaging Source Europe GmbH. Color CCD camera DFK 41BF02. 24

[49] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE International Conference on Radar and Signal Processing*, 140(2):107–113, April 1993. 71

[50] V. Grassi and J. Okamoto. Development of an omnidirectional vision system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 28:58 – 68, 2006. 20

[51] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, Feb 2007. 34, 50, 100, 104

[52] G. Grisetti, C. Stachniss, and W. Burgard. Non-linear constraint network optimization for efficient map learning. *IEEE Transactions on Intelligent Transportation Systems*, 10:428–439, 2009. 77, 124

[53] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems*, Atlanta, Georgia, U.S.A, 2007. 6, 69, 77, 79, 124

[54] S. Guadarrama and A Ruiz-Mayor. Approximate robotic mapping from sonar data by modeling perceptions with antonyms. *Information Sciences*, 180:4164–4188, 2010. 3

[55] E. Guerra, R. Munguia, and A. Grau. Monocular SLAM for autonomous robots with enhanced features initialization. *Sensors*, 14:6317–6337, 2014. 6

[56] J. Guivant, E. Nebot, and S. Baiker. Localization and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583, 2000. 7, 69

[57] J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, Jun 2001. 7, 69

[58] R. Guo, L. Han, and X. Cheng. Omni-directional vision for robot navigation in substation environments. In *IEEE International Conference on Robotics and Biomimetics. ROBIO 2009*, pages 1272–1275, Dec 2009. 73, 87

[59] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, Manchester, UK, 1988. 6

[60] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 21, 42, 91

[61] M Hazewinkel. Maximum-likelihood method. *Encyclopedia of Mathematics*, 2001. 78

[62] J. Hong. Image-based homing. In *IEEE International Conference on Robotics and Automation. ICRA 1991*, Sacramento, U.S.A., 1991. 20

[63] A. Hornung, K. Wurm, M. Bennewitz, C. Stachniss, and W. Bugard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. 4

[64] S. Huang and G. Dissanayake. Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on Robotics*, 23(5):1036–1049, Oct 2007. 8

[65] S. Huang, Z. Wang, and G. Dissanayake. Exact state and covariance sub-matrix recovery for submap based sparse eif slam algorithm. In *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008.*, pages 1868–1873, May 2008. 85

[66] V. Ila, L. Polok, M. Solony, P. Smrz, and P. Zemcik. Fast covariance recovery in incremental nonlinear least square solvers. In *IEEE International Conference on Robotics and Automation. ICRA 2015*, pages 4636–4643, May 2015. 69

[67] G. Jang, S. Kim, J. Kim, and I. Kweon. Metric localization using a single artificial landmark for indoor mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2005*, pages 2857–2862, Aug 2005. 73, 87

[68] X. Ji, H. Zhang, D. Hai, and Z. Zheng. An incremental slam algorithm with backtracking revisable data association for mobile robots. In *IEEE International Conference on Mechatronics and Automation. ICMA 2008*, pages 831–839, Aug 2008. 80

[69] C. Joly and P. Rives. Bearing-only SAM using a minimal inverse depth parametrization. In *International Conference on Informatics in Control, Automation and Robotics. ICINCO 2010*, Funchal, Madeira, Portugal, 2010. 5

[70] M. Kaess, A. Ranganathan, and F. Dellaert. isam: Fast incremental smoothing and mapping with efficient data association. In *IEEE International Conference on Robotics and Automation. ICRA 2007*, pages 1670–1677, April 2007. 69

[71] J. G. Kang, S. Y. An, and S. Y. Oh. Modified neural network aided ekf based slam for improving an accuracy of the feature map. In *International Joint Conference on Neural Networks. IJCNN 2010*, pages 1–7, July 2010. 80

[72] R. Karlsson and F. Gustafsson. Recursive bayesian estimation: bearings-only applications. *IEEE International Conference on Radar, Sonar and Navigation*, 152(5):305–313, October 2005. 72

[73] G. Krishnan and S. Nayar. Cata-fisheye camera for panoramic imaging. In *IEEE Workshop on Applications of Computer Vision. WACV 2008.*, pages 1–8, 2008. 20

[74] S. Kulback and R. A. Leiber. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951. 84, 149

[75] S. Kullback. *Information Theory and Statistics*. Wiley, New York, 1959. 84

[76] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard. Large scale graph-based slam using aerial images as prior information. *Autonomous Robots*, 30:25–39, 2011. 8

