UNIVERSITAT POLITÈCNICA DE CATALUNYA

Programa de Doctorat:

AUTOMÀTICA, ROBÒTICA I VISIÓ

Tesi Doctoral

Monocular SLAM: Data association and sensing through a human-assisted uncalibrated visual system

Edmundo Guerra Paradas

Director: Antoni Grau
Codirector: Rodrigo Munguía

Maig de 2017

A los que fueron y nos legaron un mundo;

a los que son y nos ayudan a construir;

a los que serán y por ellos hacemos.

Abstract

The Simultaneous Localization and Mapping (SLAM) task is widely acknowledged as one of the fundamental problems to solve in perception and robotics to produce actual mobile robotic agents. The problem itself is that of how a mobile robot agent can operate in an a priori unknown environment, using the sensory systems available (usually on-board) to perceive its surroundings, build a map with this knowledge, and localize itself in the map tracking its own position.

This relevance, combined with the diversity of approaches available to solve it, and the depth of the challenges it presents, makes the SLAM problem one of the more active areas of research in robotics. One of the most complex challenges in any approach is the data association, as it generally conveys hard a trade-off between robustness and computational time required, and can impact the whole architecture of a SLAM method.

In terms of sensors used, the field was originally dominated by range finder sensors, but visual SLAM research has grown in popularity in the last decade. Camera sensors have been expanding its capabilities and specifications thanks to the consumer demand for them. As a sensor, they provide lightning measurements of the projected points at known bearings, which through computer vision can be converted into bearing measurements for visual features, which can be themselves of several levels of complexity.

The same consumer demand has also pushed technical developments in MEMS and robotic devices with a direct impact in the field of cooperative robotics and the emergence of wearable device technology, where human can wear or carry devices with several sensors in an unobtrusive way. These technologies have opened many opportunities for research in robotics, including the field of collaborative SLAM and the area of human-robot interaction (HRI).

This thesis is focused in the study and development of a visual SLAM methodology based on the delayed inverse-depth feature initialization (DI-D) monocular SLAM which can benefit from the advantages of working in a HRI collaborative framework. In order to achieve this, the research has been developed two different areas. Firstly, the known and tested DI-D monocular SLAM is studied: its procedures and algorithms detailed and analyzed; with emphasis in the data association problem (DA). The DA process is reviewed, and a new validation algorithm is introduced to strengthen and give robustness to the data association technique used.

Once the DI-D has been studied and updated, the HRI collaborative framework is introduced, with an initially focus into solving one of its inconveniences: the requirement of a scaled metric initialization with a priori knowledge. The HRI is introduced by deploying into a human being a custom built wearable device which includes a camera and some other sensors. The data from this secondary monocular sensor, whose pose is approximately known with respect to the camera used to solve the SLAM problem, allows speeding up the feature initialization process of the DI-D, and even ignoring the requirement of scale initialization.

As the introduction of the HRI framework was successful, its advantages were further expanded to the rest of the SLAM process, including the measurement and update steps. This integration was performed based in a virtual sensor methodology, where the collaborative measurement process was treated as a single sensor with its own specifications and covariances, allowing seamless fusion into the EKF-SLAM. To evaluate the specific impact of the HRI with respect to the behaviour of the secondary camera, several new metrics have been proposed and studied.

All the methods have been proved and validated through experimentation with real data. When it was found relevant, the experiments were evaluated in real-time scenarios, and several simulations have been included when needed to prove some theoretical hypothesis.

Contents

List of Figures

List of Tables

List of Algorithms

Part I
Introduction

The automata "*Clockwork Prayer"*, built around the decade of 1560 by Juanelo Turriano, Court Clock Master to Holy Roman Emperor Charles V. Currently keep at the Smithsonian Institution, is was able to move autonomously, and emulate facial gestures, prayer movements, and other devotion acts, such as kissing the cross. Will truly human-like artificial intelligence ever present traits such as faith and devotion?

The human quest to produce autonomous robots can be explained and understood in several ways. From a pragmatic point of view it can be interpreted as a testimony of its conflicting nature: the lust for power and conquest, the will to achieve greatness even at the expense of others' efforts; and the nurturing and empathic instincts which produce the ethical commitment to avoid unnecessary suffering unto the fellow man. In this same train of thought, note how the same term robot finds its origins in the medieval serfdom, where Slavic languages knew as *robota* the works done by serfs –technically free people- to serve their lieges. Thus, as groups, humans know that there are some unpleasant tasks required to achieve a functional society, but as individuals we tend to reject several of those tasks for egoist reasons. In fact, this rejection can reach the point to find it undesirable to be inflicted even upon others.

Another explanation will collude how automatization of tasks could be easily attributed to instincts embedded into to human behaviour by evolution. Conserving energy, finding the best way to achieve the desired results, and dealing with task which shall produce benefits only in the long term are behaviours found even in common animal species, though in many cases they are purely of instinctual nature (isn't it funny when evolution works as an optimality solver?).

But robots are not only created and imagined to perform the *robota*, but also to bring human achievements to new heights: we are still decades far from sending humans into another planet, perhaps centuries away from traveling beyond the solar system; but there are already robotic devices exploring the surface of Mars and going deep into space, sending back home images of stars and planets far beyond the distance human mind can actually comprehend and figure. If we look toward back into the Earth, robotic manipulators process and move products in factories at speeds several orders of magnitude beyond human capabilities, while small exploratory robots survey crisis sites where human life expectancy is only a few hours, to put a few example. Humans, as individuals, cannot conquer all the challenges in the world, but we are good at producing the means to achieve them. This generally means that these robots are limited, even the so called smart ones, and will be useful in a reduced set of problems, being useless when generality is required.

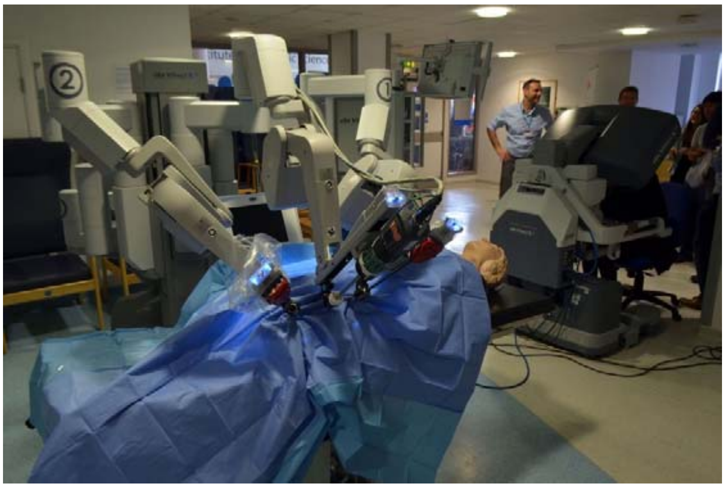
Sometimes need and will marry to birth designs and ideas: the need of mercenaries and soldiers, and the will to have the best possible ones must have been driving forces behind the multiple designs through history of "automata soldiers", like Leonardo's mechanical knights, and several other attempts. Although given the state of the art at the time, most of these automata, like those machines with simple special effects commonly found in temples and sites of adoration, would only serve to produce illusions and fool uneducated lower classes, resulting lacking in the fields of warfare.

Anyway, we are still far from fully replacing humans, especially as there are no clear definitions for many of the features and characteristics that differentiate humans and other beings: science is still unable to provide a comprehensive and coherent definition for intelligence across different fields of the same disciplines; and there has been never an agreement upon the existence of free will. Without clear objectives, any pretension of building artificial humans belongs into science fiction instead of science. Nevertheless, robotics research still has to address several challenges, even to build not so intelligent robots.

One of these key challenges is to give robots the spatial sense of self, i.e., make then capable of telling where they are with respect to an environment where they have been operating for a time. This in turn means that the robots have to be able to localize itself in an environment, and produce a representation of said environment. These two different challenges, giving a robot the capability to produce a map of an unknown environment and making it able to perceive the same environment and localized itself with respect to a reference (generally a point in the map), are the tasks that compose the SLAM problem. Although the description is pretty simple and clear enough, some thought on it reveals many issues and challenges: to begin with, the description provided fits exactly a chicken and egg dynamic, that is, a map is needed to perform localization, and the location must be known to build a map. In its apparent simplicity, the SLAM problem encapsulates several challenges to be solved, many of them depending on specifics related to the sensors used, the environment explored, the type of robot, the movements it perform, and many other factors. As such, the SLAM problem is probably one of the most complex and diverse challenges in robotics research, while at the same time being of capital importance not only for autonomous robotics, but also for any other field were exploration and mapping constitute a relevant task.

Chapter 1

Motivation and Objectives



The da Vinci Surgical System is a medical robot, commonly referred as Leonardo, why allows performing complex surgery in a minimally invasive way. According to its developers, Intuitive Surgical, over 200.000 surgeries are perfomed yearly. Though it is essentially a complex surgery tool, the social impact of this robotic device is beyond any doubt.

1.1 Motivation

As it has been already discussed, SLAM is one of the most important problems to address in robotics, especially in autonomous robotics, as it is the only way to enable robotic agents to operate in a priori unknown environments. This relevance, combined with the diversity of approaches available to solve it, both in physical (i.e. hardware) and mathematical terms, and the depth of multiple of the challenges it presents, makes the SLAM problem one of the more active and valued areas of research in robotics. It is worth noting that the most successful SLAM solutions as of today tend to rely in range finder sensors, but as it will be discussed during this dissertation, they present multiple weaknesses that make them less versatile than other approaches.

In this sense of versatility and, to a minor degree, robustness, I like to think that as the general goal of autonomous robotics points towards seamless human replacement and/or collaboration, and human capabilities should be the design guidelines and benchmarks to beat. So, although the human spatial awareness is closely related to the sense of self, and there are several senses beyond vision relevant to the task, like the vestibular sense and the kinesthetic perception, human deal with mapping and localization essentially through vision. Even if we open up to general biological existence, we can observe how evolution has clearly judged the vision based approaches the way to go for the SLAM problem, while the range-finder solutions are found only in specific biological niche scenarios. Even the famed echolocation of bats is pushed aside and in open spaces or properly illuminated scenarios most species of bats use vision[1] for navigation. While applying design decision found in life forms to decide on robotics research disjunctives may look unsubstantiated, we have to consider which criteria and circumstances brought forth this judgements. Both for hardware sensors and living being, vision sense is a much more versatile perception system, which can solve not only the SLAM problem but be useful in many other tasks, as it provides not only geometric but also photometric information. Vision is, also for both cases, generally a passive sensor: from a technical point of view, a passive sensor is generally cheaper, as it is built of a receptor only, and simpler; and from a biological and evolutionary point of view, a passive sense is simpler and avoid emitting signals, which could be subjected to disruption attacks and reveal the position in a hostile environment.

Beyond the biological inspired criteria to decide for vision as the sensing technology to solve the SLAM problem, there is still the matter of the goal of autonomous robotics: as we strive to build robot replacements or companions to humans, these entities will necessarily operate in environments designed or transformed to fit human needs and capabilities. This implies that as we operate using sight as our main sensor, robots presenting good vision based perception capabilities will be able to extract the most data from these environments, being knowledge derived from this data the key enabler of versatility action. Note that as we are still far from achieving truly intelligent robotic devices with the adaptability and generality of human capabilities, cooperation between robots and humans will probably be the most common scenario in the next decades. Under these assumptions, the humans not only condition the SLAM problem so that the preferred solutions will be based on vision, but also the emergence of collaborative frameworks where human and robot work towards the same objectives.

[1] Bats' blindness is largely a myth, and although smaller species present poor vision compared to human sight, the bigger species' sight is considered notably better than human vision.

Solutions to the SLAM problem under collaborative frameworks with humans are of critical relevance to human society, as their current main area of application is that or search and recovery operation under emergency situations. As such, their performance could be determinant to save lives or simply improve the management of an emergency situation. Then, it is clear that solutions to the SLAM problem should contemplate and explore all the options available. Under this pretense, it is worth noting that most of the collaborative approaches to SLAM focus the cooperative efforts in the data fusion process at map level, be it based on recursive filtering or optimization algorithms. So, the most frequent architectures ignore the opportunity to exploit the advantages of the collaborative framework during the sensing and measurement process unchecked. Thus, although the cooperative SLAM problem has been solved in several works in the classical sense, there is still much work to do in this field, given the societal impact that any breakthrough could imply.

1.2 Objectives

The primary objectives of this thesis are the study and analysis of the delayed feature initialization inverse depth monocular SLAM technique, and refine it to explore its viability to produce a collaborative SLAM framework where human perception forms part of the solution. In an initial phase the research is focused in studying and comparing how other techniques deal with several challenges within the SLAM problem, and bringing the knowledge into the DI-D monocular SLAM.

This initial study in the DI-D framework should focus in those of the core challenges in SLAM which can be addressed and improved in a in a monocular framework; i.e.; firstly the data association problem, including data validation, and the feature initialization. This study and the solutions provided should be aware of higher level problem in SLAM which are dependent on them, like large map management, loop closing, and place recognition. Any solution provided to lower level problem should also be proved to provide improvements into (or at least not disrupt) solutions of the higher level problems, as even if they are not explicitly solved into this thesis, they are an integral part of any SLAM solution aimed at solving a real problem.

Once the DI-D framework has been thoroughly studied, and probably some developments have been introduced, the focus will be shifted into the study of the opportunities that collaborative exploration scenarios provide. Ideally the collaboration should focus into the perception part of the framework, enhancing the data extracted from the monocular sensors. Though other sensors will probably be present, we consider that it would be most interesting trying to minimize their impact into the general estimation procedure, and try to exploit them into the perceptive procedures. This will mean working over *virtual sensor* based architectures.

Note that the collaborative framework will probably provide additional opportunities to address problems that were not entirely solved under the strictly monocular DI-D SLAM. This opportunities should be exploited when possible, studying the different trade-offs of working them under the pure monocular and the collaborative sensing solutions.

Sub-objectives and other considerations:

- Study the general data association problem, and analyse the impact of the different data association solutions available within the monocular EKF framework. A specific solution to the data association validation problem is required and will be introduced or developed.
- Any solution or development introduced into the delayed inverse-depth monocular SLAM should keep the scale estimation or improve it. The ability to produce scaled maps is one of the greatest advantages of the DI-D when compared to other approaches, and as such, it should be conserved.
- As SLAM based in bearing-only monocular cameras is a partially observable problem, the initialization of features constitutes one of the most complex problems, as they cannot be fully observed in a frame. Under the DI-D framework, the requirement of an scaled metric initialization removes chances to introduce many approaches based on map splitting, sub mapping, and in general, any kind of technique which allows for reinitialization of any part of the state. This should be addressed explicitly, as it would open the possibility to use many well-known solutions to the higher level SLAM challenges.
- The impact of the changes introduced during the eventual development of the collaborative sensing framework should be studied and validate, especially with respect to the human component of the system.
- Loop-closing and large map management problems should be accounted for, as any full SLAM solution should aim to include them. So even if they are not explicitly addressed, they cannot be ignored, and any development should contribute towards the future inclusion into the DI-D framework.

1.3 Thesis Dissertation Outline

The present thesis has been structured in five different blocks, which group the different chapter which compose this work:

Part I presents the necessary context and information to fully assimilate and study the presented research. After framing the research and mission of this thesis, the initial research objectives are presented in Chapter 1.

In Chapter 2, several aspects of the SLAM problem are studied and described. A survey on the main hardware technologies sets the context to discuss the advantages and weaknesses of monocular cameras as sensors for SLAM. Then, the SLAM problem itself is discussed form a mathematical point of view, discussing its roots and the convenience of the development of SLAM as a single problem to address localization and mapping concurrently. After the general formulation is presented, several of the most challenging issues and problems to address in SLAM are listed. This allows introducing the state of the art and solutions to several of these challenges, while introducing multiple notions and concepts that will be relevant further down the dissertation.

Chapter 3, after the general discussion into the SLAM problem and the most relevant challenges, focuses the dissertation into the visual SLAM field. As such, it starts with a brief review of some key concepts of projective geometry and point based image processing. In terms of projective geometry, formulation for the ‘camera’ models is provided and detailed, including the pinhole camera model and discussion on the distortion models; followed by a brief summary of key concepts relevant in epipolar geometry, which will be useful in same Chapter 3 and in part III of the dissertation. To conclude Chapter 3, one of the main research precedents in visual SLAM developed at the Vision and Intelligent Systems research group (VIS) is described and analysed, i.e., the delayed feature initialization inverse-depth monocular EKF SLAM (DI-D monoSLAM). The formulation, though extensive, is far from complete, and focuses on the key aspects required to understand the general procedure of the DI-D monocular SLAM and the works presented in the dissertation. Emphasis is put in the augmented state initialization process, which is responsible for setting an initial metric scale through a priori known landmarks.

Part II of the dissertation focuses in the data association problem, presenting the block with a more detailed review of the main challenges within the association problem, and discussing several taxonomies to classify the solutions applied. Within this block, Chapter 4 presents the research performed in the data association problem in the DI-D SLAM. After a study of the state of the art and main procedures to deal with data association in EKF visual SLAM, the active search technique, used in the DI-D SLAM is described and studied. The study includes an evaluation of several patch-correlation operators available, in order to determine which offers better performance and robustness to the most usual aberrations and issues found during the data association step. Notice that this will be especially relevant as the utilization of CMOS-based rolling shutter cameras intensify some of the visual artifacts and introduce other, as seen in Chapter 2. After review of the active search methodology, the issue of validating the results of the data association process is addressed. As the results of the standard batch validation procedure proved unsatisfactory, leading to frequent combinatorial explosions in terms of computational time cost, we developed a new algorithm to perform batch validation of the data association based upon the joint compatibility notion. The joint compatibility notion is largely based on measurement of the squared Mahalanobis distances, which normally requires inversion operations over the covariance matrix. As such, our proposed algorithm, the HOHCT, tried to minimize the number of these measurement required by exploiting heuristic assumptions made upon the characteristics of the DI-D SLAM and its feature initialization process. This allowed our algorithm to reduce the exponential tendency with respect to the number of landmarks in the map to almost linear most of the time.

Part III introduces the second block of research of this thesis. As this block deals with the research done with the objective of developing a collaborative SLAM framework where human becomes part of the perception system, Part III starts with a brief review of the main lines of research in human-robot interection (HRI) SLAM, and a short survey on the field of collaborative SLAM. Once the state of the field has been presented, the works done in the framework is presented in Chapter 5 and 6.

Chapter 5 presents the initial works performed to develop the collaborative sensing framework, the challenges that appeared, and the main criteria used to decide on the strategies to solve them. As our study during the first research block of this thesis (Part II) could not provide a way to skip the metric scale initialization step without sacrificing the known-scale characteristic of the DI-D monoSLAM, this problem is explicitly addressed, and a new solution is proposed and tested. The proposed approach shares information from a secondary camera sensor carried by a human as part of a wearable device. The data from this secondary camera is exploited to produce instant depth estimations for the landmarks to be initialized. The different possible approaches to solve the correspondence problem are discussed, focussing on the apparent strengths and weaknesses of an stereo vision system. As the proposed approach introduces an additional camera sensor, a procedure to predict the expected utility of any given frame from the secondary camera, based on geometrical modelling of the expected fields of view, is used to avoid unnecessary computational efforts. This same procedure also helps to optimize the correspondence problem by defining subregions on the images where to solve the problem, ignoring the rest of the image. The mathematical formulations of the new inverse observation model and its Jacobian are presented, under the assumption that the landmark is initialized as unified inverse depth point coming from a *‘virtual sensor’* which can provide depth estimation w.r.t. to the camera position. The ability to measure the candidate landmarks prior to the initialization according to different methods means that the initialization process itself becomes more complex, and a new multiple criteria algorithm, based on the delayed feature initialization of the DI-D monocular SLAM is also presented. To validate the presented approach, a set of experiments were performed, evaluating the results according to the impact in the position estimation error, and the performance and behaviour of the feature initialization process. This required recording synchronized data sequences using a robotic platform and a wearable device which are described and validated through simulated experiments.

Chapter 6 further expands the collaborative sensing framework. Based upon work described in the previous Chapter, the *‘virtual sensor’* methodology used initially only for depth estimation during feature initialization is fully integrated during the residuals computation of the EKF filter. This means updating the formulation of the observation models, deriving the new Jacobians required, and modifying the update step algorithms so that they can work with landmark observations measured in different spaces, i.e., landmarks can be measured as pixels or as 3D world points. The approach is evaluated from a theoretical point of view in terms of the gains with respect to the augmented state observability, which is greatly improved, with the state becoming fully observable under some circumstances. A set of indoor and outdoor sequences captured with the hardware described in the previous chapter is then used to test the approach against the non-collaborative DI-D approach. The results are studied in terms of accuracy and the impact of the incidence of overlapping field of view during the trajectory, with special emphasis in singular trajectories, which normally are avoided during SLAM experimentation. In order to further study the effects of the collaborative sensing and its availability frequency and distribution several new metrics have been formulated and evaluated.

The dissertation is closed by Part IV, which contains Chapter 7 and the bibliography. Chapter 7 provides a list of all the publications related to the research presented detailing the contributions, and presents the final conclusions of this thesis, with discussion of future works and the author's expectations for the future of the research in the SLAM field.

Part V contains some annexes with listings and materials useful to read several sections of the dissertation.

Chapter 2

Background and mathematical foundations

Willard S. Boyle and George E. Smith, 2009 Nobel Laureates for the invention of the CCD sensor. Picture taken in Bell Laboratories, 1974. It has been claimed their original intention was to use CCD as a memory circuit and application to imaging was proposed by Eugene Gordon and Michael Tompsett.

2.1 Introduction

The SLAM problem has been a subject of study in the robotics field for some decades now. It states how a mobile robot can operate in an unknown environment by means of only onboard sensors to build a map of its surroundings and use it to localize itself inside the environment.

The works defining the roots of the field can be traced to (Smith and Cheeseman, 1986) and (Durrant-Whyte, 1988), which established how to describe the relationships between landmarks while accounting for the geometric uncertainty through statistical methods. These eventually led to the breakthrough presented in Smith's work (Smith et al., 1987). In that work the problem was presented for the first time as a combined problem with a joint state composed of the robot pose and the landmarks estimations. These landmarks were considered correlated due the common estimation error on the robot pose. This work would lead to several works and studies, being (Durrant-Whyte et al., 1996) the first work to popularize the structure and acronym of SLAM as known today.

In the years following SLAM gained weight in the robotics field, as there came the realization that it was one of the most important problems to solve in order to build truly autonomous mobile robots. Plenty of techniques and algorithms have been developed to address the different problems in a given SLAM approach (Durrant-Whyte and Bailey 2006)(Bailey and Durrant-Whyte, 2006), but most of them rely on estimating features of the environment through one or more sensors, and use them to produce the map. The utilization of different sensors generally defines which kind of filtering or estimation techniques can be used, which different problems may arise at each SLAM step, and how they can be addressed. Thus, in an applied SLAM problem, the sensors to be used are of capital interest.

2.2 Sensors used in mapping and localization

In robotic systems all relations between the system and the physical environment are performed through transducers, which are the devices responsible of converting one kind of energy into another. In robotics there are basically two broad types of transducers: sensors and actuators. Actuators use energy from the robotic system to produce physical effects, like forces and displacements, sounds, and lightning. Sensors are the transducers responsible for sensing and measuring by way of the energy conversion they perform: turning the energy received into electrical signals, which can be coded into useful information. Note that the classification into actuators or sensors accounts only the functionality of the transducer, as many types of sensors also produce emissions as part of the sensing process. These sensors generally operate measuring the perceived reflection of the emitted energy, and thus are considered active, while those which operate based only on measuring environmental energies and effects (that they have not produced) are classified as passive sensors.

The sensors used in SLAM, just like in any other fields of robotics, can be classified according to several criteria. From a theoretical point of view, one of the most meaningful classifications is that if the sensor is of proprioceptive or exteroceptive nature.

Proprioceptive (that is,'*sense of self*') sensors are those generally responsible for measuring values internal to the robot system, like the position of a joint, the remaining battery charge, or a given internal temperature. On the other side, exteroceptive sensors measure different characteristics and aspects of the environment, normally with respect to the sensor itself.

2.2.1 Non-vision based sensing for the SLAM problem

There are plenty of sensors used in the SLAM problem that would fall out of the category of *'vision based'*. In fact the most successful SLAM approaches, with applications in real life scenarios, generally rely on a combination of sensors, generally including both proprioceptive and exteroceptive sensors, with pairings of range finders and encoders being very popular. This section will discuss briefly some of the sensors commonly used in SLAM, excluding those based in artificial vision, summing up the main features in TABLE 2.1 for reference.

The encoders are proprioceptive sensors, responsible for measuring the position or movement of a given joint. Though there are linear encoders, only the rotary encoders are used with frequency in the SLAM problem (Armesto and Tornero, 2004). These encoders can measure directly the position of the rotary axis, in terms of position if they are 'absolute encoders' or in terms of movement for the 'incremental encoders'. The great accuracy when measuring rotation allows computing the exact distance traveled by a wheel, if the radius is known. Still they present several problems related to the nature of how they measure: the derived odometers assume that all the translation of a given wheel is transformed into rotation at a constant and exact rate, which is false in many circumstances. This makes them vulnerable to irregular and sliding surfaces. As a proprioceptive sensor, with no exterior feedback, the error of a pure odometry based SLAM approach will grow unbounded, suffering the drift due to dead reckoning.

Rangefinders are exteroceptive sensors which measure distances between them and a point in the environment. They use a variety of active methods to measure distance, sending out sound, light, or radio waves, and then listening to the returning waves. Generally these are known as sonar, laser, or radar systems. The devices destined to robotics applications generally perform scans (FIGURE 2.1), where a set of measurements is performed concurrently or over such a short time that they are considered all simultaneous. This approach produces data that generally take the form of point planes or point clouds.

Sonar systems use sound propagation through the medium to determine distances (Diosi et al., 2005). Active sonar creates a pulse of sound (a ping) and listens for its reflections (echoes). The time in-between the transmission of the pulse and its reception is measured and converted to distance by knowing the speed of sound, thus acting as a time-of-flight measurement. Laser scan rangefinders (Ila et al., 2010) (also known as LIDAR) can work on different principles, using time-of-flight measurements, interferometers or the phase shift method. As the laser rays are generally more focused compared to other types of waves, they tend to provide higher accuracy measurements, but they can also be disrupted more easily, as discussed at the survey (Pomerleau et al., 2012). Radars (Checchin et al., 2010) also employ electromagnetic waves, using time-of-flight measures, frequency modulation, and phased array method to produce measurements. They generally produce a repeated pulse at a given frequency (RPF), which sets its range.

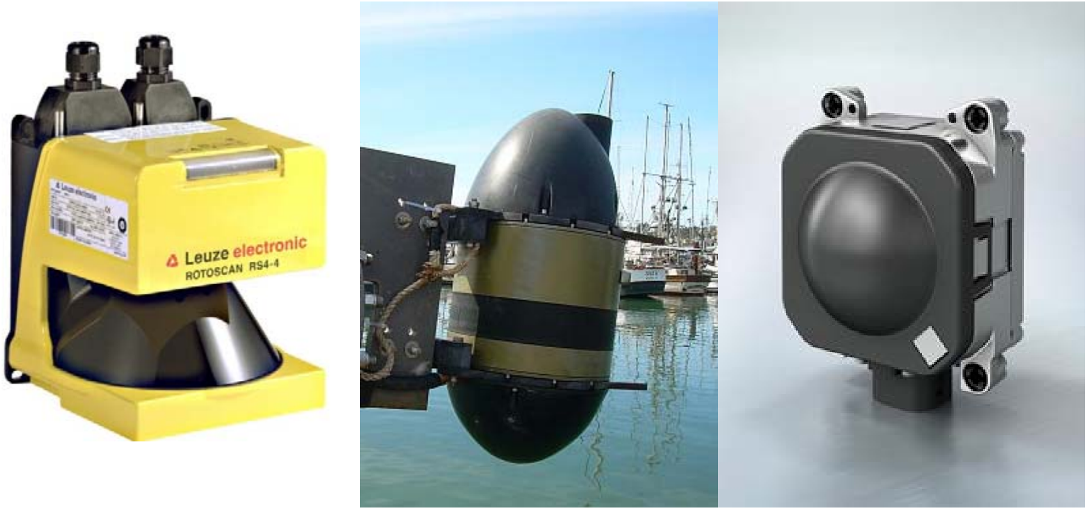


FIGURE 2.1: *Three scan range finder.* **Left:** *LRF for security in industrial robots (courtesy of Leuze).* **Centre:** *Submarine robotics sonar (courtesy of University of Oregon).* **Right:** *Car safety radar (courtesy of Bosch).*

These sensors can have great accuracy given enough time (the trade-off between data density and frequency is generally punishing), and as they capture the environment they do not suffer from dead reckoning effects. On the other side, the data they provide (point planes and point clouds) are just a set of distance at given angles, so these data need to be interpreted and associated, requiring cloud matching methodology (like Iterative Closest Point (ICP)(Besl and McKay, 1992) and other derived techniques), which is computationally expensive. Besides they have all their specific weaknesses: sonar has limited usefulness outside the water given how sound works on the air; LIDAR are vulnerable to ambient pollutants (dust, vapors) that may distort the lightning processes of the measurement; radar has very good range but tends to be lacking in accuracy compared to the other rangefinders.

The GPS (Global Positioning System) (Kotani et al., 1998) is an exteroceptive sensor (see FIGURE 2.2 for a size reference of the *receptor* chipset) based on synchronizing radio signals received from multiple satellites. With that information it can compute the coordinates and height position of the sensor on any point of the world with up to 10m margin. This 10m error grows rapidly as less satellites are visible (direct radio wave reception is required), making it useless on closed environments, like urban canyons (Joerger and Pervan, 2006), etc.… Besides the weakness to satellite occlusion and wide error margin, the GPS presents other challenges, like a rather slow update rate for most of the commercial solutions. An enhancement available to the GPS is the differential GPS (DGPS), which improves the accuracy up to 10cm in the best implementations. This enhancement is obtained thanks to a network of ground-based fixed reference stations that broadcast the difference between the satellite measurement and the actually known fixed positions. Because of this, the availability of DGPS signal is even more limited than GPS signal.

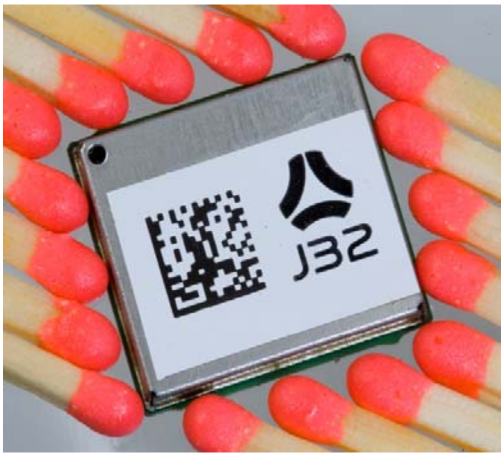


FIGURE 2.2: *A highly integrated GPS receiver microchip, with matches for size reference.*

There are two main alternatives to GPS which work under similar principles: GLONASS and Galileo. GLONASS (*Globalnaya navigatsionnaya sputnikovaya Sistema*) is the Russian response to the development of GPS, and although having suffered from a slow deployment, it is widely reported as being more accurate, with an error within the 2m range instead of 10m like GPS. On the other side, Galileo, the European analogous initiative, is still being deployed, though is expected to provide 1m accuracy when completed, around 2019. Though listed as alternatives, and being designed to be fully operational alone, the different global navigation satellite system (GNSS) can be combined to improve the availability of visible satellites, and thus the accuracy of the sensor (Dale et al., 1989).

The inertial measurement unit (IMU) is a proprioceptive sensor that combines several sensing components to produce estimations of the linear and angular velocities and the forces of the device. They generally integrate linear and angular accelerometers, and sometimes they include also gyroscopes and magnetometers, producing the sensory part of an inertial navigation system (INS). The INS includes a computing system to estimate the pose and velocities without external references. Systems derived from the IMU generally present good accuracy, but they are vulnerable to drift when used in dead reckoning strategies due their own biases. The introduction of external references can improve the accuracy, so they are frequently combined with GPS. Introduction of external references led to the development of the visual-inertial odometry field (Lupton and Sukkarieh, 2012)(Li and Mourikis, 2013), which is closely related to the SLAM (Piniés et al., 2007). Still, the accuracy gain is limited by the nature of the exteroceptive sensor added (which keeps its own weaknesses), and the IMU part of the system becomes unreliable in the presence of strong electromagnetic fields.

TABLE 2.1: FREQUENT NON-VISION SENSORS IN ROBOTICS

Sensors	Type[a]	Perception	Measurement	Features
Encoders	Passive	Propioceptive	Joint pose and/or derivatives	Widely used. Dead reckoning drift.
LIDAR	Active	Exteroceptive	Range and bearing scan, laser-based	Computationally expensive to process.
Radar	Active	Exteroceptive	Range and bearing scan, radio-based	Long range, affected by electromagnetic artifacts.
Sonar	Active	Exteroceptive	Range and bearing scan, sound-based	Short range, better suited for underwater operations.
GPS/DGPS	Passive[b]	Propioceptive	Position in global coordinates	Only for outdoor enviroments.
Inertial	Passive	Propioceptive	Specific force, angular rates	Sensors bias drifts over time.

[a] Passive sensors do not produce waves/light/sound/mechanical forces.

[b] The receptor is passive. The GPS infrastructure (satellites, radio networks, etc.) is active.

2.2.2 Vision based sensing and measurement

Vision based sensors are exteroceptive sensors which measure the environment through the reflection of light on it, capturing a set of rays conformed as a matrix, thus producing images. The most common visual sensor is the camera, which captures images of the environment observed in a direction, similarly to the human eye. Still, there are many types of cameras, depending on the technology which they are based on, which light spectra they capture, how they convert measurement into information, etc. A standard camera can generally provide colour or greyscale information as an output, at 25 frames per second (fps) or more, being generally focused on the wavelength range visible by the human eye, and presenting that information in a pleasant way to the human eye. Nevertheless, specific cameras can be designed for different scenarios or uses as a target, thus capturing other spectra not seen by human eye (IR, UV…), producing vastly higher fps rates, etc.…

One of the main weaknesses of cameras within the context of the SLAM problem is that they produce bearing-only data (from a geometry point of view): each element of the matrix which composes an image shows the information about a point where a ray (which theoretically can extend to the infinite) finds a solid object. Thus, cameras alone cannot produce depth estimation in a given time instant. This can be solved by more specialized sensors, like time-of-flight cameras (ToF). But these camera sensors generally have lower resolutions and framerates, and present reduced dynamic ranges and overall performance, while being several times more expensive. These features made them barely used for robotics research until few years ago, except in highly funded research areas.

2.2.2.1 Capture challenges

Standard digital cameras sensors are mainly based on two technologies: CCD (charge-couple device) and CMOS (complementary metal-oxide semiconductor). CCD technology was developed earlier, and still offers better image quality, with reduced noise and greater light sensitivity (enabling near-infrared light, night-vision, and zero or near zero-lux devices). A critical advantage of using CCD-based sensors is that they work natively using '*global shutter*' instead of the '*rolling shutter*' usually found in CMOS cameras. In a camera using '*global shutter*' all the pixels are captured simultaneously, thus providing a clear image without distortions or artifacts due to movements (unless the movement are of speed and magnitude significant w.r.t. to the capture time). On the other side, '*rolling shutter*' captures pixels consecutively, i.e., one-by-one or row-by-row, introducing additional artifacts (see FIGURE 2.3 and FIGURE 2.4).

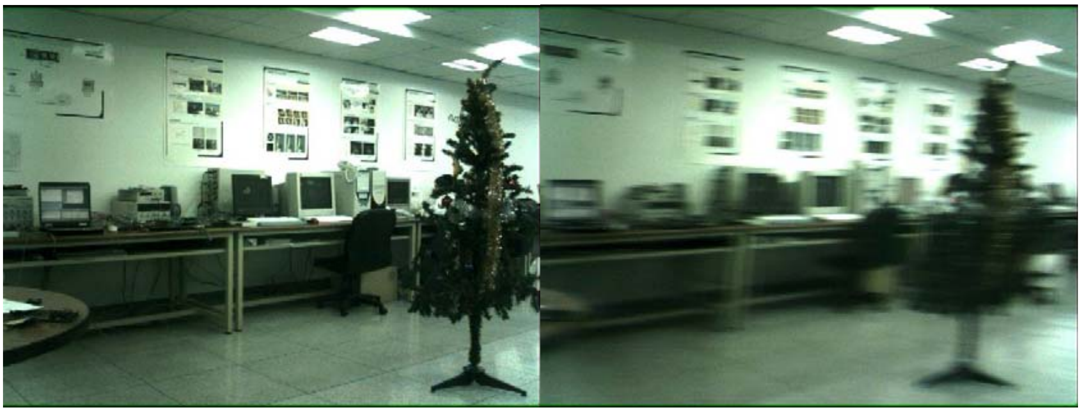


FIGURE 2.3: **Left:** *static image for reference.* **Right:** *same office scene captured with movement, showcasing artifacts induced by rolling shutter: not only is the image is blurred due movement, but the straight lines are distorted presenting a slight curvature.*

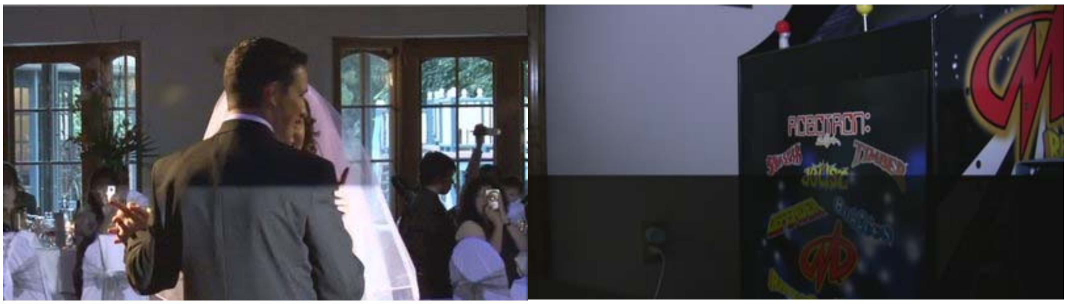


FIGURE 2.4: *Rolling shutter in a CMOS sensor with non-continous lightning.* **Left:** *effects of a camera flash from another device in a professional CMOS camera.* **Right:** *image taken under a fluorescent light with incorrect capture synchronization timing in the camera.*

Although on an ideal case the different shutter methods should not affect, there are several circumstances that can make 'rolling shutter' a problem. In dim light environments, the increase in exposition time (more acute in CMOS sensors presenting lesser light sensibility) can be combined with movements to skew the image: straight lines can be combed, and some objects might present blur during panning (Liang et al., 2008). Shifts in lighting can also disturb the capture process: a sudden flash can make an image appear divided in two regions, one much brighter than the other; and the flicker of fluorescents is usually a challenge, as several periodic artifacts and intensity distortions appear in video sequences.

Still, CMOS is becoming the standard digital imaging sensor technology in general consumer markets, as CCD production requirements make it much more expensive[2]. CCD is still the default technology for applications were budgeting is a lesser concern than the capabilities and performance, such as well-funded scientific research, industrial applications, medical fields, defense technologies, etc.

2.2.2.2 Complex Vision Based Sensors

There are several other types of camera sensors based on the technology of CCD/CMOS sensors, introducing additional hardware to modify the data captured, as shown in TABLE 2.2. Omnidirectional cameras open the horizontal field of view to 360º by using a smart trick to project all of the surrounding environment through a mirror into the sensor (see FIGURE 2.5 *left*). The frame captured by the sensor will be a concentric projection of the environment (see FIGURE 2.5 *right*), generally with a blind spot at the centre. They have been used successfully within the SLAM problem, both in filtering (Tardif et al., 2008) and bundle adjustment approaches (Lukierski et al., 2015), and are well suited for optical flow applications.

[2] For the same resolution and framerate the difference can be almost an order of magnitude.

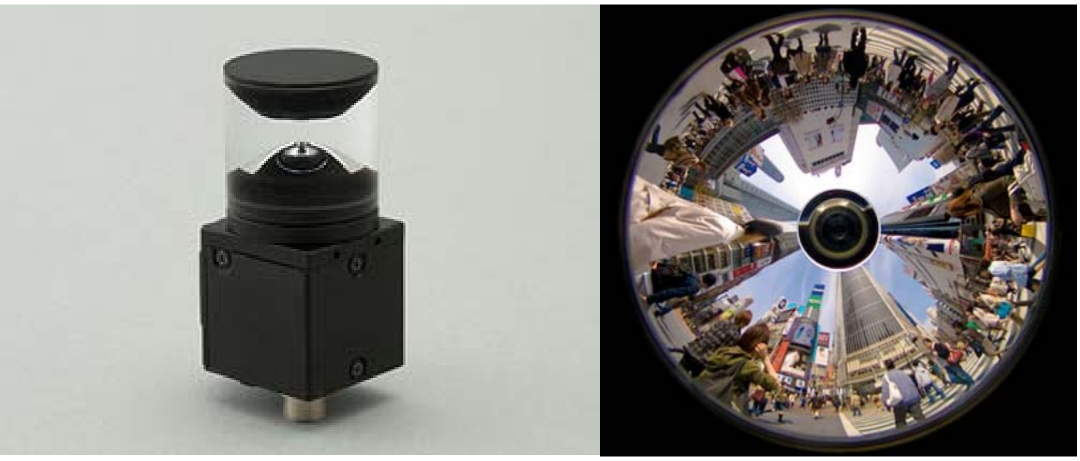


FIGURE 2.5: *Omnidirectional camera (courtesy of Olympus).* **Left:** *Camera sensor with mirrors to project 360^o around yaw axis.* **Right:** *Sample shot from an omnidirectional camera.*

Stereo cameras are based in building upon the basic digital camera: usually 2 similar camera sensors are set on a '*stereo rig*' configuration, where the precise geometry between their poses is known with great accuracy. This configuration produces stereo images, that is, pairs of calibrated frames, which can be processed by epipolar-based stereo vision algorithms. The same camera sensor can include the hardware required to compute a dense depth map, which would convert the camera into a depth sensor (known as RGB-D camera). Although the default assumption for stereo systems is using 2 cameras in a fixed configuration with coplanar projective planes, it is possible to use other setups (see FIGURE 2.6), for example introducing more cameras (Gallup et al., 2008), or variable geometry (Fanto, 2012).

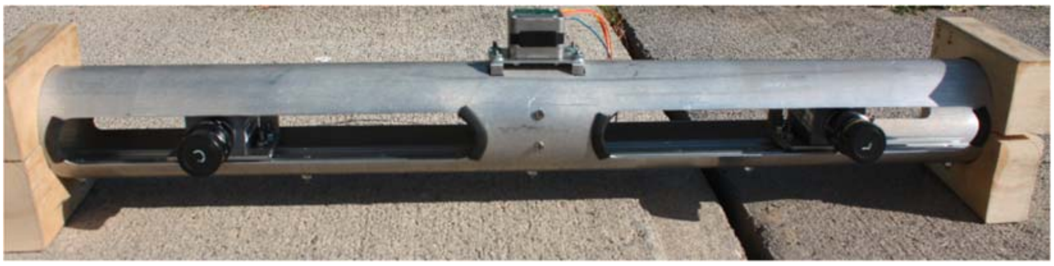


FIGURE 2.6: *Stereo camera, with variable geometry, from (Fanto, 2012).*

RGB-D cameras are a wide category of different technologies which deliver the same results: an image frame with depth measurement at pixel level. This kind of sensors can be based on stereo cameras (see FIGURE 2.7 *left*) with heavy image processing embedded at hardware level. Others rely on time-of-flight (ToF) cameras, where the entire frame is capture by a suitable sensor each time a laser pulse is emitted. Several devices based on analogous technologies have been made available to the general public since the launch of

Kinect®[3], which has helped popularizing the technology to the wider public, lowering the entry requirements for RGB-D technology in general (see FIGURE 2.7 *right*). SLAM approaches relying on RGB-D technologies have proven to be successful, as shown in (Endres et al., 2012) and (Kerl et al., 2013), to name some examples.

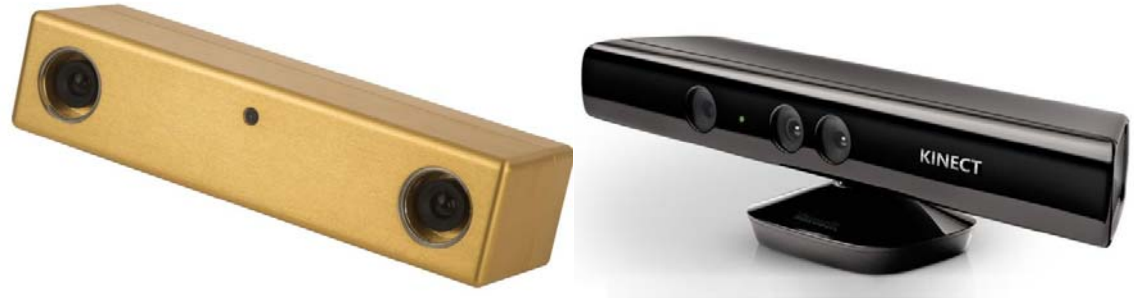


FIGURE 2.7: **Left:** *stereo based RGB-D camera (courtesy of PointGrey).* **Right:** *Kinect, with its IR emitter, IR depth camera, and colour camera all visible (courtesy of Microsoft).*

There are other approaches to vision based sensing which produce visual data and images in alternative ways. Event based cameras optimize the bandwidth available to transfer data minimizing the data sent. This is achieved by avoiding full frame synchronous images, and instead each pixel is sent asynchronously when a given variation threshold is reached. This technology has been applied to visual odometry (Censi and Scaramuzza, 2014) and SLAM (Weikersdorfer et al., 2014), and has been proved to offer a good trade-off in terms of accuracy versus computational power required in tested scenarios.

The ability to produce imaging sensor chips with higher resolutions, and the advances in fabrication of microlenses with much higher quality have opened many options based on field of light imaging (Chebira et al., 2003), making the plenoptic camera (Ives, 1930) a reality. In a plenoptic camera, a microlens array placed between the sensor and the main lens allows to decompose the scene captured into 'subimages', i.e, the same scene seen from slightly different points of view. This way the scene is capture through different images, forming a *'field of light'*. This *field of light* as sensor measurement requires processing to produce conventional images (see FIGURE 2.8). The great advantage is that the *lightfield* data captured allows refocusing images, modifying the depth of field the images, focusing on different depths and elements. The novelty and pricing of the technology used makes plenoptic vision still a not-so-well studied approach in robotics research, but it has been already applied to visual odometry (Zeller et al., 2016).

[3] Which, although originally conceived as a videogame accessory, has been a critical success in multiple research fields.

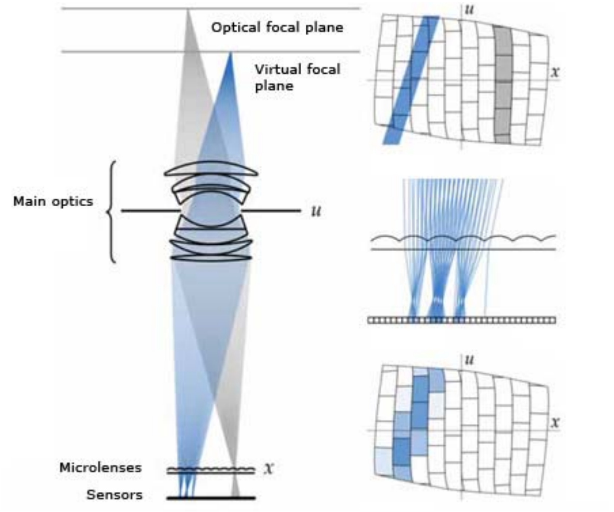


FIGURE 2.8: *Schema showing how the light is projected through an array of optical lenses into the plenoptic camera, and how microlenses refocus the rays into different regions of the sensor, building a mosaic of different subimages for the same capture (courtesy of Lytro).*

TABLE 2.2: CAMERA SENSORS IN ROBOTICS

Sensors	Type	Features
CCD	Passive sensor	First digital imaging sensor. Global shutter, higher sensibility and price.
CMOS	Passive sensor	Cheaper technology, presents rolling shutter and the related artifacts.
Stereo	Multiple sensor	Multiple camera in a rig. Allows for depth estimation using CV/photogrammetry.
Omnidirectional	Sensor & hardware	Sensor using mirror based projection of the whole environment. Complex projection.
RGB-D (stereo)	Two sensors, extra processing unit	As stereo, the photogrammetry part is integrated at hardware level, images and point clouds as output.
RGB-D (ToF)	Active sensor	ToF sensor, with optional RGB-D, sensor produces cloud points and images.
Event Based	Sensor	Sensor outputs pixel variations asynchronously instead of full frames.
Plenoptic	Sensor	Special lens to capture field of light. Images are produced through CV approaches, can be refocused later.

2.3 Simultaneous Localization and Mapping as a Filtering Problem

The SLAM problem is characterized by uncertainty and noise on the sensor input, so, many algorithms for SLAM are of probabilistic nature. As stated earlier, the objective in the SLAM problem is to estimate the environment and the localization with respect to it with the available data; and given that the data available will grow with each measurement, this data should be incorporated into the solution. A good fit to this problem thus is an incremental Bayesian filter approach.

In general, the objective of filtering methodologies is refining the knowledge available about the state of a given system with the information available through measurements. Note that refining is a different process, unlike simply adding new information and replacing older data. This can be observed in FIGURE 2.9, where an example trajectory in an environment with landmarks is illustrated. At the *top left* image the actual trajectory and environment are shown, with the trajectory perceived by the proprioceptive sensors of the robot in light grey shading. This measured trajectory presents drift and errors, like those that could be expected (see section 2.2.1 for more details on sensors). At the *top right* image the different measurements that the exteroceptive sensors of the robot would produce are shown, linking each pose of the robot with the different landmarks that would be observed. Assuming that the distance measurements between the robot and the landmarks in the map present no errors, such measurements are added to the map referenced with respect to the instant of the trajectory when they were observed, in the *bottom left* image, and introduced into a SLAM framework on the *bottom right*. This leads to the differences seen in FIGURE 2.9 *bottom,* the dead reckoning at the *left* produces a drift that cannot be corrected, and the landmarks are placed without accounting for their correlations, while in the right the iterative refining keeps the initial error bound (though this error can be observed in the landmark estimations).

A filtering methodology usually requires both previous knowledge to the process itself like the mathematical models representing the system and the initial state data, and later data like the measurements and inputs obtained during the process. It is worth noting that while the models representing the system are deterministic in nature, both the initial state and the data obtained through measurements are uncertain, thus, of stochastic nature. Then, it is required to note mathematically both known (or assumed to be known) models, as equations, and solve a problem stated as finding at each given time instant the best estimation given all the sequence of measurements since the initial state.

Note how, after each measurement, it is desired to have the best possible estimation given a growing sequence of data, so there are two available approaches:

- After each measurement, analyze and use all the information available.
- When a new measurement is available, refine the estimation obtained after the last previous measurement, in an incremental way.

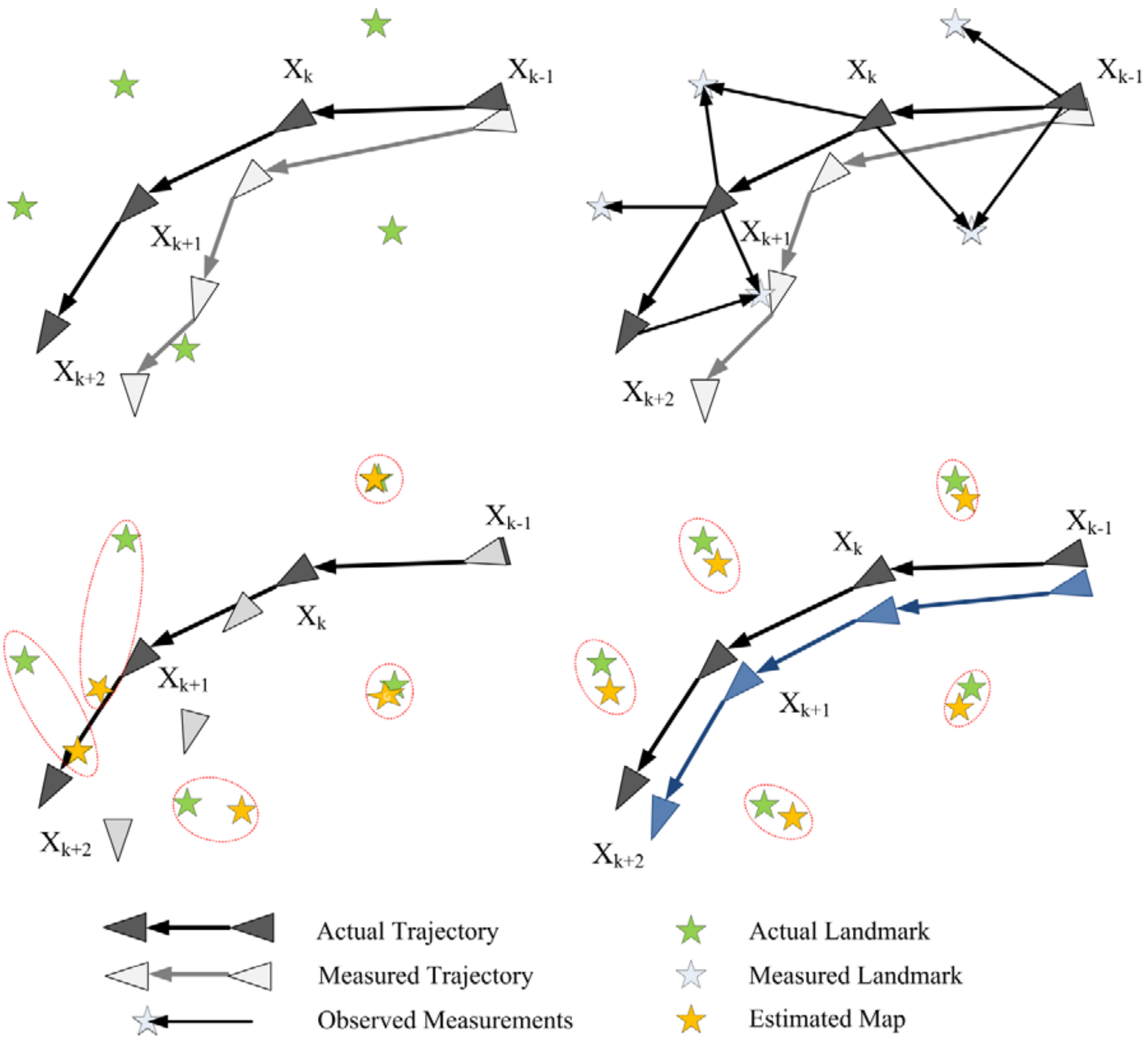


FIGURE 2.9: *Why SLAM is necessary?* **Top Left:** *the environment (green stars), with the actual and measured trajectories (black and light grey respectively).* **Top Right:** *measurements of the environment landmarks.* **Bottom Left:** *trajectory and measurements using a dead reckoning strategy. Red ellipses envelope the actual and estimated position of a landmark.* **Bottom Right:** *trajectory and measurements using a SLAM strategy.*

It is obvious that dealing with all the information at each step would suppose a great cost, thus, a filter able to use the second approach will have to deal only with a bounded amount of information, thus making it more suitable for on-line operations. The incremental nature of this approach allows using recursive formulations, which in turn can be converted into iterative algorithms.

To formulate the problem, suppose that we describe as the state to estimate the pose of a given robotic device, noted as **x**, and a map **m** that will be built incrementally with each measurement. Index *k* denotes the discrete time sequence, and it is assumed that for each *k* instant there will be new measurements and control signals for the robot. Under the assumption of a static world, the problem formulation assumes that the probability distribution *P* in equation (2.1) will be computed at each *k* time:

$$P\left(\mathbf{x}_k, \mathbf{m} \middle| \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0\right). \tag{2.1}$$

This probability distribution describes the joint posterior[4] density of vehicle **x** at time *k* and the map **m**, given the measurement observation history $\mathbf{Z}_{0:k}$ and the control signal sequence $\mathbf{U}_{0:k}$, considering $\mathbf{x}_0$ as the initial state of the system. This equation denotes all the stochastic information of the system, which is estimated and uncertain. But to solve this problem it is required to assume that some knowledge is considered certain, or deterministic, and can be used to model the relations between the data. This knowledge is represented by two different models, the state transition model (how the robot relates with the environment) and the observation model (how the measurements, so the robot, relate with the estimated map).

The state transition model describes, as seen in equation (2.2), the motion of the robotic device according to a given control signal $\mathbf{u}_k$ from the control signal sequence **U**:

$$P\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1}, \mathbf{u}_k\right). \tag{2.2}$$

As mentioned before, the problem is to be dealt in such way that only the most recent state estimation is needed to avoid dealing with all the information in the sequences **U** and **Z**. So the state transition process is assumed to satisfy the Markovian property[5], where the current $\mathbf{x}_k$ state depends on the preceding state $\mathbf{x}_{k-1}$ and the last applied control signal $\mathbf{u}_k$, becoming independent from the measurements, the map, and all the other previous states.

The observation model describes the probability distribution of the measurements to be obtained at *k*. As such, assuming that the state of **x** at *k* is known, only probabilities of the already known parts of the environment (thus, already available in the map) can be described. So, this model can be noted in the form:

$$P\left(\mathbf{z}_k \middle| \mathbf{x}_k, \mathbf{m}\right). \tag{2.3}$$

Introducing recursive filtering, we can use these models to link the posterior of a probability distribution at *k-1* with that of *k*. To achieve this, following the Bayes filter framework, a two-step process is used. Firstly, a prediction step is used to propagate the posterior density at *k-1* into a prior density of $\mathbf{x}_{k-1}$ at *k* by using the knowledge of the system dynamics represented by the state transition model. Then, this prior density is refined into a corrected posterior using the new information added through the measurements.

[4] The posterior (*a posteriori*) and prior (*a priori*) convention is used w.r.t. the measurement at instant *k*.

[5] Which can be summarized as the future is independent from the past given the current state.

2.3.1 Prediction Step

The prediction step (also known as *time update*) computes the prior density for time *k* propagating the posterior at *k-1* by using the knowledge about the system and the inputs dynamics described in equation (2.4).

$$\begin{aligned} P\left(\mathbf{x}_k,\mathbf{m}\left|\mathbf{Z}_{0:k},\mathbf{U}_{0:k},\mathbf{x}_0\right.\right) = \int P\left(\mathbf{x}_k\left|\mathbf{x}_{k-1},\mathbf{u}_k\right.\right) \\ \times P\left(\mathbf{x}_{k-1},\mathbf{m}\left|\mathbf{Z}_{0:k-1},\mathbf{U}_{0:k-1},\mathbf{x}_0\right.\right)d\mathbf{x}_{k-1} \end{aligned} \tag{2.4}$$

This means that the Markovian evolution that it is expected to take place according to the dynamics described by the transition model is applied to the knowledge considered certain about the system (the posterior at *k-1*).

2.3.2 Correction Step

The correction step (or *measurement update*), gives the posterior probability density at *k* by adding the information obtained through the measurement process. This step, described by equation (2.5), uses the Bayes Theorem based on the observation model distribution and the prior distribution to compute for the robot position distribution and map, thus obtaining the joint posterior distribution seen in equation (2.1).

$$P\left(\mathbf{x}_k,\mathbf{m}\left|\mathbf{Z}_{0:k},\mathbf{U}_{0:k},\mathbf{x}_0\right.\right) = \frac{P\left(\mathbf{z}_k\left|\mathbf{x}_k,\mathbf{m}\right.\right)P\left(\mathbf{x}_k,\mathbf{m}\left|\mathbf{Z}_{0:k-1},\mathbf{U}_{0:k-1},\mathbf{x}_0\right.\right)}{P\left(\mathbf{z}_k\left|\mathbf{Z}_{0:k-1},\mathbf{U}_{0:k}\right.\right)} \tag{2.5}$$

The dependence of the observations w.r.t. to both the robot and the map is explicity noted in the observation model, equation (2.3). This dependence precludes the simplification of the observation model through partition in separate terms for the probabilities of the map and the robot, as seen in equation (2.6):

$$P\left(\hat{\mathbf{x}}_k,\mathbf{m}\left|\mathbf{z}_k\right.\right) \neq P\left(\hat{\mathbf{x}}_k\left|\mathbf{z}_k\right.\right)P\left(\mathbf{m}\left|\mathbf{z}_k\right.\right). \tag{2.6}$$

This is coherent with results already predicted and presented in early mapping literature like (Smith and Cheeseman, 1986) and (Durrant-Whyte, 1988), where it is discussed how such partitioning would lead to estimation inconsistencies.

2.3.3 Emergent properties of SLAM as a Filtering Problem

The incremental structure of the solution implies that the map **m** must be built adding chunks of new information and refining previous data. Each new individual datum m_n is usually referred as a *'landmark'*, with the map acting essentially as a collection of landmarks. Another consequence of the incremental nature of the problem is that errors in the posterior distributions will be propagated into the following priors, meaning that errors in the robot pose have an impact in all the landmarks being observed. This effect can be mitigated in Bayesian filtering, as additional observations over the same datum m_n reduces

its uncertainty, and even without this mitigation, it is a better solution than just directly introducing data (see dead reckoning result in FIGURE 2.9 *bottom left*).
The effects of the error on the robot pose estimates can be observed in several of the landmarks. This means that the errors in the landmarks estimates are highly correlated, with the consequence that for a given set of landmarks observed concurrently, the relative pose between them can be estimated with low errors while at the same time the absolute position of each of them can be widely inaccurate. In terms of probabilities this means that the joint distribution for a given pair of landmarks $P(\mathbf{m}_i, \mathbf{m}_j)$ can be valuable as information source even if their marginal probability distributions are sparse. This becomes more apparent when considering that most of the time the biggest sources of error are introduced by a spurious estimation of the robot pose, which affects the landmark measurements, and is propagated through the filter. This can be seen in FIGURE 2.9 *bottom right*, which shows how the initial error in the robot pose is propagated, but bounded.

As more observations are performed, the correlations between landmarks increase, meaning that the relative pose between landmarks generally tends to improve. In (Dissanayake et al., 2001) the monotonically increasing nature of the correlations between landmarks was proved for the linear Gaussian case. This implies that, regardless of the movement of the robot, new observations introduce measurements of the relative poses between landmarks of nearly independent nature, meaning that the relative measurement between landmarks can be considered almost independent from the position of the robot. Considering this, it becomes intuitive the fact that the more the robot travels, more relative measurements between landmarks are available, increasing the correlation between them, thus, improving the believe estimation.

An analogy with a spring network was discussed in (Durrant-Whyte and Bailey, 2006). This analogy presented the map seen as a spring network, with each relative measurement between positions available represented with a spring. As the landmarks are observed from different positions due the robot movement, new springs are added, and those re-observed become stiffer as the correlations grow. This analogy shows in an intuitive way that the same way the spring network would become eventually rigid, an accurate map of landmarks' relative positions would be produced.

2.4 Kalman Filter and Extended Kalman Filter

The most used technique in monocular SLAM is the extended Kalman Filter (EKF), which is derived from the Kalman Filter (KF). The Kalman Filter considers that all uncertainties are of Gaussian nature, and adds a restriction assuming linear evolution and observation models, which allows for a finite formulation of the prediction-correction loop that can be solved for the optimal result. The derived EKF (McElhoe, 1966) relaxes this restriction, requiring only that the models used are locally linearizable around the last estimation of the state (prior or posterior).

2.4.1 Kalman Filter

KF is a Bayes filter whose distributions are Gaussians, and makes a series of assumptions:

- Both the motion model, or state transition function, and the observation model, must be linear with added Gaussian noise.
- The initial uncertainty is Gaussian.
- And last, the state $\mathbf{x}_k$ is dependent on $\mathbf{x}_{k-1}$, but no other previous states (satisfying the Markovian property).

The KF was fully proposed around 1960 (Kalman, 1960), being the first optimal estimation filter for linear system models with additive independent white noise. Some of its theoretical basis had been proposed by (Swerling, 1959) earlier, with the most known form of the recursive filter presented in (Kalman and Bucy, 1961).

There are plenty of works which deal with the details, not only the seminal literature mentioned above, but extensive works like (Hargrave, 1989), and those more recent and focused into specific applications (Fischer et al., 2013). As the focus of this dissertation is to deal with different aspects of the SLAM problem, we will describe and discuss the EKF directly, while inviting the reader to check the proposed literature.

2.4.2 Extended Kalman Filter

As both the state transition and observation model are normally governed by nonlinear trigonometric functions and have non-Gaussian noises, the KF is rarely used in the SLAM problem. Instead, through linearization of the relevant models, the EKF accommodates the nonlinear terms.

Thus, the EKF-SLAM formulation uses equation (2.7) to describe the robot motion

$$P\left(\mathbf{x}_k, \mathbf{m} \middle| \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_k\right) \Leftrightarrow \mathbf{x}_k = \mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_k\right) + \mathbf{w}_k \tag{2.7}$$

where $\mathbf{f}$ models the robot movement, and $\mathbf{w}_k$ are additive, zero mean uncorrelated Gaussian motion disturbances with covariance $\mathbf{Q}_k$. The observation model is described in the form

$$P\left(\mathbf{z}_k \middle| \mathbf{x}_k, \mathbf{m}\right) \Leftrightarrow \mathbf{z}_k = \mathbf{h}\left(\mathbf{x}_k, \mathbf{m}\right) + \mathbf{v}_k, \tag{2.8}$$

where $\mathbf{h}$ describes the geometry of the observation and $\mathbf{v}_k$ are additive, zero mean uncorrelated Gaussian observation errors with known covariances.

With these definitions, the standard EKF filter can be applied to compute the mean estimation and covariance, as equations (2.9) and (2.10), of the joint posterior distribution seen in equation (2.1).

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = E\begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix} \mathbf{Z}_{0:k} \tag{2.9}$$

$$\mathrm{P}_{k|k} = \begin{bmatrix} \mathrm{P}_{xx} & \mathrm{P}_{xm} \\ \mathrm{P}_{xm}^{T} & \mathrm{P}_{mm} \end{bmatrix}_{k|k} = E\left[\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m}_k - \hat{\mathbf{m}}_k \end{pmatrix}\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m}_k - \hat{\mathbf{m}}_k \end{pmatrix}^{T} |\mathbf{Z}_{0:k}\right] \tag{2.10}$$

In order to apply the prediction step to equations (2.9) and (2.10) time is updated increasing *k*, thus previous measurement become noted as *k-1*, and the estimations are update according to equations (2.11) and (2.12), for state and covariance respectively.

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}\left(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k\right) \tag{2.11}$$

$$\mathrm{P}_{xx,k|k-1} = \nabla\mathbf{f}\mathrm{P}_{xx,k-1|k-1}\nabla\mathbf{f}^{T} + \mathrm{Q}_k\left(\hat{\mathbf{x}}_{k-1|^\wedge k-1}, \mathbf{u}_k\right) \tag{2.12}$$

∇f is the Jacobian of **f** evaluated at the estimation point at instant *k*. Note that equation (2.12) explicitly requires linearization of the model, and for models with very strong non-linearities constitutes a source of error through misrepresentation. The assumption of static environment allows considering the landmark to remain in the same place.

After time is updated in the prediction step, data from the new measurements are introduced in the estimation during the correction step, refining the estimation mean and covariance with equations (2.13) and (2.14).

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \hat{\mathbf{m}}_{k-1} \end{bmatrix} + \mathbf{W}_k \mathbf{g}_k \tag{2.13}$$

$$\mathrm{P}_{k|k} = \mathrm{P}_{k|k-1} - \mathbf{W}_k \mathrm{S}_k \mathbf{W}_k^{T} \tag{2.14}$$

In these equations **g***k* denotes the Kalman innovation vector (also known as *residuals*), which contains the difference between the actual measurements, **z***k*, and predicted measurement of the landmarks, as seen in equation (2.15). The covariance of this innovation, denoted S*k*, is computed according to equation (2.16). The term **W***k* is usually known as ‘*Kalman gain*’ or ‘*gain*’, and modulates the impact of the innovation into the mean estimation. Note that for a given state where the uncertainties, annotated in P*k|k-1*, are of great magnitude, the Kalman gain will also be of great magnitude, and this will imply that the innovation can have a much bigger impact.

$$\mathbf{g}_k = \mathbf{z}_k - \mathbf{h}\left(\hat{\mathbf{x}}_{k|k-1}, \hat{\mathbf{m}}_{k-1}\right) \tag{2.15}$$

$$\mathrm{S}_k = \nabla\mathbf{h}\mathrm{P}_{k|k-1}\nabla\mathbf{h}^{T} + \mathrm{R}_k \tag{2.16}$$

$$\mathbf{W}_k = \mathrm{P}_{k|k-1}\nabla\mathbf{h}^{T}\mathrm{S}_k^{-1} \tag{2.17}$$

It is worth noting that the term **∇h,** which describes the Jacobian of the observation model **h** evaluated at the estimated joint posterior distribution, appears both in equations (2.16) and (2.17). Once again, non-linearities in the models are always undesirable, but in the observation model case, it becomes twofold: not only they produce misrepresentation errors in the linearization steps, but as they are frequently related with angular magnitudes, most of the time they also increase the Abbe error.

Through this formulation, the EKF can be used to perform SLAM, being commonly the adopted solution. Still, there are some aspects that must be considered. For starters, EKF-SLAM uses linearized models in several parts, both in the prediction and correction steps. Problems due linearization errors may arise, as it has been commented at several points. This may lead to inconsistent solutions and disrupt the filter convergence. Moreover, the cost of the EKF SLAM grows rather rapidly, as fast as quadratically with respect to the number of landmarks in the map. Several optimizations can improve this performance, being a problem that has received much attention, leading to some alternative formulations, commented in the next section.

2.5 Other Filtering approaches

The EK-EKF framework has become one of the most relevant filtering frameworks in general estimation problems. Its wide utilization has led to the development of several variants and derived techniques, with many of them finally being used in the SLAM problem. Some of them can be seen as simple upgrades or modifications, which allow an easy transition from the EKF framework, while others require essentially a full reformulation of the SLAM problem. The most impactful are discussed in this section.

2.5.1 Gaussian Sum Filter

The Gaussian Sum Filter (GSF) (Alspach and Sorenson, 1972) may be viewed as a further extension of the EKF. The main idea behind it is the approximation of the *pdfs* by Gaussian mixtures, i.e. sums of weighted Gaussians, each one of them adequately fulfilling the EKF requirements on linearization. Each Gaussian will be processed by a different EKF, and an additional method will be needed to adequately update the Gaussian weights during predictions and corrections.

As each of the weighted Gaussian distributions used will be dealt with an EKF, it will be necessary to maintain a bank of EKFs. The advantages of this filter lie in that it weakens greatly two restriction of the EKF:

- The initial distributions and uncertain knowledges (like noise) can be described as a Gaussian mixture instead of a purely Gaussian distribution, thus fitness of knowledge representation improves.
- The restriction of local linearizability around the estimates becomes more relaxed, so extreme non-linearities which could disrupt easily the EKF can be treated.

As the methodology keeps the estimation of a bank of EKFs, to obtain the actual estimation of the state an additional step to fuse the distributions is required. Moreover, given how the prediction step is performed, the size of the EKF bank grows at each prediction-correction cycle, making the filter computationally unfeasible rather fast. Thus, an additional procedure is required to keep the size of the EKF bank manageable.

2.5.2 Information Filter

The EKF accommodates the nonlinearities from the real world, by approximating the robot motion model using linear functions. An alternative approach, but still closely related, would be the utilization of the information filter (IF) or the extended information filter (EIF). The IF is implemented by propagating the inverse of the state error covariance matrix. There are several advantages of the IF filter over the KF. Firstly, the data is filtered by simply summing the information matrices and vector, providing more accurate estimates (Thrun and Liu, 2003). Secondly, IF are more stable than KF (Thrun et al., 2004). Finally, the main feature is that in the information form the information matrix is approximately-sparse, with weaker correlations having really small values, which can be marginalized.

However, the IF has also some important limitations. A primary disadvantage is the need to recover a state estimate in the update step when applied to nonlinear systems. This process requires the inversion of the information matrix. Further matrix inversions are required for the prediction step of the information filters. For high dimensional state spaces the need to compute all these inversions is generally believed to make the IF computationally worse than the Kalman filter. In fact, this is one of the reasons why the EKF has been vastly more popular than the EIF (Wang and Dissanayake, 2010). Nevertheless, as the information matrix is approximately sparse, with lower values far from the diagonal eliminated, the matrix can be dealt as an sparse graph. This approach has allowed developing methods to make the updates in an iterative fashion, with efficient costs, and better performance for large mapping than EKF filtering.

2.5.3 Unscented Kalman Filter

The Unscented Kalman Filter (UKF) (Julier and Uhlmann, 1997) addresses the approximation issues of the EKF and the linearity assumptions of the KF. The KF is used to propagate a Gaussian Random Variable (GRV) in systems with linear dynamics, and the EKF approximates the optimal terms by linearization of the dynamic models. Thus, in the EKF the state distribution is approximated by a GRV which is propagated analytically through the linearization of the system. As approximations and linearization can induce errors in the state and covariance, another approach would be to represent the state distribution by sampling the state in a reduced set of points which can capture the dynamics and covariance of the state distribution. These points are propagated through the non-linear space, avoiding linearizations and their effects. This is achieved through the unscented transformation (Julier and Uhlmann, 2004), which is used to estimate the result of applying the nonlinear models to the state distribution, characterizing with a limited subset of state points. Even if it overcomes several of the problems of the EKF, it can still produce inconsistencies, and usually presents a worse computational cost, as it was discussed in (Huang et al., 2009).

2.5.4 Particle filters and Fast SLAM

The particle filter (PF) techniques are derived from the sequential Monte-Carlo (SMC) method, tracing back to the late forties (Metropolis and Ulam, 1949). In order to represent a Bayesian posterior distribution the SMC uses a set of random point clusters, the particles. As a nonparametric method, the particle filter represents the distribution by a set of samples drawn from the same distribution. This feature allows handling great non-linearities and non-Gaussian noise. But unlike the UKF, which uses the unscented transformation to deterministically choose the samples, the PF takes a number of randomly selected samples. This procedure forces the filter to take many particles, making the computational complexity grow rapidly.

The solution adopted at FastSLAM (both 1.0 (Montemerlo et al., 2002) and 2.0 (Montemerlo et al., 2003)) was to apply *Rao-Blackwellization* to reduce the dimensionality of the state-space. The part of the state related to the robot is represented by a set of weighted sample particles, while the map accompanying each particle is composed of independent Gaussian distribution. Then the recursive estimation is achieved by applying the PF to the robot pose estimation, and standard EKF filtering for map estimation.

To do this, the algorithm first computes a *'proposal distribution'*, and starts drawing from particles this distribution. For each particle, the *'importance function'* assigns a weight, and a particle filter is applied to the robot part, with a resampling process that reassigns weights if needed. After this process is done for all the particles, the map for each particle is update as an EKF filter considering the robot part of the state (usually the pose) as known. To refine the estimation and avoid degeneration of the maps due effects of particles relevant to previous states, several strategies were introduced, such a map marginalization. This still presents issues, as the map dependence on the robot and the removal of the measurement history may induce disturbances during the resampling step (Bailey et al., 2006).

2.6 Optimization Based approaches

The Structure from Motion problem (SfM), from the computer vision field of research, has many similarities with the SLAM problem in robotics, but it presents several key differences that kept them separated for decades. In classical SfM approaches the focus was set into obtaining the best possible map (concentrating on the geometry of the environment, through photogrammetry) from a given set of images, generally in an offline computing environment. This led to a heavy focus on geometrical modelling and optimization procedures as prevalent solutions. On the other hand, SLAM approaches focused on real-time continuous estimation, especially of the sensor poses, thus leading to incremental solutions. Besides, while SfM always dealt with 3D scene reconstruction from image sequences, the SLAM problem not always is considered to include cameras as a sensor.

Still, the introduction of Bayesian methods in both fields allowed closing the divide between the SfM and SLAM problems, leading to the emergence of keyframe methods based on Bundle Adjustment (BA). On these keyframe methods the main strategy is to select a subset of image frames from the image stream, and perform optimization over them analogously to the SfM strategies. Several of these strategies still rely on filtering SLAM approaches to track the pose variations, using the filter estimation as a seed for the optimization processes.

These approaches have a rigid algorithmical structure that is tightly coupled to the hardware on which they will run. In (Klein and Murray, 2007), where PTAM (Parallel Tracking and Mapping) was first introduced, an architecture based on splitting the classical problem into two different tasks (tracking the pose and building a map) is described. Each of these tasks was mapped as a different thread on a different processing unit, and would exchange the data required asynchronously. The continuation of this work (Klein and Murray, 2008) improved the resilience of the method to sudden motions, introducing also a direct image-based method to estimate inter-frame rotations using full frames, and the utilization of edgelets as features, taken from (Eade and Drummond, 2009).

DTAM (Dense Tracking and Mapping) was presented in (Newcombe et al., 2011), acting as a real-time SfM method, based in a full direct image approach. In that work the map was modelled as a dense textured 3D mesh, estimated through keyframe optimization at pixel level, finding dense (at pixel spatial resolution) depth maps. The novelty of the work resided not only in the taken approach, but also in the level of technical development, exploiting high parallelization GPGPU[6] methods and techniques.

2.7 Mapping & Localization: Robot & Environment Representation

One of the most critical aspects of any SLAM methodology is the representation of the system, that is, how the localization of the robot is noted, and how the environment is described. Most of the filtering approaches deal in a straightforward way: the state is represented by an *augmented state vector*, which denotes consecutively the state of the robot and the *'map'*.

$$\mathbf{x} = \left[\mathbf{x}_r \;\; \mathbf{m}\right]^T \tag{2.18}$$

The robot state is generally described through its pose and velocities in a given notation. It is worth noting that for many methodologies, given the nature of the filter they use, it is desirable to employ a parametrization that produces prediction models with the lowest non-linearities possible. Thus, the position of the robot is commonly parametrized as an

[6] General-Purpose computing on Graphics Processing Units.

Euclidean point with respect to the starting point of the estimation procedure, while the orientation has been reported to be noted through several representations, generally related to the Euler angles given their intuitiveness. Still, the quaternion (Hamilton, 1844) has become the orientation representation most commonly used, given its advantages in terms of avoiding singularities and ambiguities.

The nature of the map is essentially dependent on the nature of the observations available through sensor measurements. For example, scan range-finders can produce dense and accurate maps (with limitations according to the dimensionality of their scanning process), but are limited in range, and require to store and correlate the data in some fashion. Visual based approaches rely on the qualities of the observations produced by algorithms that process the images and extract the information. As such, the images obtained in visual SLAM are processed to extract geometric primitives observable in the image projection of the environment (e.g., points and lines).

$$\mathbf{x} = \left[\mathbf{x}_r \; \mathbf{y}_0 \cdots \mathbf{y}_n\right]^T \tag{2.19}$$

As the filter based approaches deal with the knowledge acquired during the estimation in an incremental fashion, it is possible to parametrize new data into the map in a direct way: the map considered to hold data about the environment is composed of chunks of this information, in the form of landmarks. Thus, each landmark $\mathbf{y}_n$ will correspond to one key element of the environment, like the observed primitives for a visual SLAM procedure. If the element was previously observed, the information will be used to refine its representation already parametrized into the map.

2.7.1 Landmark parametrization

When speaking of landmarks in a visual SLAM context, for most of the cases the term itself is used to describe point landmarks, as they are by far the most used. Still, it is worth noting that other types of features exist and have been used successfully.

Though other approaches may use analogous representations for the different elements of the map/state/filter represented, the following section discusses landmark parametrization in the context of visual EKF or equivalent filtering SLAM approaches. This is relevant because the Bayesian nature of the EKF allows using redundant parametrization. As pointed in (Sola et al., 2012), EKF, as a Bayesian estimator, uses an initial prediction to generate the prior distributions that constrains the redundant DoF that would disrupt convergence in other approaches (such as those based on bundle adjustment or other iterative optimizations).

Euclidean points:

An Euclidean point codifies a given position in 3D space with three Cartesian coordinates (see FIGURE 2.10 *left*). They represent the simplest possible parametrization, as seen in equation (2.20), and allows for annotating rotations with quaternions, which avoids the gimbal lock problem. Moreover, rotation matrices based on quaternion rotations tend to present bilinear relations which simplify Jacobian computations.

$$\mathbf{p}_e = [x \quad y \quad z]^T \in \mathbb{R}^3 \tag{2.20}$$

Even using quaternions, the Euclidean points are unsuitable for bearing-only SLAM systems, as they introduce severe non-linearity on the models, aggravating Abbe's errors. This has been long reported and known, since (Chiuso et al., 2000).

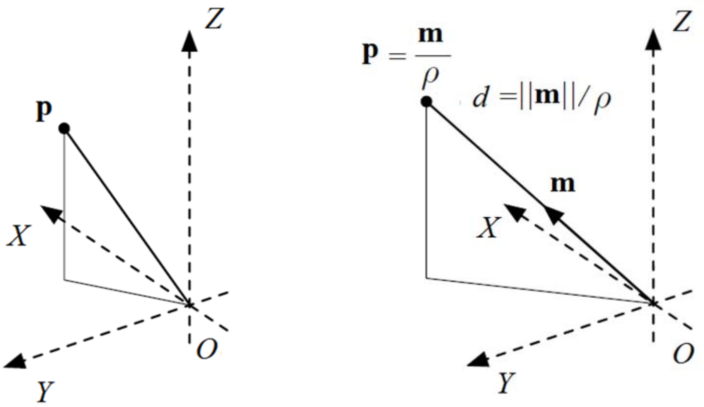

Z
p
X
Y
O
$\mathbf{p} = \frac{\mathbf{m}}{\rho}$
Z
$d = ||\mathbf{m}||/\rho$
m
X
Y
O

FIGURE 2.10: **Left:** *Visual representation of the Euclidean coordinates of point* **p**. **Right:** *Geometrical interpretation of the homogeneous coordinates for* **p**.

Homogeneous Points

Homogeneous points are coded by a vector of 4 elements, mapping a projective $\mathbb{P}^3$ space. Although this representation is widely known and used in computer vision, it is rather new in the SLAM field, first seen in (Marzorati et al., 2008). This vector is composed of a 3D vector, noted **m**, and a scalar p, the homogeneous part:

$$\mathbf{p}_h = \begin{bmatrix} p \\ \mathbf{m} \end{bmatrix} = [p \quad m_x \quad m_y \quad m_z]^T \in \mathbb{R}^4 . \tag{2.21}$$

The conversion of a homogeneous point to Euclidean coordinates is straightforward, being $\mathbf{p}_e = \mathbf{m}/p$. This means that each $\mathbf{p}_e$ can be represented by an equivalence class through proportional transformations of the 4-vector of a homogeneous point (see FIGURE 2.10 *right*). Different choices for the canonical values can produce several representations widely known in computer vision: $p = 1$ is the original Euclidean parametrization; $m_z = 1$ is the inverse-depth; and $||\mathbf{m}|| = 1$ is the inverse-distance.

As discussed in (Sola et al., 2012), the inverse-distance is isotropic, and if a given point is expressed w.r.t. to a camera sensor, **m** is the director vector of an optical ray to the point, and *p* presents linear dependency with the inverse of the distance between said sensor and the point.

Plücker Coordinates

Plücker lines codify a line in $\mathbb{P}^3$ space through 6 parameters, based on the Plücker coordinates introduced by Julius Plücker in the 19th century. Assuming, in a 3-dimensional projective space $\mathbb{P}^3$, a line *L* crossing homogeneous points $\mathbf{a}_h$ and $\mathbf{b}_h$, the Plücker coordinates can be represented as a 6-vector $\mathbf{l}_{\mathrm{p}} \in \mathbb{P}^5$. The elements of this 6-vector can be obtained through several ways, though in the context of SLAM (especially bearing-only approaches) the most used representation is that proposed at (Bartoli and Sturm, 2001), shown in equation (2.22):

$$\mathbf{l}_p = \begin{bmatrix} \mathbf{n} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} n_x \ n_y \ n_z \ v_x \ v_y \ v_z \end{bmatrix}^T \in \mathbb{P}^5 \subset \mathbb{R}^6, \tag{2.22}$$

which represents the corresponding Plücker Matrix L_p:

$$\mathbf{L}_p = \begin{bmatrix} [\mathbf{n}]_x & \mathbf{v} \\ -\mathbf{v}^T & 0 \end{bmatrix}, \quad \mathbf{n}, \mathbf{v} \in \mathbb{R}^3. \tag{2.23}$$

The general formula of the Plücker Matrix describes a given line *L* using two different points in homogeneous coordinates, and this can be used, as seen in equation (2.24), to obtain a 4x4 skew-symmetric matrix subject to the Plücker constraint, that is, the determinant must be zero.

$$\mathbf{L}_p = \mathbf{b}_h \cdot \mathbf{a}_h{}^T - \mathbf{a}_h \cdot \mathbf{b}_h{}^T \in \mathbb{R}^{4\times 4} \tag{2.24}$$

So, the representation considered in equation (2.22) allows to define vectors **n** and **v** as:

$$\mathbf{n} = \mathbf{a}_e \times \mathbf{b}_e \qquad \mathbf{v} = a_p \mathbf{b} - b_p \mathbf{a}\,. \tag{2.25}$$

This representation means that the Plücker constraint is now equivalent to the orthogonality condition $\mathbf{n}^T\mathbf{v} = 0$, and is conditioned in a way to make it easy to visualize from an Euclidean intuition point of view, as seen in FIGURE 2.11 . Vector **n** is normal to the plane which passes through the origin of coordinates and contains the line, while **v** is a director vector for the line, going from **a** to **b**. Then, the distance from the origin of coordinates to the line can be computed as $||\mathbf{n}||/||\mathbf{v}||$.

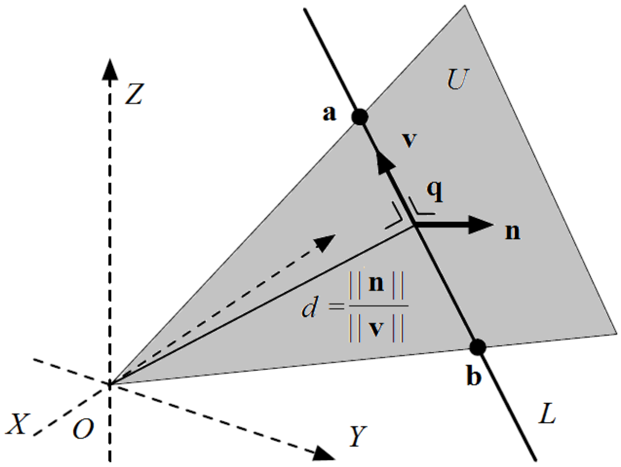


FIGURE 2.11: *Visual representation of the Plücker line coordinates. Line defined by points* **a** *and* **b** *is denoted by the vector* **v** *and* **n** *in Plücker coordinates, lying on the plane U (shaded in grey).*

Unified Inverse Depth Parameterization

In (Civera et al., 2006) a new approach to parametrize point features was introduced. The '*inverse depth points'* (IDP) method presents a formulation which combines characteristics from the homogeneous points representation and from earlier works on simplified polar coordiantes (Aidala and Hammel, 1983). A given point **p** is parametrized as $\mathbf{p}_{idp}$ according to equation (2.26): through an anchor $\mathbf{p}_0 = (x_0,y_0,z_0)$, a director vector **m**, and a distance to the point, codified through the inverse of its value, ρ.

$$\mathbf{p}_{idp} = \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{m} \\ \rho \end{bmatrix} = \begin{bmatrix} x_0 & y_0 & z_0 & \theta & \phi & \rho \end{bmatrix}^T \tag{2.26}$$

To find the Euclidean coordinates of a point under IDP notation, the director vector **m** must be applied to the inverse of ρ, and translated to the anchor, as illustrated in FIGURE 2.12 seen in equation (2.27):

$$\mathbf{p}_e = \mathbf{p}_0 + \frac{1}{\rho}\mathbf{m}(\theta,\phi) \tag{2.27}$$

where

$$\mathbf{m}(\theta,\phi) = \begin{bmatrix} \cos\phi\sin\theta & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}^T . \tag{2.28}$$

In a visual SLAM context, the anchor $\mathbf{p}_0$ is generally set to be the Euclidean coordinates of the camera optical centre when the point was first parametrized. This allows decoupling the uncertainty of the term multiplying the most uncertain value, the distance to the point (inverse distance in our case). The notation of the distance to the point through its inverse is especially fit for visual approaches: ranges up to near infinite can be codified within a bounded range, and representing the uncertainty with a low range of values also makes the filtering approaches more numerically stable.

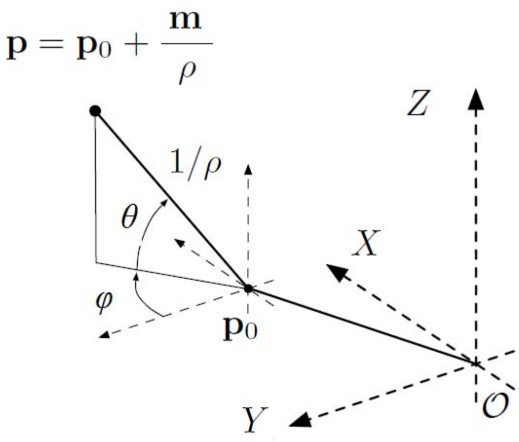


FIGURE 2.12: *Visual representation of the inverse depth points (IDP) parametrization for a given point* **p** *anchored a* $\mathbf{p}_0$ *with director vector* **m** *at distance 1/ρ.*

Introducing the anchor $\mathbf{p}_0$ improves the accuracy of the representation in the long term: the uncertainty between the camera pose and position $\mathbf{p}_0$ is initially low, and while the camera is near $\mathbf{p}_0$ the uncertainty remains low, but as soon as the camera moves from the anchor, the relative uncertainty grows quickly between camera and anchor. Without this anchor, the isolated position uncertainty would have a big impact in the director vector, **m**, which would be modified so that it is considered to have origin in the moving camera optical centre.

2.7.2 Other Representations

The Graphical SLAM approaches represent the whole state as a graph of poses, where the pose nodes are related through spatial constraints acting as edges. This kind of methodologies often rely on the sensors ability to perform scan measurements, like many types of range finders, associating each scan measurement with a node. The last node added to the graph is the last pose of the robot where a full scan measurement was performed, and can be linked to several other nodes. To build the map, a Graphical SLAM technique will initially store the whole trajectory and all the relevant measurements, then all this data can be used to retrieve a visualizable map. Graphical SLAM approaches rely heavily on detection of loop closures, as in passing near a previously recorded pose, which allow optimizing the whole trajectory, refining and removing redundant data. This data has been generally stored as vectors and sparse matrices, but as the trajectory grows it becomes inconvenient. Several graphical SLAM algorithms have been designed to used alternative representations, like (Paskin, 2003). In that work the thin junction tree filter (TJTF) is used, treating a long trajectory as a set of coupled maps, analogously to a submapping approach, where each cluster of the junction tree could be seen as a submap.

More recently, in (Kaess et al., 2010), a new data structure specifically designed for graphical SLAM approaches was proposed. Loosely based on clique trees (Blair and Peyton, 1993), the Bayes tree is used to codify factored probability densities, with directed edges to map the information matrix, and as shown in (Kaess et al., 2012), it outperforms the TJTF without omitting belief information.

In DTAM, the map is represented by a dense textured 3D mesh with millions of points, obtained by multi-view reconstruction over sets of keyframes, enabling subpixel resolutions. Then localization of the camera is performed through image registration of obtained frames against said 3D mesh which models the scene.

2.7.3 Map Management and Loop closing

Some SLAM approaches can deal with large and complex maps with relative ease, like those based on the information form of the Kalman Filter (Thrun and Liu, 2003), also known as the inverse covariance filter. This is possible because the information matrix is approximately-sparse in information form, with weaker correlations having really small values, which can be marginalized. As commented earlier, it can deal with relatively large maps through the utilization of several techniques to linearize the cost of succesive matrix inversion operations over time.

With regard to EKF based SLAM and similar techniques, the most usual approaches to deal with complexity over large maps are based on partial updates or directly dividing the map, improving greatly the performance. This allows performing the update step of the filter, which is the most expensive computationally, over a reduced state. The compressed EKF method (Guivant and Nebot, 2001) restricts the state to be updated to the vehicle position and a subset of the nearest and most recently observed features.

A brief survey of submap based techniques can be found in (Bailey and Durrant-Whyte, 2006). Most of them rely on sequentially creating different maps, and joining them at a later point. There are some criteria to classify them, but the most relevant would be how the position of the different submaps is treated: in global submap methods the position of each submap is represented with respect a global reference frame (FIGURE 2.13 *left*), while the relative submap methods (FIGURE 2.13 *right*) work with the position of each submap in relation to other close submaps.

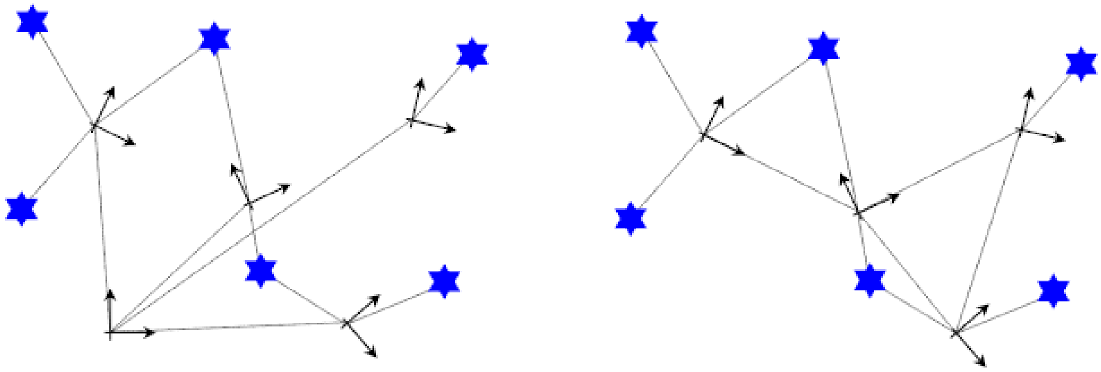


FIGURE 2.13: **Left**: *globally sharing an initial reference point.* **Right**: *locally referenced submaps, each one referenced with respect to other maps.*

Hierarchical SLAM (Estrada et al., 2005) creates different submaps, with features added relative to the point where each map is started. Joining operations of the different submaps are performed when a loop closure is detected, optimizing then the location of the submaps. This allows hierarchical SLAM to reduce the computational dependence from quadratic time with respect the number of landmarks to linear or constant time. There are other techniques based on similar approaches, like the constant time SLAM (Leonard and Newman, 2003), achieving also constant time. The downside is that to achieve this reduced cost, the local maps are related to a common reference frame, leading to bigger linearization errors due an increased uncertainty.

On the other side, the local map joining (Tardós et al., 2002) builds independent, totally separated maps: once a given size of map is reached, an entirely new EKF filter is initiated, with a new covariance matrix. The divide and conquer SLAM (Paz et al., 2008) introduces complex policies to manage the submaps, joining preferably smaller maps and delaying the more expensive operations of joining the larger maps. The conditionally independent SLAM also relies on keeping different submaps, but they are not required to be completely independent, thus, they can share information, which will be useful during the joining step. An earlier work, the constrained local submap filter (CLSF) schedules explicitly when the global covariance matrix must be updated. This way it can maintain several relative submaps, and obtain map and vehicle estimates matching those of an EKF-SLAM without a submapping technique (Williams et al., 2002).

A common approach to describe the relationships between the submaps in the relative submapping methods is creating a graph. This would work in a similar way to Graphical SLAM (Folkesson and Christensen, 2004), and other graph-based SLAM approaches. The Atlas framework (Bosse et al., 2003) implements a submapping methodology independent from the technique used to create the different submaps. In general, the relative submaps framework presents the advantage of creating locally optimal maps, numerically stable, while keeping the computational complexity reduced. As the updates are performed locally, with a reduced number of features, it also reduces the cost of the association and loop closing problems; and it lowers the error from the possible linearization assumptions done with respect to the global submap methods.

2.8 Conclusions

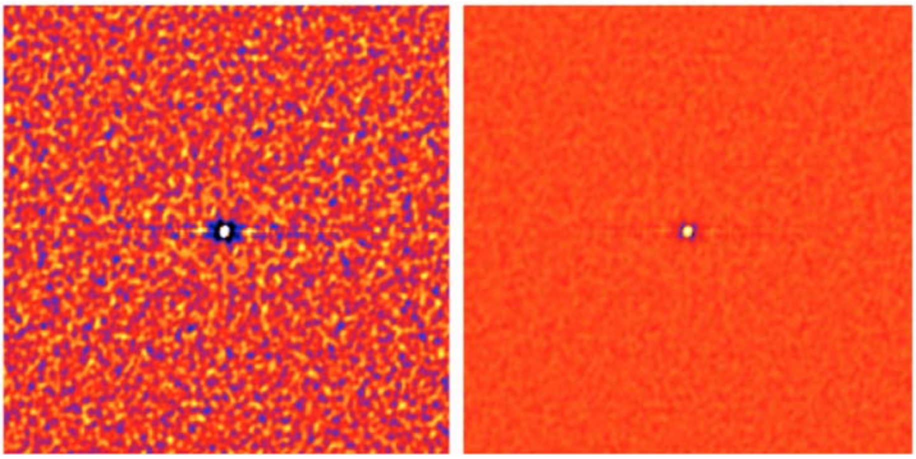
In this chapter the prerequisites to have a wide vision of the SLAM problem, both from the technical and mathematical points of view, have been presented and discussed. From a technical point of view, a review of the most used sensors in the SLAM research community is presented. As this dissertation deals with aspects of the visual SLAM problem, the vision-based sensors have been discussed in detail, in terms of capabilities, weaknesses and convenience of each type of sensor. Additional focus was put on discussing the features of low-price CMOS sensors, and the additional challenges they present, as the experiments described in other chapters of this work were based in this technology.

The mathematical foundations of SLAM as a probabilistic recursive estimation problem have been described, studying its origin and most important solutions. The most influential solution to the SLAM problem, the Extended Kalman Filter, has been detailed in terms of Bayesian probabilities, detailing how this can be translated into a general formulation for the SLAM problem. The main alternatives to the EKF have been commented and discussed, with emphasis on their advantages and weaknesses when compared to the EKF in the context of the visual SLAM problem.

Several other aspects of the general and visual SLAM problems have been also discussed beyond the estimation method, such as modelling aspects, including how to describe the maps and the elements present in them. Special attention was placed in the parametrization of landmarks, discussing key points of the different mathematical characterizations and some of their properties which have an impact in the SLAM problem. The map management was also discussed, in terms of structure and classification of methods, including the detection of loops (place recognition problem) and management of extended maps and long trajectories.

Chapter 3

Computer Vision and research antecedents in SLAM



A set of reconstructed 692 nm high resolution specle images of TRAPPIST-1, with linear and logarithmic flux scale, courtesy of (Howell et al., 2016). The star observed was later found to present 7 orbiting telluric planets, 3 of them within its habitable zone. Several of them could present water, making them habitable.

3.1 Introduction

Although there are plenty of sensors that can be used to solve the SLAM problem, as we just reviewed in Chapter 2, cameras are one of the most usual sensory devices. Though traditionally this meant stereo or monocular vision, recent developments in hardware and integration technology have opened the option to work with cameras of more complex nature, like the different types of RGB-D sensors, or the omnidirectional cameras. This chapter presents an in depth review of the DI-D monocular SLAM approach, which was used as the basis for this thesis, preceded by the foundations on computer vision and projective geometry required through this dissertation.

3.2 Projective Geometry and Camera-Based Artificial Vision

As we stated when formulating the SLAM problem, the objective of any solution is to build a map and perform localization w.r.t. this same map. The landmarks composing this map have to be modeled in mathematical terms, as it was seen in Section 2.7. That section discussed the notation of the landmarks according to different models working on real world coordinates, but in fact they are to be perceived as data extracted from a subset of pixels in an image. The data are extracted from the image through computer vision, applying techniques which detect significant points that present certain qualities, known as point features; while the spatial relation between the observed environment and the data extracted is provided through projective geometry.

3.2.1 Mathematical Monocular Cameras

From a mathematical point of view, a camera is a mapping between 3D real world and a 2D image space. Although there are several camera models, most of them are, including all based in the central projection, specializations of the known projective camera. In this section, we will focus in the well-known pinhole camera model, which idealizes the thin lens model[7].

This pinhole camera model maps the points through a *projection* operation, which can be seen as an injective application *P*, noted as:

$$P:\mathbb{R}^3\rightarrow\mathbb{R}^2;\ (X,Y,Z)\rightarrow(x,y)=P(X,Y,Z). \tag{3.1}$$

This *projection P* is assumed to follow the central projection, so we can consider that the centre of projection **C** is the origin of an Euclidean coordinate system, and that there is a plane $Z = f$ which act as the *image plane* or *focal plane*. Then, a point **r** in the 3D space is mapped into the image point $\mathbf{r}_{img}$; which lies in the intersection of *image plane* and the line joining the camera centre **C** with point **r**, as seen in FIGURE 3.1.

[7] Real cameras are built and operated according to the thin lens model, as actual pinhole cameras offer very poor specifications due physical limitations.

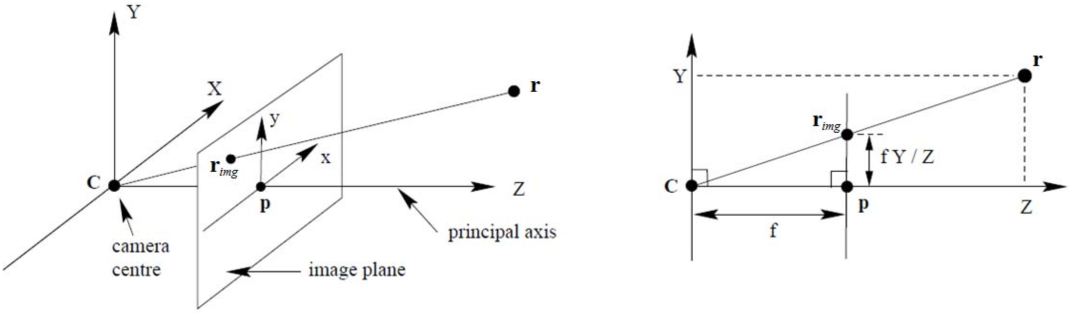


FIGURE 3.1: *Diagram of the pinhole camera model geometry.* **C** *is the camera centre, origin of coordinates and a ray advancing towards Z axis, the principal axis. This ray ends intersecting the plane image in the principal point* **p**.

It is easy to see then that a given point in coordinates $(X, Y, Z)^T$ will be projected into the *image plane* into a set of coordinates according to the following equation:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \mapsto \begin{pmatrix} f\dfrac{X}{Z} \\ f\dfrac{Y}{Z} \end{pmatrix}, \tag{3.2}$$

thus mapping it from 3D to 2D coordinates. In this model, the *camera centre* **C** is also known as *optical centre*, and the ray from it crossing the image plane perpendicularly is known as *principal axis* or *principal ray*, intersecting the plane at the *principal point* **p**.