[77] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, and A. Ng. On optimization methods for deep learning. In *International Conference on Machine Learning*, pages 265–272. Omnipress, 2011. 77

[78] L. K. Lee, S. Y. An, and S. y. Oh. Efficient visual salient object landmark extraction and recognition. In *IEEE International Conference on Systems, Man, and Cybernetics. SMC 2011*, pages 1351–1357, Oct 2011. 69

[79] S. J. Lee and J.-B. Song. A new sonar salient feature structure for ekf-based slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2010*, pages 5966–5971, Oct 2010. 3

[80] J. Leonard and H. Jacob. A computationally efficient method for large-scale concurrent mapping and localization. *Robotics Research: The Ninth International Symposium*, 2000. 69

[81] C. Leung, S. Huang, and G. Dissanayake. Active slam in structured environments. In *IEEE International Conference on Robotics and Automation. ICRA 2008.*, pages 1898–1903, May 2008. 73

[82] Y. Li, S. Li, Q. Song, and H. Liu. Fast and robust data association using posterior based approximate joint compatibility test. *IEEE Transactions on Industrial Informatics*, 10(1):331–339, Feb 2014. 94

[83] F. Linaker and M. Ishikawa. Rotation invariant features from omnidirectional camera images using a polar higher-order local autocorrelation feature extractor. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2004*, volume 4, pages 4026–4031, Edmonton, Canada, Sept 2004. 19

[84] M. Liu, C. Pradalier, and R. Siegwart. Visual homing from scale with an uncalibrated omnidirectional camera. *IEEE Transactions on Robotics*, 29(6):1353–1365, Dec 2013. 30

[85] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1985. 41

[86] D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision. ICCV 1999*, Kerkyra, Greece, 1999. 8

[87] A. Martinelli and R. Siegwart. Exploiting the information at the loop closure in slam. In *IEEE International Conference on Robotics and Automation. ICRA 2007*, pages 2055–2060, April 2007. 6

[88] L. H. Matthies. *Dynamic Stereo Vision*. PhD thesis, Pittsburgh, PA, USA, 1989. 47

[89] J. Neira and J. D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 2001. 8, 94, 123

[90] D. Nistér. An efficient solution to the five-point relative pose problem. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2003*, Madison, USA, 2003. 49

[91] D. Nistér. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications*, 2005. 47

[92] J. Mochnac, S. Marchevsky, and P. Kocan. Bayesian filtering techniques: Kalman and extended kalman filter basics. In *19th International Conference on Radioelektronika. RADIOELEKTRONIKA 2009.*, pages 119–122, April 2009. 72

[93] M. Montemerlo. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003. 6

[94] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: a factored solution to the simultaneous localization and mapping problem. In *18th National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. 4, 6, 69

[95] Hans Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover*. PhD thesis, September 1980. 47

[96] F. Moura-Pires and A. Steiger-Garcao. A decision tree algorithm with segmentation. In *International Conference on Industrial Electronics, Control and Instrumentation. IECON 1991*, volume 3, pages 2077–2082, Oct 1991. 80

[97] J. Mullane, B. N. Vo, M. D. Adams, and B. T. Vo. A random-finite-set approach to bayesian slam. *IEEE Transactions on Robotics*, 27(2):268–282, April 2011. 71

[98] H. Nagara, Y. Yagi, and M. Yachida. Wide field of view head mounted display for tele-presence with an omnidirectional image sensor. In *Conference on Computer Vision and Pattern Recognition Workshop*, Toronto, Canada, 2003. 19

[99] S. K. Nayar and S. Baker. Catadioptric image formation. In *DARPA Image Understanding Workshop*, New Orleans, U.S.A., 1997. 20

[100] R. M. Neal. Regression and classification using gaussian process priors (with discussion). *Bayesian Statistics*, 6:475–501, 1999. 80

[101] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2004.*, volume 1, pages I–652–I–659 Vol.1, Washington, U.S.A., June 2004. 47

[102] David Nister, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23:2006, 2006. 47

[103] C.F. Olson, L.H. Matthies, M. Schoppers, and M.W. Maimone. Robust stereo ego-motion for long distance navigation. In *IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000.*, volume 2, pages 453–458, U.S.A., 2000. 47