This *projection P* can be easily described as a linear mapping if the different points are represented by homogeneous vectors denoting their homogeneous coordinates. Thus, the central projection can be annotated as a matrix multiplication:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}; \tag{3.3}$$

which in turn, if we consider points **r** and $\mathbf{r}_{img}$ to be represented from in homogeneous coordinates, with 4 coordinates for the 3D point **r** and 3 coordinates for the 2D projection $\mathbf{r}_{img}$, we can write also as $\mathbf{r}_{img} = P\mathbf{r}$. Now we have that the projection operation is represented by ways of a 3-by-4 *camera projection matrix P*. This projection matrix may also be written as $P = \mathrm{diag}(f, f, 1)$ [I | 0], where $\mathrm{diag}(f, f, 1)$ is a 3-by-3 diagonal matrix, I is a 3-by-3 identity matrix, and the “0” term represents a vector of zeros.

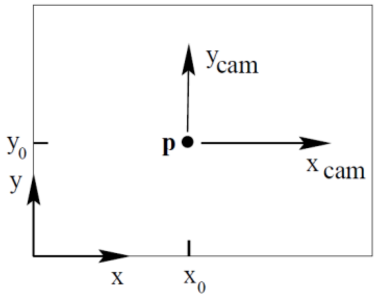


FIGURE 3.2: *Image plane coordinates (x,y) at the origin of the plane, and camera coordinates (x_{cam},y_{cam}), originated a principal point* **p**.

This camera projection matrix assumes that the origin of coordinates of the image plane is at the principal point, but this is not necessarily true. As it is shown in *FIGURE 3.2* , an image plane is usually considered to have the origin of coordinates in a corner, so equation (3.3) is rewritten as follows:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{3.4}$$

in order of include the offset of the principal point **p**, with coordinates (p_x, p_y). Note that the term diag(f, f, 1) in the previously shown decomposition now presents a triangular matrix structure, and it is known as the *camera calibration matrix K.*

$$K = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \tag{3.5}$$

The formulation until now assumes that the coordinates of **r** are noted with respect to a coordinate frame originated in the camera optical centre. If said assumption is removed, the projection operation can be compactly described as:

$$\mathbf{r}_{img} = K\left[I_{3\times3} \mid 0_{3\times1}\right]\mathbf{r}_{cam}, \quad \text{where} \quad \mathbf{r}_{cam} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \tag{3.6}$$

Where $\mathbf{r}_{cam}$ is the homogeneous vector annotating the position of a given point **r** in a 3D Euclidean space in a coordinate frame centered in the optical centre of the camera with the Z axis aligned along the principal axis. This system of coordinates is generally known as the *camera coordinate frame*[8]. This notation comes handy when working with cameras and other elements noted as points in space, as it is common that each element has its own coordinate frame, while there is a shared one, known as *world coordinate frame*. This world coordinate frame is related to the camera coordinate frame by means of a rotation R and a translation $\mathbf{t}$, as seen in FIGURE 3.3.

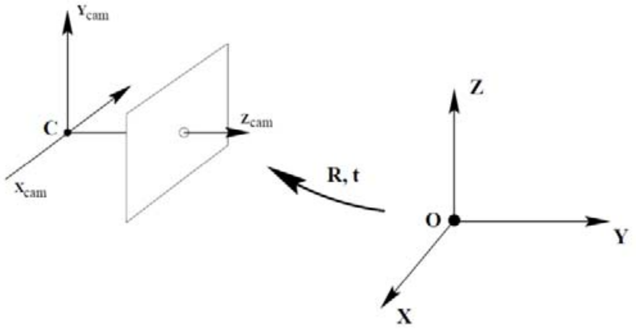


FIGURE 3.3: *Transformation between camera coordinate frame and world coordinate frame, as a rotation R and a translation* **t**.

The projection matrix P for $\mathbf{r}$ in this case would depend on 3 parameters from the K matrix, 3 parameters describing the position of the optical centre **C** through the translation $\mathbf{t}$, and 3 parameters to describe the rotation R. The parameters in K are known as the *internal* or *intrinsic* camera parameters, as they belong to the camera, while those used to describe the position and orientation of the camera, R and $\mathbf{t}$, are known as *extrinsic* or external camera parameters. The main difference relies in the fact that intrinsic camera parameters rarely vary on most artificial vision systems[9], while the extrinsic camera parameters can vary easily as they depend on the pose of the camera. This partition leads to equation (3.7), used to compute camera projection matrix P, which is similar to (3.6), but the 'empty' terms now contain the relevant parameters to model the pose of the camera.

$$\begin{gathered} P = K\left[R \mid \mathbf{t}\right] \\ \mathbf{r}_{img} = P\mathbf{r} \end{gathered} \tag{3.7}$$

Note that the equations up to this point assume that the coordinates in the image plane will be in the same units as the real world coordinates, and that the scale is uniform in both X and Y image axis. But it is known that in vision systems these assumptions are not usually satisfied: camera sensors quantize space in the tiny regions covered by each of their pixels, ignoring which metric unit is used to measure world space; besides, due technical limitations and historical reasons related to the broadcasting industry and practices, is not uncommon to find still today some cameras presenting uneven scaling between the vertical and horizontal axes, i.e. non-square pixels.

[8] Also known as camera frame for short when there is no risk of ambiguities.

[9] Unless the vision system has varying focal length, or can alter other relevant properties of the sensors.

Then, assuming that the pixel count per distance unit for directions x and y are m_x and m_y respectively, K in (3.5) is multiplied by diag(m_x, m_y, 1). Then, including all the elements and terms, the calibration matrix K for camera sensors to use in equation (3.7) can be written as follows:

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_x & y_0 \\ & & 1 \end{bmatrix} \tag{3.8}$$

where $\alpha_x = f{\cdot}m_x$ and $\alpha_y = f{\cdot}m_y$ denote the focal length in pixels for each x and y direction; and $x_0 = m_x{\cdot}p_x$ and $y_0 = m_y{\cdot}p_y$ describe the principal point **p** in pixel dimensions. The final parameter, s, denotes the skew, that is, models the defects of the pixels in terms of presenting an angle between the x and y axis differing from the right angle. This produces a shear transformation, but commonly this angle is considered to be zero[10].

As the resulting coordinates for $\mathbf{r}_{img}$ obtained through equation (3.7) with the calibration matrix from equation (3.8) describe a 2D point in an image plane with dimensions in axis adjusted according to pixel size in homogeneous coordinates, its representation is a 3-element vector. These coordinates then are converted to the plane $Z = 1$, considering the equation:

$$\mathbf{r}_{pix} = (u, \upsilon)^T = \begin{pmatrix} h_x / h_z \\ h_y / h_z \end{pmatrix} \text{ where } \mathbf{r}_{img} = \begin{pmatrix} h_x \\ h_y \\ h_z \end{pmatrix}. \tag{3.9}$$

The pixel coordinates of $\mathbf{r}_{img}$, $\mathbf{r}_{pix}$, represent the ideal projection according to the intrinsic and extrinsic camera parameters contained in matrix P according to (3.7) and (3.8). This assumes that the optical lenses are perfect from a geometrical and projective point of view, matching exactly the thin lens model, and it is not the case. In fact, most of the efforts related to optics development are focused in producing better lenses for photography, i.e., improving the photometrical aspects of the lens, but not the geometrical ones that are interesting for photogrammetry.

One of the aberrations present in real lenses is the *optical distortion*. Optical distortion evidences the fact that the camera sensors cannot be constructed as ideal pinhole cameras: the points in the image are not in a straight line crossing the optical centre toward the origin spatial point which the map. Although there are several types of distortions, the most commonly found distortions are known as *radial distortions*, as they present radial symmetry due to the symmetric nature of the lens. The radial distortion patterns usually observed include the barrel and the cushion distortions, where the scale with respect to the

[10] Although it is also common that automated calibration algorithms that consider skew end producing values infinitesimally different to zero due errors and depending on how the optimization is perfomed.

optical axis decreases and increases respectively, as seen in FIGURE 3.4. Several works have studied radial distortion and how to model it, in (Henrique Brito et al., 2013) the authors review many previous works on radial distortion in order to study and produce a self-calibration algorithm for this optical aberration.

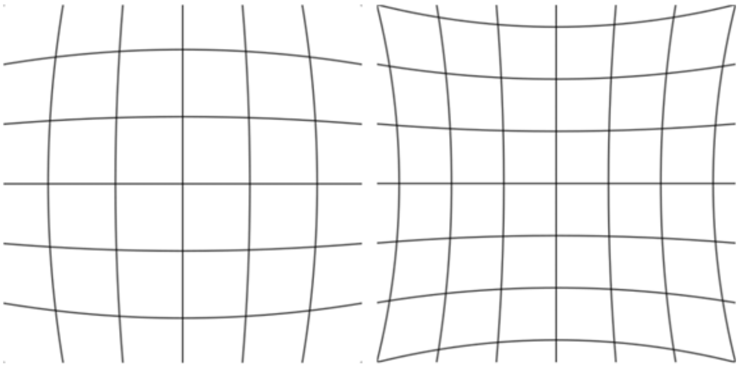


FIGURE 3.4: *Radial distortion patterns.* **Left:** *Barrel distortion.* **Right:** *Pincushion distortion.*

Another distortion pattern found recurrently in computer vision is the *fisheye distortion*. This is found in *fisheye lenses*, which because of being panoramic wide-angle lenses produce not only heavy barrel distortion but also *projective distortion*. Although some of the cameras used during the experiments described in chapters 4 through 6 of this dissertation were wide-angled, they did not present projective distortions to fall under fisheye category, so a simplified radial distortion model like the one presented in (Davison et al., 2004) was used.

3.2.2 Epipolar geometry

The epipolar geometry is commonly defined as the intrinsic projective geometry between two views. It is of interest for the present dissertation, as in several occasions the geometry between two views studying the same scene is considered. It is worth noting that the epipolar geometry itself, being of intrinsic projective nature, depends only on the cameras internal parameters and their relative pose[11].

[11] Note that the term camera is referred in the mathematical sense just detailed, with each camera being w.r.t. one of the multiple views or images. So it is possible to consider the epipolar geometry between different views captured by the same camera.

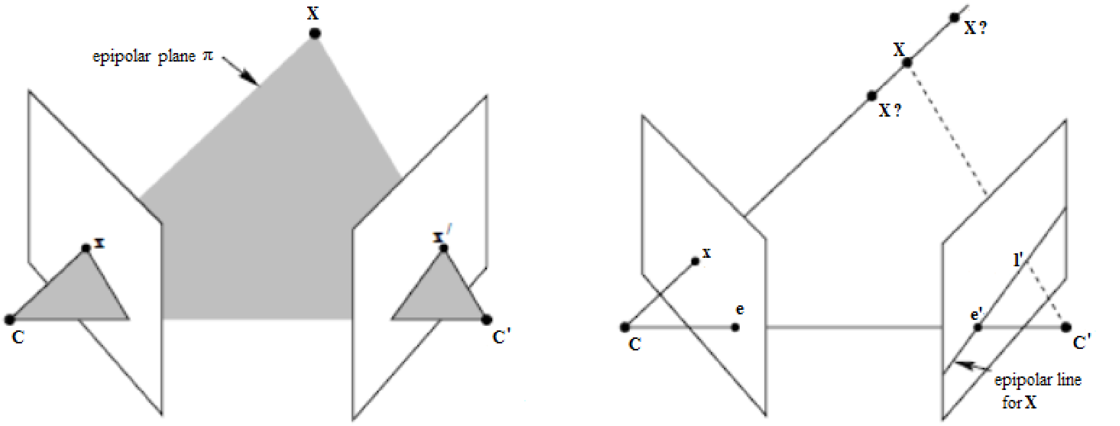


FIGURE 3.5: *Epipolar geometry and point correspondence. The camera centres* **C** *and* **C'** *determine the baseline and the location of the epipoles.* **Left:** *The ray projected from* **C** *through* **x** *points toward* **X**. **Right:** *All the possible projection of* **X** *(found along the ray traced from* **C** *to* **x***) onto the right image lie into the epipolar line* **l'** *originated at the epipole* **e'**.

Thus, assuming two views, with known relative poses, the epipolar geometry studies the geometry of the intersection of the images and a set of *epipolar planes*. These epipolar planes are a set of planes which contain the *baseline*, which is the line that joins the camera centres **C** and **C'**. For any given point **X** in the Euclidean scene observed in both views, an epipolar plane π will be defined between **X** and its projections on the pixels **x** and **x'**. The epipoles, **e** and **e'**, are found at the intersection of the baseline and the images, and are contained by the *epipolar lines* **l** and **l'**, which are the intersections between the views and the epipolar plane π.

The most interesting feature of the epipolar geometry from the point of view of robotic vision and photogrammetry is that for any given pair of views taken with known cameras, with known relative pose, the epipoles can be computed; and this fact allows, for any given point **X** seen as **x** at the image captured in **C**, to determine the epipolar line **l'** where it is to be found on the image captured by **C**'. Thus, searching matching points between different views can be reduced to searching in a linear region instead of full image explorations.

Generally, the epipolar geometry of a given set of cameras or views can be represented in matrix form under the *fundamental matrix F*. This fundamental matrix can be computed in several ways, dependent on the transformation between the camera poses, and requires knowledge of the camera calibration matrix *K* of both cameras/views. A survey of the fundamental methods to compute *F* can be found in (Armangué and Salvi, 2003).

3.3 Feature Based Image Processing: detectors and descriptors

The camera can provide bearing-only information about the different points present on the image. But then, some criteria are needed to choose which points are of interest, as they will eventually be the landmarks composing the map. So they should be meaningful an easy to find and track. There are several techniques to detect and identify these points of interest, also known as features. The most commonly used techniques range from simple approaches, like Harris detector (Harris and Stephens, 1988) or SUSAN (Smith and Brady, 1995), to more complex approaches, like SIFT (Lowe, 2004), SURF (Bay et al., 2006), and FAST (Rosten et al., 2005). More recent techniques include BRIEF (Calonder et al., 2010), BRISK (Leutenegger et al., 2011), FREAK (Alahi et al., 2012) and ORB (Rublee et al., 2011). In (Krig, 2014) the authors offer an extensive review including most of the known feature detectors and descriptors, including discussions on their fitness according different criteria; while in (Kashif et al., 2016) several descriptors are compared in the context of biomedical imaging applications. Still, SIFT remains the most accurate descriptor according to literature consensus, with SURF and BRISK providing a good trade off in terms performance.

The Harris corner detector relies on finding salient corners (Harris and Stephens, 1988), that is, starting from a corner the image intensity will change largely in multiple directions following certain patterns. This can alternatively be formulated by examining the changes of intensity due to shifts in a local window: when the window is centered on corner point, the image intensity will change greatly when the window is shifted in an arbitrary direction. This allows using a second moment matrix, known as the autocorrelation matrix, to evaluate if there is a point of interest or not. The family of Harris-based feature detectors remains popular as it is very effective in structured and artificial environments, and it is used commonly when the problem of tracking or discerning matches for the detected point at posterior times is not a critical aspect to consider.

The SUSAN method is also a corner detector (Smith and Brady, 1995), though it was patented in 1994, which did not help its popularity and spread. It is based on segmenting image features based on local areas of matching intensity, producing bimodal features. The technique itself uses an Univalue Segment Assimilating Nucleus (USAN), which creates areas of similar intensity by comparing pixels within a given radius from a reference pixel. The FAST (Fast Accelerated Segment Test) family of methods (Rosten et al., 2005) is partially derived from SUSAN, as they aim to detect segments based on bimodal segmentation. Instead of dealing with all the pixels inside the circle determined by the reference radius, only those in the Bresenham circle[12] around the point of interest are considered.

The scale-invariant feature transformation, or SIFT (Lowe, 2004), is widely considered to be the most accurate modern feature descriptor, dealing also with the detection. It is based on producing a database of features through a four step process (scale-space extrema detection, keypoint localization, orientation assignment and keypoint descriptor). SIFT features are reported to present the highest discerning rate, presenting lower false positives rates. Still they have had relatively low use in the monocular SLAM field, because of two

[12] An approach to rasterization of curves well-known in computer graphics.

characteristics: SIFT descriptors are time consuming to compute; and the detector used by SIFT tends to find too many points of interest in environments with rich textures, thus dampening performance. These problems have been addressed in some works like (Suzuki et al., 2011) or (Chekhlov et al., 2006), where they use the SIFT as base to create multi-resolution descriptors. Anyway SIFT descriptors are rarely used in monocular SLAM. Another approach taken in relation to SIFT is combining it with principal components analysis (PCA). This PCA technique is an standard technique for dimensionality reduction, thus making SIFT feature vectors smaller, and improving its efficiency, but making it weaker against image blurring, as reported in (Juan and Gwun, 2009).

One of the main problems of SIFT, the computational cost, was addressed by the Speeded Up Robust Features (SURF) detector (Bay et al., 2006). This method, inspired by SIFT uses an integer approximation to the determinant of Hessian blob detector, which can be computed extremely quickly with an integral image. Though reported to be faster than SIFT (Juan and Gwun, 2009), it still produces an excessive amount of features that will lead to association problems between frames, and SIFT is still more accurate and has a better matching rate.

Binary robust independent elementary features (BRIEF) is a binary descriptor based on intensity comparisons. These comparisons are performed between a set of chosen pixels found in a patch around a given interest point. For each set of pixels, a binary value is set into the descriptor depending on which presents higher intensity. This fact makes them resilient to illuminance changes, but weak to rotation and scale variations. An updated version, Binary robust invariant scalable keypoints, or BRISK, solved the orientation problem, defining 2 subsets, with a long distance set of comparisons used to determine the orientation computing local intensity gradients. This allowed to rotate the patches after determining the orientation, and using a set of short distances comparison analogously to the BRIEF descriptor. The Fast Retina Keypoint descriptor, FREAK, is also partially derived from BRIEF and BRISK, introducing biologically inspired concepts. A retinal sampling pattern is used to compute binary values through intensitiy comparison, like BRISK, but introducing symmetric fields instead of long distance comparison.

The ORB (Oriented FAST and rotated BRIEF), combined both FAST detector and BRIEF descriptor to produce an alternative to SIFT. While in terms of accuracy it is not able to match SIFT, ORB is about an order of magnitude faster than SURF and 2 orders of magnitude than SIFT (Rublee et al., 2011). This fact has made ORB popular, and a successful ORB based monocular SLAM method is described in (Mur-Artal et al., 2015).

3.4 Delayed Inverse-Depth feature initialization monocular SLAM

This section describes previous works by the research group Vision and Intelligent Systems (VIS) in the field of monocular SLAM with delayed feature initialization, with works like (Munguia and Grau, 2007a), (Munguia and Grau, 2007b), and (Munguia and Grau, 2012).

These works constitute the base upon the developments presented on this thesis started. The diagram on FIGURE 3.6 shows a block schema describing the different subprocesses performed and the different flows of information between said processes. The most relevant features of these processes and data flows will be described in the subsequent subsections, with focus on the mathematical basis that will be required on further chapters.

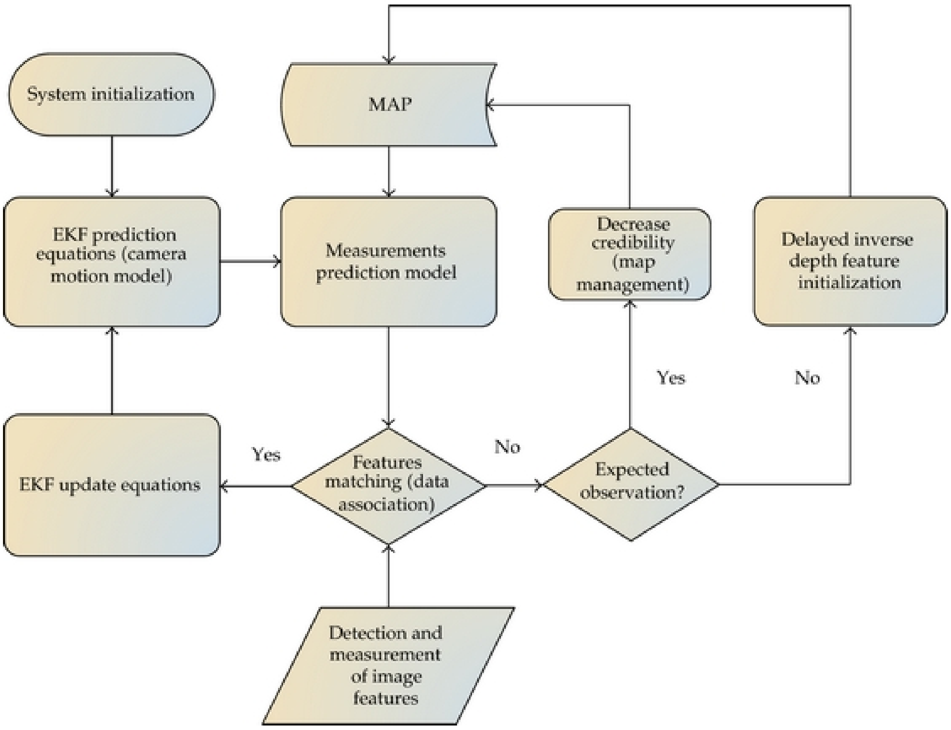

System initialization
MAP
EKF prediction equations (camera motion model)
Measurements prediction model
Decrease credibility (map management)
Delayed inverse depth feature initialization
Yes
No
EKF update equations
Yes
Features matching (data association)
No
Expected observation?
Detection and measurement of image features

FIGURE 3.6: *Block diagram of the delayed inverse-depth feature initialization Monocular SLAM, detailing the different processes and how they relate to build the MAP* **m** *though iterative EKF filtering.*

3.4.1 System Parameterization

The system state is denoted by an augmented state vector, as described in Chapter 2:

$$\hat{\mathbf{x}} = \left[\hat{\mathbf{x}}_{\mathbf{v}}, \hat{\mathbf{y}}_{\mathbf{1}}, \ldots \hat{\mathbf{y}}_{\mathbf{n}}\right]^{T} \tag{3.10}$$

where $\hat{\mathbf{x}}_{\mathrm{V}}$ represents the state of a free robotic camera moving in any direction in $\mathbb{R}^3$, with 6 DoF, and the map is denoted as a set of landmarks represented by feature points in the 3-dimensional environment. These feature points are parametrized according the inverse-depth model, described in section 2.7.1.

The state of the robotic camera, $\hat{\mathbf{x}}_v$, can be decomposed into pose, including both position and orientation, and the instantaneous speeds, both linear and angular, as seen in equation (3.11)[13]:

$$\hat{\mathbf{x}}_v = \begin{bmatrix} \mathbf{r}^{WC} & \mathbf{q}^{WC} & \mathbf{v}^{W} & \boldsymbol{\omega}^{W} \end{bmatrix}^T , \tag{3.11}$$

where

$$\mathbf{r}^{WC} = \begin{bmatrix} x_v & y_v & z_v \end{bmatrix}^T \tag{3.12}$$

$\mathbf{r}^{CW}$ denotes the optical centre of the robotic camera in Cartesian coordinates and

$$\mathbf{q}^{WC} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix}^T \tag{3.13}$$

$\mathbf{q}^{CW}$ denotes the orientation of the camera with respect to the global reference frame using a unit quaternion form parametrization. Quaternions, although they do not present the intuitiveness of Euler angles in terms of interpretation, they present a composition formulation simpler, while avoiding the gimbal lock. When compared with rotation matrices, they present better numerical stability and efficiency in terms of parametrization, being these features critical for filtering techniques like the EKF.

The terms $\boldsymbol{\omega}^W$ and $\mathbf{v}^W$ in equation (3.11) denote linear and angular velocities respectively, descomposed according to equation (3.14):

$$\begin{aligned} \mathbf{v}^{W} &= \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}^T \\ \boldsymbol{\omega}^{W} &= \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T \end{aligned} . \tag{3.14}$$

In these equations the speeds are denoted w.r.t. global reference coordinates, even though they are applied to the robot camera as local combination of translation and location.

The rest of the augmented state vector is composed of features, which are annotated according to equation (3.15). Note that said features are parametrized under the inverse-depth parametrization (discussed in section 2.7.1, equation 2.27 and 2.28). In FIGURE 3.7 the features codification according to this method and the structure of the state initialization (described in the next section) are illustrated.

$$y_i = \begin{bmatrix} x_i & y_i & z_i & \phi_i & \theta_i & \rho_i \end{bmatrix}^T \tag{3.15}$$

[13] The superscripts are used to denote the reference frame relevant, so the *W* superscript denotes the *world* reference, thus global coordinates, while *C* will be used to denote the camera reference frame. When the superscripts denote multiple reference frames, they denote a transformation (be it rotation, translation, or full homogeneous transformation) from the first reference frame to the ending frame.

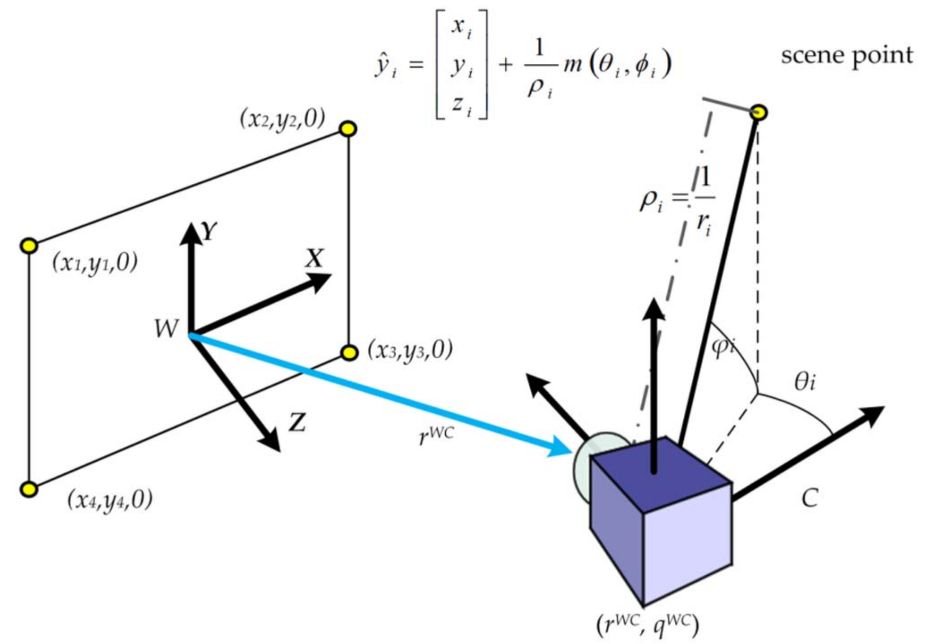


FIGURE 3.7:*Inverse depth parametrization of points, with detail of the set of a priori known coplanar points used for system initialization.*

3.4.2 System Initialization

Pure monocular SLAM approaches largely ignore the scale problem and focus on producing scaleless reconstructions, even the most advanced and recent techniques, like (Engel et al., 2014) and (Mur-Artal et al., 2015). Still, obtaining the metric scale of the map is necessary if the map reconstructed is to be used later in localization and navigation applications. Monocular SLAM approaches generally ignore the scale of the observed world given the nature of the camera, which as a bearing-only sensor, only produces angular data, but no distance information. Thus, additional information from other sensors or previous knowledge about the dimensions of a given reference to be found is required to retrieve the scale of the world.

The system metric initialization process is solved analogously to the n-point perspective problem (PnP) (Chatterjee and Roychowdhury, 2000). In PnP the challenge relies on finding the position and orientation of a camera with respect to a set of n known points. A solution for the case with 4 known points (thus P4P problem) used, solving it completely under the assumption that the four points are coplanar. A similar approach is used to solve the initialization problem.

Thus, to initialize the system, a set of 4 coplanar known points are used. These points can be present in any object (present a priori or specifically added) as long as 2 criteria are met:

- The exact geometrical relations between the coplanar point are known.

- The points are identified by a process which guarantees that there is zero probabilities of mismatch or false positives.

For this given set of points and a calibrated camera, the extrinsic parameters R^{CW} (world to camera rotation) and **t** (translation vector from world coordinate origin to camera optical centre) are computed. As discussed previously in this chapter, considering a camera as *calibrated* implies knowing the intrinsic parameters of the camera, namely: focal distance *f*, the optical centre (i_0,j_0), and the radial distortion parameters k_1 … k_n. The 4 coplanar points, with coordinates (x_i,y_i,0) for *i* = [1 .. 4] are assumed to lie at the same distance from the camera projection plane, in a plane with world coordinates z = 0 , thus putting the world coordinates origin in the same plane.

The method used is based on (Ganapathy, 1984), thus a system of linear equations is built, as described in equation (3.16), with an unknown vector **b**:

$$\begin{bmatrix} x_1 f & y_1 f & 0 & 0 & -i_1 x_1 & -i_1 y_1 & f & 0 \\ 0 & 0 & x_1 f & y_1 f & -j_1 x_1 & -j_1 y_1 & 0 & f \\ x_2 f & y_2 f & 0 & 0 & -i_2 x_2 & -i_2 y_2 & f & 0 \\ 0 & 0 & x_2 f & y_2 f & -j_2 x_2 & -j_2 y_2 & 0 & f \\ x_3 f & y_3 f & 0 & 0 & -i_3 x_3 & -i_3 y_3 & f & 0 \\ 0 & 0 & x_3 f & y_3 f & -j_3 x_3 & -j_3 y_3 & 0 & f \\ x_4 f & y_4 f & 0 & 0 & -i_4 x_4 & -i_4 y_4 & f & 0 \\ 0 & 0 & x_4 f & y_4 f & -j_4 x_4 & -j_4 y_4 & 0 & f \end{bmatrix} \mathbf{b} = \begin{bmatrix} i_1 \\ j_1 \\ i_2 \\ j_2 \\ i_3 \\ j_3 \\ i_4 \\ j_4 \end{bmatrix}, \tag{3.16}$$

where

$$\mathbf{b} = \begin{bmatrix} \frac{r_{11}}{t_3} & \frac{r_{12}}{t_3} & \frac{r_{21}}{t_3} & \frac{r_{22}}{t_3} & \frac{r_{31}}{t_3} & \frac{r_{32}}{t_3} & \frac{t_1}{t_3} & \frac{t_2}{t_3} \end{bmatrix}^T . \tag{3.17}$$

This linear equation system is solved for

$$\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \end{bmatrix}^T ; \tag{3.18}$$

then t_3 is determined as

$$t_3 = \sqrt{\frac{f^2}{b_1^2 + b_3^2 + b_5^2}} . \tag{3.19}$$

Once a solution for t_3 is found, solutions for R^{CV} and **t** are computed according to equations (3.20) and (3.21).

$$R^{CW} = \begin{bmatrix} t_3 b_1 & t_3 b_2 & (R_{21}R_{32} - R_{31}R_{22}) \\ t_3 b_3 & t_3 b_4 & (R_{31}R_{12} - R_{11}R_{32}) \\ t_3 b_5 & t_3 b_6 & (R_{11}R_{22} - R_{21}R_{12}) \end{bmatrix} \tag{3.20}$$

$$\mathbf{t} = \begin{bmatrix} t_3 b_7 & t_3 b_8 & t_3 \end{bmatrix}^T \tag{3.21}$$

The terms R_{ij} found at the third column of R^{CW} refer to the elements of the same R^{CW} matrix. Once the extrinsic parameters of the camera, that is, the camera pose with respect to the world coordinates (which have been set with the origin in the plane defined by the coplanar points), the augmented state is initialized prior to starting the EKF filtering.

This state initialization follows equation (3.22), with estimated state $\hat{\mathbf{x}}_{ini}$ containing the initial pose and velocities of the camera and the four points (x_i,y_i,0) initialized as $\hat{\mathbf{y}}_i$.

$$\hat{\mathbf{x}}_{ini} = \begin{bmatrix} \mathbf{r}^{WC}_{ini} & \mathbf{q}^{WC}_{ini} & \mathbf{v}^{W}_{ini} & \boldsymbol{\omega}^{W}_{ini} & \hat{\mathbf{y}}_1 & \hat{\mathbf{y}}_2 & \hat{\mathbf{y}}_3 & \hat{\mathbf{y}}_4 \end{bmatrix}^T \tag{3.22}$$

$\mathbf{r}^{WC}{}_{ini}$ will be the same vector **t** describing the camera extrinsic translation, while $\mathbf{q}^{WC}{}_{ini}$ will be the quaternion obtained[14] from transposed R^{CW}, from equation (3.20), and the camera will be considered to be initially inmobile, all as describe in equation (3.23):

$$\mathbf{r}^{WC}_{ini} = t\,,\quad \mathbf{q}^{WC}_{ini} = q\left(\left(R^{CW}\right)^T\right),\quad \mathbf{v}^{W}_{ini} = \left[0_{3\times1}\right],\quad \boldsymbol{\omega}^{W}_{ini} = \left[0_{3\times1}\right] \quad . \tag{3.23}$$

As the map part contains the initial features $\hat{\mathbf{y}}_i$, each of these are initialized following equations (3.24) and (3.25), where [g_1 g_2 g_3] are the coordinates for the landmark w.r.t. an origin of coordinates translated from the plane where the coplanar points lie to the initial position of the camera:

$$\hat{\mathbf{y}}_i = \begin{bmatrix} \mathbf{r}^{WC}_{ini} & \arctan_2\left(g_1, g_3\right) & \arctan_2\left(-g_2, \sqrt{g_1^2 + g_3^2}\right) & \dfrac{1}{\left\|\mathbf{r}^{WC}_{ini}\right\|} \end{bmatrix}^T , \tag{3.24}$$

$$\begin{bmatrix} g_1 & g_2 & g_3 \end{bmatrix} = \begin{bmatrix} x_i & y_i & 0 \end{bmatrix} - \mathbf{r}^{WC}_{ini} . \tag{3.25}$$

Finally the initial covariance matrix is set. Note that although it is entirely possible to fill it with zeros, it generally provides better results to try to characterize the initial belief, thus normally arbitratry small values are chosen, according to the knowledge about and experience with the system:

$$P_{ini} = \left[\varepsilon_{37x37}\right]. \tag{3.26}$$

3.4.3 Prediction Step: camera model and prediction equations

Following the EKF SLAM schema described in chapter 2, each iteration starts with the prediction or time-update step. An unconstrained prediction model for a constant-acceleration camera motion, defined in equation (3.27) is applied to the $\hat{\mathbf{x}}_\mathrm{v}$ part of the state vector:

[14] See equation (V.2) at the annexes for quaternion from a rotation matrix.

$$f_v = \begin{bmatrix} \mathbf{r}_{k+1}^{WC} \\ \mathbf{q}_{k+1}^{WC} \\ \mathbf{v}_{k+1}^{W} \\ \boldsymbol{\omega}_{k+1}^{W} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k^{WC} + \left(\mathbf{v}_k^W + \mathbf{V}_k^W\right)\Delta t \\ \mathbf{q}_{k+1}^{WC} \times \mathbf{q}\left(\left(\boldsymbol{\omega}_k^W + \Omega_k^W\right)\Delta t\right) \\ \mathbf{v}_k^W + \mathbf{V}_k^W \\ \boldsymbol{\omega}_k^W + \Omega_k^W \end{bmatrix} \tag{3.27}$$

Δt is the time increment between each k instant, defined by the rate of the sensors, in this case, the framerate of the camera. The velocities $\boldsymbol{\omega}^W$and $\mathbf{v}^W$are modied at each step k by an input vector u, which contains linear and angular accelerations $a^W$and α^W, each them being a zero-mean Gaussian random process, as seen in equation (3.28). The final orientation of the camera is obtained by quaternion multiplication of the current orientation with the quaternion $\mathbf{q}((\boldsymbol{\omega}^W + \Omega^W)\ \Delta t)$, derived from converting the orientation increment vector into quaternion[15].

$$u = \begin{bmatrix} \mathbf{V}^W \\ \Omega^W \end{bmatrix} = \begin{bmatrix} a^W \Delta t \\ \alpha^W \Delta t \end{bmatrix} \tag{3.28}$$

These unknown linear and angular velocities entries V^W and Ω^W are introduced into the system covariance by the process noise covariance matrix U:

$$U = \begin{bmatrix} \left(\sigma_{\mathrm{V}}\Delta t\right)^2 I_{3\times 3} & 0_{3\times 3} \\ 0_{3\times 3} & \left(\sigma_{\Omega}\Delta t\right)^2 I_{3\times 3} \end{bmatrix}. \tag{3.29}$$

Thus, after defining the time-update model for the camera movement at equation (3.27), and the uncertainty characterization of the process at equation (3.29), the augmented state vector and the covariance matrix can be updated following equations (3.30) and (3.31):

$$\hat{\mathbf{x}}_{k+1} = \begin{bmatrix} f_v(\hat{\mathbf{x}}_v) \\ \hat{\mathbf{y}}_1 \\ \vdots \\ \hat{\mathbf{y}}_n \end{bmatrix}, \tag{3.30}$$

$$P_{k+1} = \nabla F_x P_k \nabla F_x^T + \nabla F_u Q \nabla F_u^T \ . \tag{3.31}$$

Notice that the map part **m** in equation (3.30) follows the static world assumption, i.e., the landmarks do not move thus they present no variation at the time-update step, as detailed in section 2.4.2. This means that the Jacobians ∇F_x and ∇F_u w.r.t. the process and the input signal must consider the partitioned state, as shown in equation (3.32), with n being the size of the parametrization of the landmarks present in the map **m**:

[15] See equation (V.1) at annexes for converting orientation vector to quaternion.

$$\nabla F_x = \begin{bmatrix} \frac{\partial f_v}{\partial \hat{\mathbf{x}}_v} & 0_{13\times n} \\ 0_{n\times 13} & I_{n\times n} \end{bmatrix}, \quad \nabla F_u = \begin{bmatrix} \frac{\partial f_v}{\partial u} & 0_{13\times n} \\ 0_{n\times 6} & I_{n\times n} \end{bmatrix}, \quad Q = \begin{bmatrix} U & 0_{6\times n} \\ 0_{n\times 6} & I_{n\times n} \end{bmatrix}. \tag{3.32}$$

The main component of the Jacobian of the camera motion model ∇*F*$_x$, δ*f*$_v$/δ*fv* is in turn built as a block matrix, following the structure in equation (3.33):

$$\frac{\partial f_v}{\partial \hat{\mathbf{x}}_v} = \begin{bmatrix} I_{3\times 3} & 0 & \frac{\partial \mathbf{r}^{WC}}{\partial \mathbf{v}^{W}} & 0 \\ 0 & \frac{\partial \mathbf{q}^{WC}_{k+1}}{\partial \mathbf{q}^{WC}_{k}} & 0 & \frac{\partial \mathbf{q}^{WC}_{k+1}}{\partial \boldsymbol{\omega}^{W}_{k+1}} \\ 0 & 0 & I_{3\times 3} & 0 \\ 0 & 0 & 0 & I_{3\times 3} \end{bmatrix}, \tag{3.33}$$

with the block components detailed in equations (3.34), (3.35), and (3.38). Equation (3.34) describes the derivatives of the position $\mathbf{r}^{WC}$ w.r.t. to the linear velocities:

$$\frac{\partial \mathbf{r}^{WC}}{\partial \mathbf{v}^{W}} = \begin{bmatrix} \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & \Delta t \end{bmatrix}. \tag{3.34}$$

Equation (3.35) details the derivatives of the orientation quaternion with respect to its previous state:

$$\frac{\partial \mathbf{q}^{WC}_{k+1}}{\partial \mathbf{q}^{WC}_{k}} = \frac{\partial q_3}{\partial q_2} = \begin{bmatrix} q_R & -q_X & -q_Y & -q_Z \\ q_X & q_R & q_Z & -q_Y \\ q_Y & -q_Z & q_R & q_X \\ q_Z & q_Y & -q_X & q_R \end{bmatrix}, \tag{3.35}$$

where

$$\begin{bmatrix} q_R \\ q_X \\ q_Y \\ q_Z \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\mu_x \\ \sin(\theta/2)\mu_y \\ \sin(\theta/2)\mu_z \end{bmatrix} \tag{3.36}$$

and

$$\theta = \left\|\omega^W \Delta t\right\|, \ \mu = \frac{\omega^W \Delta t}{\left\|\omega^W \Delta t\right\|}. \tag{3.37}$$

Finally, equation (3.38) denotes the derivatives of the orientation (as a quaternion) w.r.t to the angular velocities in the Brownian motion model:

$$\frac{\partial \mathbf{q}_{k+1}^{WC}}{\partial \boldsymbol{\omega}_{k+1}^{W}} = \frac{\partial \mathbf{q}_{k+1}^{WC}}{\partial \mathbf{q}\left(\left(\boldsymbol{\omega}_k^W + \Omega_k^W\right)\Delta t\right)} \frac{\partial \mathbf{q}\left(\left(\boldsymbol{\omega}_k^W + \Omega_k^W\right)\Delta t\right)}{\partial \boldsymbol{\omega}_{k+1}^{WC}}, \tag{3.38}$$

where

$$\frac{\partial \mathbf{q}_{k+1}^{WC}}{\partial \mathbf{q}\left(\left(\boldsymbol{\omega}_k^W + \Omega_k^W\right)\Delta t\right)} = \frac{\partial q_3}{\partial q_1} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \tag{3.39}$$

The term ($\delta\mathbf{q}(\boldsymbol{\omega}^W{}_k + \boldsymbol{\Omega}^W{}_k)\ \Delta t\ )/\delta\boldsymbol{\omega}^W{}_{k+1}$ can be computed from the conversion formula from a rotation vector to quaternion representation. This conversion is noted in equation (3.40):

$$\mathbf{q}(\omega) = \begin{bmatrix} \cos\frac{\theta}{2} & \sin\frac{\theta}{2}\frac{\boldsymbol{\omega}^T}{\theta} \end{bmatrix}^T, \text{ where } \theta = \|\omega\| . \tag{3.40}$$

The derivatives for this expression can be then separated by components,

$$\frac{\partial \mathbf{q}\left(\left(\boldsymbol{\omega}_k^W + \Omega_k^W\right)\Delta t\right)}{\partial \boldsymbol{\omega}_{k+1}^{WC}} = \begin{bmatrix} \frac{\partial q_0\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_x^{WC}} & \frac{\partial q_0\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_y^{WC}} & \frac{\partial q_0\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_z^{WC}} \\ \frac{\partial q_1\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_x^{WC}} & \frac{\partial q_1\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_y^{WC}} & \frac{\partial q_1\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_z^{WC}} \\ \frac{\partial q_2\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_x^{WC}} & \frac{\partial q_2\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_y^{WC}} & \frac{\partial q_2\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_z^{WC}} \\ \frac{\partial q_3\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_x^{WC}} & \frac{\partial q_3\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_y^{WC}} & \frac{\partial q_3\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_z^{WC}} \end{bmatrix}, \tag{3.41}$$

where the components are computed according to:

$$\frac{\partial q_0\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_i^{WC}} = -\frac{\Delta t}{2}\frac{\omega_i}{\theta}\sin\left(\theta\frac{\Delta t}{2}\right), \tag{3.42}$$

$$\frac{\partial q_i\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_i^{WC}}\Big|_{i\neq 0} = \frac{\Delta t}{2}\left(\frac{\omega_i}{\theta}\right)^2 \cos\left(\theta\frac{\Delta t}{2}\right) + \frac{1}{\theta}\left(1 - \left(\frac{\omega_i}{\theta}\right)^2\right)\sin\left(\theta\frac{\Delta t}{2}\right), \tag{3.43}$$

$$\frac{\partial q_i\left(\boldsymbol{\omega}_k^W \Delta t\right)}{\partial \boldsymbol{\omega}_j^{WC}}\Big|_{i\neq 0, j\neq i} = \frac{\Delta t}{2}\frac{\omega_i\omega_j}{\theta^2}\cos\left(\theta\frac{\Delta t}{2}\right) - \frac{1}{\theta}\sin\left(\theta\frac{\Delta t}{2}\right). \tag{3.44}$$

Finally, $\delta f_v/\delta u$ denotes the Jacobian of f_v w.r.t the input noise process Q, built with blocks already computed, namely equations (3.34) and (3.38):

$$\frac{\partial f_v}{\partial u} = \begin{bmatrix} \frac{\partial \mathbf{r}^{WC}}{\partial \mathbf{v}^W} & 0 \\ 0 & \frac{\partial \mathbf{q}_{k+1}^{WC}}{\partial \boldsymbol{\omega}_{k+1}^W} \\ I_{3\times 3} & 0 \\ 0 & I_{3\times 3} \end{bmatrix}. \tag{3.45}$$

3.4.4 Measurement prediction model

As it was discussed in Chapter 2, one of the procedures which an EKF methodology will perform during the update step is the measurement prediction. Using data from the augmented state posterior to the time-update step, the measurement prediction model forecasts the distorted pixel coordinates in the image plane for all the landmarks in $\hat{\mathbf{x}}_k$:

$$\begin{bmatrix} u_d \\ \upsilon_d \end{bmatrix} = h_i\left(\hat{x}_v, \hat{y}_i\right). \tag{3.46}$$

Thus, for each landmark $\hat{\mathbf{y}}_i$ the correspondent (u_d,υ_d) coordinates are found with the following process: the observation model of any given landmark, parametrized according to section IDP notation in section 2.4.2, defines a ray h^c w.r.t. the camera frame:

$$h^c = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = R^{CW}\left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right) - \mathbf{r}^{WC}\right) \tag{3.47}$$

where R^{CW} is the transformation from global reference to camera reference frame, and $\mathbf{m}$ is the director vector described in equation (2.28), within the IDP parametrization (see equation (2.27)). This vector representation h^c of the ray between the landmark and the camera optical centre is projected into the camera coordinates appliying the following equation:

$$\begin{bmatrix} u \\ \upsilon \end{bmatrix} = \begin{bmatrix} \frac{h_x}{h_z} \\ \frac{h_y}{h_z} \end{bmatrix}, \tag{3.48}$$

which produces the pixel coordinates w.r.t. the ideal camera optical centre. These coordinates in turn are converted into actual undistorted pixel coordinates applying the offset of the actual camera centre:

$$\begin{bmatrix} u_u \\ \upsilon_u \end{bmatrix} = \begin{bmatrix} u_0 - fu \\ \upsilon_0 - f\upsilon \end{bmatrix}, \tag{3.49}$$

producing (u_u,υ_u), which in turn are applied a radial distortion model to obtain the distorted pixel coodinates. Equation (3.50) describes the applied distortion model, originally proposed in (Davison et al., 2004), using the distortion coefficient k_l:

$$\begin{bmatrix} u_d \\ \upsilon_d \end{bmatrix} = \begin{bmatrix} \dfrac{u_u - u_0}{\sqrt{1-2k_1 r^2}} + u_0 \\ \dfrac{\upsilon_u - \upsilon_0}{\sqrt{1-2k_1 r^2}} + \upsilon_0 \end{bmatrix} \quad , \; r = \sqrt{(u_u - \upsilon_0)^2 + (\upsilon_u - \upsilon_0)} \quad . \tag{3.50}$$

If the distorted pixel coordinates for the prediction of a given landmark $\hat{y}_i$ fall within the image, that is, both u_u and υ_u have values between 0 and the image height and width respectively, then a match for the landmark prediction will be searched. Note that if the landmark falls within the image, but its u_u,υ_u coordinates are near the image edges, the search will be skipped due the possibility that the actual measured landmark will lay out of the image. This is especially risky given how the matching process is performed, described in the next section, as the chances of false positives increase in these cases.

3.4.5 Feature Matching and measurement update

For the features found in the image, within a safe distance from the edges, the data association problem is solved, that is, the actual pixel coordinates for landmark $\hat{\mathbf{y}}_i$ at instant k are to be found. These coordinates constitute the measurement matched to the predicted landmark, corresponding to one of i parts of the measurement vector $\mathbf{z}_k$ found in equation (2.15) and described in section 2.4.2. Note that the predicted u_u,υ_u coordinates are also present in the same equation, being the term $\mathbf{h}(\hat{\mathbf{x}}_{k|k-1},\hat{\mathbf{m}}_{k-1})$.

The strategy to solve the data association is based on and active search technique (Davison and Murray, 2002), which is widely described and discussed in Chapter 4. To summarize it, a matching point is to be found in a search area defined as a function of the innovation of the covariance matrix, as shown in equations (4.4) to (4.6). In order to determine which of the pixels lying in the area is the match, a likehood function is applied between the aspect of a patch captured around the landmark when it was first seen, and current image appearance, as described in section 4.3.

Once the data association problem is fully solved, the measurement update step takes places. The Kalman innovation covariance, see equation (3.51), required to apply the active search technique, is computed using the Jacobian $\nabla \mathrm{H}_k$ of the observation model, described in the previous section. This Jacobian is decomposed as the set of partial derivatives noted in equation (3.52), and applied to each one of the $\hat{\mathbf{y}}_i$ landmarks.

$$S = \nabla H \, P_{k+1} \nabla H^{T} + R \tag{3.51}$$

$$\nabla H_i = \begin{bmatrix} \frac{\partial h_i}{\partial \hat{\mathbf{x}}_v} & \cdots & 0_{2\times 6} & \cdots & \frac{\partial h_i}{\partial \hat{\mathbf{y}}_i} & \cdots & 0_{2\times 6} & \cdots \end{bmatrix} \tag{3.52}$$

$\delta h_i/\delta\hat{\mathbf{x}}_v$ contains the derivatives of the measurement prediction model for the landmark with i index w.r.t. the camera state, and $\delta h_i/\delta\hat{\mathbf{y}}_i$ the derivatives w.r.t. the same landmark. Note that for all the other landmarks the Jacobian $\nabla\mathrm{H}_i$ is zero, so for each one of them, $\nabla\mathrm{H}_i$ will contains a submatrix of zeros of rank 2×6 (the measurement contains 2 coordinates and the IDP parametrization presents parameters).

After the data association is solved the procedure described in 2.4.2 can be fully applied, so the Kalman innovation or residuals are computed:

$$\mathbf{g} = \mathbf{z} - H\left(\hat{\mathbf{x}}\right); \tag{3.53}$$

and, as the innovation covariance was previously solved, the Kalman gain can be calculated:

$$W = P_{k+1} \nabla H^{T} S^{-1}. \tag{3.54}$$

With all the data available, the measurement-update, or correction step, can be completed by updating both the state and the covariance matrix, with equations (3.55) and (3.56).

$$\hat{\mathbf{x}}_{\mathrm{k}} = \hat{\mathbf{x}}_{\mathrm{k+1}} + \mathrm{W}\mathbf{g} \tag{3.55}$$

$$P_k = P_{k+1} - WSW^{T} \tag{3.56}$$

3.4.6 Delayed Inverse-Depth Feature Initialization

The key difference between the most common approaches to SLAM and the delayed approach discussed lays in how the feature depth is initialized. In order to initialize a feature, it must have been detected and tracked during enough frames to achieve significant parallax so that the depth can be estimated.

In order to achieve this, points are detected using the Harris detector in areas where there are no landmarks or candidates points. When a good candidate λ_i is detected, it is stored into a database, with the following parametrization:

$$\lambda_i = \left(r_\lambda, \sigma_r, q_\lambda, u_1, v_1\right), \tag{3.57}$$

where r_λ is the optical centre of the camera r^{WC} when the point is detected

$$r_\lambda = \left(x_1, y_1, z_1\right) = \left(x_v, y_v, z_v\right); \tag{3.58}$$

σ_r denotes the covariances of r^{WC}, taken from P_k:

$$\sigma_r = \left(\sigma_{x1}, \sigma_{y1}, \sigma_{z1}\right) = \left(P_k(2,2), P_k(2,2), P_k(3,3)\right); \tag{3.59}$$

q_λ is the same quaternion $\mathbf{q}^{WC}$ denoting the orientation of the camera, with σ_r noting its covariances

$$\sigma_q = \left(\sigma_{1q1}, \sigma_{1q2}, \sigma_{1q3}, \sigma_{1q4}\right) = \left(P_k(4,4), P_k(5,5), P_k(6,6), P_k(7,7)\right); \tag{3.60}$$

and u_1, v_1 denotes the pixel coordinates of the landmark where the candidate λ_i was detected for the first time. Those candidates stored in the database are tracked between frames, until there is need for new landmarks to be introduced into the EKF filter. Many of the candidates λ_i will not be introduced into the filter, as they will be pruned during the tracking following several criteria.

For a candidate λ_i to become a new landmark $\hat{\mathbf{y}}_{new}$, it must be tracked until achieving parallax α_{min}. To compute this parallax, the data stored in λ_i and the current state of the camera is used. Notice that as λ_i contains r_λ and the posterior state holds $r^{WC}{}_k$, it is possible to estimate the baseline of the displacement b, thus the parallax estimation results reminiscent of epipolar geometry, as observed in FIGURE 3.8.

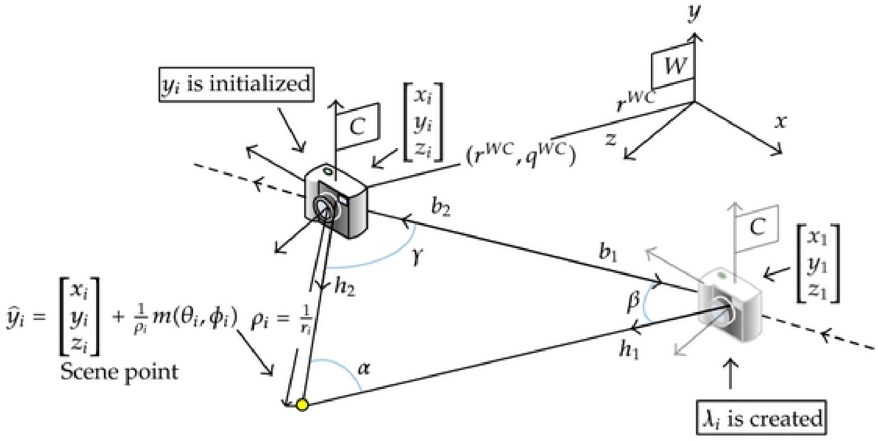

y_i is initialized
C
$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}$
(r^{WC}, q^{WC})
y
W
r^{WC}
x
z
b_2
γ
b_1
β
h_2
h_1
C
$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$
$\hat{y}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} m(\theta_i, \phi_i)$ $\rho_i = \frac{1}{r_i}$
Scene point
α
λ_i is created

FIGURE 3.8: *Feature initialization process diagram, with parallax estimation schema.*

So, for each candidate λ_i the parallax α is computed when a match z_i in a new frame is detected, as

$$\alpha = \pi - \left(\beta + \gamma\right) \tag{3.61}$$

where angles β and γ are determined form the respective director vectors h_1 and h_2 , and the baseline director vectors b_1 and b_2 (seen in FIGURE 3.8). So, the director vector h_1 in equation (3.62)

$$\beta = \cos^{-1}\left(\frac{h_1 \cdot b_1}{\|h_1\| \cdot \|b_1\|}\right) \tag{3.62}$$

is the vector annotating the projective ray $h_1 = [h_{1x}\ h_{1y}\ h_{1z}]$ in Euclidean world coordinates. This ray is computed from the camera position and the pixel coordinates stored in λ_i, as seen in equation (3.63): applying the transformation $R^{WC}(\mathbf{q}_\lambda)$ from camera reference (when λ_i was detected) into the global frame reference to the directional vector between the camera optical centre and the pixel coordinates, $\mathbf{h}^C{}_1(u_{1u}, \upsilon_{1u})$.

$$h_1 = R^{WC}\left(q_\lambda\right) h_1^C\left(u_{1u}, \upsilon_{1u}\right) \tag{3.63}$$

This $\mathbf{h}^C{}_1(u_{1u}, \upsilon_{1u})$ vector is computed per equation (3.64)

$$h_1^C\left(u_{1u}, \upsilon_{1u}\right) = \begin{bmatrix} \dfrac{u_0 - u_{1u}}{f} & \dfrac{\upsilon_0 - \upsilon_{1u}}{f} & 1 \end{bmatrix}; \tag{3.64}$$

which uses the undistorted pixel coordinates (u_{1u}, υ_{1u}), These undistorted pixel coordinates are obtained by applying the *undistortion* model (the inverse of the distortion model) at equation (3.65) to the pixel coordinates stored in λ_i.

$$\begin{bmatrix} u_{1u} \\ \upsilon_{1u} \end{bmatrix} = \begin{bmatrix} \dfrac{u_1 - u_0}{\sqrt{1 - 2k_1 r^2}} + u_0 \\ \dfrac{\upsilon_1 - \upsilon_0}{\sqrt{1 - 2k_1 r^2}} + \upsilon_0 \end{bmatrix}, \quad r = \sqrt{\left(u_1 - u_0\right)^2 + \left(\upsilon_1 - \upsilon_0\right)^2} \tag{3.65}$$

The baseline director vector, b_1, is computed as the difference between the current position of the camera optical centre, $\mathbf{r}^{WC}{}_k = [\mathrm{x}_k\ \ \mathrm{y}_k\ \ \mathrm{z}_k]^T$, minus the camera optical centre when λ_i was first observed, r_λ:

$$b_1 = \left[\left(x_k - x_1\right)\ \ \left(y_k - y_1\right)\ \ \left(z_k - z_1\right)\right] \tag{3.66}$$

The computation of γ is analogous to that of β, so the angle is estimated from the director vector h_2 and baseline directional vector b_2 according to equation (3.67):

$$\gamma = \cos^{-1}\left(\frac{h_2 \cdot b_2}{\|h_2\| \cdot \|b_2\|}\right) \tag{3.67}$$

where h_2 is the director vector in world coordinates describing a ray which is the projection of the match found to the candidate λ_i in the current frame. This h_2 is found by applying the transformation from current camera frame into world reference $R^{WC}(\mathbf{q}^{WC}{}_k)$ to the vector ray $\mathbf{h}^C{}_2(u_{2u}, \upsilon_{2u})$.

$$h_2 = R^{WC}\left(q^{WC}\right) h_2^C\left(u_{2u}, \upsilon_{2u}\right) \tag{3.68}$$

This $\mathbf{h}^{C}{}_{2}(u_{2u},\upsilon_{2u})$ ray is obtained from the undistorted coordinates of the current observation z_i of the candidate λ_i:

$$h_2^C\left(u_u,\upsilon_u\right)=\begin{bmatrix}\dfrac{u_0-u_u}{f} & \dfrac{\upsilon_0-\upsilon_u}{f} & 1\end{bmatrix}, \tag{3.69}$$

where the undistorted coordinates are found applying the inverse distortion model to the pixel coordinates of the last measurement of λ_i, $z_i = (u,\upsilon)$:

$$\begin{bmatrix}u_u\\ \upsilon_u\end{bmatrix}=\begin{bmatrix}\dfrac{u-u_0}{\sqrt{1-2k_1r^2}}+u_0\\ \dfrac{\upsilon-\upsilon_0}{\sqrt{1-2k_1r^2}}+\upsilon_0\end{bmatrix}\ ,\ r=\sqrt{\left(u-u_0\right)^2+\left(\upsilon-\upsilon_0\right)^2}\ . \tag{3.70}$$

In turn, director vector b_2 is just a vector with same direction and module as b_1, but opposite sense, so $b_2 = -b_1$, and the baseline b is just the Euclidean norm of b_1 or b_2.

$$b_2=b_1\ ;\ b=\left\|b_1\right\|=\left\|b_2\right\| \tag{3.71}$$

After estimating β and γ, α is tested against α_{min}: if it is greater, $\hat{\mathbf{y}}_{new}$ is introduced into the state vector $\hat{\mathbf{x}}$ as part of the map $\mathbf{m}$, applying the inverse observation model g, in equation (3.72):

$$\hat{\mathbf{y}}_{new}=g(\hat{x}_k,\lambda_i,z_i)=\begin{bmatrix}x_i & y_i & z_i & \theta_i & \phi_i & \rho_i\end{bmatrix}^T\ . \tag{3.72}$$

The position of the camera optical centre is taken from the current $\mathbf{r}^{WC}{}_{k}$ camera position, as equation (3.73).

$$\begin{bmatrix}x_i\\ y_i\\ z_i\end{bmatrix}=\begin{bmatrix}x_k\\ y_k\\ z_k\end{bmatrix} \tag{3.73}$$

The attitude and heading are computed from the director vector h_2:

$$\begin{bmatrix}\theta_i\\ \phi_i\end{bmatrix}=\begin{bmatrix}a\tan 2\left(-h_{2y},\sqrt{h_{2x}^2+h_{2z}^2}\right)\\ a\tan\left(h_{2x},h_{2z}\right)\end{bmatrix}; \tag{3.74}$$

and the inverse depth is computed as:

$$\rho_i=\frac{\sin\left(\alpha\right)}{b\cdot\cos\left(\beta\right)}\ . \tag{3.75}$$

Once the feature $\hat{\mathbf{y}}_{new}$ has been added at the end of the state vector as $\hat{\mathbf{y}}_{n+1}$ through equation (3.10), the covariance matrix needs to be updated. As a new feature has been added, the new covariance matrix P_{new} has to include new elements to describe the uncertainties related to the new landmark:

$$P_{new} = \nabla\Upsilon \begin{pmatrix} P & 0 \\ 0 & R_j \end{pmatrix} \nabla\Upsilon^T \tag{3.76}$$

where

$$R_j = \begin{pmatrix} \sigma_{u1}^2 &&&&&&&&&& \\ & \sigma_{v1}^2 &&&&&&&&& \\ && \sigma_u^2 &&&&&&&& \\ &&& \sigma_v^2 &&&&&&& \\ &&&& \sigma_{x1}^2 &&&&&& \\ &&&&& \sigma_{y1}^2 &&&&& \\ &&&&&& \sigma_{z1}^2 &&&& \\ &&&&&&& \sigma_{1q1}^2 &&& \\ &&&&&&&& \sigma_{1q2}^2 && \\ &&&&&&&&& \sigma_{1q3}^2 & \\ &&&&&&&&&& \sigma_{1q4}^2 \end{pmatrix}. \tag{3.77}$$

R_j contains the variances of the pixel at which the landmark was seen for the first time, (σ_{u1}^2, σ_{v1}^2); the variances of the pixel coordinates when initialized, (σ_u^2, σ_v^2); the variances for the camera optical centre during the first observation, (σ_{x1}^2, σ_{y1}^2, σ_{z1}^2); and the variances of the quaternion describing the orientation of the camera optical centre, (σ_{1q1}^2, σ_{1q2}^2, σ_{1q3}^2, σ_{1q14}^2).

The Jacobian ∇Y of the initialization model follows a block structure:

$$\nabla\Upsilon = \begin{bmatrix} I_{n\times n} & 0 \\ \dfrac{\partial g}{\partial \hat{\mathbf{x}}_v}, 0_{6\times6}, \dots, 0_{6\times6} & \dfrac{\partial g}{\partial R_j} \end{bmatrix} \tag{3.78}$$

where $\delta g/\delta\hat{\mathbf{x}}_v$ contains the partial derivatives of the inverse observation model g (equation (3.72)) w.r.t the state of the camera, $\hat{\mathbf{x}}_v$, and $\delta g/\delta R_j$ are the derivatives of the same inverse observation model w.r.t. to the covariance parameters of the initialization process, as noted in the matrix R_j.

Alternatively, it is possible to introduce landmarks according the undelayed method (Civera et al., 2006). Though undesirable, this may be needed in scenarios where the environment is deprived of robust features in short distances, but which present strong far landmarks. This is because the far landmarks may never achieve enough parallax to satisfy the condition $\alpha > \alpha_{min}$, but they are still valuable to estimate the orientation of the camera. In order to enable this, a set of heuristic rules can be implemented, so that once the estimated baseline b for a given candidate λ_i has reached a value that satisfy $b > b_{min}$, it can be initialized, even if $\alpha < \alpha_{min}$. A method to adjust the values of α_{min}, b_{min} and the initial uncertainties modelled in the covariance matrices is proposed in (Munguia and Grau, 2012).

3.4.7 Map Management

As the computational costs of any EKF based SLAM methodology grow with the number of landmarks present in the state, a common strategy also applied in this method is to remove the older features. By doing this the EKF-SLAM modifies its behaviour to resemble that of a visual odometry technique. It is also pretty common to store those features removed from the EKF state, and deal with them with other strategies related to the problems of place recognition and big map management (discussed in section 2.7.3).

In a related work (Munguia and Grau, 2009), researchers at VIS proposed a virtual sensor architecture, where the real-time EKF-SLAM process works similarly to visual odometry, and a different slower SLAM process decoupled from the camera rate is used to deal with the global mapping and localization.

With regards to the monocular EKF-SLAM itself and the work developed in this thesis, the strategy used is assuming that a similar architecture to that described at (Munguia and Grau, 2009), or any other posterior long trajectory map management technique is available. Under this assumption, the focus is to produce locally robust trajectory and map estimations, as the introduction of global mapping techniques will maintain the robustness, as discussed in (Strasdat et al., 2010).

So, once a certain threshold of features is reached, older features, and those that are predicted but not observed during the matching step, are removed from the state. This process is much simpler than the initialization of the same features, just requiring to remove those columns and rows which contains its correlations; e.g.:

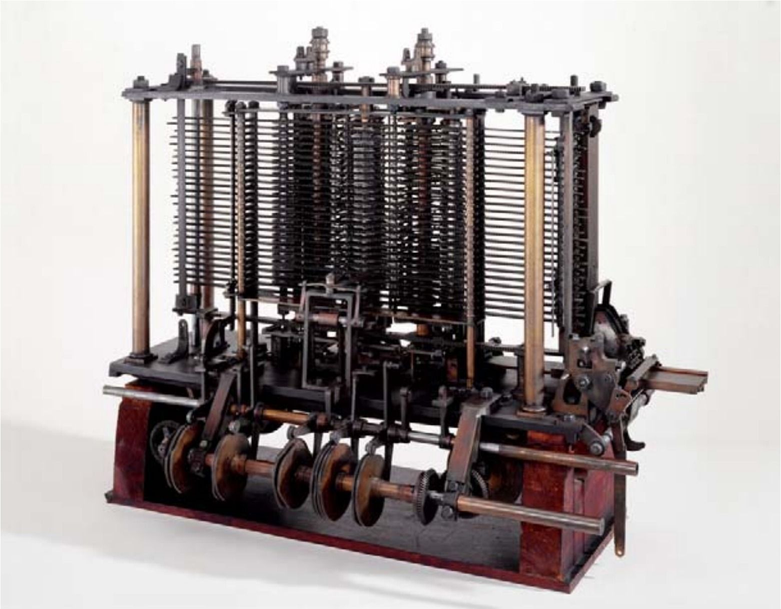
$$\begin{bmatrix} \mathbf{x}_v \\ \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{x}_v \\ \mathbf{y}_1 \\ \cancel{\mathbf{y}_2} \\ \mathbf{y}_3 \end{bmatrix} \rightarrow \begin{bmatrix} \mathbf{x}_v \\ \mathbf{y}_1 \\ \mathbf{y}_3 \end{bmatrix} ; \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_2} & P_{xy_3} \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_2} & P_{y_1y_3} \\ P_{y_2x} & P_{y_2y_1} & P_{y_2y_2} & P_{y_2y_3} \\ P_{y_3x} & P_{y_3y_1} & P_{y_3y_2} & P_{y_3y_3} \end{bmatrix} \rightarrow \begin{bmatrix} P_{xx} & P_{xy_1} & P_{xy_3} \\ P_{y_1x} & P_{y_1y_1} & P_{y_1y_3} \\ P_{y_3x} & P_{y_3y_1} & P_{y_3y_3} \end{bmatrix} . \qquad (3.79)$$

3.5 Conclusions

In this chapter the basics of projective geometry for computer vision have been reviewed, both for the single camera approach and for multiple view/camera scenario, and the main point feature detectors and descriptors have been discussed. Discussion on these items was presented leading to a detailed description of the delayed inverse depth feature initialization approach to monocular SLAM (DI-D monocular SLAM). This monocular SLAM technique, derived from the undelayed monocular SLAM approach popularized by works like (Davidson et al., 2006) and (Clemente et al., 2007), was developed in the Vision and Intelligent Systems (VIS) research group, and presented and discussed in (Munguía and Grau, 2012). Although the delayed visual SLAM approach is a monocular SLAM approach, thus working with only a camera, the introduction of delayed initialization through parallax estimation uses concepts based on epipolar geometry and multiple view geometry. Many of these concepts are further commented and discussed during the development presented in Part III of this dissertation.

Part II

The Data Association Problem



Trial model of the Analytical Engine, proposed by Charles Babbage in 1837, and considered the first Turing-complete general-purpose computer, displayed at the Science Museum, London.

II.A Introduction

The *data association problem*, also known as the *correspondence problem*[16], is that of finding the correspondence relations between new data available from the sensors and the previous data about the environment already present in the map. This means that the objective is to associate new uncertain data (under the guise of new measurements or

[16] Within computer vision community the term *correspondence problem* is widely extended due the influence of the stereo vision field; at the same time the *data association problem* expression remains used in the SLAM community, probably inherited from the tracking methodologies in detection and ranging sensors.

observations) to known environmental landmarks, thus concluding that both measurements correspond to the same physical object/feature in the world.

For any given system which relies on solving the data association problem, there is a set of critical factors that define which will be the suitable approaches to solve it. The single most defining feature is the kind of sensor/s to be deployed. This will define how are the landmarks that can be detected as features, and in turn, configures the set of landmark parametrizations available. For example, it is common for multimodal systems having to address the data association in different spaces or multiple times in order to associate readings from the various sensors, as seen in (Atrey et al., 2010). Besides, the specifications of sensors and landmarks are the defining factors in two other problems: the landmark detection and, depending on the sensors, the filtering of false positives. The landmark detection problem is pretty straight forward for most of the approaches, as once it is clear what is defined as a landmark, it becomes just an issue of signal processing and feature detection in the sensor space.

On the other hand, filtering false positives is a harder challenge, as not only these false positives may arise in terms of landmark detection, but in the association step itself. The false positives for the landmark detection step may be produced by the spuriousness of the sensor, or by the characterization used to detect the landmark, e.g: in structured enviroments, like furnished rooms, it is usual that the Harris corner detector (Harris and Stephens, 1988) denotes as salient non-existent points where two orthogonal edges at different surfaces/depths overlap. The false positives due incorrect association tend to be an issue largely caused by the environment; if the detected landmarks are too similar or too close in sensor space, the probability of false associations grows. These can be the most disruptive cases for probabilistic SLAM, and many of the research in the association problem within SLAM is destined to pruning them, going as far as being preferable risk losing correct associations, as will be described in this chapter.

II.B Challenges on Data Association

In the probabilistic SLAM framework, the data association requires that the prediction step has been completed, and new sensor data are available. The problem itself can be considered as three different smaller problems or tasks: finding landmarks (as distinguishable data on the environment) in sensor reading; measurement of said landmarks, thus, interpreting their spatial relations with respect to the sensor or robot; and finally establish the correspondence relation with the previous landmarks on the map, if possible. FIGURE II.1 illustrates these tasks within the general EKF framework procedure. The prediction of the observations (that is, how the known data are expected to be perceived by the sensor), is computed with the direct observation model **h** (equation 2.8), and although it is always a critical part on EKF-SLAM approaches, it is not a required step in many data association solutions. This theoretical division within the correspondence problem is rarely as clear in practice, and it is largely dependent on the approach, e.g.: in

LRF-based SLAM approaches the most common approach to data association is the ICP (Besl and McKay, 1992), where the whole sensors scan is matched, producing association without extracting landmarks.

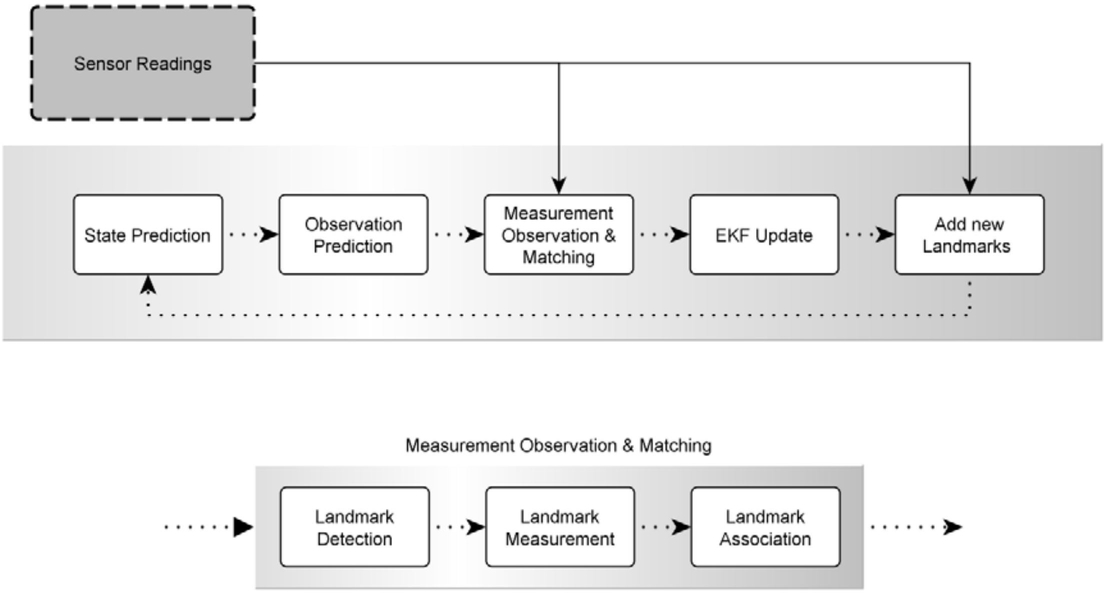


FIGURE II.1: *General EKF SLAM procedure and details on the processes considered within the Data Association Problem.*

In the context of visual SLAM, the landmark measurement step is affected by the limitations of the camera sensors: without additional information, with cameras being bearing-only sensors, the depth is not directly observable, producing incomplete measurements. Also, as will be discussed along this chapter, the final association step depends largely on the parametrization model and the landmark initialization processes (seen in sections 2.7.1 and 3.5.6 respectively).

The methods to deal with landmark detection and observation problem can be classified in two large groups: model-based and appearance-based approaches. This division is related to the parametrization of the landmarks (seen in 2.7.1), as model-based approaches (Gee and Mayol-Cuevas, 2006) work with non-point *geometric primitives*[17] used as geometric models, which allow modelling more complex elements of the environment with meaningful structure (Flint et al., 2010). On the other side, appearance-based approaches rely directly on the appearance of elements; thus, relying directly on the image data at pixel level. This means that the geometry, shape and size of an object is meaningless, but that the potentially relevant data are over the whole image, and a vision-based interpretation method is required to extract and reduce the information.

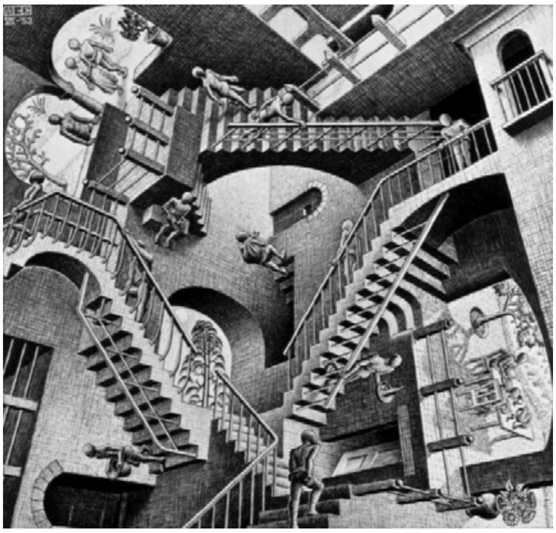
[17] As in computer graphics: atomic irreducible objects.

This information reduction is a way to remove irrelevant data, as an average camera sensor can yield hundreds of megabytes per second. The information that will be kept (let be it colour of a pixel/region, illumination, optical defocus, image-spatial relations, etc.) depends largely on the strategy used to produce the features. These strategies are usually divided in two wide categories: localized features, and whole-image features. The localized features involve sub-regions of a given image, generally denoting a pixel or/and its vicinity, while whole-image features are derived from full images (Lovegrove and Davison, 2010). Note that there are plenty of whole-image feature techniques relying on localized features to build upon using histograms (Dalal and Triggs, 2005) and other image classification techniques such as Bag of Words (Galvez-López and Tardós, 2012). These approaches are used frequently in the context of loop closing[18], where the association problem has more relaxed real-time constraints. Several examples of the approaches taken to the association problem when dealing with loop closing can be found in (Williams et al., 2009).

[18] Also known as the *place recognition problem.*

Chapter 4

Data Association & Validation for Monocular SLAM



"Relativity", by M.C. Escher (1953), one of his works, famously inspired by Mathematics. Physically impossible perspectives and gravitational pulls combine.

4.1 Introduction

The general solutions to the data association (DA) problem are based in two different approaches, the Bayesian and the non-Bayesian methods. Under the Bayesian approach, full distributions (or representative enough approximations) in the DA space are computed from prior and posterior beliefs; on the other side, the non-Bayesian approaches rely on computing maximum likehood estimates from different sets of DA solutions. This difference means Bayesian DA generally delays "solving an association" until enough confidence on the beliefs is achieved by producing complex models to represent the different distributions at each step, and frequently multiple DA hypotheses.

The main representatives of Bayesian DA, the multihypothesis tracking approaches, are closely related to object tracking problems in cluttered environments (Bar-Shalom, 1987). When association ambiguities arise new data association tracks are produced, representing different hypotheses, and maintained until the ambiguities are solved. This generally means pruning the hypotheses with the lowest likehoods, and fusing when possible those that present better odds. These approaches are computationally expensive both in terms of memory space and time to process and update all the hypothesis, thus the pruning and reconditioning are critical processes. This is especially true in the SLAM context, where each new hypothesis can represent an additional map estimation. These characteristics make the loop closing problem (Rizzini and Caselli, 2011) and the optimizations (Cummins and Newman, 2008) achieved very significant.

On the other side, non-Bayesian techniques tend to take a best effort approach (generally under the form of a greedy algorithm) to synthesize the knowledge and beliefs available at each step to produce a single DA solution with respect to the set of observations using a likehood function (i.e. after each data association step, each single landmark is associated to a single observation).

Whether they are of Bayesian nature or not, many of the DA approaches rely on the expectations on known landmarks to prune out those observations that pose a risk in terms of producing distant false positives. In order to achieve this, the measurements of the landmarks found in the map are predicted, and these *predicted observations* are used to define an area in the sensor space where the actual new observations from the sensor are expected. This area is called a validation gate (Bar-Shalom and Tse, 1975), and it is commonly based on the notion of the Mahalanobis distance (concretely the squared Mahalanobis distance, SMD), although there are other approaches (Blanco et al., 2012).

4.2 Mahalanobis distance gating

The Mahalanobis distance describes the distance between a given point and a distribution, in the DA case between an observation x and the measurement process probability $p(z)$. Thus, for a given measurement process where the measurement likehood model in equation (4.1) describes the *pdf* of the observations with predicted measurement $\hat{z}(k)$ and covariance $\hat{S}(k)$

$$p\left(z\left(k\right)\right)=\mathrm{N}\left(x\left(k\right);\hat{z}\left(k\right)\hat{S}\left(k\right)\right) \tag{4.1}$$

the Mahalanobis distance between a measurement x produced by the sensor and the predicted expectation would be described by the expression:

$$d=\sqrt{\left(x-\hat{z}\right)\hat{S}^{-1}\left(x-\hat{z}\right)}\,. \tag{4.2}$$

Conceptually, the Mahalanobis distance can be interpreted as measuring how far off is a given point from the centre of the distribution in terms of *standard deviations*. This can be observed on the example in FIGURE 4.1, where a random distribution is sampled, and the isoprobability curves describe the probability boundaries, illustrating how the distance is scaled according to the covariance. The Mahalanobis distance then presents several features which makes it useful when evaluating the goodness of measurements from an statistical point of view, namely: it accounts for different variances in each dimension of a distribution; it accounts for the covariance between variables; and if the covariance is a diagonal matrix, the Mahalanobis distance becomes the *normalized Euclidean distance*. A special case of the latter property is found when covariance matrix is not only diagonal, but also the identity matrix (*I*), then the Mahalanobis distance reduces to the *Euclidean distance*.

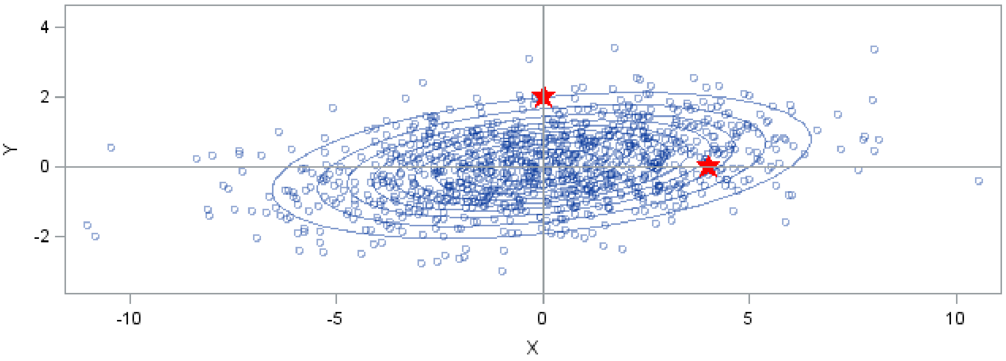


FIGURE 4.1: *Plot of random data produced by a given bivariate normal distribution, with prediction ellipses overlaid. The ellipses describe isoprobability contours, containing the 10%, 20% ....90% probabilities. Although geometrically p_1 at (0,2) is nearer to the centre than p_2 at (4,0), if we account for variances p_1 is at the 0.9 probability contour while p_2 lays near the 0.7 contour, thus presenting a much shorter distance in terms of standard deviations.*

Using the Mahalanobis distance as basis for a gating procedure requires a set of conditions and assumptions to be met (Montiel and Montano, 1998), namely:

- The known measurements (landmarks) are measured with known covariance.
- The sensors produce measurement of Gaussian nature (i.e. white noise).
- Any other source of noise or uncertainty in the system can be described through Gaussian *pdfs*.
- The function to transform from landmark or measurement representation to sensor space is known and computable, thus allowing prediction of the appearance of measurements.

The general probabilistic SLAM framework satisfies all those conditions. Within the context of visual SLAM, the Gaussian character of the sensors and mapping reversibility of the measurement model are also generally assumed, although they are only naturally satisfied in part: sensors rarely present an error function that truly behaves as white noise (although they tend to be similar enough), and the mapping gap between sensor space and representation space can be solved even if the measurement model function is not fully invertible.

Then, a validation gate G can be defined, as seen in equation (4.3). In this area it is guaranteed that the Mahalanobis distance between a measurement and its expectation is bounded by the threshold γ:

$$\begin{aligned} G(k,\gamma) &= \left\{ z : \left(z - \hat{z} \right)^T \hat{S}^{-1} \left(z - \hat{z} \right) \leq \gamma_{k,\alpha} \right\} \\ &= \left\{ z : d^2 \leq \gamma_{k,\alpha} \right\} \end{aligned} \tag{4.3}$$

The threshold γ is generally computed as the inverse χ^2 cumulative distribution at a significance level α, with k degrees of freedom. Note that this formulation means that *G(k, γ)* describes an hyper-ellipsoid where measurements are expected to appear with a known probability. Thus, the gate is essentially an iso-probability contour produced by intersecting the Gaussian *pdf* of the measurement with an hyperplane. This means that interpreting the Mahalanobis distance as a sum of squared standard normal random variables, G acts as a gate which excludes 100(1- α)% of the true measurements according to the *pdf.* The value of α is typically of 0.95 or 0.99, which are approximately equivalent to 2σ and 3σ respectively[18].

The impact of the properties of the Mahalanobis distance, and its relation with the different concepts of distance depending on the dimensionality and uncertainty management is illustrated in FIGURE 4.2. The same points present hugely different measurements depending on what distance is measured and how estimated errors and uncertainties are treated. The first plot directly measures 2D Euclidean distance, which results pretty intuitive in human terms. The second plot considers only uncertainties in term of position/measurement, achieved by estimating the Mahalanobis distance using a diagonal Covariance Matrix, ignoring the rest of the terms, and assuming that the error probability is homogenous in all the directions. Some isoprobability contours show how the pure uncertainty (without considering prior knowledge in described correlations) propagates probability equally in all directions. The third plot describes the actual effects of using Mahalanobis distance: the correlations describe different errors and scaling of the uncertainty along dimensions due previous movements/actions of the system.

[18] 2σ and 3σ have values of 95.45% and 99.73% respectively.

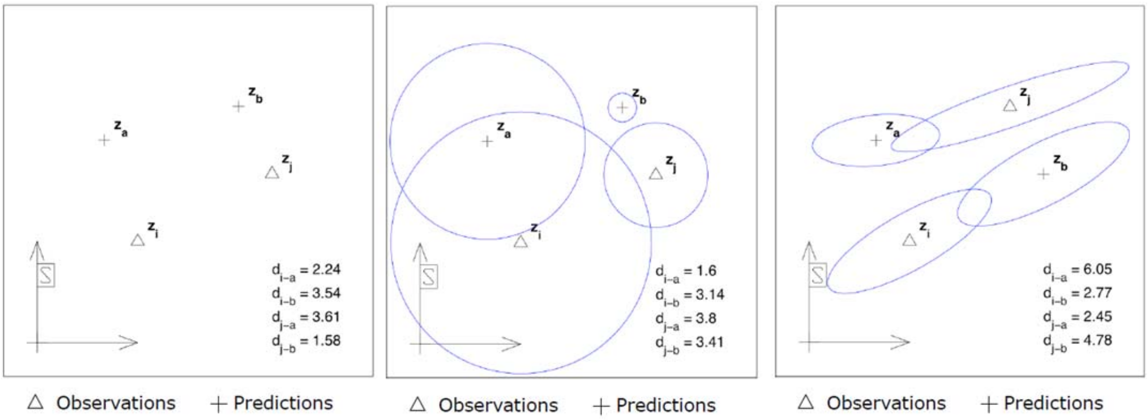


FIGURE 4.2: *Examples of 2D pairwise data association based on different distance measurements.* **Left:** *distance in terms of actual position (Euclidean distance).* **Centre:** *distance accounts positions and uncertainty (Mahalanobis distance computed only with the diagonal of Covariance Matrix).* **Right:** *positions, uncertainties and correlations are accounted (full Covariance Matrix to compute Mahalanobis distance).*

4.2.1 Nearest neighbour based matching

The simplest algorithm to perform gate-based data association was the Individual Compatibility Nearest Neighbor filter (ICNN, or simply nearest neighbour, NN) (Neira and Tardós, 2001), which works as a greedy algorithm: for each predicted landmark, it evaluates measurements in the sensor space, computes the Mahalanobis distance to each measurement, and accepts the one with the minimal distance. This approach is largely based on the simple NN algorithm, a well-known and studied technique in tracking problem (Bar-Shalom, 1987). The notion of individual compatibility reflects two important aspects about the algorithm: the association is performed individually, and no matter if a given measurement is the nearest one to the prediction, if it does not satisfy validation gate G it is rejected, thus considering the landmark to have no measurement.

This algorithm presents, given a set of m measurements for n landmarks in a map, a linear cost $O(mn)$ with respect to the size of the map in terms of compatibility tests with the Mahalanobis distance. The computation of the Mahalanobis distance requires the inversion of the covariance matrix $\hat{S}$, which can become a burden in some characterizations. Still, the greatest limitation of this method comes from the individual aspect of the algorithm: pairs of data association *individually compatible* might not be *jointly compatible*. Moreover, as a greedy algorithm, it is entirely possible that an observed measurement x_n, whose minimal Mahalanobis distance would match it to a predicted observation z_n, is associated with a different predicted landmark z_i (where $i < n$) given that a better measurement x_i is not found for z_i. Besides, when treating measurements individually, the correlations between measurements are ignored, as those present in the map (Castellanos et al., 1999) and those introduced in the predicted observations due the robot/sensor pose uncertainty, as discussed in Section 2.3.3. These weaknesses limit the suitability of the approach to setups where two conditions can be guaranteed: the pose uncertainty is smaller than the distance between features (to avoid mismatches); and the spuriousness of the sensor is low enough so that the probabilities of fake measurement appearing in the gating area is low.

To overcome the described weaknesses, several derived techniques were developed. The sequential compatibility nearest neighbour (SCNN) (Neira and Tardós, 2001), introduced the notion of joint compatibility, where the gating is performed considering not individual pairings, but a set of them. The direction of the association is reversed, i.e., for each of the new measurements found, the nearest predicted landmark which complies the validation gate is matched. Then, after each new pairing is added, the state and covariance are updated, so new pairs are tested accounting for the previous pairings. This process makes the SCNN an $O(mn)$ + $O(mn^2)$ algorithm: with *mn* tests to evaluate the Mahalanobis distances, and *m* updates of the state, which is quadratic (n^2) with respect to the size of the map.

While the SCNN guaranteed that the validation gate is accepted for all the measurements, it is still far from perfect. As a greedy algorithm, a pairing deemed compatible with those previously associated is never revaluated. This means that in early stages, it is possible to accept an spurious measurement or incorrect association, to update the state considering it correct, and to affect the rest of the associations. Moreover, as the uncertainty represented in $\hat{S}$ decreases, but not the error (given an incorrect association), the risk of producing map inconsistencies grows. Several later approaches in the NN family of algorithms overcame some of the weaknesses inherent to the original NN approaches.

4.3 Active Search and Cross-Correlation operators for data association

In the context of visual SLAM, detection of new points of interest over a whole image is a computationally demanding operation, as discussed during Part I. As such, any way to reduce the area to search produces enormous gains in terms of performance and efficiency. Proposed as a way to solve stereo correspondence in (Davison and Murray, 2002), and applied to monocular SLAM in (Davison, 2003), the active search technique solves the measurement and matching challenges through direct appearance correlation. In a similar fashion to the concept of validation gate, an area around a given landmark prediction is defined, but in sensor space. And instead of testing Mahalanobis distances (or any other likehood function), a matching measurement to the landmark is brute-forced through a correlation operator used as a likehood function. This is possible as full information about the map and the uncertainty on the present elements is propagated and updated at each iteration, thus the search area can be computed with an assigned probability for the matching feature to lie in it.

4.3.1 Active Search for Visual feature matching

The area where the correlation operator-based search is performed is defined according to the uncertainty on each landmark prediction (see FIGURE 4.3); that is, using the uncertainty terms (the diagonal of the covariance matrix *P*), and scaled according to a number of standard deviations that must be covered (i.e. assigning a probability for the landmark to lie into the area). Equation (4.4) details how to compute this area under the monocular vision assumption for a SLAM process:

$$\begin{bmatrix} S_x \\ S_y \end{bmatrix} = \begin{bmatrix} 2n\sqrt{S_{i(1,1)}} \\ 2n\sqrt{S_{i(2,2)}} \end{bmatrix} \tag{4.4}$$

where *n* is the number of standard deviations to explore at each direction, with values ranging normally between *n=2* and *n=3* (~0.95 and ~0.99 respectively). S_i is the expression of the covariance for the i^{th} feature prediction, computed according to equation (4.5), which is just an instanced expression of the standard covariance equation (2.16). For this expression, the white noise error described in terms of pixel variance, $\sigma_u^2$and σ_v^2, is assumed to take values of 1 pixel both for *u* and *v* directions.

$$S_i = \nabla H_i P_{k+1} \nabla H_i^T + R_{uv} \tag{4.5}$$

Where

$$R_{uv} = \begin{bmatrix} \sigma_u^2 & \\ & \sigma_v^2 \end{bmatrix}. \tag{4.6}$$

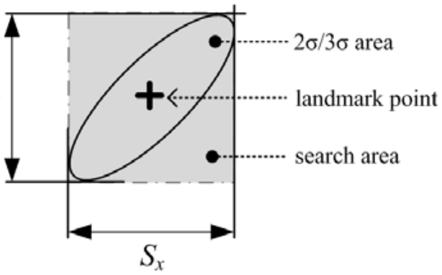


FIGURE 4.3: *Search area defined by equation (0.4), according to the 2σ or 3σ criteria. Depending on the characterization of S_{xy}, the seach area covers an excess of pixels in the vicinity. This, although not optimal, is preferable to spend computation time interpolating an ellipse to guarantee the borders of the region instead of matching points.*

The general method is described in ALGORITHM 4.1. This algorithm describes only the part of the search procedure, as there are several steps that would depend largely on the exact correlation operator used as a likehood function. So, for each landmark in the map **m** whose predicted appearance is within the current image, this search region is computed and the patch which was stored when initially observed is retrieved. As the camera pose probably has varied since the landmark was first observed the patch is linearly warped considering the scale and rotation[19] in order to maximize the outcome of the likehood function. Then, for all the pixels in the given search area, the correlation between this *warped patch* and new patch centred on each pixel is found. The best correlation score, which depends on the operator used, is considered the candidate. In the case of normalized operators, this value can be tested against a given threshold, as variance in illumination will not affect the score.

function (z_i, S_i) := ActiveSearch $(h_i, \nabla H_i, P_{k+1}, img, R_i, dB)$

Input:

P_{k+1}	estimation covariance
h_i	predicted observations
∇H_i	observation Jacobian
img	image from sequence
R_i	observation process covariance
dB	features patch database

Output:

S_i	innovation covariance matrix
z_i	matching observations found

```
$S_i := \nabla H_i P_{k+1} \nabla H_i^T + R_i$
for all predicted observations $k$ in $h_i$ do
    if $h_i(k)$ is not NULL then
        compute search $Region_k$ with $S_i$
        $patch_k$ := retrieve dB($k$).patch
        warp($patch_k$)
        $(p_x,p_y)$ := NULL;
        maxCorrelation := 0;
        for all the points 'i,j' in $Region_k$ do
            tempCorrelation := correlationOp([i,j], img, $patch_k$)
            if tempCorrelation >= maxCorrelation then
                $(p_x,p_y)$ := 'i,j'
                maxCorrelation := tempCorrelation
            end if
        end for
        if maxCorrelation > correlationThreshold then
            add 'i,j' to vector $z_i$ as the matched observation to prediction $k$
        end if
    end if
end for
return $(z_i, S_i)$
```

ALGORITHM 4.1: *Pseudocode for the general active search strategy implemented, independent from the correlation operator* 'correlationOp' *used (except for the test value).*

[19] Retrieved from the initial pose of the landmark and the predicted position w.r.t. the sensor.

4.3.2 Correlation operators used in data association

As it is described in ALGORITHM 4.1, once the area is defined to cover the possible appearance of the matching point for the predicted landmark with a given *n* probability[20], a likehood function to choose the best matching pixel is required. Different correlation based operators have been proposed over the years. In (Davison and Murray, 2002), where the active search was proposed, the sum of squared differences (SSD) (Anandan, 1987) is used, a well-known correlation correspondence detector based in cumulative aggregation of a cost function over an area around the pixel (see equation (0.8)). The main characteristics that made it a suitable choice was better discriminating performance than most of the direct cost aggregation operators (SAD, ZSAD, etc…), while having a computational cost well bounded below those based on cross-correlation, CC, like NCC and ZNCC.

Note that the correlation between a given image patch, call it I_p, and a search region within an image, I_s, is closely related to the signal process and communications theory fields. Most of the visual correspondence detectors can be traced signal processing techniques, as discussed in (Martin and Crowley, 1995), as they tend to rely on forms of cost aggregation that can be described as comparing signals over a discrete space. This discrete space tends to be quantized on the pixel notion, but it can be based on more complex spaces, like multi-resolution/scale pyramids, tensors spaces, filter banks, and many other approaches which today are linked with high level feature descriptors (Krig, 2014). In (Scharstein and Szeliski, 2002) a survey presents and discusses over 35 different correlation approaches, though focused on their features with respect to the stereo correspondence problem.

With respect to the cross-correlation based matching approaches it is worth remembering that the expression for cross-correlation can be derived directly from an inner product of two vectors, or from the sum of squared differences of two neighbourhoods. This means that the normalization approach taken can have a noticeable impact, e.g.: in (Martin and Crowley, 1995) it is shown how SSD is equivalent to performing a cross-correlation step with a suitable normalization of the pixel sets I_p and I_s.

4.3.2.1 Correlation operators for visual SLAM

The most used operators for correlation based matching can be grouped according to which criteria they apply. SAD, SSD, and ZSAD (described in equations (4.7), (4.8) and (4.9) respectively) are based on distance measurements. SSD leads essentially to a square-minimization solution to the matching problem w.r.t. distance between the image coordinates of the matched points. SAD optimizes the solution by working over the differences directly, without square minimization, but this makes it vulnerable to outliers. ZSAD introduces the zero-mean modification (used also in ZNCC) which makes the method invariant to brightness.

[20] In terms of standard deviations.

The most used cross-correlation coefficients, NCC and ZNCC (equations (4.10) and (4.11), respectively), use the correlation operator from signal processing to quantify the likeness of the patches, discussed in previous section. The advantage of NCC over the classic cross-correlation operation lies in the fact that the normalization step introduces invariance to contrast variations, as the operation is equivalent of equalizing the histograms of the patches being processed through CC. As in the SAD case, introduction of the zero-mean variant makes ZNCC also invariant to luminosity variance.

- Sum of absolute differences (SAD):

$$SAD = \sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} \left| I_p(i,j) - I_s(x+i, y+j) \right| \tag{4.7}$$

- Sum of squared differences (SSD):

$$SSD = \sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} \left(I_p(i,j) - I_s(x+i, y+j) \right)^2 \tag{4.8}$$

- Zero-Mean Sum of absolute differences (ZSAD):

$$ZSAD = \sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} \left| I_p(i,j) - \overline{I}_p(i,j,n) - I_s(x+i, y+j)\overline{I}_s(x+i, y+j,n) \right| \tag{4.9}$$

- Normalized cross-correlation (NCC):

$$NCC = \frac{\sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} I_p(i,j) \cdot \sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} I_s(x+i, y+j)}{\sqrt[2]{\sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} I_p^{\,2}(i,j) \cdot \sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} I_s^{\,2}(x+i, y+j)}} \tag{4.10}$$

- Zero-Mean Normalized Cross-correlation (ZNCC):

$$ZNCC = \frac{\sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} \left(I_p(i,j) - \overline{I}_p(i,j,n) \right) \cdot \sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} \left(I_s(x+i, y+j) - \overline{I}_s(x+i, y+j,n) \right)}{\sqrt[2]{\sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} \left(I_p(i,j) - \overline{I}_p(i,j,n) \right)^2 \cdot \sum_{i=-n_p}^{n_p} \sum_{j=-n_p}^{n_p} \left(I_s(x+i, y+j) - \overline{I}_s(x+i, y+j,n) \right)^2}} \tag{4.11}$$

4.3.2.2 Evaluation of correlation operators

A set of experiments have been performed to study the performance of the different operators. These tests were used as guidelines to determine which of the operators was convenient to use in the active search methodology within the delayed monocular SLAM. The experiments consisted in a series of sensibility analyses, to see which operators perform better as likehood functions under different circumstances.

To generate the different circumstantial perturbations, a set of images was treated with several filters and noise sources, at multiple levels of intensity or characterization for each case. FIGURE 4.4 show some examples of the effects of the perturbations induced into the image set. Each resulting disrupted image was processed matching it with the original image to evaluate the impact of the perturbations. Notice that these experiments implicitly assume that the matching process is perfect, as it is not accounted, and the metrics are computed for whole images[21].

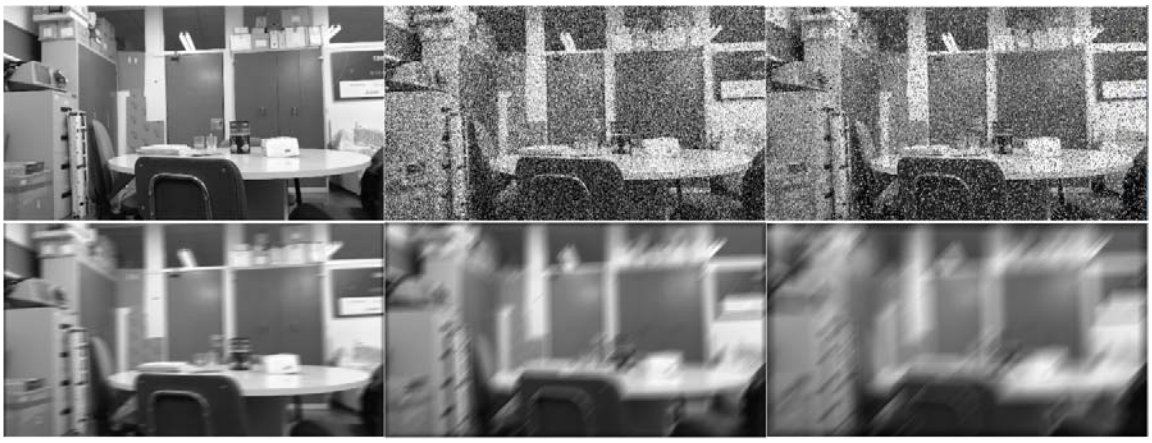


FIGURE 4.4: *Example of image treated with several disturbances.* **Top right:** *original undisturbed image.* **Top centre:** *white Gaussian additive noise (σ=0.1) introduced.* **Top left**: *salt and pepper (ratio of pixels affected = 0.2).* **Bottom left:** *motion blur (40 pixels, 10º).* **Bottom centre:** *motion blur (80 pixels, 225º).* **Bottom right:** *motion blur (150 pixels, 60º).*

The sensor disruptions studied were noise (modelled as additive zero-mean white noise) and capture artifacts (modelled as *'salt and pepper'* noise). The sensibility to movement blur was also tested, using image processing filters to simulate its effects. TABLE 4.1 shows the values studied for each disruption. Each disturbance parameter was tested with 7 different intensity levels, including the standard deviation of the white noise, the amount of pixels affected by *salt and pepper* noise, and the distances in the motion blur. For this last disturbance an additional parameter, angle, was needed, which was set at three levels in 15º increments.

[21] Which is irrelevant for normalized operators (NCC & ZNCC), but scales the values of the other tests.