[104] D. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *International Conference on Robotics and Automation. ICRA 2006*, pages 2262–2269, Orlando, Florida, U.S.A., 2006. 77, 79, 124

[105] S. Park, S. Kim, M Park, and Park S.-K. Vision-based global localization for mobile robots with hybrid maps of objects and spatial layouts. *Information Sciences*, 179:4174–4198, 2009. 6, 69, 87

[106] L. Paya, F. Amoros, L. Fernandez, and O. Reinoso. Performance of global-appearance descriptors in map building and localization using omnidirectional vision. *Sensors*, 14:3033–3064, 2014. 5

[107] L. Qing, Z. Nanning, M. Lin, and C. Hong. True single view point multi-resolution catadioptric system for intelligent vehicle. In *International IEEE Conference on Intelligent Transportation Systems, 2004.*, pages 155–160, 2004. 20

[108] H. Qingbo. Time-frequency manifold histogram matching for transient signal detection. In *IEEE International Conference on Instrumentation and Measurement Technology Conference. I2MTC 2015*, pages 584–587, Pisa, Italy, May 2015. 96

[109] F. T. Ramos, J. Nieto, and H. F. Durrant-Whyte. Recognising and modelling landmarks to close loops in outdoor slam. In *IEEE International Conference on Robotics and Automation. ICRA 2007*, pages 2036–2041, April 2007. 69

[110] A. Ranganathan and F. Dellaert. Bayesian surprise and landmark detection. In *IEEE International Conference on Robotics and Automation. ICRA 2009*, pages 2017–2023, May 2009. 71

[111] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning series. Massachusetts Institute of Technology, 2006. 80, 146

[112] Y. Rasmussen, Y. Lu, and M. Kocamaz. Integrating stereo structure for omnidirectional trail following. In *International Conference on Intelligent Robots and Systems. IROS 2011*, pages 4084–4090, San Francisco, USA, 25-30 September 2011. 5

[113] D. W. Rees. Panoramic television viewing system. *United States Patent*, (3, 5, 465), 1970. 20

[114] Mobile Robots. ARIA: Advanced Robot Interface for Applications. 34

[115] SICKMobile Robots. LMS200. 33

[116] D. Scaramuzza. Performance evaluation of 1-point RANSAC visual odometry. *Journal of Field Robotics*, 28:792–811, 2011. 49, 96

[117] D. Scaramuzza, F. Fraundorfer, and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *IEEE International Conference on Robotics and Automation. ICRA 2009*, Kobe, Japan, 2009. 47, 96

[118] D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2006*, Beijing, China, 2006. 25, 147

[119] M. Schreiber, A. M. Hellmund, and C. Stiller. Multi-drive feature association for automated map generation using low-cost sensor data. In *IEEE International Conference on Intelligent Vehicles Symposium*, pages 1140–1147, June 2015. 69

[120] J. Servos, M. Smart, and S.L. Waslander. Underwater stereo slam with refraction correction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2013*, pages 3350–3355, Tokyo, Japan, Nov 2013. 41

[121] C. E. Shannon. A mathematical theory of communication. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(1):3–55, January 2001. 84

[122] X. Shi, C. Zhao, and Tao Chen. Data association technology based on multi algorithm matching for slam. In *World Congress on Intelligent Control and Automation. WCICA 2014*, pages 934–939, June 2014. 69

[123] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *IEEE International Conference on Robotics and Automation. ICRA 1987*, volume 4, pages 850–850, Mar 1987. 69, 73

[124] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 1986. 69, 73

[125] Open source project. Robot Operating System, ROS. 33

[126] Open source project. ROSARIA: Robot Operating System-Advanced Robot ìnterface for Applications. 33

[127] C. Stachniss, G. Grisetti, D. Haehnel, and W. Burgard. Improved Rao-Blackwellized mapping by adaptive sampling and active loop-closure. In *Workshop on Self-Organization of Adaptive behavior. SOAVE 2004*, Ilmenau, Germany, 2004. 4, 34, 50, 100, 104

[128] Y.-T. Sun, C.-H. Wang and C.-C. Chang. Switching t-s fuzzy model-based guaranteed cost control for two-wheeled mobile robots. *International Journal of Innovative Computing, Information and Control*, 8:3015–3028, 2012. 8, 124

[129] J. Tardif, Y. Pavlidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2008.*, pages 2531–2538, Nice, France, Sept 2008. 47