TABLE 4.1: VALUES USED TO CHARACTERIZE THE DIFFERENT DISTURBANCES USED IN THE SENSIBILITY ANALYSIS

Disturbance	Parameter	Magnitude
White Noise	std. dev.= σ	$\sigma = 0.1i$ for i= [1..7]
Salt and Pepper Noise	pixel ratio = p	$p = 0.05i$ for i= [1..7]
Motion	pixel distance=d	$d = 10i$ for i= [1..7]
	angle=α	$\alpha = 15j$ for j= [1..3]

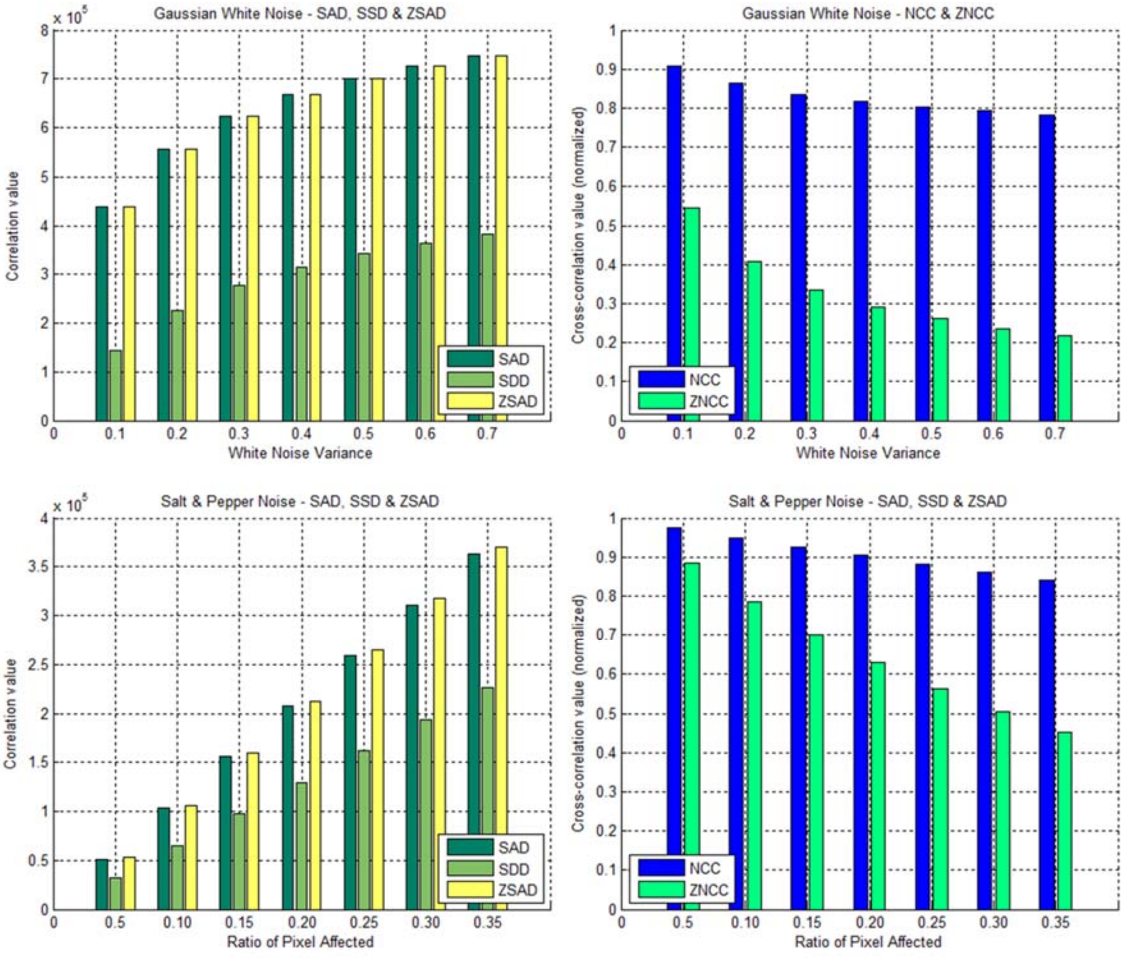


FIGURE 4.5: **Left column:** *Sensibility analisis for SAD, SSD and ZSAD (lower is better).* **Right column:** *Normalized operators: NCC and ZNCC (higher is better).* **Top row:** *Average results of the operators for zero-mean Gaussian noise with σ from 0.1 to 0.7.* **Bottom row:** *Average results of the operators for salt and pepper noise, with p = [0.05 .. 0.35].*

The application of the described disturbances over the image set produced many experimental data. Some selected results of these experiments are shown in FIGURE 4.5 and FIGURE 4.6, describing the cases for the sensor/capture noise (white noise, *salt and pepper*) and a selection of the motion blur disturbance results, respectively.

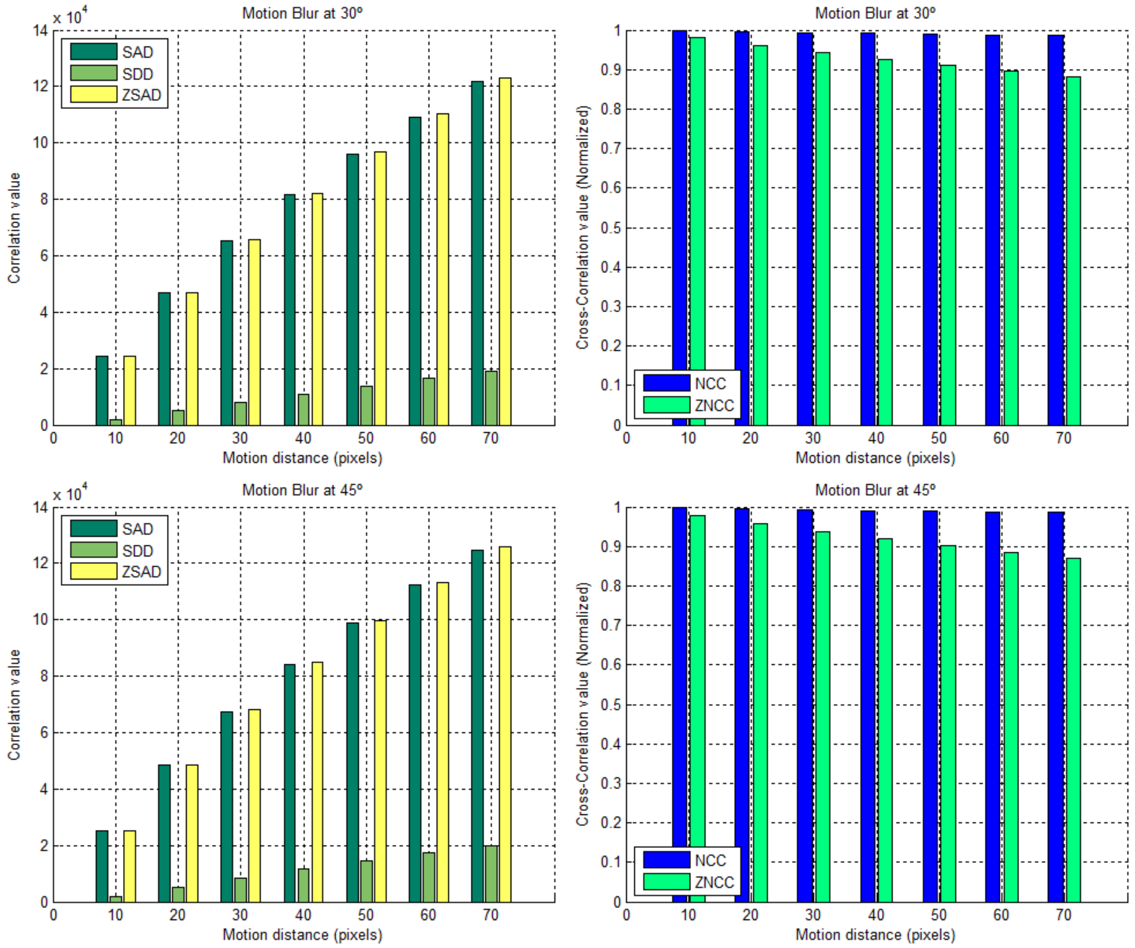


FIGURE 4.6*: Results for the motion blur disturbance.* **Left column:** *Sensibility analysis for SAD, SSD and ZSAD (lower is better).* **Right column:** *Normalized operators: NCC and ZNCC (higher is better).* **Top row:** *Average results of motion blur with $\alpha = 30^o$ and d = 10i for i= [1..7].* **Bottom row:** *Average results of motion blur with $\alpha = 45^o$ and d = 10i for i= [1..7].*

In FIGURE 4.5 *top left* it can be observed that for average cases the difference between using or not the square minimization is minimal. This was to be expected, accounting for the assumption of perfect matching and the absence of fake positives and outliers. When comparing with FIGURE 4.5 *top right*, for the normalized operators, it is worth noting that the ZNCC presents approximately the same sensibility to white noise as non-normalized operators, i.e., the difference in the score between the best and worst case is about 50%. At the same time NCC shows much more robust results against both disturbances.

In FIGURE 4.6 the difference between the normalized cross-correlated operators and those based on aggregations is even more evident. The left column reveals a much greater degree of sensibility, even with the square-based minimization SSD. On the other side, ZNCC presents much less sensibility against motion blur than against capture artifacts (in FIGURE 4.5), with a variation of between 10% to 15% between best and worst cases. The NCC shows again the strongest results, resulting practically invariant to the disruption.

The results obtained show a clear divide in sensibility: the cross-correlation methods are notably more resilient against noisy types of disturbances, and result practically invariant with respect to the motion blur. Although it is well-known that the aggregation based methods present better computational costs, as (Scharstein and Szeliski, 2002) already discussed, the difference is not of a relevant order of magnitude when considered as part of a visual SLAM process. Moreover, there are plenty of optimization techniques to avoid redundant computations, like those discussed in (Luo and Konofagou, 2010).

It is worth noting that in a given visual SLAM sequence the lighting conditions will vary, especially in an outdoor scenario. As such, once determined that in spite of the cost differences, the cross-correlation based operators offer much better results, it becomes apparent that the most suitable operator would be the ZNCC. This choice makes the data association process more resilient to illumination changes, as ZNCC adds lighting intensity invariance to the contrast invariance presented by the NCC. The resilience to the motion blur is also a desirable feature, as most of the robotics applications outside automation/health/other highly funded industries rely on CMOS-based camera sensors, with the weaknesses described before (section 2.2.2.1).

4.4 Data Association Batch Validation

The active search matching methodology just described addresses the problem of data association by allowing to produce a matching observation for each predicted landmark in the map. This yields a set of pairs, *a data association hypothesis*, composed each one of a predicted landmark and its matching feature pixel point in image. As it was discussed earlier, finding a correct association pairs list is usually a critical problem in any EKF-based SLAM system, and the active search technique with ZNCC produces accurate pairings within an acceptable computational time. Still, as there are many factors that may introduce errors, the data association pairings may be association errors even without being incorrectly matched: a moving object can be correctly matched, but produces a dynamic landamark which can disrupt the map, as this spurious landmark does not comply the static assumption (see sections 2.3 and 3.4.3). Other errors may arise when dealing with ambiguous textures and features on the mapped environment. Thus, even after solving data association through a technique analogous to a validation gate, many monocular SLAM approaches present an additional validation gate step to reject those data association pairs found that can be considered erroneous. This is especially true for EKF-based approaches (such as our base DI-D MonoSLAM), as they do not present any other mechanism to marginalise the impact of an spurious or incorrect landmark on the map.

4.4.1 Joint Compatibility Branch and Bound

In the context of classical approaches to inverse-depth (I-D) feature parametrization monocular SLAM, the undelayed I-D technique Joint Compatibility Brach and Bound (JCBB), as seen in (Clemente et al., 2007), has probably been the most influential batch validation methodology. For a decade the JCBB has been considered the golden rule of data association batch validation (Civera et al., 2009), widely reported and studied (Bailey and Durrant-Whyte, 2006). This test is based on the notion of *Joint Compatibility* (Neira and Tardós, 2001) introduced with the SCNN, and its evaluation for different data association hypotheses. The data association hypotheses, briefly mention earlier, are subsets of the set of pairs produced by the measurement matching technique, in this case, the ZNCC-active search combination. Then, this validation test just evaluates the joint compatibility using the Mahalanobis distance-based validation gate, determining if all the pairs on a given hypothesis or set are '*jointly compatible*', thus consistent and valid, or inconsistent as a whole.

The validation gate, once instanced for the monocular SLAM problem, takes the form

$$D_H^2 = \mathbf{g}_H^T S_H^{-1} \mathbf{g}_H \leq \chi^2_{dof,\alpha} \ , \tag{4.12}$$

where the squared Mahalanobis distance D_H^2 (SMD) is approximated using the values of the Kalman innovation $\mathbf{g}_H$ or residuals (see equation (3.53) at section 3.4.5), and the covariance S_H of the innovation (per equation (3.56)), for the given association hypothesis *H.* Note that for each hypothesis to evaluate, the covariance matrix has to be inverted, and the all data structures need to be update, as they may vary in size according to the number of pairings being evaluated in *H*. The Chi-square distribution will have a range equal to twice the number of measurements to test, as each observation is measured in terms of *u*,*v* coordinates in the image; with the same commented values of confidence α of 0.95 or 0.99.

Although this test appears to be computationally expensive, note that $\mathbf{g}_H$ and S_H will be already available as they are updated through the EKF methodology in the observation matching and update steps. Besides, as not all the data association pairs are taken into account in each hypothesis *H*, $\mathbf{g}_H$ and S_H will not be taken completely to obtain the Mahalanobis distance, only those rows related to the considered pair, without necessity of fully computing $\mathbf{g}_H$ and S_H again.

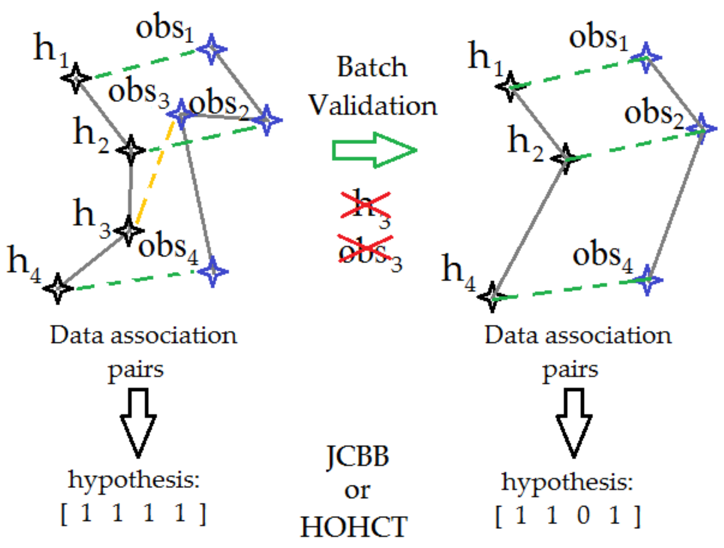
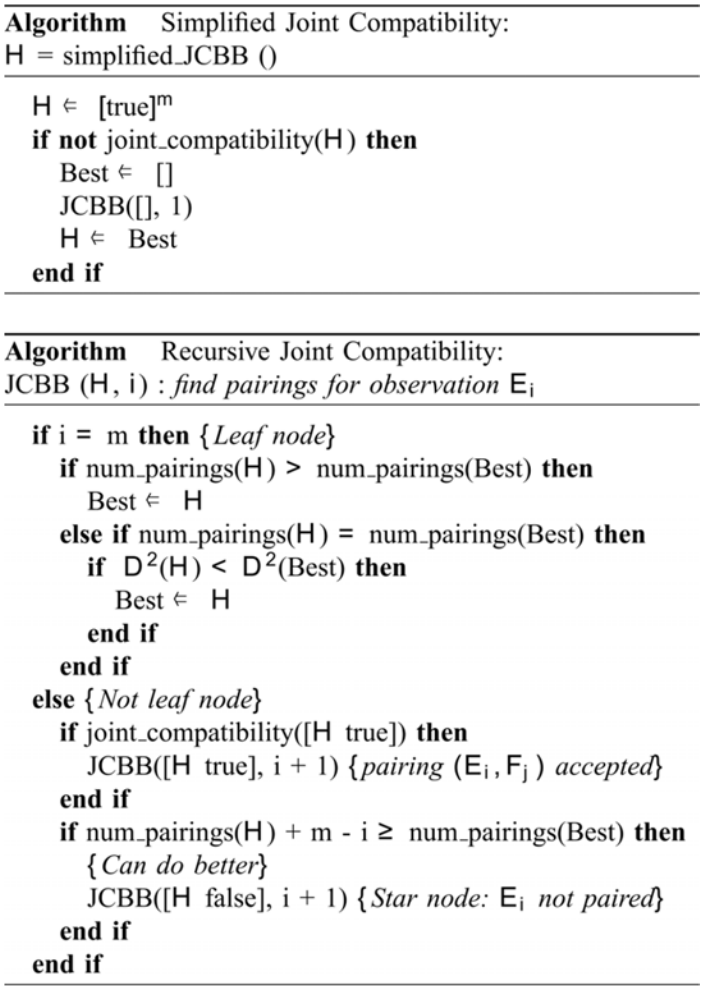


FIGURE 4.7: *Batch validation with joint compatibility illustrated. In this simplified example, the pairing between prediction* h_3 *and observation* obs_3 *is removed as the 'gain' is not compatible with the rest of the pairings.*

As the test to determine if a given data association set is joint compatible or not is available, an algorithm which enables exploration of the hypothesis space is needed. This is solved by the JCBB, exploiting the fact that the data association hypotheses to be validated are essentially a set of ordered decisions. As such, they can be represented as an array of Boolean values, as shown in FIGURE 4.7, where each found pair is accepted (*true* or "*1*") or rejected (*false* or *"0"*). If an initial optimistic hypothesis which tries to accept all the data pairings (where the full vector is *true*) fails the compatibility test, then a search for an smaller hypothesis is performed. Thus the JCBB algorithm builds a binary recursive exploration tree to make sure that it finds the best remaining hypothesis, defined as: achieving the maximal number of compatible data associations, and best compatibility between the hypotheses with the same order, i.e., presenting the lowest SMD. This *best compatibility* criterion also means that the algorithm has a conservative behaviour, as the SMD is proportional to the innovation, so the JCBB always takes the more complete but less divergent hypotheses.



```
Algorithm  Simplified Joint Compatibility:
H = simplified_JCBB ()

  H ⇐ [true]$^m$
  if not joint_compatibility(H) then
    Best ⇐ []
    JCBB([], 1)
    H ⇐ Best
  end if
```

```
Algorithm  Recursive Joint Compatibility:
JCBB (H, i) : find pairings for observation $E_i$

  if i = m then {Leaf node}
    if num_pairings(H) > num_pairings(Best) then
      Best ⇐ H
    else if num_pairings(H) = num_pairings(Best) then
      if $D^2$(H) < $D^2$(Best) then
        Best ⇐ H
      end if
    end if
  else {Not leaf node}
    if joint_compatibility([H true]) then
      JCBB([H true], i + 1) {pairing ($E_i$, $F_j$) accepted}
    end if
    if num_pairings(H) + m - i ≥ num_pairings(Best) then
      {Can do better}
      JCBB([H false], i + 1) {Star node: $E_i$ not paired}
    end if
  end if
```

ALGORITHM 4.2: *Pseudocode JCBB algorithm, from (Clemente et al., 2007)*

The pseudo-code implementation in ALGORITHM 4.2, from (Clemente et al., 2007), shows how the JCBB makes a branch and bound search on a binary tree to increasingly build the Boolean vector representing the hypothesis. The results of this process are order-independent: the algorithm will try all the hypotheses, even after a *jointly compatible* one has been found, in order to guarantee the optimality of the given result (see the *Star node* annotation on ALGORITHM 4.2). Because of this uninformed, unordered exhaustive search, the algorithm has a strong tendency to exponential cost, with no mechanism to control the growth or keeping it low beyond the branching procedure. Note that to enable this branching the JCBB requires to estimate the SMD at each *node* of the exploration tree to test the current hypothesis *H*. This penalization at each node is partially mitigated by exploiting the linearizability of the costs of successive incremental matrix inversions, as described in (Harville, 1998), and using the fact that the Mahalanobis distance can be considered as sum of squared standard normally distributed random variables to cut as early as possible bad branches of the tree.

4.5 Highest Order Hypothesis Compatibility Test

This section describes the proposed algorithm to deal with the data association batch validation problem using the SMD-based joint compatibility test.

4.5.1.1 JCBB computational costs under DI-D Monocular SLAM

The JCBB has shown good results within the context of Undelayed I-D initialization monocular SLAM techniques (Davison et al., 2007)(Williams et al., 2007)(Grasa et al., 2011), and other non-visual SLAM approaches (Fenwick et al., 2002), but it proved to be rather inefficient within the context of the Delayed I-D SLAM approach, first presented in (Munguia and Grau, 2007a) and(Munguia and Grau, 2007b). Initial tests on the MATLAB based prototypes of DI-D became unstable, with a computational time increased by a factor well over an order of magnitude, and viability issues due excessive memory consumption crashing the Java Virtual Machine. Profiling of the partial results obtained pointed clearly to the issue: the binary exploration of the data association hypothesis space produced a combinatorial explosion. It was also noticed that the JCBB achieved slightly better performances with greater number of incorrect data association pairings (noticed on worst cases experiments where the EKF fails to converge and the mapping a positioning becomes useless). Introducing the optimization of the sequential inversion of growing matrices (Harville, 1998) proved helpful in order to avoid memory management crashes, and it produced a slight improvement in performance. But the introduction of JCBB still resulted in an unfeasible performance.

With the JCBB proven a non-viable solution in the DI-D SLAM framework, it is worth noting that there are some key differences between the undelayed I-D (Davison et al., 2007) and the delayed I-D SLAM techniques (Munguia and Grau, 2007): while the undelayed approach tries to initialize a good number of features as landmarks as soon as possible with an heuristic value for depth representation, the delayed approach generally considers less features, but present greater robustness and an initial estimation of the depth obtained through stochastic triangulation (as detailed in section 3.4.6). This robustness is provided by a series of tests and conditions to be passed by candidate features to be considered landmarks, hence the initialization delay, and the requirement to be tracked correctly within a minimal number of frames achieving a parallax value greater than a minimum α_{min} to guarantee the depth estimation accuracy. This makes the delayed approach more expensive in terms of computational cost per feature, but as it holds more accurate information it requires less mapped features initialized to work, thus achieving better performance as odometry estimator (Munguia and Grau, 2009)(Munguía and Grau, 2012). Another consequence of the greater accuracy when initializing features in DI-D is that rate of incompatible data associations which present more than one incompatible pairing is very low, as DI-D generally produces more stable features. This means that the JCBB will prune out a relatively low number of branches, so the binary search tree will be fully built and expanded frequently.

4.5.1.2 Impact of the validation of data association results

It has been discussed earlier how data association validation is necessary for the undelayed approaches to reject hastily initialized landmarks which can prove themselves disruptive for the trajectory estimation and the map build process. It has also been largely discussed how the DI-D initialization generally produces *stronger* features, and the rejection rate is generally lower when compared to undelayed approaches. Considering the discussion in previous section on the cost penalization incurred by JCBB (or similar batch validation techniques), is it really worth? Which could be the actual impact?

FIGURE 4.8 illustrates some of the spurious features that were deemed incompatible in a sample run of the DI-D Monocular SLAM with JCBB (using the same environment as described in section 4.6.1.1). The star markers appear around the point or features that were initialized according to the DI-D methodology, and thus were proved *robust enough* to be tracked (at least temporarily) and achieved enough parallax (though it is evident that in many cases they are not even actual features). It is worth noting that the indoor scenery, being completely artificial, presents a favourable rate of corners and easily detected points of interest, but at the same time presents repeated textures and patterns, structured occlusions, and other inconvenient characteristics.

The frame in FIGURE 4.8 *top left* shows an example of a composite landmark (emerged at the partial occlusion between structure solids) which has been fully initialized, thus passed through the DI-D, but the batch validation of DA has deemed it incompatible and is to be removed. Note how the cords produce several early candidates of similar characteristics, and though most of them will not be initialized, they still pose a risk.

In FIGURE 4.8. *top right* the landmark rejected was deemed incompatible due the repeated pattern with the same design found around, in the cardboard file/box. Note in the same image how there are multiple *early candidates* for DI-D (marked with a *blue cross*), which clearly are not real points (in a 3-dimensional sense), but the result of partial occlusions between structured solids, as found at the border of the right desk. This kind of visual features are generally rejected by the DI-D methodology, as they are not robust enough to be tracked during sufficient frames, but sometimes they are initialized posing a risk to the filter convergence.

FIGURE 4.8 *bottom left* illustrates the risks associated to reflective and curved surfaces, and other commonly found elements with repetitive structure. Note how the landmarks were fully initialized over the desktop PC front, on a curved reflective surface. The left one could be especially disruptive, as there is also a repeated texture, so the data pairing can find a match along the whole surface wherever the reflection slides to. These cases help to remark the relevance of a data association validation, even in the context of the less spurious DI-D initialization, in spite of the JCBB being computationally unfeasible.

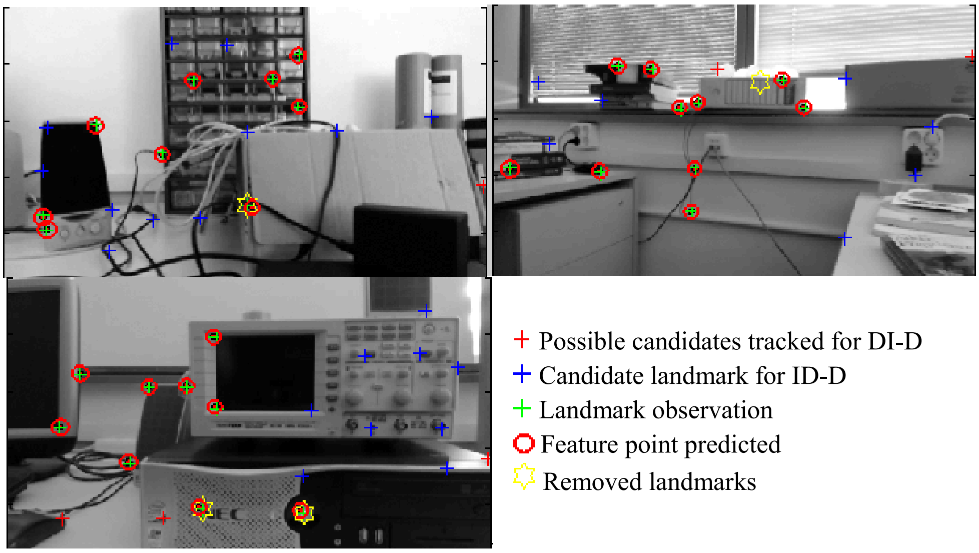

+ Possible candidates tracked for DI-D
+ Candidate landmark for ID-D
+ Landmark observation
○ Feature point predicted
☆ Removed landmarks

FIGURE 4.8: **Top left:** *Incompatibility found at composite landmark.* **Top right:** *incompatibility due repeated design.* **Bottom left:** *incompatibilities produced by reflections on materials and curved surfaces.*

4.5.2 An alternative to JCBB: the HOHCT algorithm

So accounting for DI-D initialization characteristics, a new batch validation technique was proposed, the Highest Order Hypothesis Compatibility Test (HOHCT). This new technique uses the same joint compatibility notion, but the search algorithm is built to exploit said DI-D characteristics to optimize the performance. Two criteria were used to build the algorithm, based on the JCBB prototype test results and the differential features between the undelayed and the delayed initialization of landmarks:

- As the number of incompatible landmarks in DI-D will be generally low, hypotheses with low number of rejections tend to pass the compatibility test; also the objective is to maximize the number of accepted pairs, the algorithm should deal with the *biggest*[22] hypotheses first.
- Even with the matrix inversions optimization, performing the full compatibility test at each node of an exploration tree is too expensive, so, the number of SMD tests to be performed should be minimized.

Note that, once a hypothesis with a given number of accepted pairs is produced, any effort dealing with hypotheses with a lower number of accepted pairings is wasteful, and should

[22] Meaning that they contain the greater number of accepted data pairing possible.

be avoided. JCBB is able to detect these cases, and stops considering them, but only after the compatibility test of the partial hypotheses. This can lead to explore undesirable parts of the tree, where previously found incompatible pairing are to be tested, as per JCBB star node exploration. Then, considering these criteria, the best option is to perform an ordered search, guaranteeing it deals with the *biggest* hypotheses earlier, so when a compatible hypothesis is found, only those with the same number of accepted pairs need to be tested.

The new search algorithm is implemented through a hybrid recursive and iterative algorithm, which considers said criteria to exploit the DI-D features to reduce the computational effort required during the data association step. Once the search starts, all the hypotheses containing an exact number of rejected pairing (initially one), are tested; and if none is able to pass the compatibility test, the number of rejected pairings to be searched for is increased. The pseudocode for this search procedure can be seen in ALGORITHM 4.3 and ALGORITHM 4.4.

Function $(h_i, z_i, S_i, \nabla H_i)$:= HOHCT-test $(h_i, z_i, S_i, \nabla H_i)$

Input:

z_i	matching observations found
h_i	features observation prediction
S_i	innovation covariance matrix
∇H_i	observation Jacobian

Output:

z_i	matching observations with incompatible ones excluded
h_i	features observation prediction excluding those without a match
S_i	innovation covariance matrix with compatible observations only
∇H_i	Jacobian considering only compatible observations

```
m: = Number of Matches in $z_i$
hyp := $[1]^m$                         // Grab all matches
if ~JointCompatible( hyp, $h_i$,$z_i$, $\nabla H_i$,$S_i$) then
    i := 1
    while i< m do          // Hypothesis reducer loop
        (hyp,d2) := HOHCT-Rec(m,0,[],i,$h_i$, $z_i$,$\nabla H_i$,$S_i$)
        if JointCompatible(hyp,$h_i$, $z_i$,$\nabla H_i$,$S_i$) then
            i := m
        else
            i := i + 1
        end if
    end while
    remove incompatible pairings from $h_i$ and $z_i$
    update jacobian $\nabla H_i$ and matrix $S_i$
end if
return ($h_i$, $z_i$, $S_i$,$\nabla H_i$)
```

ALGORITHM 4.3*: Initial HOHCT-test function, SMD-testing the optimistic hypothesis, and iterating the creation of n-ary search trees over the number of pairings to be rejected.*

Function (hyp_b, $d2_b$) := HOHCT-Rec (m, m_{hyp}, hyp_s, rm, h_i, z_i,∇H_i,S_i)

Input:

m	size of full hypothesis
m_{hyp}	size previously formed hypothesis
hyp_s	hypothesis built through recursion
rm	number of incompatible pairings to find
z_i	matching observations found
h_i	features observation prediction
S_i	innovation covariance matrix
∇H_i	observation Jacobian

Output:

hyp_b	best hypothesis found from hyp_s
$d2_b$	best Mahalanobis distance

```
if (rm = 0) or (m = $m_{hyp}$) then
    $hyp_b$:= [$m_{hyp}$ [1]$^{m-mhyp}$]
    $d2_b$ := Mahalanobis ($h_i$, $z_i$,$\nabla H_i$ , $S_i$)
else
    $hyp_b$:= [$hyp_s$[1]$^{m-mhyp}$]
    $d2_b$ := Mahalanobis($h_i$, $z_i$, $\nabla H_i$, $S_i$)
    for r:= ($m_{hyp+1}$) : (m-rm+1) do
        (h,d):=HOHCT-Rec (m, $m_{hyp}$+1, [$hyp_s$0],rm-1,$h_i$,$z_i$, $\nabla H_i$, $S_i$)
        if (d < $d2_b$) then
            $d2_b$ := d ;  $hyp_b$ := h
        end if
        $hyp_s$:= [$hyp_s$1] ;  $m_{hyp}$:= $m_{hyp}$ +1
    end for
end if
return ($hyp_b$, $d2_b$)
```

ALGORITHM 4.4*: Search HOHCT-Rec function, which SMD tests all the hypotheses containing the number of 'rm' rejected data association pairings.*

ALGORITHM 4.3 checks the optimistic hypothesis taking all the pairs *'m'*, and failing it, it starts an iterative process to find the lowest number of data pairs to be rejected so that the joint compatibility test is passed. At each iteration this process will perform the test searching for all the hypotheses which have an exact number of pairs, considering as rejected as much association pairs as the number of times the test has failed, noted as *'i'*. So, after the initial fail, with *'i=1'*, and *'m'* data pairs, the hypotheses tested would include only those containing exactly *m-1* accepted data pairs. As the number of test fails *'i'* increase, the recursive search test will be repeated, searching only hypotheses which include *'m-i'* accepted data pairs, thus avoiding repetition of previously tested hypotheses. Assuming that the first criterion is correct, the number of test fails *'i'*, i.e. the number of data parings to reject, should be low, avoiding having to perform too many calls to the search function.

The algorithm to perform each of the search calls mixes both recursive and iterative steps to build an n-ary tree (ALGORITHM 4.4). This n-ary tree essentially works as a binary tree but allows skipping exploration of nodes (see FIGURE 4.9). To achieve this, the iterative steps add accepted pairs into the hypothesis (noted as *'1'*), and the recursive steps introduce rejected pairs (noted as *'0'*). Thus, in the end, the search is performed on a subtree of the hypothetical binary search tree, and the compatibility test (so the SMD) is only computed on the leaf nodes. By comparison, JCBB evaluates each node of the tree to know if it should cut the branch, so the SMD estimation is computed an exponentially growing number of times. Note also how if the assumptions done at the first criterion are true (the sparse error conditions found in the DI-D initialization SLAM), the whole ordered search (after the optimistic hypothesis fails) will usually have linear cost with the number of landmarks matched (meaning that '*i=1*'), with exceptional cases achieving up to cubic cost over some frames (reaching '*i=3*'). Although cubic, this cost over the whole number of data pairings is very far from the exponential cost (with and average *m* between 15 and 25 depending on the algorithm settings) that a recursion search over a binary tree could suppose, as JCBB.

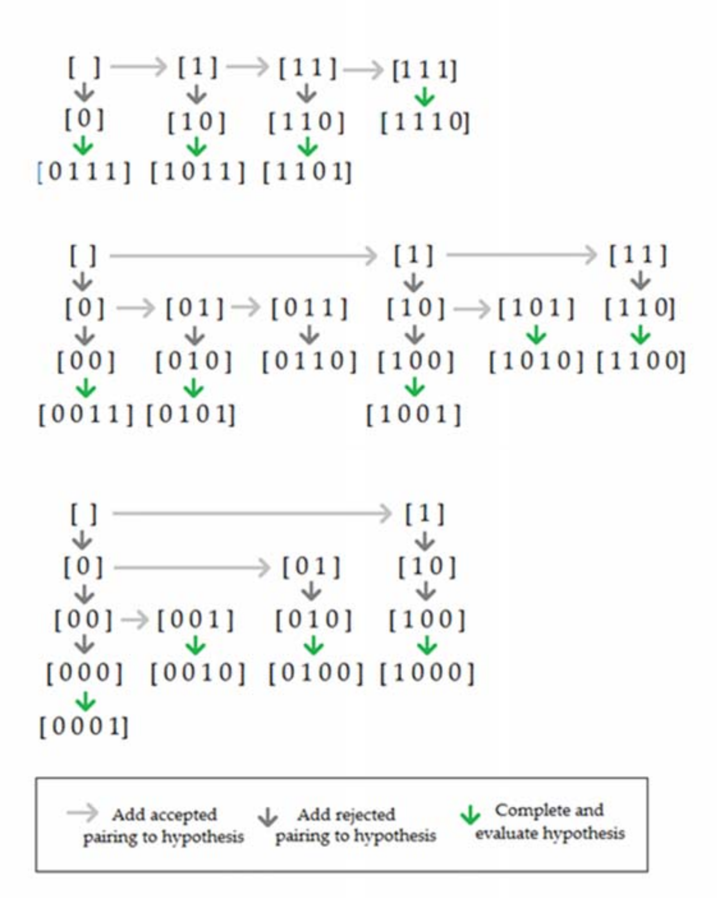


FIGURE 4.9: *Example with* m = *4 with increasing number of pairs to be rejected generating different pseudo-binary trees.*

4.6 Evaluation of the HOHCT

In order to evaluate the validity of the proposed algorithm, and study its usefulness and potential applicability, this section will present the results obtained with the HOHCT and analyse them and their impact both from theoretical and empirical point of view. To perform this study, three different sets of multiple sequences were captured, two of them indoor, with and without an exact ground truth, and one outdoor sequence set with help from a robotic device. These sets were used to experiment with different implementations in order to evaluate the system produced by combining the DI-D Monocular SLAM technique with the HOHCT algorithm; and compare it against the JCBB, the standard in batch validation of data association.

The performance achieved during experiments is described in terms of the two wide evaluation areas: quality of the results, understood as the fitness of the estimations produced for the trajectories and maps; and performance in terms of computational effort and resources required, be it time, computational complex operations with respect to the size of the data, or other metrics.

4.6.1 Mapping and trajectory estimation

A Logitech C920 HD camera was used in experiments to record the sequences. This low cost camera has an USB interface and wide angle lens. It is capable of acquiring HD colour video. The video sequences and images, including those required for the calibration process, were captured with full resolution and colour. This allowed, through image processing, to test the system at different resolutions. However, in experiments, grey level video sequences with a resolution of 480 × 270 pixels, captured at 15 frames per second, were used. It is important to note that all the sequences of video were captured at a relatively low frame rate of 15 frames per second (fps). While this frame rate would increase the difficulty of the SLAM process itself, and make it more prone to error, it would also give a bigger window of time to process each frame in an implementation aiming for real-time. So, although satisfactory results would be easier to achieve assuming 30 fps streams of image (in literature, most of the experiments are reported to be captured at least at 25 frames per second, using high cost IEEE1394 cameras), it has been considered a better option to evaluate SLAM results with at 15 fps, to eventually allow easier implementation into systems with compromised computational budgets, such as autonomous robots, and embedded or mobile systems.

In experiments, the following defaults values for the models parameters have been used: variances for linear and angular velocity respectively σ_V=4(m/s)^2, σ_Ω=4(°/s)^2, noise variances σ_u= σ_v= 1 pixel, minimum base-line b_{min}=15cm and minimum parallax angle $\alpha_{\min}$=5°. The default confidence level for the χ^2 distribution was set to τ = 0.95.

4.6.1.1 Indoor Sequences and Experiments

All the indoor video sequences were captured inside the Vision and Intelligent Systems Research Group laboratory at UPC. For the initial sequence test, a 4m rail guide was assembled in order to provide an approximate ground truth reference, as seen in FIGURE 4.10. Every video sequence on this scenario was captured by sliding the camera (manually) while looking sideways, at different swiftness, over the rail guide. The duration of the different sequences for this scenario ran from 35 seconds to 1 minute (525 to 900 frames) for different runs on the same trajectory, with the camera moved manually. Though deploying the rail to emulate having a ground truth looks like an ardours task, it was initially considered a valuable effort as it could help evaluate the accuracy of the on-line scale estimation. This differentiates DI-D Monocular SLAM from most of the works on monocular SLAM (Davison et al., 2007) (Civera et al., 2010), even when compared with newer works which produce scaleless maps (Engel et al., 2014) (Mur-Artal et al., 2015).

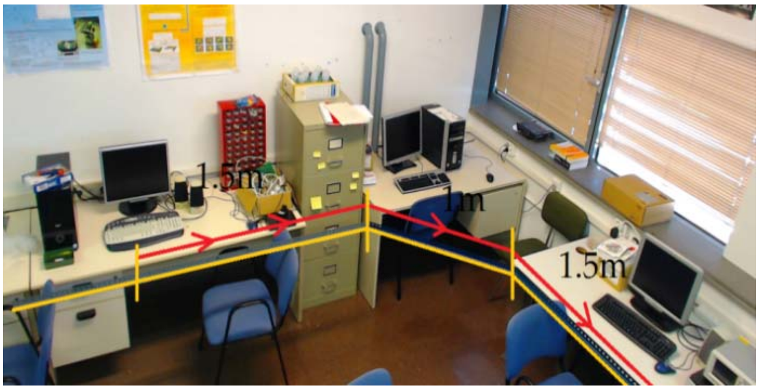


FIGURE 4.10: *First indoor experimental scenario, with ground truth reference rail.*

FIGURE 4.11 and FIGURE 4.12 illustrate the estimated map and trajectory for a selection of experiments of the first indoor scenario set. The left and right columns show the results for each sequence, experiments **a** through **d**, with and without HOHCT validation, respectively. Note that the application of JCBB or any other batch gating methodology based on the joint compatibility notion tested through the Mahalanobis distance estimation would produce similar results, albeit at a different computational cost, as long as the procedure guarantees to find the optimal hypothesis. As it could be expected, the estimations obtained with the HOHCT validation were consistently better. Case **a** shows a sequence with average results obtained without any data association validation. The final position error for the sequence is well over a meter, with a noticeable drift in the odometry estimation. Still the orientation errors are small when compared to the worst cases. On the other side, introducing data association validation reduces drastically the trajectory and map scale error, with the final position accruing an error much lower that the non-validated case (around a quarter of the error, 0.28%).

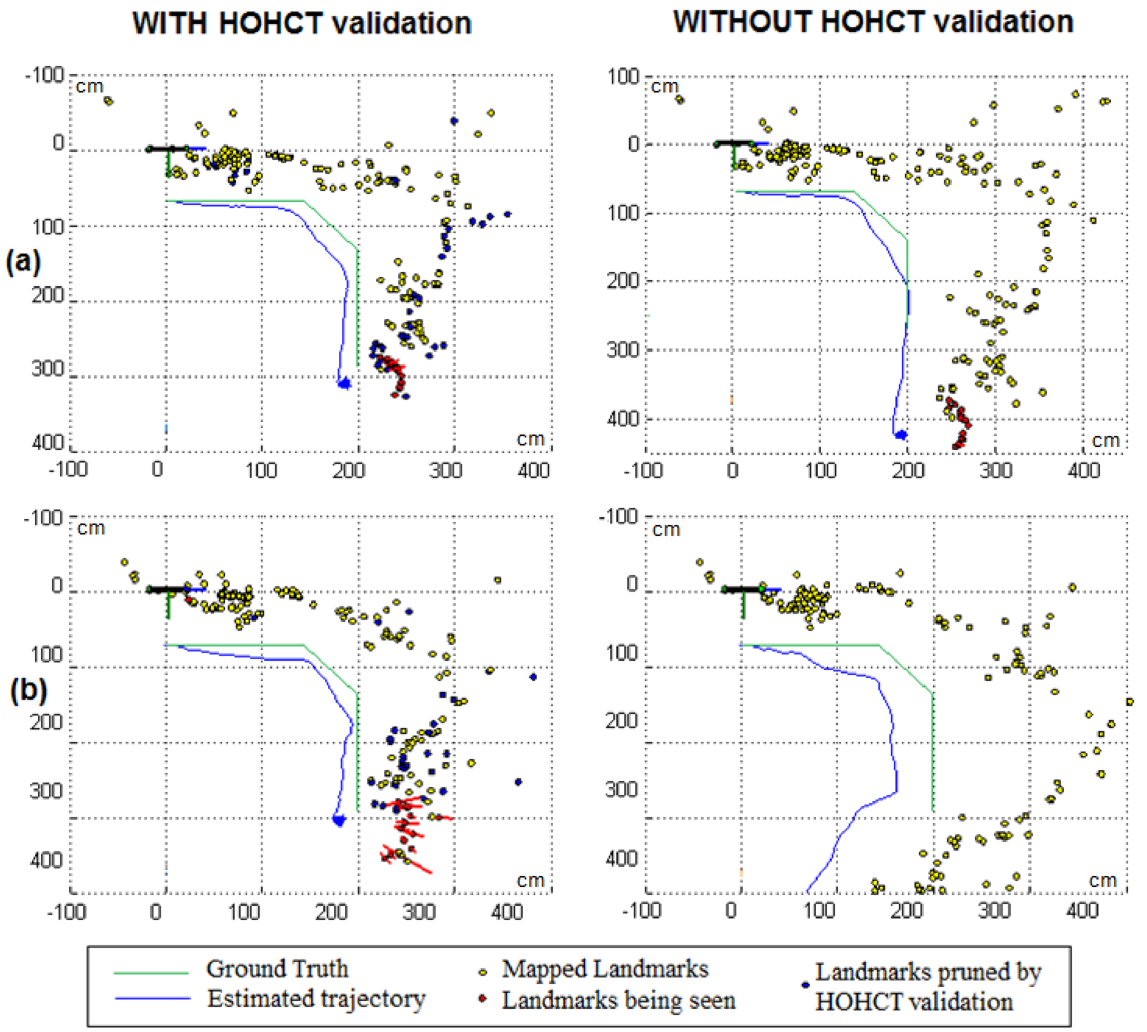


FIGURE 4.11: *Map and trajectory estimation results obtained from two sequences of video.* **Top row:** *case* ***a****, 845 frames.* **Bottom row:** *case* ***b****, 675 frames. Left column displays results using HOHCT, while right column displays results for the same sequences without using HOHCT batch validation.*

Sequence **b** shows how the drift induced by the orientation error disrupts the scale propagation through the Abbe error (Abbe, 1890). Though at the end of the first segment the trajectory estimation looks displaced but of correct magnitude, as if having travelled with wrong orientation, the successive segment introduced growing scale errors. This can be attributed to the orientation presenting enough error to disrupt the depth estimation (thus the linear component of the odometry), but still producing clearly defined straight segments, although incorrectly aligned.

The third sequence (case **c** in FIGURE 4.12) shows an example of one of the worst possible cases (before filter convergence loss, which invalidates the whole procedure), where the scale becomes irrelevant as the orientation error grows to the point of totally disruption the map. Note the gap in the right region of the estimated map without validation, and its distinctive lack of the mapped corner observed on the validated experiment, which presents a good approximation to the scale of the map, but with plenty of orientation drift. Although the batch validated map estimation would not be useful for accurate autonomous navigation, the improvement over the non-validated approach showcases the impact of the HOHCT.

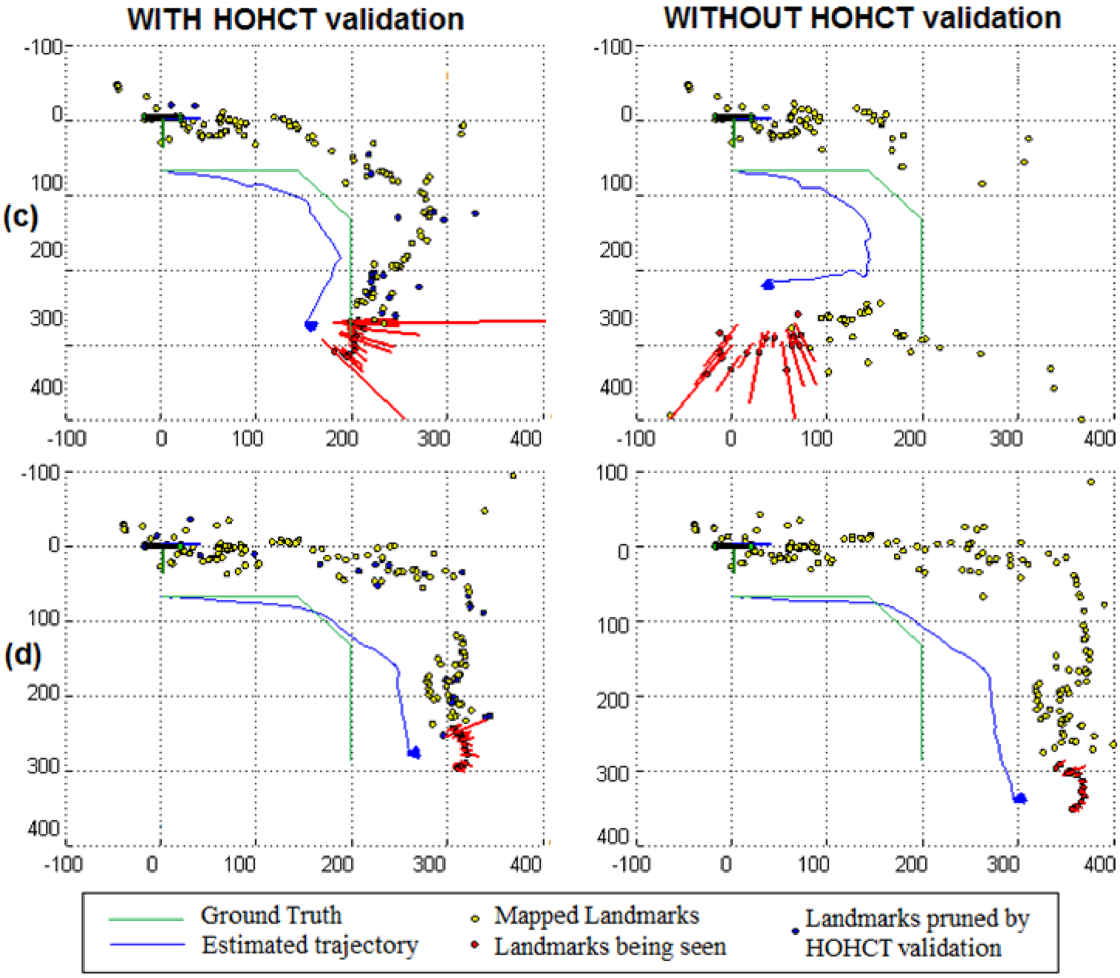


FIGURE 4.12: *Map and trajectory estimation results obtained from two sequences of video.* **Top row:** *case* **c**, *615 frames.* **Bottom row:** *case* **d**, *750 frames. Left column displays results using HOHCT, while right column displays results for the same sequences without using HOHCT batch validation.*

Sequence **d** (shown in FIGURE 4.12) helps emphasize the effects of the validation in scale estimation, showing what it could be described as an average case for the raw odometry and map estimation. In the two experiments, with and without HOHCT validation, the orientation error looks similar, with both trajectories showing the same turning underestimation pattern. On the other side, the magnitude of the linear movements is much more accurate in the validated case, especially on the third segments.

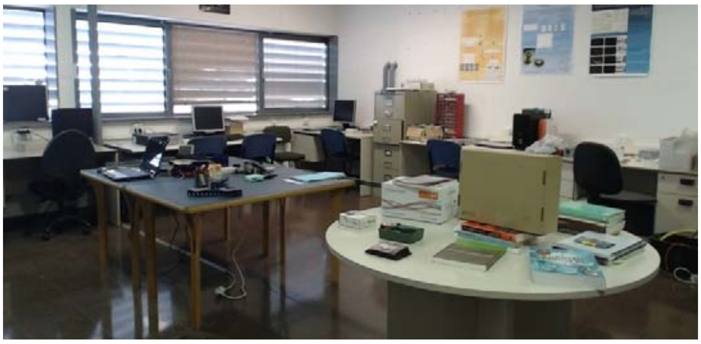


FIGURE 4.13: *Environment used to capture sequences with approximate ground truth for trajectory, second indoor experimental set.*

The results of the first set of indoor experiments showed that the proposed algorithm had a disproportionate impact on the scaling and linear aspects of the trajectory and map estimation, with a more limited effect on the orientation error. As the need for an actual ground truth (built upon rails) limited the possible trajectories, it was considered the possibility that the bias towards better scale estimation was produced by the features of the experiment. To test this, an additional set of image sequences was captured in other parts of the same environment (seen in FIGURE 4.13), presenting a wider variety of trajectories, with curves and twists.

Experiment **e**, in FIGURE 4.14, starts with a U-turn around a table with several objects (the cluttered zone on the centre of the map), and continues along a straight line of three meters. Note how the experiment with HOHCT data validation (FIGURE 4.14 *upper left* plot) displays a blue trajectory which follows approximately the described path, while the same experimental sequence without HOHCT (FIGURE 4.14 *upper right* plot) presents a clear drift in orientation, making a more open turn. Besides, once the turn is complete, the straight part of the trajectory is clearly too long on the *top right* map, probably exceeding the actually travelled distance by one third in this segment (from 3m to about 4m). Experimental sample sequence **f** shows similar results. The experimental trajectory consisted on an almost full turn around a cluttered table. The map with data association applied shows how the estimated trajectory resulted in a more open path than the ground truth; and as expected, the trajectory estimated without data validation resulted in an even greater orientation error, which in turn led to scale drift towards the end. Note that the implemented DI-D approach does not incorporate any loop closing technique, so the results were considered very solid. At the same time, it is clear that ranging from the best cases to the worst cases, the introduction of HOHCT improves the results both in terms of odometry and map estimation. As such, the indoor experimental results presented shown the importance and impact of incorporating a data association validation technique in the context of monocular SLAM. As the data validation rejects erroneous and weak matching features, it helps to reduce the drift, and in many cases, it keeps the EKF from losing convergence capabilities. Specifically, the HOHCT validation test significantly improves the algorithm robustness, by rejecting harmful matches, clearly noted in the improvement on orientation estimation, observed specially in sequences **c**, **e**, and **f**. Another improvement observed was the enhanced preservation of the metric scale on estimations, emphasized in sequences **a**, **b** and **c**. The sequences taken with slower camera movements tend to produce better results, although this can be easily attributed to the low frame rate of the camera used.

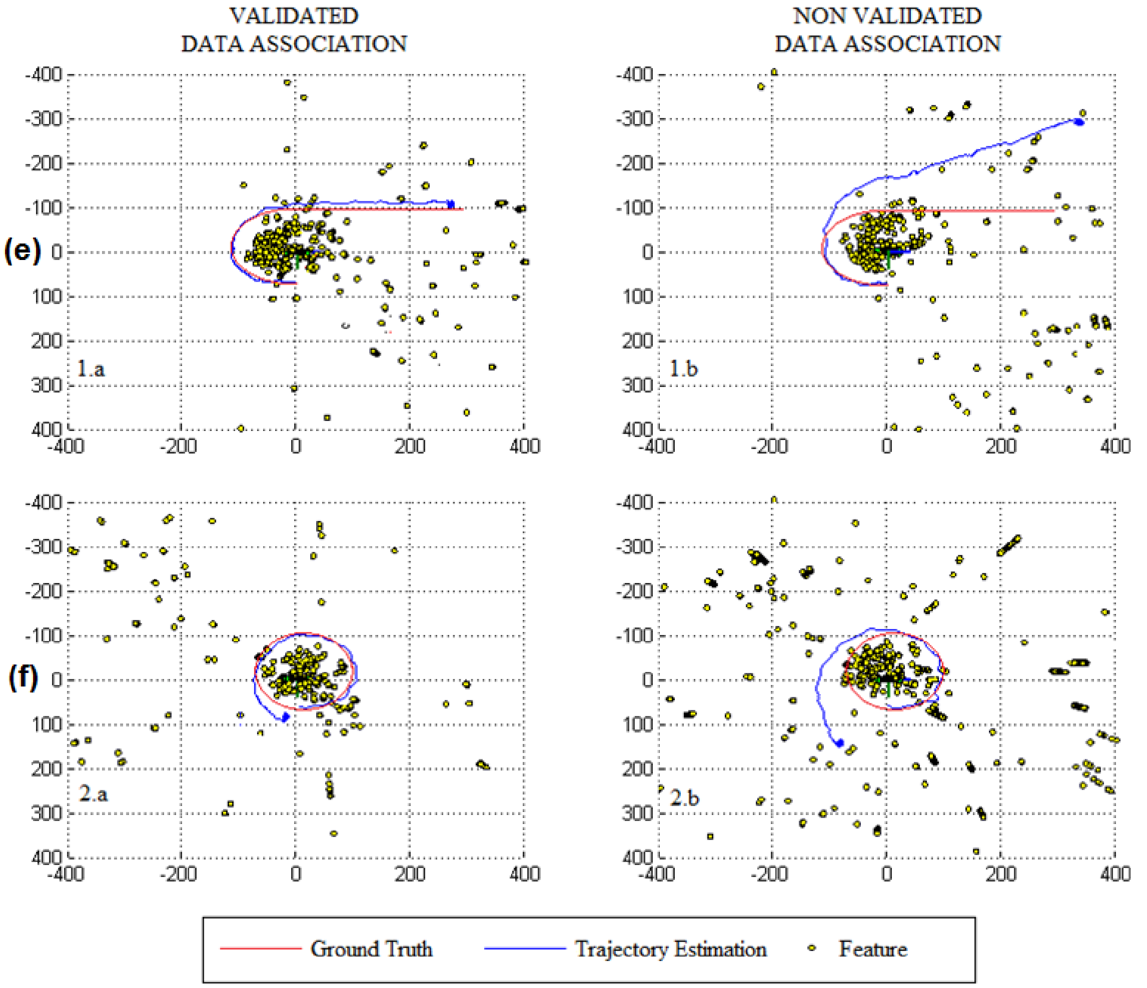


FIGURE 4.14: *Results of two trajectories,* **e** *and* **f**, *with HOHCT applied, at the left; and without at the right column.*

4.6.1.2 Outdoor Sequences and Experiments

Indoors experiments are generally a good fit for monocular SLAM approaches: structured environments, controlled lightning, and plenty of easy to detect artificial features[23]. To test the full DI-D with HOHCT approach in less controlled circumstances, a small set of outdoor image sequences was captured with the help of a robotic platform, which carried the camera looking sideways along a pre-fixed trajectory. The robot platform used was a Pioneer 3-AT, running over an Ubuntu 12.04 distribution with ROS Fuerte as middleware and in order to provide control, navigation, and other required software tools.

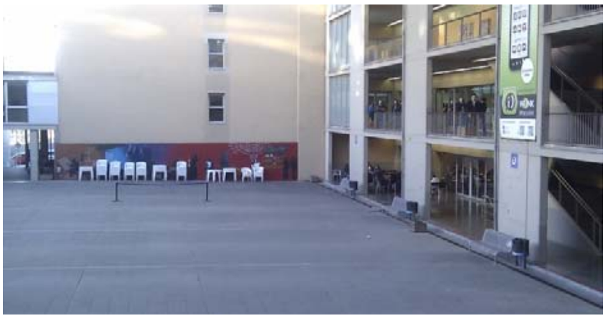


FIGURE 4.15: *Outdoor environment used to record experimental sequences.*

This robotic platform traversed repeatedly a known trajectory in a near courtyard with columns, benches and multiple reflective surfaces among other elements (FIGURE 4.15). This trajectory described an 'L' shaped course running along 12m, with a 90º turn. While going through this course, a camera installed on top of the platform captured the sequences, looking sideways. Although the physical space was still highly artificial and manmade in nature, this courtyard allowed performing outdoor tests, with open space and longer trajectories, and still presenting disturbances from uncontrolled lightning and other difficulties usually associated to outdoor environments.

The sequences were taken with the platform moving at different speeds, ranging from 0.25m/s to 1m/s. Thus, the duration of sequences went approximately from 20 seconds to 90 seconds (600 to 1300 frames) for different takes of the same trajectory. FIGURE *4.16* shows the result of off-line application of the DI-D SLAM technique, with and without application of the HOHCT, with the robotic platform moving at 0.65 m/s. The most notable difference with indoor handheld experiments is the capability to move at greater speeds while keeping filter convergence in the SLAM process. This was due mainly two facts: the robotic platform described a less spurious trajectory, with constant speeds along the straight parts, and smother turns; and the presence of objects at a wider depth range, which allowed keeping better estimation of the orientation. The effects of the HOHCT can be seen in the different trajectory estimations at FIGURE 4.16: while both the SLAM with and without HOHCT are able to estimate quite accurately the length of the trajectory, without data association validation, the estimation drifts greatly, especially in terms of orientation.

[23] Although the risk of repeated patterns make them a double-edged sword.

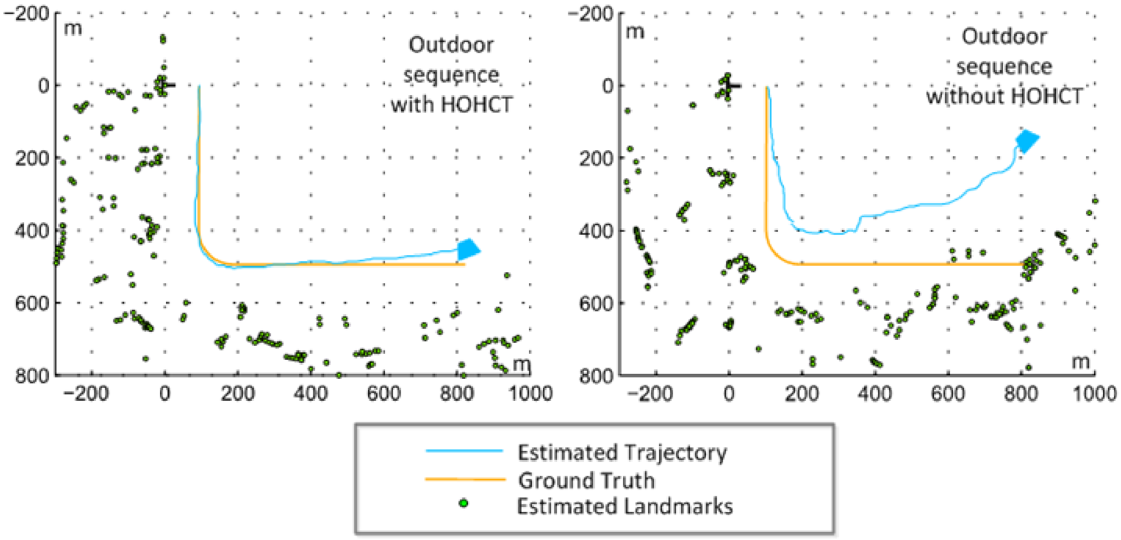


FIGURE 4.16: *Example of outdoor trajectory experiment results with navigation from the Pioneer 3-AT platform at 0.65m/s.*

4.6.2 Performance of the HOHCT

The HOHCT algorithm was developed to enable the introduction of a data association batch validation technique into the delayed inverse-depth monocular SLAM framework (Munguía and Grau, 2012), as other approaches had proved unfeasible due computational costs. The impact of batch validation techniques has been already proved, but they must be still evaluated in terms of the efficiency of the proposed approach.

4.6.2.1 Exploration of the hypothesis space: theoretical costs

The benefits of the application of the HOHCT validation comes together with the addition of the computational cost of exploring an *n*-ary tree (which can be interpreted easily as a binary tree) to build hypotheses and test them at the leaf nodes against the squared Mahalanobis distance. The number of data association pairings defines the size of the hypotheses, so it will also define the magnitude of the search space to be explored as a tree. On the other side, the JCBB estimates the SMD at each node, which includes a matrix inversion operation, unlike HOHCT that performs the test only at leaf nodes. So, although it does not account for the JCBB optimization described in (Neira and Tardós, 2001), we consider that a relevant data point to study the costs of the HOHCT algorithm is the number of SMD tests performed per batch validation process. This assumes that the initial optimistic hypothesis failed the SMD test and the search for an optimal hypothesis is performed; otherwise the cost is of a single SMD for both HOHCT and JCBB.

The cost of using JCBB in terms of number of performed SMD tests, noted as h_n^{JCBB}, can be bounded below the full cost of building the complete binary tree, 2^n, but on average tends to that same cost subtracting all the nodes present in the subtrees pruned due detected incompatibilities, as seen in equation (4.13). This equation (4.13), and equation (4.14), denote as n and r the number of observations matched by the data association process and the number of these association pairings to be deemed incompatible at the optimal hypothesis, respectively. In the upper bound and average behaviour of h_n^{JCBB}, equation (0.13), it can be noticed how the term with the biggest weight is the number of data pairs n, on term 2^n, which increases the cost exponentially. This exponential cost can be relieved by a high number of rejected data pairs, r. But as it has been previously said in this same dissertation, the delayed I-D monocular SLAM introduces low numbers of weak undesirable features that would eventually be deemed incompatible and rejected as r.

$$2^n \geq 2^n - 1 - \sum_{i=1}^{r}(2^{n-i} - 1) \cong h_n^{JCBB} \tag{4.13}$$

The cost of the HOHCT in terms of SMD tests to be performed can be predicted accurately with ease: each recursive search (ALGORITHM 4.4) will be essentially a case of permutations over n with multiplicities of n-i and i, with i being the growing number of pairings to be rejected (until the *biggest* jointly compatible hypothesis is found). Noting down this as an expression, described in equation (4.14), makes it evident that for lower r values the number of terms to be summed will be low (the hybrid search HOHCT-Rec is performed fewer times). At the same time, the cost of each search as function of the number of SMD tests will be dominated by the term $n!/(n-i)!$, becoming linear cost (n), quadratic cost (n^2), and so on for for low i values, $i=1$, $i=2$, etc.

$$h_n^{HOHCT} = \sum_{i=1}^{r} \frac{n!}{(n-i)!\,i!} \tag{4.14}$$

The difference between these different characterizations of cost is shown in FIGURE 4.17. The cost of the JCBB will be bounded above the JCBB expected average (blue), and below the red line picturing the exponential case (red line). The actual cost will lie nearer to the expected average than to the upper limit most of the time, although it is largely dependent on the order of appearance of the incompatible data pairings. On the other side, the HOHCT cost grows following a sigmoid function with respect to the number of data pairings to be rejected (green line). This can be explained observing the cost of each HOHCT recursive search for the different number of rejected pairings (green asterisks). The cost for the HOHCT at each r value would be the accumulated cost of all the HOHCT recursive searches up to itself, and it is intuitive that the bigger ‘search trees’ are built when *n-r* and *r* are balanced[24]. So the HOHCT costs grow faster when *r* is around half of the *n* data association pairings, which gives the sigmoid characteristic to the cost.

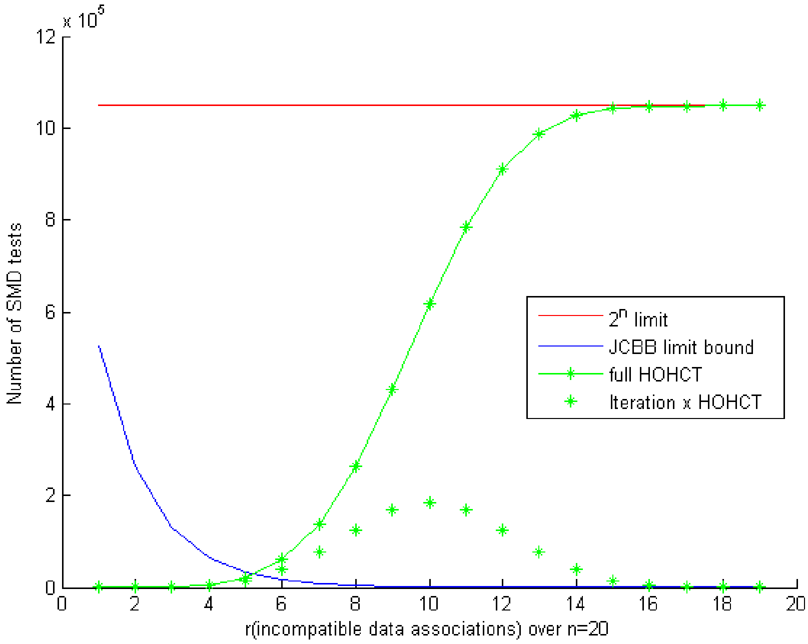


FIGURE 4.17: *Costs in terms of nodes evaluated, with each executing a joint compatibility test (SMD), for a sample group of 20 data association pairings, which contains r =[1 ... 19] incompatible matches. The number of SMD tests performed by JCBB decreases with the number of incompatible pairings, thanks to the pruning strategy. On the other hand, as long as the number of incompatible pairing is low, HOHCT presents a lower cost.*

Accounting for the cost of the HOHCT and the bounds computed for the JCBB cost, a conservative estimation would be that as long the expected number of pairing rejections *r* is bound below a third of the total *n* pairings, the HOHCT will outperform the JCBB in terms of SMD tests performed. Moreover, the HOHCT should present enough advantage to outperform the JCBB even in actual computation time although it cannot benefit from the matrix inversion optimizations.

4.6.2.2 Exploration of the hypothesis space: statistics

The total durations, number of frames, and SMD executions of a batch validation technique in the indoors experiments are shown in TABLE 4.2, accounting for 20 sequences. Although a SMD test is performed at each frame at least, to check the optimistic hypothesis that all the data association pairings are jointly compatible, for the sake of these statistics, we refer as *'application of JCBB/HOHCT/validation technique'* to the cases when said optimistic hypothesis does not comply and the algorithm to search the *biggest* compatible hypothesis is executed. On average, the relevant batch validation algorithm was performed for about a

[24] Otherwise the *n*-ary tree is very deep or wide, but presents low branch expansion.

tenth of the frames (~10.02 %). It must be noted that frequently most of the incompatibilities emerged towards the end of the trajectories when drift is already noticeable, as commented earlier. Thus the number of batch validation searches performed could still be probably reduced further with the introduction of submapping/map splitting techniques to reduce the drift.

TABLE 4.2: AVERAGE AND TOTAL STATISTICS ON SEQUENCE DURATION AND BATCH VALIDATION USE

Metric	**Average**	**Accumulated**[a]
Sequence Length	857.5 frames	17150 frames
Sequence Runtime	57.17 s	1143.5 s
HOHCT searches[b]	87.55	1751

[a]Statistics aggregated over the 20 indoor sequences.

[b]HOHCT/JCBB searches due optimistic hypothesis failing the SMD test.

The number of SMD tests performed on average per video sequence by the different techniques is shown in TABLE 4.3, accounting also for the ratio with respect to the number of average frames and searches performed. As discussed earlier, the count of SMD tests for the JCBB will be equal to the number of nodes build upon the binary three, while in the case of the HOHCT the count of SMD will equal the aggregated number of leaf nodes of the n-ary trees built during the search. The difference in the order of computational cost on average is between two and three orders of magnitude; while an average HOHCT search performed under a hundred SMD tests, the JCBB technique had to compute it over fifteen thousand.

TABLE 4.3: SMD TESTS PERFORMED AT EACH EXPERIMENTAL SEQUENCE ON AVERAGE

Metric	**HOHCT**	**JCBB**[a]
Total SMD tests	1432916	5713.21
SMD test/frame	1671.36	5.18
SMD tests/search[b]	16366.01	65.26

[a]Accounting all nodes during the expansion of the search tree.

[b]HOHCT/JCBB searches due optimistic hypothesis failing the SMD test.

This can be comprehended observing the number of features n considered each time, and the number of data pair rejections r, being the two main factors leading the complexity, shown in TABLE 4.4. Consequently, for the average case, the low number of pairs rejected on average when searching for a jointly compatible hypothesis makes the cost for the HOHCT almost linear with respect the number of data association pairs; while in the case of the JCBB the cost is still dominated by an exponential term (although rather low). This low average number is obtained from a really low counting of data pairs deemed incompatible at each search, see TABLE 4.5.

TABLE 4.4: AVERAGE DATA ASSOCIATION PAIRINGS FOUND AND REJECTED AT EACH VALIDATION SEARCH

Avg. Data Association Pairings	Quantity
Pairings present at each search	15.18
Pairings rejected per search	1.22

Most of the time HOHCT/JCBB searches had to reject only a pair, with linear cost, with a chance of less than a fifth to have to reject two pairs. As this number grows, the chances are reduced, with a very low chance of having to reject 4 pairings, for the used dataset, less than one per sequence on average. It is worth noting how in fact the cases are concentrated on a subset of video sequences representing worst case scenarios, with difficult conditions. Still, with an average number of data pairs of 15.18 at each search, these found worst case costs for HOHCT can be present a computational time requirement not dissimilar to that of the average case using JCBB.

TABLE 4.5: AVERAGE CASES OF MULTIPLE DA REJECTIONS IN A HOHCT VALIDATION STEP

Pairs rejected per search	Avg. per sequence[a]	Percentage[b]
1 pairing incompatible	73.45	83.90 %
2 pairings incompatible	9.90	11.32 %
3 pairings incompatible	3.44	3.93 %
4 pairings incompatible	0.74	0.85 %

[a] For an average sequence over the aggregated 20 indoor sequences.

[b] Over the average.

4.6.2.3 Execution and Profiling: computational performance

A profiling implementation showed that both approaches could achieve real-time performance, with results obtained shown on TABLE 4.6. This table presents the average computation times required per frame for several processes: the raw DI-D Monocular SLAM, the same DI-D approach using a batch validation technique (without the cost of the technique itself), and the penalization of the JCBB and the HOHCT. The results were obtained in an average powered laptop, running on a Linux system (Ubuntu Lucid Lynx), using OpenCV 2.1 as support library.

TABLE 4.6: AVERAGE TIMING OBTAINED BY THE DI-D ALONE, WITH BATCH VALIDATION, AND BY THE JCBB AND HOHCT PROCESSES

Process measured	**average time (ms)**	***σ***
DI-D MonoSLAM[a]	37.98	7.12
DI-D MonoSLAM with validation[b]	38.43	6.93
JCBB[c]	24.58	16.35
HOHCT[d]	2.36	4.37

[a] For an *average sequence* over the aggregated 20 indoor sequences.

[b] HOHCT/JCBB searches due optimistic hypothesis failing the SMD test.

[c,d] JCBB/HOHCT penalization added to DI-D MonoSLAM with validation (a).

It is noticeable that the Delayed I-D SLAM itself is a computationally expensive procedure, requiring by itself 37.98ms per frame on average. This statistic grows with the introduction of a batch validation technique, as they introduce a slight penalization in the form of computational effort spent in map management and state augmentation (removing the non-compliant landmarks and searching for new ones). The JCBB and HOHCT produce each one a noticeable penalization by themselves, with the penalization for the JCBB being an order of magnitude greater. This difference is lower than what TABLE 4.3 would predict, but the number of search does not account for the reduction to quadratic cost of the iterative matrix inversion required, which cannot be implemented into the HOHCT method. This optimization hugely reduces the cost of the SMD tests in the JCBB, partly compensating for the wider hypotheses space. Also, this means that the cost for each test is cubic (n^3 w.r.t. the number of landmarks) for the HOHCT, and the advantage comes from the lower number of SMD tests. Given the cumulative nature of the cost for HOHCT (in SMD tests as in 4.6.2.1) seen in FIGURE 4.17, it is clear that the advantage of the HOHCT is clearly local and depends intrinsically on the quality of the features initialization process and the robustness of the data association process.

4.7 Conclusions

The data association problem has been studied in this chapter, starting with a theoretical overview of the main procedures, until dealing with the actual problems found in the monocular DI-D SLAM. The main aspects dealt with were on one side, the correlation based search, and the suitability of the available operators, and on the other side, the need of introducing a data association validation procedure. This would lead to the development of the HOHCT algorithm, which would constitute one of the main contributions of this dissertation.

The testing of the different matching operators was performed in order to evaluate with some objective metrics not subjected to sampling bias or unknown effects produced by probing them inside the monocular SLAM technique. The robustness of the delayed I-D monocular SLAM is known, and would have impacted any study, so the tests were performed independently. As many of the literature (commented in section 4.3.2) already discussed the operators performance and efficiency against lighting variations, our tests focused on the disturbances that were expected to have greater impact: the different kinds of noises expected to be introduced by the utilization of inexpensive sensors; and the motion blur, typically aggravated by the CMOS technology. The tests modelled said disturbances over a set of images, from insignificant levels to those at the limit of what could be realistically expected, and tested each operator. As it was expected, the cross-correlation based operators outperformed those based on aggregation/squared aggregation. To conclude, the chosen operator was ZNCC, which presented the robustness to motion blur of the cross-correlation technique, with the invariance to contrast variation from the normalization and to illumination due the zero-mean modification.

As it is discussed and probed in section 4.5.1.2, even a robust SLAM technique like the DI-D monocular SLAM can benefit from the introduction of a validation method. The main characteristic of the DI-D monocular SLAM (discussed in Chapter 3) is that landmarks are only introduced into the EKF once the depth estimation is accurate enough, finding this estimation through the parallax effect. This introduces a slight computational burden on the algorithm, compensated by the fact that as the information about landmarks present at the map and filter is more accurate, the filter can proceed with fewer landmarks mapped than in the undelayed approach. Although the landmarks mapped are highly precise, it is still needed a data association gating technique to treat with multiple disruptions that may arise from incorrect or inconsistent matching obtained through active search.

The computational issues produced by the JCBB introduction (the golden rule in data association for a decade) in the early tests motivated the development of the proposed HOHCT, the Highest Order Hypothesis Compatibility Test. This techniques is largely based in the Squared Mahalanobis Distance test, just like the JCBB, but optimized the search to prioritize hypotheses based on the order of the solution.

Both the effectiveness and the efficiency of the HOHCT have been validated theoretically and experimentally, including indoor and outdoor experiments. These experiments results show how the introduction of the batch validation based on joint compatibility improves the technique resilience to erroneous data association and false features or landmarks, produced by difficult illumination and feature detection errors. At the same time, the HOHCT costs have been studied and compared to that of JCBB. While having worst case scenario of exponential cost, just like the JCBB, the HOHCT has been probed to tend most of the time to the linear, quadratic and cubic cases. This tendency to linearity of cost has been probed experimentally, compiling statistics over tens of sequences. It is worth noting that, as it is discussed in sections 4.6.2.1 and 4.6.2.3, the structure of the JCBB enables using matrix inversion optimization, while HOHCT algorithm makes this optimization much more complex to apply, with greater memory requirements. In any case, the HOHCT clearly outperforms JCBB in the context of the DI-D monocular SLAM by over an order of magnitude in the average case.

Part III

Collaborative sensing for Visual SLAM

Robot rebellion scene in the play *R.U.R.,Rossum's Universal Robots*, by Karel Capek(1921). Though conceived as organically built machine-workers, the term robot (‘*robota*’) was first coined referring to this caste of slaves.

III.A Introduction

The general monocular EKF-SLAM procedure is based on detecting points of interest, chosen between those considered to be landmarks and introduced into the EKF, and tracking them through frames, estimating both their pose and camera odometry, as described during Part I of this dissertation. The estimation process is based on probabilistic filtering, where an initial prediction step makes a prediction of the movement, and a further update (or correction step) compares the predicted observations obtained according to the movement prediction with actual observations from the sensor. While the undelayed approaches try to chose the points to become landmarks and initialize them when seen for the first time, the delayed approaches generally rely on obtaining a previous depth estimation. These two types of strategies define many characteristics of the SLAM procedures. As undelayed approaches try to use point features as landmarks just after have been seen, the points are quickly introduced into the filter, accepting many outliers that have to be validated later. This validation step generally removes many points, thus an undelayed approach needs to add constantly new feature points. On the other side, delayed approaches track and estimate the points before using them. Although a validation process is still required, the used landmarks are generally more stable and reliable. This way, while the undelayed approaches put less effort per landmark during initialization, the performance is similar compared with delayed approaches, using less points and with computationally cheaper validation algorithms, as shown during Chapter 4. The diagram in Figure III.1 shows the process steps of the delayed I-D EKF SLAM, as an example of a delayed feature initialization EKF architecture, including the initial state initialization through sythetic features.

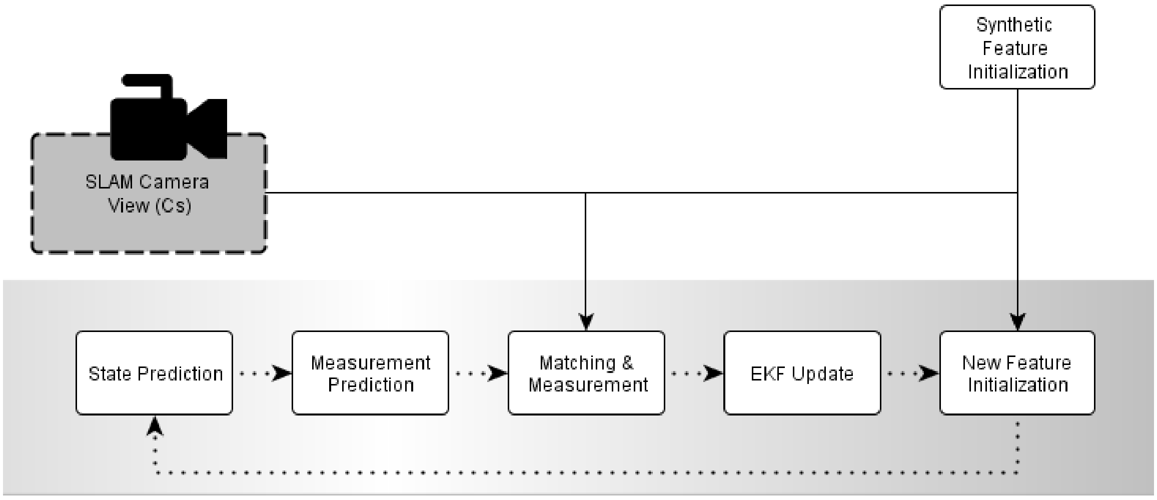


Figure III.1:*Delayed inverse-depth (DI-D) Monocular EKF-SLAM.*

All the developments commented, including those presented earlier in this dissertation, generally assume that the robotic devices where the mapping and localization tasks are performed are autonomous robots operating as stand-alone units. This paradigm, though clearly inherited from the most utilitarian conceptualization of robots[25], is still challenged both from a practical and a philosophical point of view. Although fully autonomous robots with capabilities to substitute human elements in a given system have been developed, these work only for constrained problems. Even the most advanced robots lack the generality and adaptability required to match human versatility, both in terms of physical

[25] Where a robot replaces an individual human.

actuation power and mobility in relation to mass, and in terms of '*intelligence*'[26]. From a philosophical perspective, the idea of designing machines able to match mankind both physically and cognitively still makes the general public and many members of the academia uncomfortable.

One of the answers to these challenges comes from the Human-Robot Interaction (HRI) field: note, that while a human can do many different things, robots can generally outperform them in those scenarios and tasks for what they were designed. Thus, it is only natural that the SLAM problem would be studied from a HRI perspective, where robots and humans collaborate in exploratory tasks, exploiting the human adaptive nature and the accuracy and repeatability of robotic systems to measure the environment.

Another widely used form of dealing with the limitations of robots when treating the SLAM problem is the introduction of multiple agents: both as repeated instances of a generic robot, which can explore the environment through different pathways, or as sets of different devices, each one with different features and capabilities. These approaches, collectively known as collaborative or cooperative SLAM, generally rely on each of the robotic devices solving locally the SLAM problem so that they produce an initial solution which is used as a basis to join the different estimations and compute a global solution which joins data form all of them.

III.B HRI in SLAM

In HRI collaborative context, the SLAM problem has been studied by several works in the domains of emergency response and companion/assistant robotics. While the specific properties in each domain may vary according to the application, the need for indoor and outdoor SLAM techniques where the human component is present as a key factor is clear. In (Kleiner et al., 2007), for example, large areas are explored by a wide group of persons and robots, where the human carry a wearable device, while in (Fallon et al., 2012) a human explores and maps a building while carrying a mapping robotic device. Introducing HRI within the classical SLAM framework usually means increased complexity, like having to deal with dynamic objects in the vicinity and increasing the multimodal range of sensors. All these issues have responses in the SLAM research, thus it is better to concentrate on the new possibilities, such as trying to improve the depth estimation, or overcome other challenges. Thus, exploratory HRI opens the door to improve known mapping techniques exploiting the opportunities provided by the human component.

[26] However it is defined. Although scientific community finds agreeing on a definition for intelligence a hard to solve problem, it is indeed agreed that AI research has struggled to deliver on the promises it made in the 1950s decade.

Another field of application where SLAM approaches are tightly integrated within HRI frameworks is the assistance robotics. In (Cheein et al., 2010) a semi-autonomous robotic wheelchair combined an EKF-SLAM with LRF and a muscle-computer interface (MCI) adapted to the disabilities of the user. This allowed the chair mapping the environment in real-time and at the same time learn to interpret electromyographic signals obtained through the MCI, allowing semi-autonomous navigation guided by the user.

III.C Collaborative SLAM

The collaborative SLAM problem is closely related to other problem in multi-robot system, such as multi-robot tracking (Mazo et al., 2004), cooperative localization (Spletzer et al., 2001), distributed multi-view reconstruction (Seitz et al., 2006), etc., generally englobed in the fied of distributed perception and estimation. Many solutions, not only to collaborative SLAM, but to the other cited problems, are based in decentralized data fusion (DDF) approaches (Durrant-Whyte et al., 2001), where robot and mobile sensors are to infer the variables from measurements and other data communicated by nearby robotic devices. These kind of frameworks present many advantages, like good scalability and resilience to failure of a component thanks to being of decentralized nature.

When dealing specifically with the SLAM problem, one of the first works dealing with cooperative estimation and positioning was (Kurazume et al., 1994), where two groups of robots alternated in moving and taking measurements of each other. In (Roumeliotis and Bekey, 2002) another of the seminal works was presented, where a fully distributed EKF estimation algorithm was used satisfactorily under the assumption that robots could take relative pose estimation w.r.t each other and a global frame. Similarly to the general SLAM problem, early works were based on EKF methodologies, which remain the most popular with newer non-linear optimization based approach, but there are also plenty of works based on other techniques like particle filters, as (Fox et al., 2000) and (Carlone et al., 2011).

The early works discussed and many newer approaches, like (Bailey et al., 2011), present one common feature: they deal with observations considering robot states from the same time instant. The inherent restrictions of the SLAM problem refer to space: in a multi-robot/multi-agent SLAM scenario what is needed is that the different observations pertain to the same scene. But there is no inherent time restriction to the SLAM problem: a robot could stop for a period of time, and retake the SLAM task from the same place, or a near place if the SLAM approach can solve the loop closure/place recognition problem. Then, collaborative SLAM approaches that can work without temporal concurrency restrictions need to work with observations not formulated w.r.t. the robot states at a given time, but referenced to other variables. These variables can be landmarks or set of robots states at different time instants.

The research into this scenario has produced approaches based in well-known techniques. The SAM (Smoothing and Mapping) framework (Dellaert and Kaess, 2006) has been expanded to deal with multiple agents by several authors. In (Andersson and Nygards, 2008) the authors presented the collaborative SAM, C-SAM, which developed support for the multi-robot case based in a centralized framework, while in (Cunningham et al., 2010) an extended formulation of the SAM problem within the DDF framework was presented.

Other works have dealt with specific challenges within the problem, such as unreliable communications and initialization requirements. In (Walls and Eustice, 2013) and (Walls et al., 2015) the authors developed a cooperative localization method based in the decentralized extended information filter (DEIF) considering low-bandwidth and very unreliable communications for underwater operation. With respect to initialization related challenges, one of the most usually assumed restrictions is that the robots known the exact spatial relations between each other at the start, thus enabling a shared reference frame. Works like (Howard, 2004) and (Zhou and Roumeliotis, 2006) started to work towards removing these constraints, and have also helped develop the previously commented branch of multi-robot SLAM with agents separated in time.

Chapter V

Collaborative Sensing in Feature Initialization

Mercury and Argos (*'Mercurio y Argos'*) by Diego Velázquez (1659), depicting the hundred-eyed giant Argos Panoptes, the *all-seeing one*. The idea of perfect observability has fascinated scholars of multiple disciplines, introducing derivatives of the *panoptic* concept in geometry, architecture, and sociology, not always for the betterment of humankind (e.g., Bentham's Panopticon and Foucault's Panopticism)

5.1 Introduction

Some of the challenges in the delayed feature initialization (Munguía and Grau, 2012) discussed in previous chapters could be dealt with a different approach in HRI cooperative context. After improving the robustness of the DI-D monocular SLAM through the introduction of a batch validation technique with better computational costs than the competing approaches in Chapter 4, the requirement of an initial scaled state estimation is one of these challenges that can benefit from solutions considering a collaborative framework. Work on this requirement can improve the generality of the delayed I-D SLAM approach, at the cost of producing a more specialized version dependent on the features of the designed collaborative framework.

As it is discussed in III.B, several works have considered the option of introducing the human factor into solutions to the SLAM problem, especially in the field of urban search and rescue robotics (USAR). In search and rescue (SAR) operations, it is common for humans to wear robotized equipment, in the form of wearable smart sensors. These sensors usually include a camera device with streaming capabilities so that the image feed can be observed and recorded. The robot platforms normally deploy themselves a combination of exteroceptive and proprioceptive sensors to perform localization and measurements of the environment.

For the sake of this work, a sample robotic platform system is considered, which will work as a part of a human-robot collaborative exploration team. As the robot is assumed to operate at least in a partially autonomous manner, it must have the sensors required to perceive the environment, which can be used to measure where the human component is w.r.t. the robot. At the same time, the human is supposed to wear robotized equipment, which includes a camera and an AHRS. The data of these sensors is to be used to solve the initial scale initialization challenge and improve the general feature initialization procedure. In order to produce a solution which can be exported to other framework, it is required that is decoupled from the general delayed EKF-SLAM as much as possible. Thus, the sensors deployed on the human are considered part of a *virtual sensor* which is not always available, which enables depth estimation of features for initialization. The general EKF-SLAM formulation will remain largely the same except for the initialization of features, where this *virtual sensor* is considered to switch on and be used to produce the depth measurements.

In this chapter, besides a new solution to the initial scale initialization challenge, a non-constant multiple view estimation of depth technique for feature initialization is proposed within an EKF monocular SLAM process. After framing the basics of the solution within the delayed I-D problem and the merits and disadvantage of stereo vision for the problem proposed, the new processes introduced are described in theoretical and implementation terms. To conclude, the system is validated through theoretical and experimental results.

5.2 Problem statement

Let us assume the presence of a secondary camera device worn by the human, which moves freely (thus noted as free camera, or C_f), without any way to predict position or orientation; but with an approximately known translation[27] w.r.t. to the camera performing SLAM (known as SLAM camera or C_s), $\mathbf{r}^{Cs}{}_{Cf}$, measured through the robot sensors. If it is also assumed that we the orientation of C_f in global coordinates, $\mathbf{q}^{WC}{}_{cf}$ is known, with an AHRS rigidly solidary to the camera, the whole pose (as position and rotation) can be retrieved

[27] Or an approximation deemed good enough.

both w.r.t. the global frame and w.r.t. C_s frame. As this camera C_f will be used only as part of the *virtual sensor* for depth estimation operation, and its pose is retrieved from other sensors, it does not produce explicit observations into EKF-SLAM filter, so its state is not modelled into it.

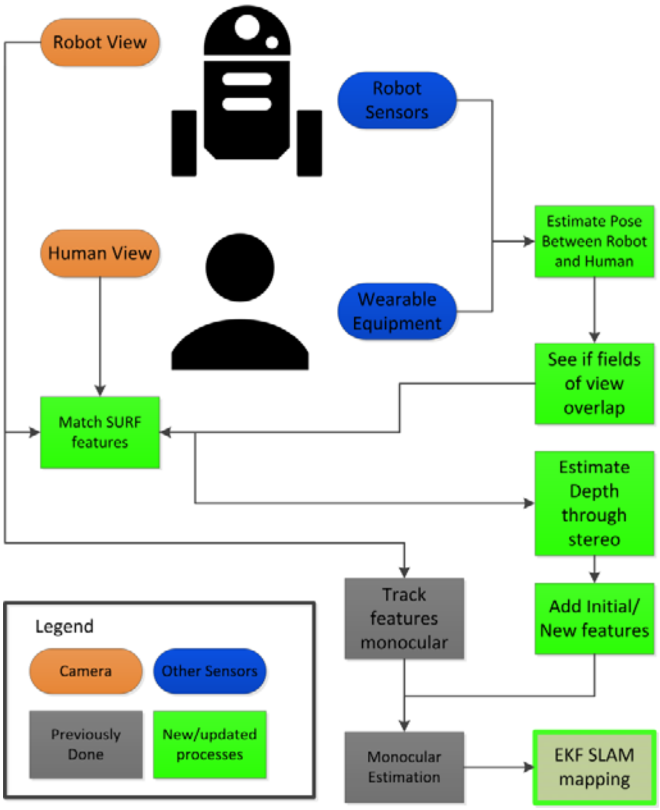


FIGURE 5.1: *System diagram with the general structure of the human-robot team assumed and the steps to produce the EKF SLAM solution. Green boxes denote the new processes added to the original delayed I-D monocular SLAM.*

The new augmented state vector will consider $\hat{\mathbf{x}}_{cs}$ and the landmarks, in a similar manner to the method described in Chapter 2 and 3:

$$\hat{\mathbf{x}} = [\hat{\mathbf{x}}_{c_s} \quad \hat{\mathbf{y}}_1 \quad \dots \quad \hat{\mathbf{y}}_n]^T, \tag{5.1}$$

$$\hat{\mathbf{x}}_{c_s} = \left[\mathbf{r}_{c_s}^{WC} \quad \mathbf{q}_{c_s}^{WC} \quad \mathbf{v}_{c_s}^{W} \quad \boldsymbol{\omega}_{c_s}^{W}\right]^T ; \tag{5.2}$$

where $\hat{\mathbf{x}}_{cs}$ contains the position $\mathbf{r}^{WC}{}_{cs}$, orientation $\mathbf{q}^{WC}{}_{cs}$ and velocities $\boldsymbol{\omega}^{W}{}_{cs}$ and $\mathbf{v}^{W}{}_{cs}$. The pose of C_f w.r.t to world coordinates is found using equation (5.3), where $\mathbf{r}^{Cs}{}_{cf}$ denotes the position of C_f measured w.r.t. C_s, and $R^{CW}(\mathbf{q}^{WC}{}_{cs})$ denotes the rotation matrix obtained from the orientation of C_s to transform the coordinates to world frame reference:

$$\mathbf{r}_{c_f}^{WC} = \mathbf{r}_{c_s}^{WC} + R^{WC}(\mathbf{q}_{c_s}^{WC})\mathbf{r}_{c_f}^{C_s} \tag{5.3}$$

Once the pose of both the cameras is estimated with the sensors deployed, the strategy to follow consists in: firstly, join observations from both cameras by solving the correspondence problem[28] if both camera sensors are capturing concurrent sections of the environment; then, proceed to use this data to initialize the feature. The effect achieved by this strategy is that of replacing the temporal separation of the observations of a given landmark seen by the same sensor with spatial separation obtained by introducing a secondary camera in the *virtual sensor*.

To implement the strategy just described there are 3 different problems that need to be addressed in order to use the multiple view data to initialize the features:

- Determine if the different camera sensors measurements pertain to the same scene.
- Find the matches between the point features of relevant landmarks in the different images.
- Compute the depth exploiting the data from both sensors accounting for the actual knowledge about them, producing a *virtual sensor* that temporarily enhances the monocular camera C_s with depth measurements.

This procedure operates as a non-constant multiple view estimation of depth, which can provide the initial scaled state required in the delayed I-D monocular SLAM without relying on synthetic or previously known features. Note that the most common strategies used to deal multiple view measurement and estimation usually rely in epipolar geometry-based stereo vision. Thus, the main advantages and inconveniences of using stereo vision within the context of the studied problem are discussed.

5.3 Applicability of Epipolar Geometry-based Stereo Vision

Classical stereo approaches, like (Loop and Zhang, 1999), (Fusiello et al., 2000), and (Howard, 2008) rely on epipolar geometry to create a calibrated camera rig with multiple geometrical constraints. These constraints typically include that both cameras projection planes need to be coplanar in world coordinates, and generally with parallel axes. These configurations allow optimizing the correspondence problem as the match on an image of another's image pixel will lie in the corresponding epipolar line, and rectification can turn them into straight-lines, parallel to the horizontal axis, which in turn is parallel to the baseline. Several works have dealt with rectification of stereo images for unrestricted pose cameras, both calibrated (Fusiello et al., 2000) and uncalibrated (Fusiello and Irsara, 2008), (Kumar et al., 2010).

In (Fusiello et al., 2000), the author detailed the first method to rectify stereo pairs with any given pairs of calibrated cameras. The method is based on rotating the cameras until they have one of their axis aligned to the baseline, and forcing them to have their projective planes contained within the same plane to achieve horizontal epipolar lines. Other works

[28] Noted as in the field of stereo vision as it is essentially a multiple view scenario.

have proposed similar approaches to rectify stereo pairs assuming calibrated, uncalibrated, or even multiple view stereo configurations (Kang et al., 1995) (Gallup et al., 2008), including automatized variable stereo rigs (Fanto, 2012). These approaches need to warp both images according to the homography found (see FIGURE 5.2 versus FIGURE 5.3), and in some cases producing great variations in terms of orientation and scale (FIGURE 5.4).

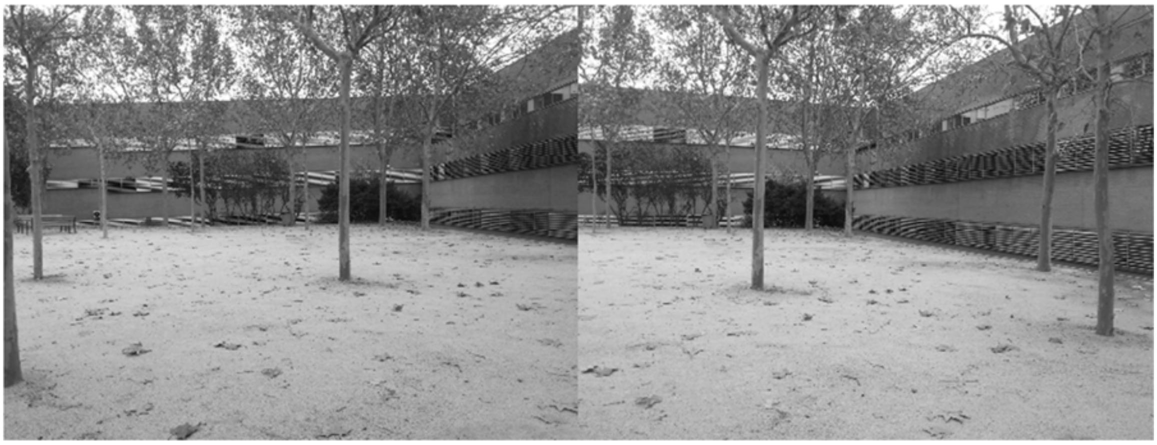


FIGURE 5.2: *Pair of images captured in an outdoor environment.*

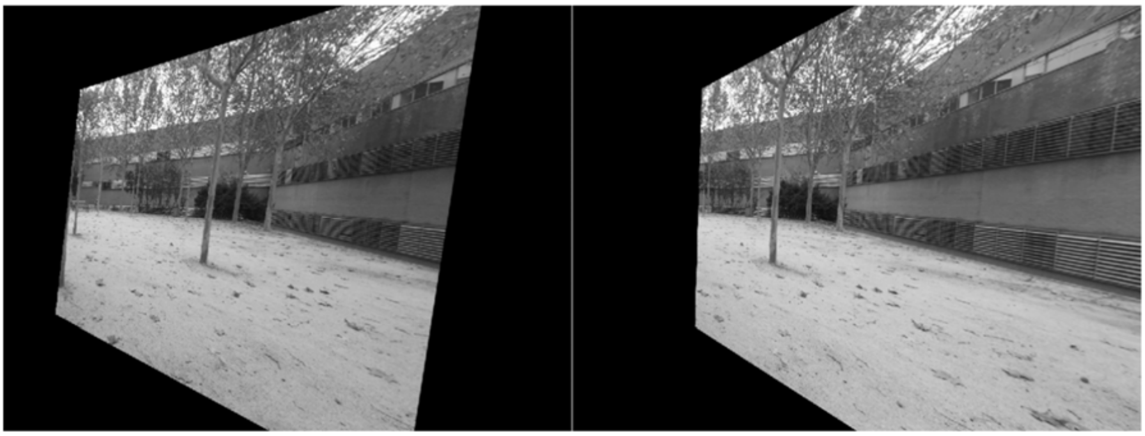


FIGURE 5.3: *Pair of images rectified according to the method described in (Fusiello et al., 2000).*

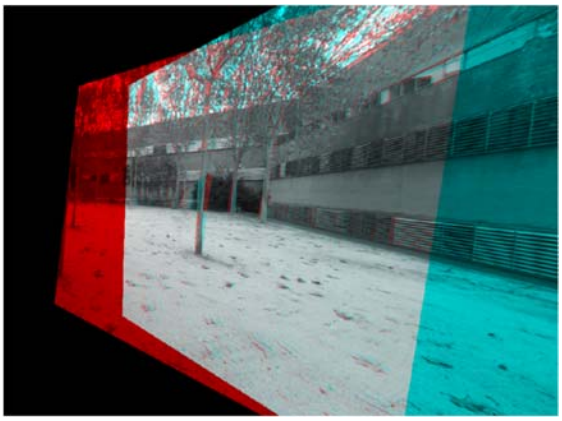


FIGURE 5.4: *Pair of images rectified and matched.*

These characteristics make the introduction of stereo vision (based in epipolar geometry) to enable the collaborative initialization of features in the proposed system less desirable: the main advantage of said strategies would be optimizing the computational efforts required to solve the correspondence problem by working over the reprojected images, where the correspondence problem would be limited to the area around the epipoles, which can be easily predicted. At the same time, this implies computing the homographies required to reproject images, and perform the reprojection operations for the whole images. This would mean that the reduction in computational time required to solve the correspondence problem would be just spent in image processing operations to perform the reprojection and additional data processing to ensure that estimation of the relative pose between cameras is accurate enough.

Then, considering the characteristics of the base monocular SLAM framework studied, dealing with multiple view features without stereo vision-based correspondence should not prove specially challenging, as the multiple view feature initialization operation will be performed sparsely. Working directly over the same frames, without the stereo photogrammetry, will also allow optimizing the process exploiting direct geometric intuitions based on the parallax threshold.

5.4 Feature initialization under unreliable multiple view sensing

5.4.1 Multiple view sensing for scaled feature initialization

The requirement of metric scale initialization of the DI-D method can be solved assuming the presence of the described cooperating camera C_f. The previously used methodology (Munguía and Grau, 2012) required the presence of a set of known, easily identifiable features to estimate them initially through the PnP problem (Section 3.4.2). Then, assuming that at the start of the exploration a cooperating, free moving camera is near, the data from this camera (C_f) can produce the depth estimations required through multiple view photogrammetry. A diagram of this multiple view estimation process is shown in FIGURE 5.5.

Initially, the poses of the SLAM camera C_s and the free camera C_f are known, as camera C_s starts at the origin of the map[29], and the different sensors present in the system allow estimating the translation from C_f to C_s in world coordinates, $\mathbf{r}^{WC}{}_{cf}$, as described in equation (5.3). Then, a limit to bound the distance l at which all landmarks with minimal parallax α_{smin} can lie at most can be approximated:

[29] Like most of the vision-based SLAM approaches, the map is generated w.r.t. to the camera so initially is at the origin of the *world coordinates* of the map built.

$$l = \left\| \mathbf{r}_{c_f}^{WC} \right\| / \sin\left(\alpha_{s_{min}}\right). \tag{5.4}$$

This *l* distance is used to scale a model of the field of view of each camera, built using the respective Kcf and Kcs intrinsic camera matrices, and their known poses. Each field of view is modeled as a pyramid in $\mathbb{R}^3$, that is, a set of points with the apex points positioned in the respective optical centres of Cf and Cs, and the bases parallel to each camera projection plane, at a distance *l* along the visual axis.