[130] S. Thrun. Bayesian landmark learning for mobile robot localization. *Machine Learning*, 1998. 71

[131] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping. In *IEEE International Conference on Robotics and Automation. ICRA 2000*, volume 1, pages 321–328 vol.1, 2000. 3

[132] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005. 36

[133] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 2000. 6

[134] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A. Ng. *Algorithmic Foundations of Robotics V*. Springer, Berlin, Germany, 2004. 85

[135] D. Valiente, A. Gil, L. Fernandez, and O. Reinoso. A comparison of EKF and SGD applied to a view-based slam approach with omnidirectional images. *Robotics and Autonomous Systems*, 62:108–119, 2014. iii, xv, 11, 90, 169

[136] D. Valiente, A. Gil, L. Fernandez, and O. Reinoso. A modified stochastic gradient descent algorithm for view-based slam using omnidirectional images. *Information Sciences*, 279:326–337, 2014. iii, xv, 10, 77, 169

[137] D. Valiente, A. Gil, L. Fernandez, and O. Reinoso. View-based maps using omnidirectional images. In *International Conference on Informatics in Control, Automation and Robotics ICINCO 2012*, volume 2, pages 48–57, Rome, Italy, 28-31 July 2011. 8

[138] David Valiente, Maani Ghaffari Jadidi, Jaime Valls Miró, Arturo Gil, and Oscar Reinoso. Information-based view initialization in visual slam with a single omnidirectional camera. *Robotics and Autonomous Systems*, 72:93 – 104, 2015. 12

[139] J. Valls-Miró, G. Dissanayake, and W. Zhou. Towards vision based navigation in large indoor environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2006*, Beijing, China, 2006. 6

[140] R. Vazquez-Martin, P. Nunez, J. C. del Toro, A. Bandera, and F. Sandoval. Adaptive observation covariance for ekf-slam in indoor environments using laser data. In *IEEE Mediterranean Electrotechnical Conference. MELECON 2006.*, pages 445–448, May 2006. 73

[141] M. Walter, R. Eustice, and J. Leonard. Exactly sparse extended information filters for feature-based slam. *International Journal of Robotics Research*, 26(4):335–359, April 2007. 85

[142] Z. Wang and G. Dissanayake. Efficient monocular slam using sparse information filters. In *5th International Conference on Information and Automation for Sustainability. ICIAFs 2010*, pages 311–316, Dec 2010. 85

[143] J. Weingarten and R. Siegwart. Ekf-based 3d slam for structured environment reconstruction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2005*, pages 3834–3839, Aug 2005. 73

[144] C. Weinrich, C. Vollmer, and H. M. Gross. Estimation of human upper body orientation for mobile robotics using an svm decision tree on monocular images. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2012*, pages 2147–2152, Oct 2012. 80

[145] O. Wijk and H.I. Christensen. Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data. *Robotics and Autonomous Systems*, 31(12):31 – 42, 2000. 4

[146] C. K. I. Williams. *Gaussian Processes*. The Handbook of Brain Theory and Neural Networks. Massachusetts Institute of Technology, 2002. 80

[147] S. B. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *IEEE International Conference on Robotics and Automation. ICRA 2000*, volume 2, pages 1793–1798, 2000. 69

[148] L. Wu and D. W. C. Ho. Fuzzy filter design for Itô stochastic systems with application to sensor fault detection. *IEEE Transactions on Fuzzy Systems*, 17:233–242, 2009. 8, 124

[149] Y. Yagi and S. Kawato. Panoramic scene analysis with conic projection. In *IEEE/RSJ International Conference on Robots and Systems. IROS 1990*, Ibaraki, Japan, 1990. 20

[150] K. Yamazawa, Y. Yagi, and M. Yachida. Obstacle detection with omnidirectional image sensor hyperomni vision. In *IEEE International Conference on Robotics and Automation. ICRA 1995.*, volume 1, pages 1062–1067, 1995. 20

[151] R. Yang, H. Gao, and P. Shi. Delay-dependent robust H_∞ control for uncertain stochastic time-delay systems. *International Journal of Robust and Nonlinear Control*, 20:1852–1865, 2010. 8, 124

[152] S.-E. Yu and D Kim. Image-based homing navigation with landmark arrangement matching. *Information Sciences*, 181:3427–3442, 2011. 5