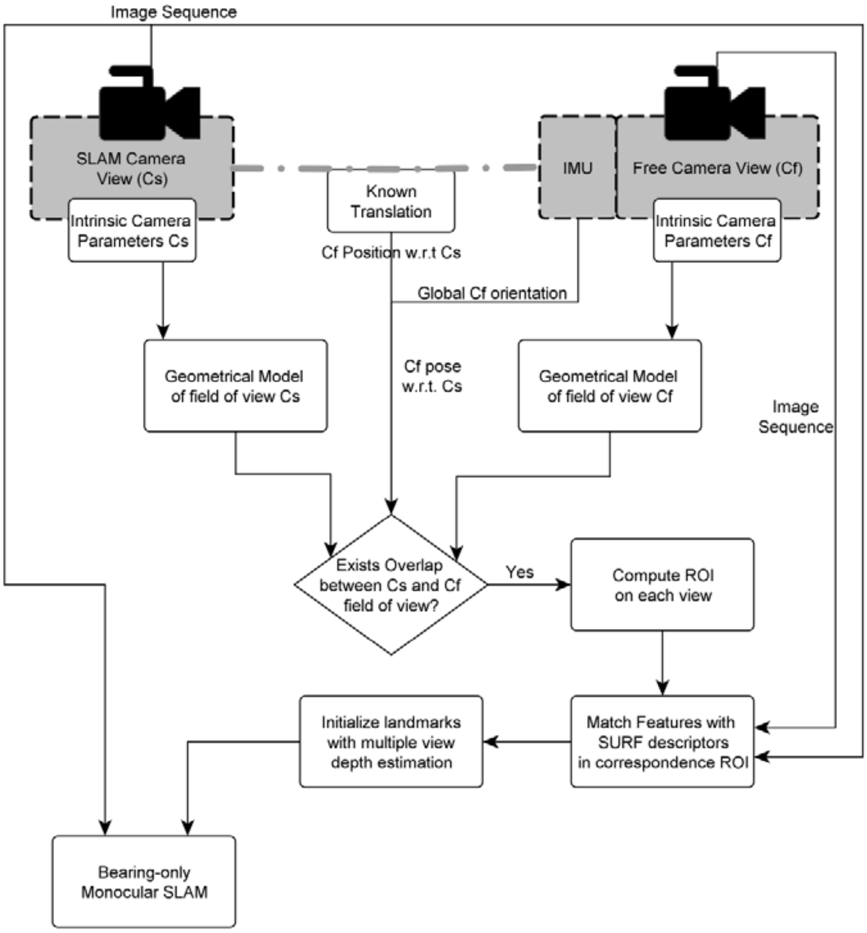


Image Sequence
SLAM Camera View (Cs)
Intrinsic Camera Parameters Cs
Known Translation
IMU
Free Camera View (Cf)
Intrinsic Camera Parameters Cf
Cf Position w.r.t Cs
Global Cf orientation
Geometrical Model of field of view Cs
Cf pose w.r.t. Cs
Geometrical Model of field of view Cf
Image Sequence
Exists Overlap between Cs and Cf field of view?
Yes
Compute ROI on each view
Initialize landmarks with multiple view depth estimation
Match Features with SURF descriptors in correspondence ROI
Bearing-only Monocular SLAM

FIGURE 5.5: *Block diagram of the multiple view overlap detection and correspondence ROI estimation.*

Then it can be assumed that any point with parallax –between cameras- equal or greater than α_{smin} lies in the space intersected by the two $\mathbb{R}^3$ polyhedron modelling the fields of view, as seen FIGURE 5.6. Note that the accuracy of this process is heavily correlated with that of the estimated pose between cameras, so a strategy which allows adjusting margins of error is followed in the next steps. The intersection between the different polygons composing the field of view models is computed as a set of segments, represented as tuples of two $\mathbb{R}^3$ points as described by ALGORITHM 5.1. Once all the segments are known, their ends are projected into the 2D projective space of C_s and C_f respectively, and a search region is adjusted around them in each image plane, determining the regions of interest (ROI), or correspondence regions, where the multiple view correspondence may provide useful matches.

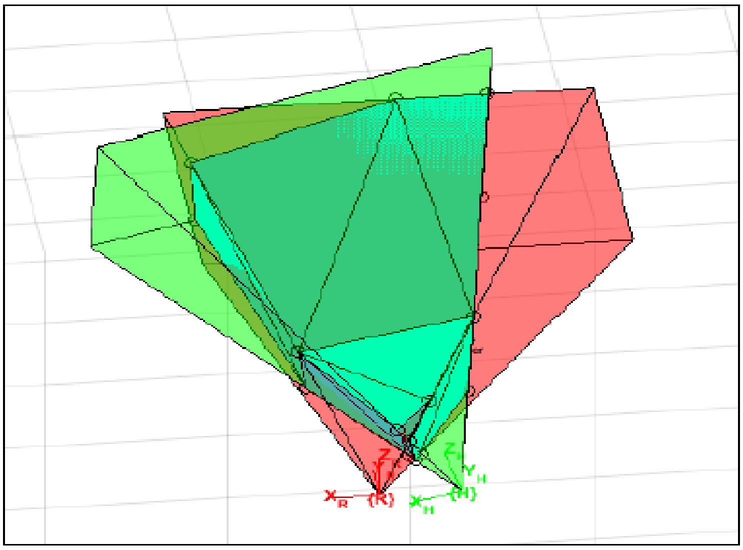


FIGURE 5.6: *Polyhedron found intersecting fields of view extended until a set depth where minimum parallax α_{smin} could be found.*

This adjustment can be fitted in several ways, permitting the introduction of offsets or tolerances to compensate possible errors: *exact fitting* defines a polygonal region with the convex hull of the approximate projection of the segments as edges[30]; while *bounding box* defines the minimal rectangle which fully envelopes all the projections.

The use of this procedure to determine if there is an area where salient features can be matched between images allows reducing/avoiding the computational effort at the initial estimation process notably, as it avoids trying to find correspondences between views when it is predicted that no useful data may be retrieved, skipping the cost altogether, and removing completely the chance of false positives.

[30] Note that due radial distortions, it is possible that the actual projection of an edge presents parts outside the estimated projection.

```
Function: ($ri_s$, $ri_f$):=find-Stereo-ROI ($C_s$, $C_f$, $\alpha_{smin}$)

Input:
$C_s$            SLAM camera calibration model data
$C_f$            free camera calibration model data
$\alpha_{smin}$  desired minimum parallax for depth estimation
Output:
$ri_s$           correspondence region in $C_s$ image
$ri_f$           correspondence region in $C_f$ image

distance := FindDistance ($C_s$.pose, $C_f$.pose)
PyramidDepth := FindMaxDepth (distance, $\alpha_{smin}$)
Py1 := ModelFoV($C_s$,PyramidDepth)
Py2 := ModelFoV($C_f$,PyramidDepth)
intersection = Ø;
for each polygon_i in Py1
    segment := Ø
    for each polygon_j in Py2
        segment := Intersect(polygon_i, polygon_j)
        if ¬ (segment = Ø)
            intersection.add(segment)
            segment := Ø
        end if
    end for
end for
$ri_s$ := Ø; $ri_f$ := Ø
if ¬(intersection = Ø) then
    $ri_s$ := Envelope(ProjectTo2D($C_s$.pose, intersection.points))
    $ri_f$ := Envelope(ProjectTo2D($C_f$.pose, intersection.points))
end if
return ($ri_s$, $ri_f$)
```

ALGORITHM 5.1: *Process to predict the occurrence of concurrent field of views and determining the correspondence region of interest for the matching process.*

The whole procedure to find the areas of interest for the correspondence problem is described in ALGORITHM 5.1. Once this procedure determines if the camera sensors observe concurrent scenes, and the regions of interest in the views of C_s and C_f are found, a search for correspondences based on feature point descriptors is performed (see Section 3.3). The first time that features are initialized, substituting the synthetic features initialization process, up to ten features are initialized in the EKF state vector, analogously to the process described in section 3.4.2, as reflected on FIGURE 5.5 diagram. SURF (Bay et al., 2006) is chosen over SIFT and FAST due the more convenient trade-off offered in terms of matching accuracy and efficiency (Juan and Gwun, 2009).

$$\mathbf{p}_{\mathbf{y}_{new}}^{c_s} = \left(u_{c_s}, \upsilon_{c_s}\right), \quad \mathbf{p}_{\mathbf{y}_{new}}^{c_f} = \left(u_{c_f}, \upsilon_{c_f}\right) \tag{5.5}$$

Points outside of the relevant region of interest in their respective images are ignored in this matching process. Then, the pixel coordinates in each image of the matched feature descriptors, (u_{cs},υ_{cs}) in C$_s$ and (u_{cf},υ_{cf}) in C$_f$ as shown in equation (5.5), are used to estimate the world coordinates of the landmark detected through stochastic triangulation. The landmarks are backtraced from C$_s$ through $\mathbf{p}^{Cs}{}_{ynew}$, and the ray from C$_f$ through $\mathbf{p}^{Cf}{}_{ynew}$ is used to determine the depth from C$_s$ to $\hat{\mathbf{y}}_{new}$. Then, the set of landmarks found and estimated are introduced in the monocular EKF according to the inverse depth parametrization.

5.4.2 Parametrization of features initialized through uncalibrated multiple-view estimation

New features detected with the multiple-view initialization process are also introduced into the EKF augmented state vector, similarly to equations (3.10) and (3.11), under the IDP model discussed in section 2.4.2, noted in equation (5.6).

$$\hat{\mathbf{y}}_{new} = \begin{bmatrix} \hat{x}_i & \hat{y}_i & \hat{z}_i & \hat{\theta}_i & \hat{\phi}_i & \hat{\rho}_i \end{bmatrix}^T \tag{5.6}$$

For the sake of simplicity, all features are still annotated into the EKF under the IDP w.r.t. the camera C$_s$, meaning that from the point of view of the EKF filter methodology, all the models and operations not related with the initialization process remain the same. So for $\hat{\mathbf{y}}_{new}$ the optical centre of the camera annotated will be that of the camera performing the SLAM process, $\mathbf{r}^{W}{}_{cs}$:

$$\begin{bmatrix} \hat{x}_i \\ \hat{y}_i \\ \hat{z}_i \end{bmatrix} = \begin{bmatrix} x_{c_s} \\ y_{c_s} \\ z_{c_s} \end{bmatrix}, \text{ where } \begin{bmatrix} x_{c_s} \\ y_{c_s} \\ z_{c_s} \end{bmatrix} = \mathbf{r}^{WC}_{c_s} \tag{5.7}$$

The parameters to define the director vector **m** from equations (2.27) and (2.28) are retrieved using equation (5.8) on the values obtained from the directional ray vector equation $\mathbf{h}^{\mathrm{W}}{}_{cs}$.

$$\begin{bmatrix} \theta_i \\ \phi_i \end{bmatrix} = \begin{bmatrix} \operatorname{atan2}\left(-h_{c_s y}, \sqrt{h_{c_s x}^2 + h_{c_s z}^2}\right) \\ \operatorname{atan2}\left(h_{c_s x}, h_{c_s z}\right) \end{bmatrix} \tag{5.8}$$

This projection ray vector $\mathbf{h}^{W}{}_{cs}$ (similar to those described in section 3.4.6, equations 3.63) is referenced to the world frame W, originating on position of the optical centre of C$_s$, to be used as anchor. Then, R^{WC} is the transformation matrix form camera C$_s$ to world reference frame, which is derived from the quaternion $\mathbf{q}^{WC}$ with C$_s$ orientation:

$$\mathbf{h}^{W}_{c_s} = \begin{bmatrix} h_{c_s x} \\ h_{c_s y} \\ h_{c_s z} \end{bmatrix} = R^{WC}(\mathbf{q}^{WC})\mathbf{h}_{c_s}(u_{u_s}, \upsilon_{u_s}). \tag{5.9}$$

The projection ray vector $\mathbf{h}_{\mathrm{cs}}(u_{us},\upsilon_{us})$ describes the same ray as $\mathbf{h}^{W}{}_{cs}$ but with respect to the same camera, pointing from the optical centre of the camera to the position of the landmark, and can be found from the undistorted pixel coordinates (u_{us},υ_{us}) and C_{s} calibration data:

$$\mathbf{h}_{c_s}(u_{u_s},u_{u_s}) = \begin{bmatrix} \dfrac{u_0 - u_{u_s}}{f_{c_s}} & \dfrac{\upsilon_0 - \upsilon_{u_s}}{f_{c_s}} & 1 \end{bmatrix}. \tag{5.10}$$

These undistorted pixel coordinates (u_{us},υ_{us}) are found applying the inverse of the distortion model (described in sections 3.2.1 and 3.4.6):

$$\begin{bmatrix} u_{u_s} \\ \upsilon_{u_s} \end{bmatrix} = \begin{bmatrix} \dfrac{u_{c_s} - u_{c_0}}{\sqrt{1-2k_1 r^2}} + u_0 \\ \dfrac{\upsilon_{c_s} - \upsilon_{c_0}}{\sqrt{1-2k_1 r^2}} + \upsilon_0 \end{bmatrix} \;;\; r = \sqrt{\left(u_{c_s} - u_{c_0}\right)^2 + \left(\upsilon_{c_s} - \upsilon_{c_0}\right)^2}\;. \tag{5.11}$$

The inverse depth is computed as the inverse of the norm the vector between $\mathbf{r}^{W}{}_{cs}$ and the intersection point between the rays $\mathbf{h}^{W}{}_{cs}$ and $\mathbf{h}^{W}{}_{cf}$, $\mathbf{h}_{cs}{}^{cs}$:

$$\rho_i = \frac{1}{\left\| \mathbf{h}_{c_s}^{c_f} - \mathbf{r}_{c_s} \right\|}\,. \tag{5.12}$$

Note that the computation of the ray coordinates $\mathbf{h}^{W}{}_{cf}$ is analogous to that for $\mathbf{h}^{W}{}_{cs}$, being the director vector of the ray originated at $\mathbf{r}^{WC}{}_{cf}$ through pixel (u_{cf},υ_{cf}), computed similarly to equations (5.9) to (5.11). The intersection between the two rays is computed according to equation (5.13), so $\mathbf{h}_{cs}{}^{cf}$ will be the nearest point to $\mathbf{h}^{W}{}_{cf}$ lying in the ray $\mathbf{h}^{W}{}_{cs}$.

$$\mathbf{h}_{cs}^{cf} = \mathbf{r}_{c_f}^{WC} + \left(\mathbf{h}_{c_s}^{W} - \mathbf{r}_{c_s}^{WC}\right)\cdot\frac{\left(\left(\mathbf{h}_{c_f}^{W} - \mathbf{r}_{c_f}^{WC}\right)\times\left(\mathbf{r}_{c_s}^{WC} - \mathbf{r}_{c_f}^{WC}\right)\right)\cdot\left(\left(\mathbf{h}_{c_s}^{W} - \mathbf{r}_{c_s}^{WC}\right)\times\left(\mathbf{h}_{c_f}^{W} - \mathbf{r}_{c_f}^{WC}\right)\right)}{\left(\left(\mathbf{h}_{c_s}^{W} - \mathbf{r}_{c_s}^{WC}\right)\times\left(\mathbf{h}_{c_f}^{W} - \mathbf{r}_{c_f}^{WC}\right)\right)\cdot\left(\left(\mathbf{h}_{c_s}^{W} - \mathbf{r}_{c_s}^{WC}\right)\times\left(\mathbf{h}_{c_f}^{W} - \mathbf{r}_{c_f}^{WC}\right)\right)} \tag{5.13}$$

Once all the parameters of $\hat{\mathbf{y}}_{new}$ have been computed, the new feature is added to the state vector, at the end of the map.

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_{c_s} \\ \hat{\mathbf{y}}_1 \\ \cdots \\ \hat{\mathbf{y}}_n \end{bmatrix} \Rightarrow \hat{\mathbf{x}}_{new} = \begin{bmatrix} \hat{\mathbf{x}}_{c_s} \\ \hat{\mathbf{y}}_1 \\ \cdots \\ \hat{\mathbf{y}}_n \\ \hat{\mathbf{y}}_{new} \end{bmatrix} \tag{5.14}$$

As the state vector grows, the covariance matrix P is updated, using equation (5.15), where R_j is the covariance matrix of the initialization measurement process, found at equation (5.16), and $\nabla\Upsilon$ is the Jacobian of the initialization process described earlier.

$$P_{new} = \nabla\Upsilon \begin{pmatrix} P & 0 \\ 0 & R_j \end{pmatrix} \nabla\Upsilon^T \tag{5.15}$$

Matrix R_j contains the measurement error variances σ_u^2 and σ_υ^2 in pixel units, and the covariance of the depth estimation process σ_ρ^2.

$$R_j = \mathrm{diag}\left(\left[\sigma_u^2 \quad \sigma_\upsilon^2 \quad \sigma_\rho^2\right]\right) \tag{5.16}$$

The Jacobian $\nabla\Upsilon$ is composed of an identity matrix with size equal to the prior covariance matrix P, $I_{m\times m}$, the derivatives of the initialization model with respect to the camera C$_s$ position $\delta\hat{\mathbf{y}}/\mathbf{r}_{cs}^{WC}$, w.r.t. C$_s$ orientation $\delta\hat{\mathbf{y}}/\mathbf{q}_{cs}^{WC}$, and the derivatives $\delta\hat{\mathbf{y}}/R_j$ with respect to the parameters of the covariances matrix R_j.

$$\nabla\Upsilon = \begin{bmatrix} I_{m\times m} & 0 \\ \dfrac{\partial\hat{\mathbf{y}}}{\partial\mathbf{r}_{c_s}^{WC}}, \dfrac{\partial\hat{\mathbf{y}}}{\partial\mathbf{q}_{c_s}^{WC}}, 0, \ldots, 0 & \dfrac{\partial\hat{\mathbf{y}}}{\partial R_j} \end{bmatrix} \tag{5.17}$$

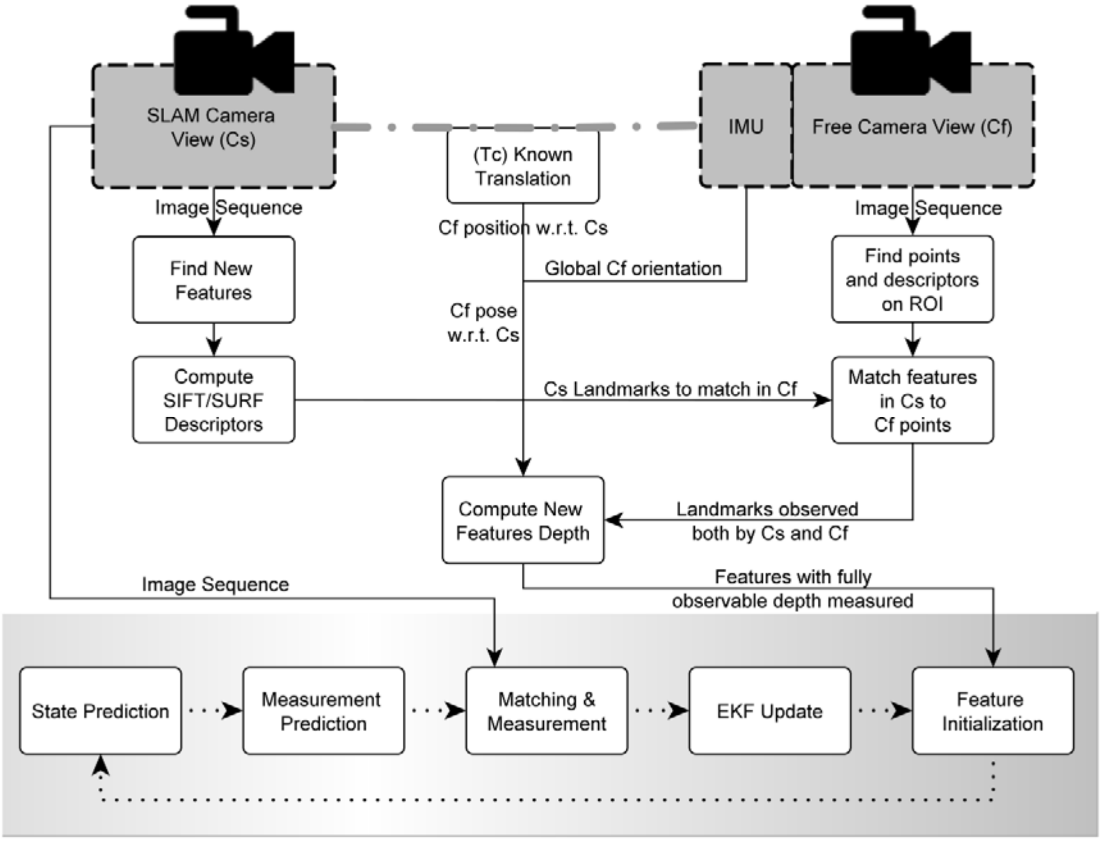


FIGURE 5.7: *Block diagram of the proposed feature initialization process for the case using the non-constant multiple view sensing, thus assuming that there is multiple view correspondence, and the correspondence region of interest presents landmarks to be initialized.*

5.4.3 State augmentation step: introduction of new landmarks under multiple view sensing

The original DI-D initialization, proposed in (Munguia and Grau, 2009) (and discussed in Section 3.4), adds new landmarks into the map part **m** of the state vector $\hat{\mathbf{x}}$ when a feature achieves enough parallax. This process is easily disrupted if the features cannot be tracked long enough due to motion blur, illumination problems, trajectory irregularities, etc., probably disrupting the filter performance and convergence.

Although the introduction of data association validation generally improves convergence of the filter (as it was discussed in Part 2), it may also deprive the filter of features, as it is possible that landmarks are removed faster than they are initialized. These problems can be reduced under the assumption of the temporary multiple view correspondence between cameras C_s and C_f just discussed, introducing the features much earlier with accurate depth estimation, using the non-constant multiple view I-D feature initialization presented.

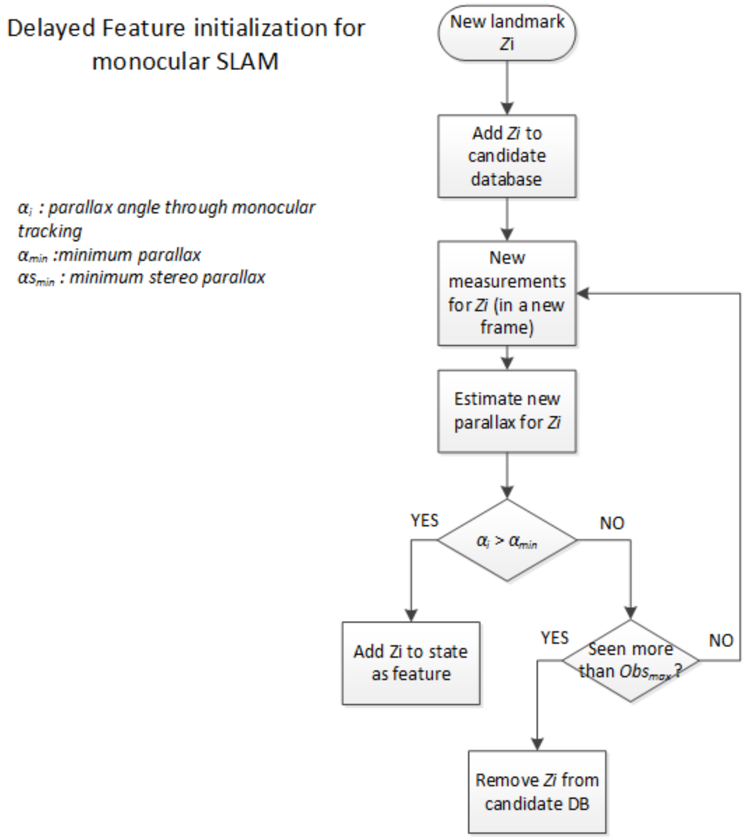

Delayed Feature initialization for monocular SLAM
α_i : parallax angle through monocular tracking
α_{min} :minimum parallax
αs_{min} : minimum stereo parallax
New landmark Zi
Add Zi to candidate database
New measurements for Zi (in a new frame)
Estimate new parallax for Zi
$\alpha_i > \alpha_{min}$
YES
NO
Add Zi to state as feature
Seen more than Obs_{max}?
YES
NO
Remove Zi from candidate DB

FIGURE 5.8: *Feature initialization process according to the single monocular camera approach.*

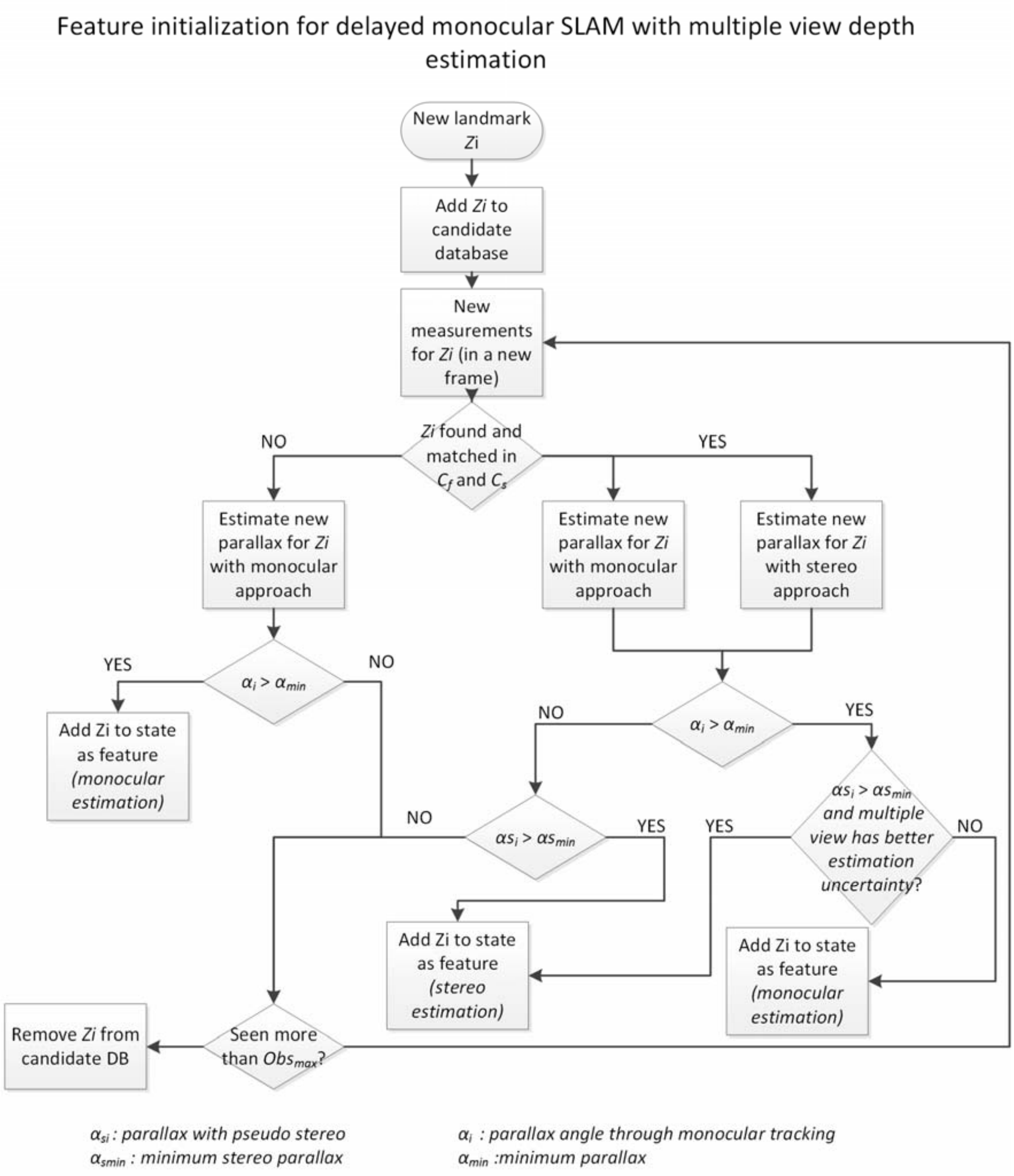

Feature initialization for delayed monocular SLAM with multiple view depth estimation
New landmark Zi
Add Zi to candidate database
New measurements for Zi (in a new frame)
Zi found and matched in C_f and C_s
NO
YES
Estimate new parallax for Zi with monocular approach
Estimate new parallax for Zi with monocular approach
Estimate new parallax for Zi with stereo approach
YES
$\alpha_i > \alpha_{min}$
NO
Add Zi to state as feature (monocular estimation)
NO
$\alpha_i > \alpha_{min}$
YES
NO
$\alpha s_i > \alpha s_{min}$
YES
YES
$\alpha s_i > \alpha s_{min}$ and multiple view has better estimation uncertainty?
NO
Add Zi to state as feature (stereo estimation)
Add Zi to state as feature (monocular estimation)
Remove Zi from candidate DB
Seen more than Obs_{max}?
α_{si} : parallax with pseudo stereo
α_{smin} : minimum stereo parallax
α_i : parallax angle through monocular tracking
α_{min} :minimum parallax

FIGURE 5.9: *Proposed feature intialization method with non-constant multiple view feature initialization.*

The previous delayed I-D feature initialization procedure and the one proposed exploiting the multiple-view depth estimation are shown in FIGURE 5.8 and FIGURE 5.9, respectively. The schema for the standard delayed I-D approach follows the strategy of storing and tracking candidate landmarks, detecting them through the Harris salience operator, seing if enough parallax is reached ($\alpha_i > \alpha_{min}$) within a given number of frames (Obs_{max}), and then proceed to initialize them with the estimated depth value. On the other side, the proposed scheme shows how the non-constant multiple view approach presents several chances to optimize the initialization process, and how it can work without an initial set of known features.

If during the correspondence prediction step a correspondence region of interest is not found, the process to introduce new features will try to work using the delayed I-D approach. If a correspondence region is found, the matching feature descriptor in C_f (with pixel coordinates u_{if},v_{if}) will be searched for those candidates whose pixel coordinates u_{is},v_{is} lay in the ROI ri_s. The parallax α_i will be also computed, so the landmarks which comply ($\alpha_i > \alpha_{min}$) and ($\alpha_i > \alpha_{smin}$) while having a match u_{if},v_{if} in ri_f will be given priority. For these candidates, the feature will be initialized with the parametrization which presents lower uncertainty. This allows performing an additional validation check, and if the depth discrepancy is too large between the two methods, the candidate will be ignored during the current iteration.

If not enough candidates were found, those that present a multiple view match and comply with ($\alpha_i > \alpha_{smin}$) will be initialized following the multiple view depth estimation approach, with a penalization to the σ_ρ^2 variance, as the depth estimation could not be validated between the two methods.

As an additional last effort, if the candidates available through the database are not enough to fill the minimum required number of features required into the EKF after a set number of frames, the stereo matched regions will be searched for new features to initialize, just as in the state vector initialization process described in section 5.4.1.

5.5 Experimental setup study and validation

To test the feature initialization methodology presented several experiments, both with real and synthetic data, were performed. In the case of the real data experiments, multiple sequences of synchronized data were captured, with each sequence consisting in a collaborative exploration of the environment at low speeds, including a human and a robotic platform. Each one of them was equipped with the monocular sensors assumed earlier, C_f for the human and C_s for the robotic platform, respectively. The data collected include the monocular sequences, odometry estimation from the robot (to have an approximate ground truth), estimation of the human pose with respect to the robot, and the orientation of the camera.

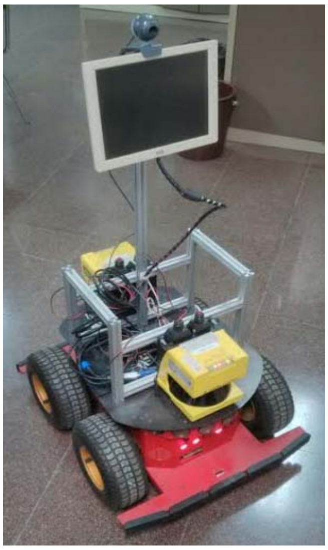


FIGURE 5.10: *Robotic platform Pioneer AT3 with a test webcam and laser range finders.*

The robot used was a robotic platform based on the Pioneer 3 AT, shown in FIGURE 5.10. The platform runs ROS Fuerte robotics middleware over an Ubuntu 12.04 LTS distribution, and it was equipped with a pair of laser range finders Leuzer RS4-4 and a Logitech C170 webcam. This webcam is able to work at 1024x768 pixels (XGA). The sensors worn by the human were deployed on a helmet, including a C170 camera and a Xsens AHRS. All the sensors, both in the robotic platform and the helmet, produce streams of data captured and synchronized by tools available in the ROS middleware.

To estimate the pose of C_f, orientation data from the AHRS are combined with the approximate pose of the human, estimated with the range finders, as presented by (Sanfeliu et al., 2010) and (Ferrer et al., 2013). The final position of the camera is computed geometrically as a translation from the estimated position of the Atlas and Axis vertebrae (which allow most of the freedom of movement of the head). These vertebrae are considered to be at a vertical axis over the person position estimated with the range finders, with height modeled individually for each person. In this work, it is assumed that the environment is a flat area, reducing the perturbations when trying to compose the estimated poses of the human and the camera C_f. FIGURE 5.11 depicts the helmet with the deployed sensors, and the coordinate frames considered for the transformations to compute the pose of C_f.

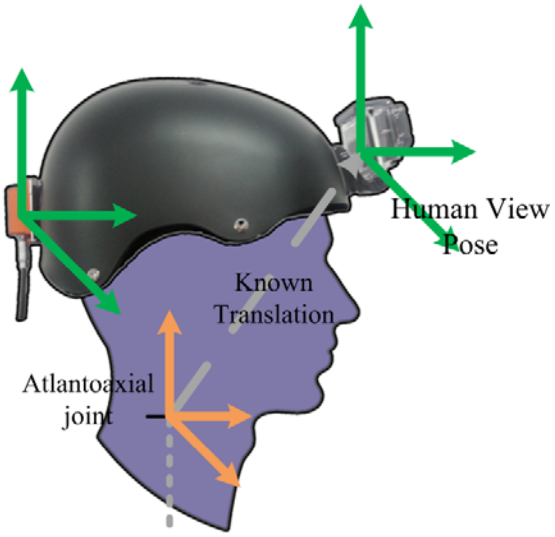


FIGURE 5.11: *View of the helmet with the camera and AHRS unit placement detail.*

Note that for the described method, accounting for the hardware available, the pose of the camera worn by the human respect to the SLAM camera is not assumed to be perfectly known. Instead, it is considered that when needed, a '*noisy*' observation of the pose of C_f respect C_s is available by means of the methodology described above. The inherent error to the observation process is modeled assuming that the observation is corrupted by Gaussian noise. The value of the parameters used to model the inaccuracies for computing the pose of C_f were obtained statistically by comparing actual and estimated values. It is also important to note that an alternate method could be used for computing the relative pose of C_f, for instance using different sensors. However, even with the use of a more reliable methodology the errors would not be completely eliminated.

To estimate the impact of the errors introduced by the multiple view *virtual sensor*, and its effects in the system accuracy, a Monte Carlo test was performed. This test showed that the errors introduced had little impact in the system, with the test consisting in simulating the initialization of a single feature using side-by-side:

- the ID-delayed monocular method, and
- the pseudo-calibrated stereo rig approach.

In the simulation, camera C_s is located at [x,y]=[0,0] at instant *k*. C_f is located at [x,y]=[2,0] at instant k. Thus, it is assumed that the base-line between C_s and C_f is equal to 2m. A landmark is located at [x,y]=[0.21,5]. For comparison purposes it is assumed that C_s was moved (at some instant *k*+*t*) to its right to [x,y]=[0.42,0] in order to generate a parallax equal to 5 degrees. This amount is a typical value used as a threshold in the ID-Delayed method for initializing new features.

In the simulation, the drift associated with the estimated displacement of C_s is modeled adding Gaussian noise with standard deviation σ=0.1m to the actual location of C_s at instant *k*+*t*. The angular measurements provided by C_s are modeled adding to its actual value a Gaussian noise with σ=0.5º. In order to model the inaccuracies associated with the multiple view approach hardware, the estimated location of C_f was modeled adding a Gaussian noise with σ=0.3m to its actual location. The errors introduced by the AHRS device have been taken into account by considering that the angular measurements provided by C_f are corrupted by Gaussian noise with σ=1.5º.

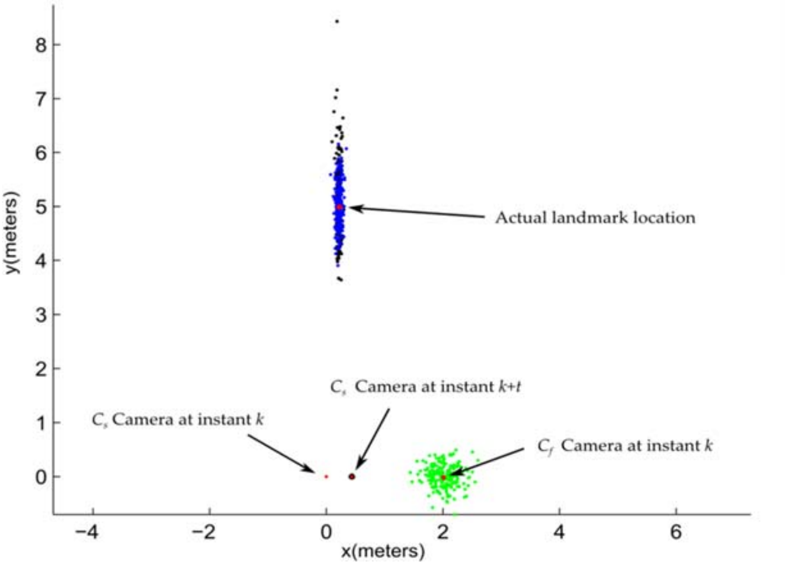


FIGURE 5.12: *Initialization of a single landmark using:* **i:** *the delayed I-D monocular method (black dots), and* **ii:** *the multiple view approach (blue dots). The actual position of the cameras and the landmark are indicated by red dots. Green dots show the estimated poses of camera C_f in different runs.*

Using the above conditions, the location of the landmark was estimated by stochastic triangulation with the location of C_s (at instant $k + t$) and with C_f. In both cases the location of C_s (at instant *k*) was used as common pivot. The experiment was carried out 200 times.

For the experimental setup, even considering that the location of C_f is estimated with many uncertainties, the likelihood region obtained with the multiple view approach is always smaller than the likelihood region obtained with the delayed I-D approach. This is because landmark depth estimation is heavily dependent on parallax. In the shown case, the parallax at the feature for the multiple view initialization approach is about 22º.

5.6 Results and Discussion

5.6.1 Experimental Results

The introduction of an auxiliary monocular sensor which can provide non-constant multiple view information was tested with multiple sequences. One of the disadvantages discussed on previous chapters was the need to manually introduce an initial metric scale, which is removed with the proposed methodology. This grants more autonomy to any SLAM technique, exploiting capabilities in a multiple-element/multimodal scenario, and enabling the generation of scaled vision-based maps. The additional sensing capabilities exploited come from the implicit human-robot interaction captured by the sensors worn by the human. In addition to skip the prior knowledge requirement (i.e. artificial or known landmarks), thanks to multiple view system initialization, the scale propagates generally in a smoother way with reduced drift as the proposed method can introduce more features into the initial state because it is not limited by the prior knowledge.

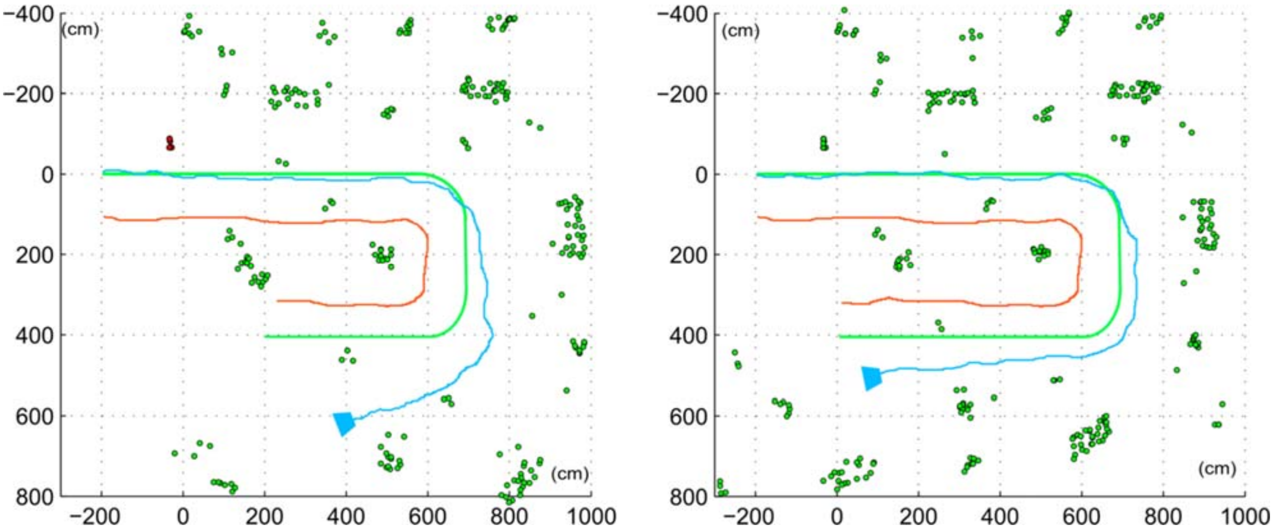


FIGURE 5.13: **Left:** *trajectory estimated with DI-D monocular SLAM.* **Right:** *trajectory estimated with the multiple view feature initialization approach. Green line denotes robot ground truth, orange line denotes C_f ground truth, and the estimated C_s trajectory is shown in blue. Red features (only left) have been artificially calibrated and introduced to have an initial scale estimation for the delayed I-D.*

FIGURE 5.13 shows results for one of the experimental trajectories, with and without the utilization of the proposed non-constant stereo I-D feature initialization approach, *right* and *left* maps respectively. The introduction of multiple-view initialization allows state augmentation where reliable depth estimation is achieved in a shorter time, thus making the system more resilient to quick view changes, such as turning. This can be seen on FIGURE 5.13 right, where the orientation drift is visibly minor. On the left trajectory estimation, the accumulated drift forces estimations so distant from actual observations within the data validation algorithm that most of the features are rejected. These rejections, combined with the drift itself, disrupt the estimation. On the other side, the trajectory estimated with the non-constant multiple view procedure minimizes the drift and orientation deviation, thus keeping an accurate estimation even after the U-turn. Results obtained are consistent through several runs, with multiple examples obtained shown on FIGURE 5.14 and TABLE 5.1.

TABLE 5.1: FINAL POSE ESTIMATION ERRORS AT THE END OF THE TRAJECTORY.

Experiment	**Original DI-D**				**Multiple View**			
& FIGURE	**|x|(m)**	**|y|(m)**	**d (m)**	**Angle(º)**	**|x|(m)**	**|y|(m)**	**d (m)**	**Angle(º)**
1 (5.11)	2.02	2.14	2.93	26	0,93	1,02	1,39	13.6
2 (5.12.a)	1,53	2,92	3,29	-51	0,89	0,71	1,14	-29.4
3 (5.12.b)	1,30	1,89	2,30	43	0,62	0,93	1,12	37.6
5 (5.12.c)	2,15	1,78	2,79	48	1,33	1,25	1,83	31.2

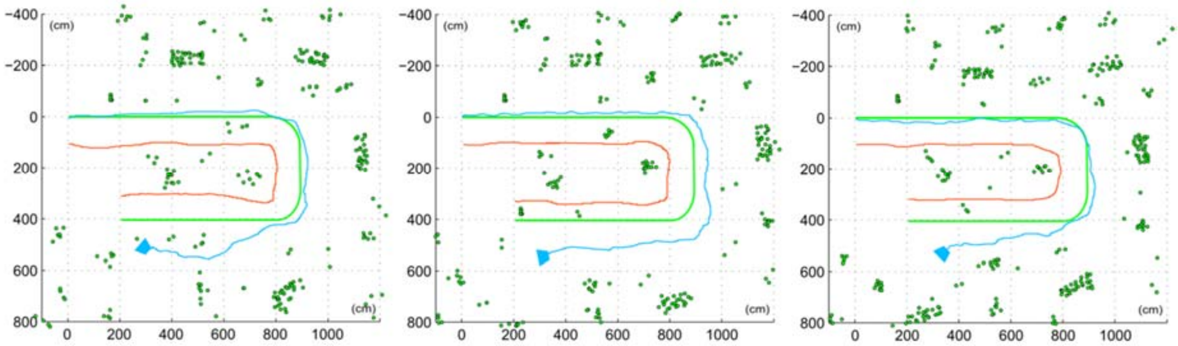


FIGURE 5.14: *Estimations obtained for the rest of captured sequences, performing the trajectory several times and processed with the proposed approach.*

The proposed approach allows using features which normally would be rejected in the DI-D approach after being unable to achieve enough parallax in a given time. There are two critical cases were features are usually unable to achieve enough parallax: the first one is when they are distant features, and the camera would have to travel great distances in certain ways to see parallax; the second one affects features which lie in projective rays near the visual axis when camera moves in singular trajectories, like forward. Several works, like (Civera et al., 2006) and (Clemente et al., 2007), have dealt with distant features initializing them with heuristic values, as in the undelayed general approach, as they tend to be useful to estimate orientation (Munguia and Grau, 2009). In the presented approach, the number of distant features used increases with respect to the DI-D approach (up to a range limited by the minimum parallax required and the distance between C_s and C_f), but those lying near the visual axis are the most benefitting, as they will always present more parallax between C_f and C_s than that achieve through temporal separation over a singular movement.

5.6.2 Costs and Analysis

The apparent increase of the computational effort that would suppose the utilization of the presented approach could be hard to justify within the field of filter based SLAM, which tries to keep reduced computational costs. But the cost increase is bounded and could be further reduced. For our C_s sequence set, made of a total of 7120 frames in all sequences, only 38.22 % (2721 frames) presented field of view overlap with the C_f camera. While this overlap ratio supposed an overhead of processing almost 40% more images, the exploration area was reduced with the search of the corespondence ROI. It is also interesting how the newly proposed approach made less effort per feature to be initialized in terms of number of frames requiring it to be tracked, compensating the larger number of features used.

TABLE 5.2: STATISTICS OF FEATURES USED AND TRACKING DUE DELAYED INITIALIZATION FOR ORIGINAL DI-D MONOCULAR AND FOR MULTIPLE VIEW APPROACH

Metric	DI-D Monocular	Multiple view
Features initialized (total)	1487	1549
Features initialized (avg.)	297.4	309.8
Average tracking period	24.6	10.4

TABLE 5.2 shows the features used on each feature initialization approach, and the tracking effort required (measured in number of frames where the feature is tracked) until the initialization of the features. For the experimental set, the multiple view approach uses about 4% more features, but the time required to initialize them is smaller. This is because many features that are being tracked are instantly initialized through the multiple view method once they lay in the overlapped field of view. This is advantageous because it allows introducing features known to be strong (enough to be tracked) directly, avoiding the computational costs of tracking them, offsetting the additional costs introduced by by the multiple view approach.

Furthermore, in real-time applications employing this technique, the C_f sensor could be upgraded to an *intelligent* sensor, i.e., presenting processing capabilities. This approach would integrate the image processing in the C_f sensor, allowing parallel processing of SURF features, and sending only extracted features, minimizing communications delay. This processing step could be done while the robotic camera C_s makes the general EKF-SLAM process, and thus it would be possible to have the SURF landmarks information after the EKF update, in time for the possible inclusion of new features.

5.7 Conclusions

In this chapter a new approach for feature initialization in SLAM has been proposed and discussed. A multiple view depth estimation procedure is proposed for feature initialization under a collaborative sensing assumption. The approach is based on the DI-D technique discussed in Chapter 3 (Munguia and Grau, 2012) and expanded in Chapter 4 (Guerra et al., 2013), being heavily focused towards human-robot interaction frameworks, under the form of collaborative explorations of the environment. The human collaboration has been introduced through a monocular sensor with total freedom of movement and approximately known pose, which is a set of assumptions generally satisfied in collaborative SAR robotics. As the different monocular sensors move freely, sometimes their fields of view will be concurrent: both cameras observing the same elements of the environment, producing non-constant multiple view measurements of them. As the relative pose between the cameras and the calibration matrices of each one of them are known, the fundamental matrix of a stereo system could be found. Even though this would allow the utilization of stereo-based rectification to ease the correspondence problem, it was deemed inconvenient for the approach, and descriptor-based feature matching was considered a better option.

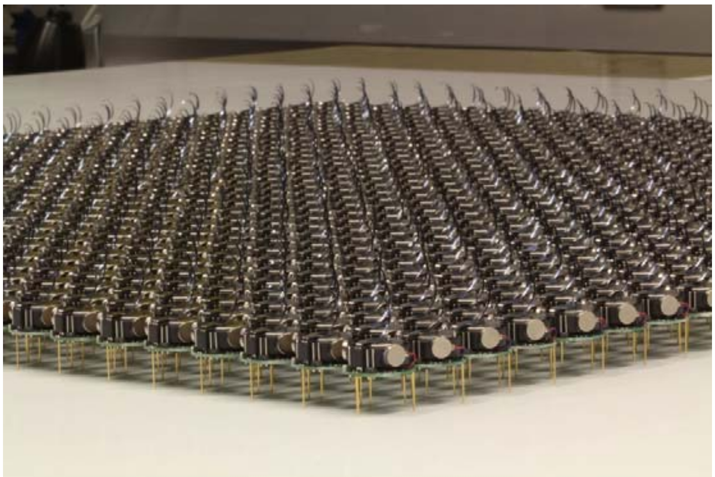
Utilization of non-constant multiple view depth estimation allows improving the performance of two specific aspects in the local scale EKF-SLAM framework. Firstly, the requirement of an initial metric scale introduced through synthetic features can be removed, substituted by the initialization of a set of features with collaborative depth estimation. This depth estimation has proven to have a multiple advantages: the number of features introduced initially is not limited to four coplanar points; and the use of a larger number of features presenting diverse depth values makes the metric scale propagation smoother. Secondly, the introduction of later landmarks through multiple view depth estimation enables utilization of far distance features with real depth estimation, instead of the heuristically assigned value used in previous works, and the initialization of frontal landmarks when the camera C_s moves forward and other singular trajectories. These changes have produced a locally strong and robust SLAM approach, thus enabling its future utilization on larger scale SLAM, as commented on section 2.7.3. Using the proposed approach in an SLAM framework considering loop closure and large map management would further reduce the drift of the estimated trajectory, thanks to the covariance reduction produced by loop closure.

As the viability of the proposed approach has been demonstrated, research could focus on maximizing the advantages obtained from the HRI, while studying in depth the costs of the proposed technique. In terms of exploiting the HRI, the multiple view depth estimation could be introduced the measurement and update step of the EKF SLAM. This would probably require a general overhaul of the prediction and observation models currently used, but it should improve the accuracy of the approach. In line with this overhaul, the use of non-constant stereo allows to reinitialize a metric scale whenever the field of view overlaps, permitting the introduction of submapping techniques and other methods related with large map management to and achieve larger trajectories, including loop closing.

The proposed technique could be also expanded, with some techniques taken from the collaborative SLAM field, to deal with more C_f agents, e.g.: a group of different humans could explore an environment accompanied by a robot mapping their surroundings with data from the sensors deployed on the humans. While this approach would require much more computational power and an insightful architecture, it would be of great interest due its resemblance to hypothetical real cases where not a human alone, but a team, would explore new zones with robotic assistance.

Chapter 6

Multiple-view sensing for Monocular SLAM



A few hundreds of Kilobots. Developed by Harvard's Self-Organizing Systems Research Group, swarms of these 3.3cm robots, up to a thousand, can cooperate to execute tasks impossible for a single unit, like shape self-assembly, human-swarm interaction and collective transportation.

6.1 Introduction

The main contribution of the multiple view feature initialization approach presented in the previous chapter relied on the presence of a secondary monocular sensor (C_f) worn by a human[29] which satisfies three conditions:

- its pose with respect to the camera performing the SLAM process, C_s, is approximately known/measured;
- it produces data of similar nature to $C_{s;}$

[29] Though it could be carried by another robot.

- during some frames C$_f$ would observe the same scenes as C$_s$.

These three conditions, frequently satisfied in collaborative robotics environments, have allowed speeding up the feature initialization process, reducing the number of required observations of an interest point along frames. Under this collaborative initialization process a feature whose multiple view depth estimation is available can be initialized without delays, like in the undelayed approach, but with an actual depth estimation instead of a heuristic value. As features can be initialized instantly, the risk of spurious/dynamic points increases, thus a data association validation step, like the HOCHT discussed in Chapter 4, becomes a critical component of the SLAM method. The feature initialization process designed in Chapter 5 also permitted keeping the EKF architecture at the core largely unchanged. And as the landmarks initialized through the described method presents better initial depth estimations, the uncertainty components in the covariance matrix *P* are usually lower, producing a more aggressive rejection threshold on the HOHCT gate (as this gating area is adjusted as a function of the innovation covariance, which in turn depends on the state covariance *P*).

This chapter describes how to expand the benefits of the collaborative, sensing under the framework proposed in the previous chapter, so that the *virtual sensor* is exploited during the measurement update step, removing the limitation of using it only during the state augmentation step. The described collaborative SLAM method is studied from a theoretical point of view, considering the gains in terms of state observability and evaluating the increased accuracy in depth estimation through simulation. Additional experiments are used to test the technique in locally relevant challenges and generalist short trajectories. These tests allow proposing new methodologies and metrics to study how the behaviour of the collaborating sensor affects the performance of the SLAM methodology, and measure the additional computational effort required versus the classical approach.

6.2 Full non-continuous collaborative Monocular SLAM

6.2.1 Kalman Update with full-observability based residuals

In Chapter 3 the inverse depth parametrization is the basis for the observation model $\mathbf{h}^c$. The inverse depth model for camera C$_s$, $\mathbf{h}^{cs}$ in equation (6.1), is largely identical to $\mathbf{h}^c$ in equation 3.47, producing 3D world coordinates with respect to the camera, (h_x,h_y,h_z). These can be projected into a camera plane with equation (6.2), and obtain undistorted pixel coordinates $(u^{cs}{}_u,v^{cs}{}_u)$ in the projective plane. These coordinates are to be converted to pixel space, and distorted to produce, following the same equations (3.47) and (3.48), measurement predictions for the landmarks in map $\mathbf{m}$. These predictions of the observations in pixel coordinates are used during the measurement and correction processes at the update step of the EKF methodology (described for the general approach in Section 2.4.2 and for the delayed approach in 3.5.5) as the predicted observations for the map; and the matches to them found are used as measurements obtained from the sensors (per active search, section 4.3.1).

$$\mathbf{h}^{c_s} = \begin{bmatrix} h_x^{c_s} \\ h_z^{c_s} \\ h_z^{c_s} \end{bmatrix} = R^{CW}\left(\left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} - r_{c_s}^{WC}\right) + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \phi_i\right)\right) \tag{6.1}$$

$$\begin{bmatrix} u_u^{c_s} \\ \upsilon_u^{c_s} \end{bmatrix} = \begin{bmatrix} \dfrac{h_x^{c_s}}{h_z^{c_s}} \\ \dfrac{h_y^{c_s}}{h_z^{c_s}} \end{bmatrix} \tag{6.2}$$

As the collaborative *virtual sensor* architecture (see FIGURE 6.1) allows the SLAM process to work with observations both in pixel space and in real world coordinates, it is possible to compute the Kalman update using the residuals from fully observed features measured through the multiple view *virtual sensor*. Utilizing these measurements requires that a new specific measurement prediction model is derived for said features in world coordinates. This way, the multiple view observation data sparsely available can be used not only for the state augmentation step (introduction of new features), but also for the estimation and measurement. The new feature measurement prediction model, noted in equation (6.3), is to be used to predict the measurement of the landmarks satisfactorily observed and measured with the multiple view *virtual sensor*.

$$\mathbf{h}_i^{xyz} = \begin{bmatrix} x_{hi} \\ y_{hi} \\ z_{hi} \end{bmatrix} = R^{CW}\left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i}\mathbf{m}\left(\theta_i, \varphi_i\right) - r^{WC}\right) \tag{6.3}$$

Although the idea is simple enough, that is, if it is possible to exploit the multiple view data in the monocular SLAM to initialize features, then it is possible to do the same during the update step, it presents many ramifications and require multiple modifications into the EKF methodology (which in Chapters 4 and 5 remained largely unchanged). Under standard monocular SLAM approaches, measurement and data association processes are performed in one step with the active search (Section 3.5.5), but to support measurement and observations in world space (w.r.t. the C_s camera) an additional pipeline is required to solve the correspondence problem, produce the measurements, and adapt the EKF to work with an innovation vector containing two different types of landmarks.

The matching and state update procedure with the new pipeline integrated into the monocular SLAM is initialized as in previous chapters: the known features are matched through active search between frames in the sequence obtained with camera C_s in order to keep tracking accuracy consistent when only monocular data is available. The features matched through active search found in the correspondence ROI determined to be observed by both C_s and C_f are modelled with an additional descriptor. To this end SURF descriptors are used similarly to the process described in section 5.4.1. Afterwards, possible points of interest are detected in the correspondence ROI found at the matching frame from C_f, and the SURF descriptors are build.

These SURF point descriptors from C_f are then matched with the SURF descriptors from the known features found in the correspondence ROI of C_s. Known features without a match in C_f will be treated as only-bearing features, using the pixel position on image as measurement, and thus will use the same procedures described in Section 3.5 during the state update step, as in the standard delayed monocular SLAM. Those with a matching point in C_f, are measured in terms of world coordinates with respect to C_s through stochastic triangulation (Section 5.4.2). This in turn implies that from that point on, all the processes and equations must account that these landmarks are measured with actual depth, and thus described as Euclidean points w.r.t. to C_s frame, and their measurement prediction model will be through equation (6.3) instead of equations (3.47) and (3.48).

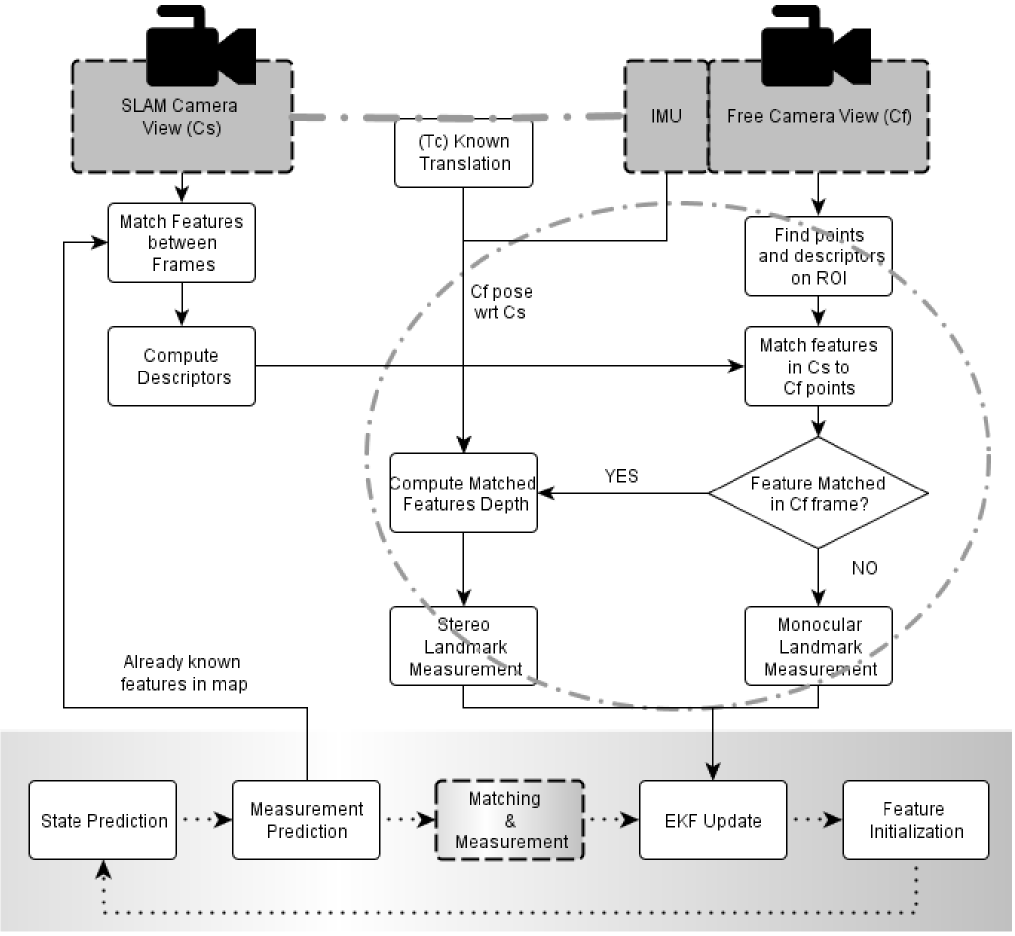

SLAM Camera View (Cs)
(Tc) Known Translation
IMU
Free Camera View (Cf)
Match Features between Frames
Compute Descriptors
Cf pose wrt Cs
Find points and descriptors on ROI
Match features in Cs to Cf points
Compute Matched Features Depth
YES
Feature Matched in Cf frame?
NO
Stereo Landmark Measurement
Monocular Landmark Measurement
Already known features in map
State Prediction
Measurement Prediction
Matching & Measurement
EKF Update
Feature Initialization

FIGURE 6.1: *Monocular EKF-SLAM with complete multiple view collaborative sensing, including measurement and matching.*

As it was detailed in section 3.5.5, during the update step of EKF there are several computations performed in matrix that model relations of the landmarks and their measurements with the state (equation (3.51) to equation (3.56)). These matrices are generally consistent in the sense that they present a repeating structure, as described earlier, and one of their dimensions is dependent on the size of the innovation vector **g** (equation 2.15 and equation 3.53). Note that innovation vector is essentially a stack with the residuals of the observations obtained and the predicted observations from known features; and the new update procedure considers some measurements as observable through only-bearing data while others are considered fully observable. This means that **g** size will vary accordingly to the number of seen features, as previously, and how these features are observed (be it as pixel coordinates or as Euclidean points in space).

6.2.1.1. Measurement Prediction Jacobian

Note that the augmented state vector and the covariance matrix will remain the same, as the features $\hat{\mathbf{y}}_i$ are keep in the filter parametrized as IDPs. But the Jacobian matrices of the observation model, used on several equations, like (2.16), (2.17), (3.54) and (3.56), will change not only in size, but in the way they are built. In previous works (Munguía and Grau, 2012), as $\mathbf{h}^c$ is to be projected into C$_s$ and distorted once in pixel coordinates, the only relevant information was the bearing represented by the director vector **m** to define a director ray w.r.t. C$_s$.

This means that the general approach is replacing equation (3.47) with (6.4), as a means to simplify the symbolical computation of the Jacobian ∇H in equation (3.52).

$$\mathbf{h}^c = \begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = R^{CW}\left(\rho_i\left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} - r^{WC}\right) + \mathbf{m}(\theta_i, \phi_i)\right) \tag{6.4}$$

This change allows for a simpler derivation of the required Jacobian for the bearing-only observation case, but also implies that its partial components are not reusable to compute the Jacobian of $\mathbf{h}^{xyz}$ as the coordinates of $\mathbf{h}^C$ describe a ray, with no consideration of the depth required to produce the full observation measurement. Hence, to complete the full procedure, the Jacobian ∇H_i^{xyz} is formulated, as noted in equation (6.5), derived from equation (3.47) without any simplification:

$$\nabla H_i^{xyz} = \begin{bmatrix} \frac{\partial h_i^{xyz}}{\partial \hat{\mathbf{x}}_{C_s}} & \cdots & 0_{3\times6} & \cdots & \frac{\partial h_i^{xyz}}{\partial \hat{\mathbf{y}}_i} & \cdots & 0_{3\times6} & \cdots \end{bmatrix}. \tag{6.5}$$

The partial derivatives in ∇H_i^{xyz} are w.r.t. the camera C$_s$ state and to the noted feature $\hat{\mathbf{y}}_i$ parametrization, expanded in equations (6.6) and (6.20) respectively.

$$\frac{\partial h_i^{xyz}}{\partial \hat{\mathbf{x}}_{c_s}} = \begin{bmatrix} \frac{\partial h_i^{xyz}}{\partial \mathbf{r}_{c_s}^{WC}} & \frac{\partial h_i^{xyz}}{\partial \mathbf{q}_{c_s}^{WC}} & 0_{3\times 6} \end{bmatrix} \tag{6.6}$$

The measurement model w.r.t. to Cs, $\delta h_i^{xyz}/\delta\hat{\mathbf{x}}_{cs}$, can be decomposed into those dependent in the position, $\delta h_i^{xyz}/\delta\mathbf{r}^{WC}{}_{cs}$, and $\delta h_i^{xyz}/\delta\mathbf{q}^{WC}{}_{cs}$, according to equation (6.6). These are respectively defined in equations (6.7) and (6.8). Note that the derivatives of h_i^{xyz} w.r.t. the velocities in the state of $\hat{\mathbf{x}}_{cs}$ are zero, as seen in the third block of equation (6.6), as the direct observation model does not consider them.

$$\frac{\partial h_i^{xyz}}{\partial \hat{\mathbf{x}}_v} = -R^{CW} \tag{6.7}$$

The Jacobian with respect to the orientation is split into two pieces:

$$\frac{\partial h_i^{xyz}}{\partial \mathbf{q}_{c_s}^{WC}} = \frac{\partial h_i^{xyz}}{\partial \mathbf{q}_{c_s}^{CW}} \frac{\partial \mathbf{q}_{c_s}^{CW}}{\partial \mathbf{q}_{c_s}^{WC}} , \tag{6.8}$$

where $\delta\mathbf{q}^{CW}{}_{cs}/\delta\mathbf{q}^{WC}{}_{cs}$ described the derivative of the orientation quaternion $\mathbf{q}^{WC}{}_{cs}$ w.r.t to its conjugate quaternion $\mathbf{q}^{CW}{}_{cs}$:

$$\mathbf{q}_{c_s}^{CW} = conj(\mathbf{q}_{c_s}^{WC}) = \begin{bmatrix} q_0^{CW} & q_1^{CW} & q_2^{CW} & q_3^{CW} \end{bmatrix}, \tag{6.9}$$

$$\frac{\partial \mathbf{q}_{c_s}^{CW}}{\partial \mathbf{q}_{c_s}^{WC}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}; \tag{6.10}$$

and the partial derivatives of the model w.r.t. the conjugate quaternion can be divided into the derivatives w.r.t. each of the components of the quaternion $\mathbf{q}^{CW}{}_{cs}$.

$$\frac{\partial h_i^{xyz}}{\partial \mathbf{q}_{c_s}^{CW}} = \begin{bmatrix} \frac{\partial h_i^{xyz}}{\partial q_0^{CW}} & \frac{\partial h_i^{xyz}}{\partial q_1^{CW}} & \frac{\partial h_i^{xyz}}{\partial q_2^{CW}} & \frac{\partial h_i^{xyz}}{\partial q_3^{CW}} \end{bmatrix} \tag{6.11}$$

Each of the derivatives w.r.t. each component only affects to the rotation matrix R^{CW}, obtained according to equation (V.3) found at the annexes.

$$\frac{\partial h_i^{xyz}}{\partial \mathbf{q}_0^{CW}} = \frac{\partial R^{CW}}{\partial \mathbf{q}_0^{CW}} \left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} \mathbf{m}(\theta_i, \varphi_i) - r_{c_s}^{WC} \right) \tag{6.12}$$

$$\frac{\partial h_i^{xyz}}{\partial \mathbf{q}_1^{CW}} = \frac{\partial R^{CW}}{\partial \mathbf{q}_1^{CW}} \left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} \mathbf{m}(\theta_i, \varphi_i) - r_{c_s}^{WC} \right) \tag{6.13}$$

$$\frac{\partial h_i^{xyz}}{\partial \mathbf{q}_2^{CW}} = \frac{\partial R^{CW}}{\partial \mathbf{q}_2^{CW}} \left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} \mathbf{m}(\theta_i, \varphi_i) - r_{c_s}^{WC} \right) \tag{6.14}$$

$$\frac{\partial h_i^{xyz}}{\partial \mathbf{q}_3^{CW}} = \frac{\partial R^{CW}}{\partial \mathbf{q}_3^{CW}} \left(\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} + \frac{1}{\rho_i} \mathbf{m}(\theta_i, \varphi_i) - r_{c_s}^{WC} \right) \tag{6.15}$$

Then, the derivatives of the matrix R^{CW} with respect to the quaternion components (assuming that the conversion is performed with equation V.3), can be defined as:

$$\frac{\partial R^{CW}}{\partial \mathbf{q}_0^{CW}} = \begin{bmatrix} 2q_0^{CW} & -2q_3^{CW} & 2q_2^{CW} \\ 2q_3^{CW} & 2q_0^{CW} & -2q_1^{CW} \\ -2q_2^{CW} & 2q_1^{CW} & 2q_0^{CW} \end{bmatrix}, \tag{6.16}$$

$$\frac{\partial R^{CW}}{\partial \mathbf{q}_1^{CW}} = \begin{bmatrix} 2q_1^{CW} & 2q_2^{CW} & 2q_3^{CW} \\ 2q_2^{CW} & -2q_1^{CW} & -2q_0^{CW} \\ 2q_3^{CW} & 2q_0^{CW} & -2q_1^{CW} \end{bmatrix}, \tag{6.17}$$

$$\frac{\partial R^{CW}}{\partial \mathbf{q}_2^{CW}} = \begin{bmatrix} -2q_2^{CW} & 2q_1^{CW} & 2q_0^{CW} \\ 2q_1^{CW} & 2q_2^{CW} & 2q_3^{CW} \\ -2q_0^{CW} & 2q_3^{CW} & -2q_2^{CW} \end{bmatrix}, \tag{6.18}$$

$$\frac{\partial R^{CW}}{\partial \mathbf{q}_3^{CW}} = \begin{bmatrix} -2q_3^{CW} & -2q_0^{CW} & 2q_1^{CW} \\ 2q_0^{CW} & -2q_3^{CW} & 2q_2^{CW} \\ 2q_1^{CW} & 2q_2^{CW} & 2q_3^{CW} \end{bmatrix}. \tag{6.19}$$

Once all the partial derivatives w.r.t. the camera state are computed, the only remaining step is to compute the derivatives with respect to the parametrization of the landmark:

$$\frac{\partial h_i^{xyz}}{\partial \hat{\mathbf{y}}_i} = \begin{bmatrix} R^{CW} & R^{CW} \frac{1}{\rho_i} \begin{bmatrix} \cos\theta\cos\phi \\ 0 \\ -\cos\phi\sin\theta \end{bmatrix} & R^{CW} \frac{1}{\rho_i} \begin{bmatrix} \sin\theta\sin\phi \\ -\cos\phi \\ -\sin\phi\cos\theta \end{bmatrix} & R^{CW} \frac{-\mathbf{m}}{(\rho_i)^2} \end{bmatrix} \tag{6.20}$$

6.2.2 State aumentation and the Covariance Matrix

When a new landmark is introduced as a feature in the EKF state vector the data describing the landmark uncertainty and relations with previous estimations must be introduced into the covariance matrix. The general EKF SLAM methodology introduces the new data using equation (3.76), updated as equation (6.21):

$$P_k = \nabla Y \begin{pmatrix} P_k & 0 \\ 0 & R_j \end{pmatrix} \nabla Y^T \tag{6.21}$$

where R_j is a diagonal matrix containing the error variance parameters of the sensor and the parameters stored for the new landmark, and ∇Y is the Jacobian of the inverse observation model. The inverse observation model is used to compute the characterization of an observed landmark as an inverse-depth feature, using data from the sensors and the current estimates of the system. In the previous chapter, and in (Guerra et al., 2014), the features initialized through the delayed method used the DI-D initialization process (section 3.5.6), while those added through the multiple view estimation used a classic monocular inverse-depth model, as proposed by (Civera et al., 2006), with an accurate depth estimation, as described in section 5.4.2. This fact supposes an underrepresentation of the uncertainty related to the secondary camera C_f. This problem can be addressed modifying this method, so that matrix R_j and Jacobian ∇Y used to add features to matrix P (augmented state covariance matrix) account for the uncertainties derived from both cameras.

This modification over the approach presented in the previous chapter means having a total of 11 parameters to represent uncertainty instead of 3. Thus, instead of updating the covariance matrix for the new features with depth estimation using equation (5.15) and (5.16), this step will be performed according to equation (3.76), so Jacobian ∇Y is analogous to the delayed feature initialization (discussed in section 3.5.6, equation (3.78)), accounting for the origin coordinates for two different rays. This in turn means that matrix R_j will be formed as shown in equation (6.22):

$$R_j = \begin{pmatrix} \sigma_{us}^2 &&&&&&&&&& \\ & \sigma_{vs}^2 &&&&&&&&& \\ && \sigma_{uf}^2 &&&&&&&& \\ &&& \sigma_{vf}^2 &&&&&&& \\ &&&& \sigma_{x\lambda}^2 &&&&&& \\ &&&&& \sigma_{y\lambda}^2 &&&&& \\ &&&&&& \sigma_{z\lambda}^2 &&&& \\ &&&&&&& \sigma_{q0\lambda}^2 &&& \\ &&&&&&&& \sigma_{q1\lambda}^2 && \\ &&&&&&&&& \sigma_{q2\lambda}^2 & \\ &&&&&&&&&& \sigma_{q3\lambda}^2 \end{pmatrix} \tag{6.22}$$

where (σ_{ui}^2, σ_{vi}^2) denotes the pixel uncertainty for the cameras, ($\sigma_{x\lambda}$, $\sigma_{y\lambda}$, $\sigma_{z\lambda}$) denotes the uncertainty of the position of C_f, and ($\sigma_{q0\lambda}$, $\sigma_{q1\lambda}$, $\sigma_{q2\lambda}$, $\sigma_{q3\lambda}$) denotes the uncertainty in C_f orientation quaternion.

6.3 Theoretical validation

The intuitive impact of the introduction of the multiple view *virtual sensor* in the measurement and update steps of the EKF is clear: the landmarks $\mathbf{y}_i$ that compose the map $\mathbf{m}$ are to be observed as 3-dimensional points instead of pixel coordinates, which leads to an immediate perception of depth. In turn, this ability to estimate the actual depth instantly allows to keep the scale map without the prior knowledge initialization, and without suffering the drift introduced by the degeneration of the accuracy of the initial calibration. Still, we desire to study and confirm these theoretical hypotheses. The results related to accuracy and physical realities are easy to confirm experimentally, as they will, but to test other aspect, different studies are required.

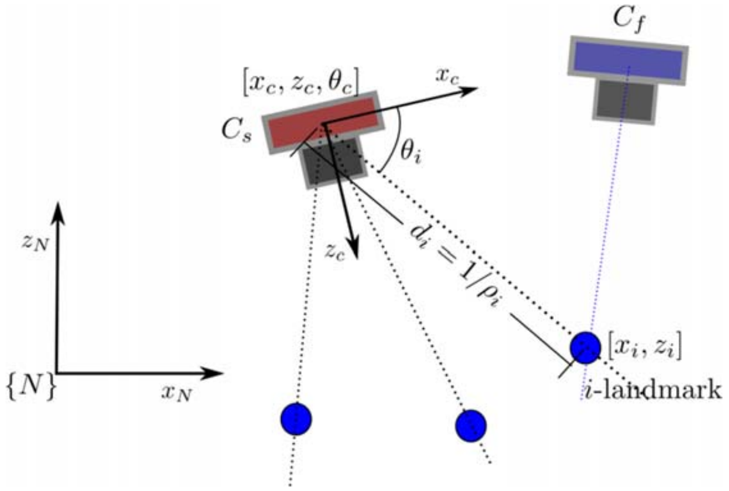


FIGURE 6.2: *A 2DOF simplified version of the proposed system used for performing an observability test.*

One point of interest is to study the observability of the system. As it was discussed in chapter 2, bearing-only sensors like camera are unable to fully observe[30] the real world as they perceive only the orientation aspects of the geometrical relations or transformations, lacking the capabilities to perceive scale in translation transformations. In this section an observability analysis is carried out. This analysis will show that the observability of the system is improved when multiple view measurements are incorporated into the SLAM system, as some landmarks in the map are considered to be fully observed.

A system is defined as observable if the initial state x_0 at any initial time t_0 can be determined given the state transition and observation models of the system and observations $y[t_0,t]$ from time t_0 to a finite time t. When a system is fully observable, the lower bound of the error in our estimate of its state will only depend on the noise

[30] In the sense of complete measurement of the pose without additional data or processing.

parameters of the system and will not be reliant on initial information about the states. This has important consequences in the context of SLAM.

In order to carry out the analysis, a simplified version of the proposed system is assumed (see FIGURE 6.2). Assuming the following unconstrained camera model $\dot{x}_c = f(x,u)$ for the camera C_s:

$$\begin{array}{lll} \dot{x}_c = v_x & \dot{z}_c = v_z & \dot{\theta}_c = \omega_z \\ \dot{v}_x = V_x & \dot{v}_z = V_z & \dot{\omega}_c = \Omega \end{array} \tag{6.23}$$

where x_c=[x_c, z_c, θ_c, v_x, v_z, ω_c] is the system state of camera C_s. [x_c, z_c, $\theta_{c,}$] represent the position and orientation of the camera, and [v_x, v_z, ω_c] their first derivatives. In this model, it is assumed an unknown input, u = [V_x, V_z, Ω], of linear and angular accelerations with zero-mean and known covariance Gaussian processes. It is also assumed that the camera C_s it is capable of detecting and tracking feature points with perfect matching accuracy, coded in their inverse depth. In this case, the measurement process is modelled by equation (6.24):

$$y_i = h_{\theta i}(x) = \arctan 2\left(\frac{z_c - z_i}{x_c - x_i}\right) - \theta_c \tag{6.24}$$

where [x_i, z_i] is the Euclidean position of a i^{th} feature coded by its inverse form:

$$\begin{array}{l} x_i = \left(1\big/\rho_i\right)\cos(\theta_i) + x_{0i} \\ z_i = \left(1\big/\rho_i\right)\sin(\theta_i) + z_{0i} \end{array} \tag{6.25}$$

The state of the i^{th} feature w_i is defined by w_i=[x_{0i}, z_{0i}, θ_i, ρ_i], where [x_{0i}, z_{0i}] is the position of the camera C_s when the feature was first detected, θ_i is the first bearing measurement, and $\rho_i = 1/d_i$ is the inverse of the feature depth d_i. Because, [x_{0i}, z_{0i}, θ_i] is given directly when the i^{th} feature is initialized, it is assumed that the system state to be estimated, $\hat{x}$, is composed of the state of the camera C_s and the inverse depth of the features. Hence $\hat{x}=[\hat{x}_c,\hat{\rho}_1,\hat{\rho}_2,...,\hat{\rho}_n]$.

Fully observable measurements from the multiple view *virtual sensor*, which are available when there is some overlapping of the FoV of both cameras C_s and C_f, provide information about the feature depths. Thus, a multiple view measurement of a i^{th} feature is modelled by equation (6.26):

$$y_i = h_{\rho i}(x) = \frac{1}{\rho_i}. \tag{6.26}$$

Thus, for n landmarks being measured by the camera C_s, and assuming that $m \leq n$ multiple view measurements are available, the system output is defined as y = [$h_{\theta 1}$, ..., $h_{\theta n}$,$h_{\rho 1}$,...,$h_{\rho m}$]^T.

In (Hermann and Krener, 1977) it is demonstrated that a nonlinear system is *locally weakly observable* if the observability rank condition *rank* (*O*) = *dim*(*x*) is verified. The observability matrix *O* is computed from equation (6.27):

$$O = \left[\frac{L_f^0\left(h_{\theta 1}\right)^T}{\partial x} \ \frac{L_f^1\left(h_{\theta 1}\right)^T}{\partial x} \ \ldots \frac{L_f^0\left(h_{\theta n}\right)^T}{\partial x} \ \frac{L_f^1\left(h_{\theta n}\right)^T}{\partial x} \ \ldots \frac{L_f^0\left(h_{\rho 1}\right)^T}{\partial x} \ \frac{L_f^0\left(h_{\rho n}\right)^T}{\partial x} \right]^T \tag{6.27}$$

Where $L^i{}_f(h)$ is the i^{th} order Lie Derivative (Slotine and Li, 1991) of the scalar field of the measurement *h* with respect to the vector field *f*. Note that in equation (6.27) the zero-order and first-order Lie Derivatives are used for each bearing measurement $y_i = h_{\theta i}(x)$. In the case of multiple view measurements $y_i = h_{\rho i}(x)$ only the zero-order Lie Derivative is used.

In particular, it is investigated the case when bearing measurements $y_i = h_{\theta i}(x)$ of four landmarks are available. Hence $\hat{x} = [\hat{x}_c, \hat{\rho}_1, \hat{\rho}_2, \hat{\rho}_3, \hat{\rho}_4]$, and dim($\hat{x}$) = 10. The observability matrix *O* was symbolically computed for three different assumptions: i) no multiple view measurements are available, ii) one multiple view measurement is available, iii) two multiple view measurements are available. These assumptions gave the following results:

- First case, when there is no availability of multiple view measurements the *rank*(*O*) = 8, so there are two non-observable modes in the system.
- Second case, with a unique multiple view measurement, *rank*(*O*) = 9, which makes one more mode observable.
- Third case, when two multiple view measurements are available, *rank*(*O*) = 10, then the whole system becomes fully observable.

The above result is interesting because shows that the system could become fully observable even if only a sub-set of the landmarks seen by camera C_s is also detected by camera C_f. Also, as it could be expected, the observability of the system is improved by incorporating multiple view measurements.

6.4 Results and discussion

The whole data fusion process described, together with the inclusion of additional data during both the EKF update step and the initialization of new landmarks, was evaluated and tested within multiple environments. The tests have shown clearly how monocular SLAM approach can greatly benefit from the sparsely distributed in time data provided by the secondary camera sensor, C_f. This C_f camera, acting as an auxiliary bearing-only monocular sensors deployed as a wearable device by a human, helps composing a *virtual sensor* with instant depth estimation capabilities, creating a new hybrid monocular SLAM approach with greater accuracy and reliability.

Special focus was put in singular trajectories and worse case scenarios, like front advance in corridors and indoor closed turns. This are commonly found in areas designed for human use, and present specific challenges. Note how monocular SLAM approaches rely most of the time in side-ways movement to avoid the singular –forward advance– trajectories, and avoid close turns, expanding them to long curves. Another recurrent issue, not only on indoor visual mapping, but in structured environments, is the appearance of texture, repeated patterns, or simply, similarly looking objects, which raise the challenge of the data association problem from 'looking for a good match' to 'discriminating the correct match between the good ones'.

6.4.1 Experimental Simulations

To study performance of the method in terms of depth estimation, a robotic camera was simulated, assuming that this camera moved in a trajectory approximately parallel to a wall with known points that can be used by the system as visual landmarks (FIGURE 6.3). The orientation of the camera varies a few degrees, but it is maintained approximately perpendicular to the landmarks. In the simulations it is assumed that camera is able to track without error all the landmarks inside its field of view. The objective of the experiment is to evaluate the benefits obtained from incorporating multiple view measurements into the system for short periods of time.

The following parameters were used in simulations for the SLAM camera C_s: noise for angular measurements σ_{Cs} = 1°, field of view FoV = 70°. Multiple view measurements, which are available when there is some overlapping of the FoV of both cameras C_s and C_f, are emulated by assuming noisy measurements of range and bearing. In this case, the noise for angular measurements is $\sigma_{Csf} = 6°$, and the noise for range measurements is $\sigma_r = 0.5$ m. In the simulated experiments, the camera was moved approximately 14 meters during 30 seconds of simulation time. For two periods of time, from the second 8th to the 9th, and from the second 17th to the 19 th, it was assumed that multiple view measurements were available for being incorporated into the system.

The upper plot of FIGURE 6.3 shows the results obtained from a run of the simulation when no multiple view measurements are available (pure monocular DI-D SLAM). In this case it can be clearly appreciated a huge drift in the error of the estimated map and trajectory. In this plot it is also noticeable the degradation of the metric scale in the estimations. The lower plot of FIGURE 6.3 shows the results obtained when multiple view measurements are incorporated into the system. It is worth noting that *virtual sensor* measurements were available only during two short periods, yet this was enough to improve the estimation. FIGURE 6.4 shows the average MAE (mean absolute error) in scale (*top*) and camera position (*bottom*), obtained after 20 Monte Carlo runs of simulation. The degradation of the metric scale was measured using the function at equation (6.28):

$$s = \left\| 1 - mean\left(\frac{d_i}{\hat{d}_i} \right) \right\|, \tag{6.28}$$

where d_i is the actual depth of a feature, and the set i={1,2,..n} represents the features seen by the camera at a given time instant. The variable $\hat{d}_i$ is the estimated depth for the same i feature. In this case a relation $d_i / \hat{d}_i = 1$ represents that the metric scale of a feature has been perfectly recovered. The above expression is only computed for those features with a small covariance where it is assumed that the estimated depth has converged. Hence, in equation (6.28), small values of s imply that the metric scale is correctly propagated by the system.

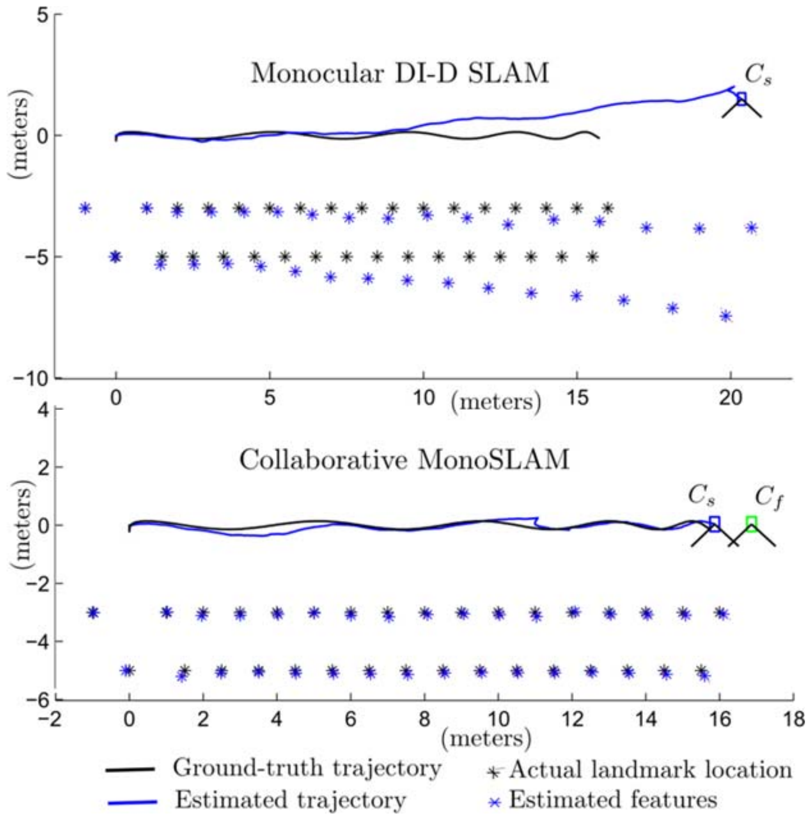


FIGURE 6.3: *Estimated map and trajectory obtained.* **Top:** *with monocular DI-D SLAM.* **Bottom:** *with collaborative monoSLAM.*

In FIGURE 6.4 it can be clearly appreciated how both the drift in the metric scale and the error in position are minimized just after the inclusion of multiple view measurements into the system. Note that the above effect is especially notorious during the second period where C_f and C_s observe concurrent scenes.

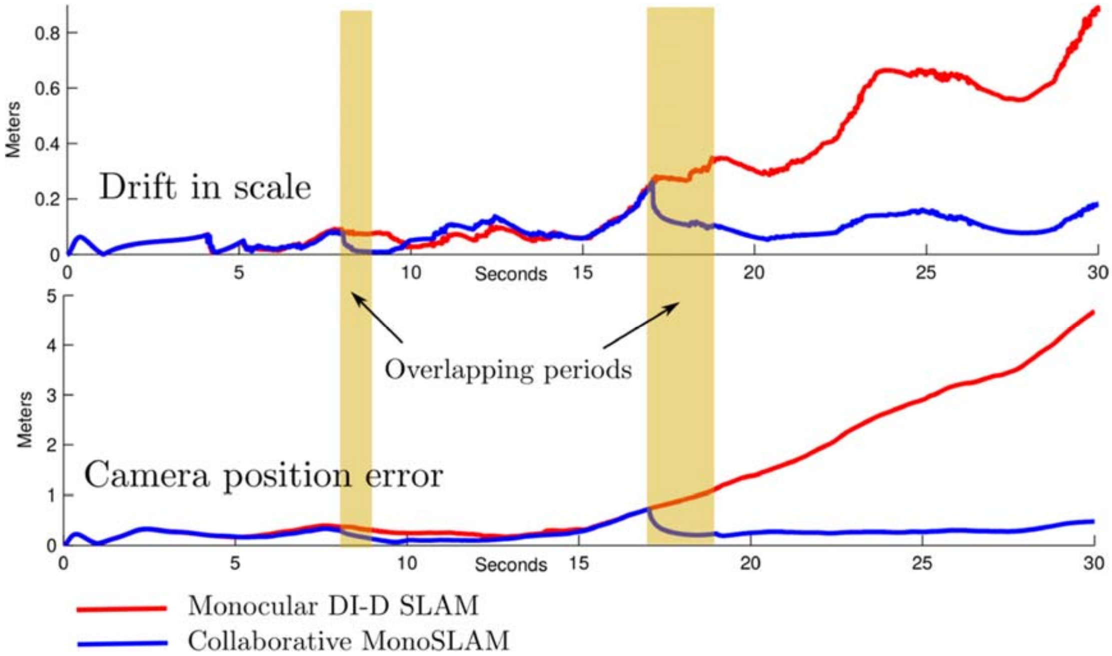


FIGURE 6.4: **Top:** *Average MAE for drift in scale.* **Bottom:** *Average MAE for camera position. For the results obtained with collaborative SLAM, the translucent rectangles indicate periods of time during of which multiple view measurements are available. Note how MAE is minimized just after that the above occur.*

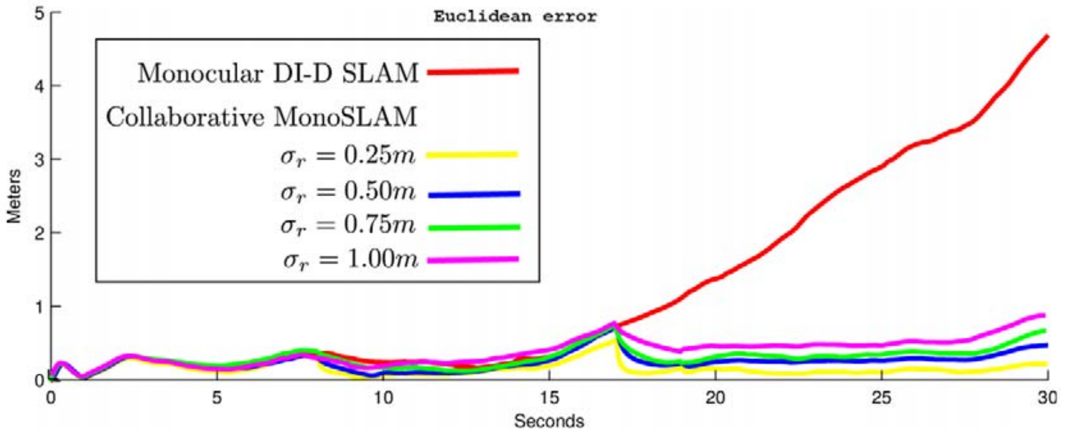


FIGURE 6.5: *Average MAE computed from camera position for different values of uncertainty σ_r in multiple view measurements. Note that even with a considerable value of uncertainty in estimates of depth provided by the virtual sensor, the MAE is well bounded compared with the purely monocular approach.*

FIGURE 6.5 shows the average MAE in camera C_s position when parameter σ_r is varied. The objective is to investigate the effectiveness of the proposed approach for different values of uncertainty in *virtual sensor* measurements. As it can be appreciated from this experiment, even, if noisy multiple view measurements are incorporated into the system, the error in the estimates can be considerably mitigated, thus proving the expected robustness of the described approach.

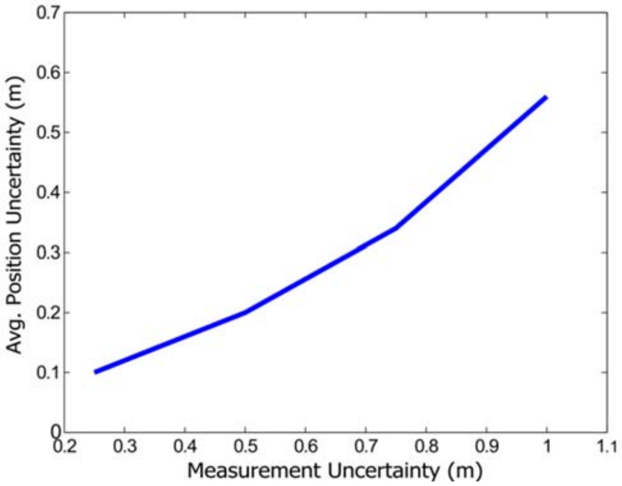


FIGURE 6.6: *Relationship between depth measurement uncertainty and average MAE position uncertainty.*

In order to see the relationship between the measurement uncertainty and the camera trajectory estimation uncertainty, the average MAE for the trajectories with varying σ_r was computed. FIGURE 6.6 shows the different average MAE for the trajectories, whose measurement uncertainty varies between 0.25m and 1m. The plot shows a strong correlation between the uncertainties in the measurement process and the estimation of the trajectory. Thus, we can conclude that an improvement in the accuracy of the depth estimation should provide a strong improvement in the general estimation of the map, reducing the uncertainty inside the EKF.

6.4.2 Singular trajectories and movements

A set of front advancing sequences were captured through ROS running over Ubuntu 12.04, and processed offline with the described technique. During the recording, the exploration team composed of a human and the robotic platform travelled a straight corridor. Note that under movements aligned with the camera depth axis only really long trajectories produce enough parallax to enable landmark depth measurement, thus these are the worst cases for delayed monocular approaches, on which this work is based. At the same time, long movements generally produce the effect that the relative perceived size of the elements on the environment vary, inducing scale variability, which combined with reflective phenomena and possible repetitive textures, reduces robustness and reliability. Many works, both in delayed and undelayed approaches, like (Clemente et al., 2007) and (Munguía and Grau, 2012), exploit distant features, initializing them with heuristic values, and rely on them to reduce the effects of noise on orientation estimation and improve stability. Though similar to this case, note that in singular movements, especially in corridors, most of the solid landmark candidates will be found as unreliable to be fully initialized under a delayed approach in a reasonable number of frames.

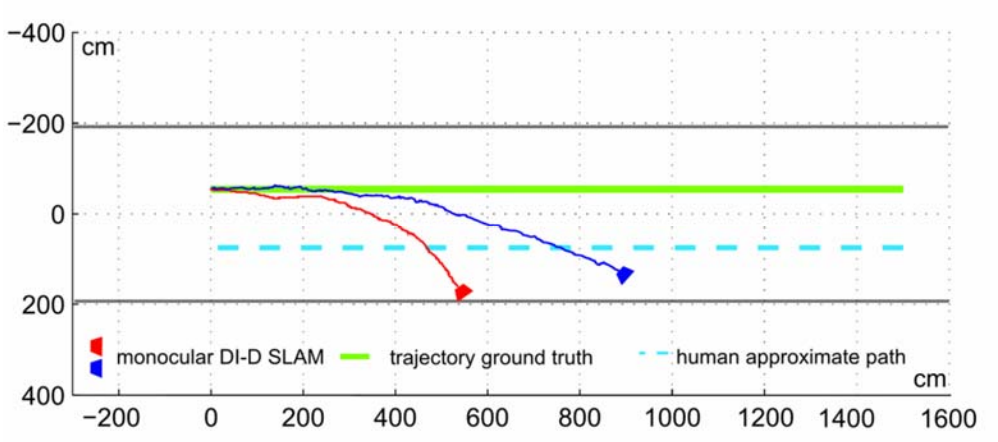


FIGURE 6.7: *Worst (red) and average (blue) cases for standard monocular DI-D SLAM within a corridor in singular trajectory.*

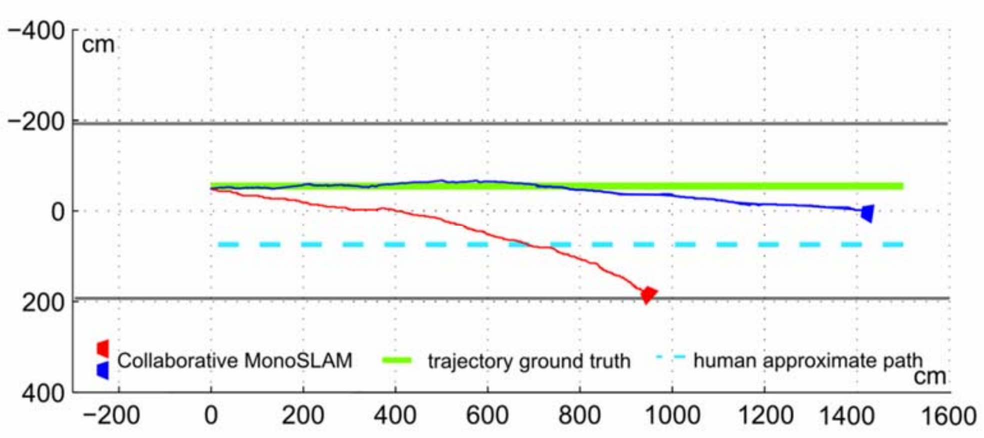


FIGURE 6.8: *Worst (red) and average (blue) cases (same sequences as* FIGURE 6.7*) for collaborative monocular SLAM within a corridor in singular trajectory.*

The battery of tests consisted in a series of several sequences captured in similarly looking corridors, trying to obtain a 15-meter trajectory map without using any of the classic large map management techniques (Bailey and Durrant-Whyte, 2006). The robot speed was adjusted to approximately match that of a walking person, between 0.75m/s and 1.5m/s. FIGURE 6.7 and FIGURE 6.8 show the estimated odometry results (the sequence of camera optical centre $\mathbf{r}^{WC}{}_{cs}$ values) for 2 of the cases: one of the worst scenarios (red) and the average case scenario, with FIGURE 6.7 showing the trajectories for the monocular SLAM and FIGURE 6.8 those for the proposed collaborative SLAM. In the worst case trajectory (red line in both figures) both approaches underestimated the displacement and achieve a huge orientation error. Still, in the proposed approach errors are lesser, with almost double the distance advance along the depth camera axis, traveling almost 60% of the 15m. For the blue trajectory (average case in both figures), the standard procedure manages to advance a notable 8.9 meters, but still incurs in a noticeable orientation error, which in larger scenarios could make all the process useless given that it was supposed to be a straight trajectory. On the other hand, the proposed approach falls short of the target by less than 1m with minimal orientation error (about 9.5º). Average error metrics from the whole set of sequences are found in TABLE 6.1.

TABLE 6.1 AVERAGE METRICS FOR DI-D MONOSLAM AND COLLABORATIVE MONOSLAM.

Technique	Avg. accumulated position error ε_{acc} (m)	Avg instantaneous position error(m)	Avg final position error(m)	Avg. overlap time ratio(s/s)
DI-D MonoSLAM	694	5.56	6.78	-
Collaborative MonoSLAM	276	2.21	3.17	0.387

The accumulated and instantaneous position errors are computed according to equations (6.29) and (6.30) respectively, with the averages for all the 15m long experiments shown on TABLE 6.1. ε_j denotes the sum of the position error for each estimated point $i=\{1..k\}$, in a given trajectory j, and ε_{acc} denotes the average ε of the different sequences. At the same time, $\bar{\varepsilon}_j$ computes the average position error for all the k steps in sequence j, and $\bar{\varepsilon}_{acc}$ accumulates this same value on average for all the 10 sequences. The average error metrics in TABLE 6.1 show how the collaborative approach has a strong advantage over the classical approach in singular movements.

$$\varepsilon_j = \sum_{i=1}^{k}\left(\left\|\mathbf{r}_i^{WC} - \hat{\mathbf{r}}_i^{WC}\right\|\right) \ , \ \varepsilon_{acc} = \frac{1}{10}\sum_{j=1}^{10}\varepsilon_j \ . \tag{6.29}$$

$$\bar{\varepsilon}_j = \frac{1}{k}\sum_{i=1}^{k}\left(\left|\mathbf{r}_i^{WC} - \hat{\mathbf{r}}_i^{WC}\right|\right) \ , \ \bar{\varepsilon}_{acc} = \frac{1}{10}\sum_{j=1}^{10}\bar{\varepsilon}_j \ . \tag{6.30}$$

All the error metrics observed produce noticeable lower values for the proposed approach than the classical DI-D approach metrics. As the drift accumulates, with locally long trajectories (without map splitting or similar approach), the error grows faster the longer it runs, so for both approaches we see that the final position error is notably over the average instantaneous error.

6.4.3 High angular speeds within small view spaces

Another recurrent issue detected in monocular SLAM approaches is that during turns the observable environment changes very quickly, frequently producing an scenario where all the features $\hat{\mathbf{y}}_i$ available in the map $\mathbf{m}$ are no longer seen in a matter of fractions of seconds (which translate into few frames). This problem is very present in the delayed feature initialization approaches: while the undelayed approaches will initialize landmarks with inaccurate depth estimations, it is entirely possible that a delayed approach is not able to find and initialize new features as quick as those in the map become no longer visible. When the number of features seen in an environment drops below a threshold (which depends on several factors, as the movement and rotation speeds, the quality of the detected features, etc.)[31], the EKF loses convergence quickly, leading to completely distorted trajectories, or in some cases, estimated trajectories which do not match the actual ones even in direction. When combined with forward aligned movements w.r.t. the camera visual axis, turn and twist become an even worse issue (see FIGURE 6.9).

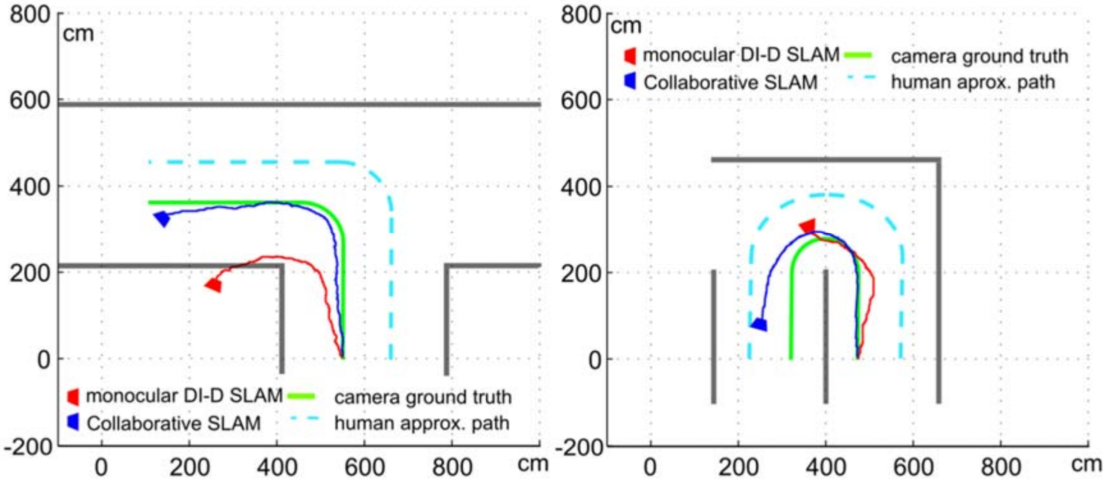


FIGURE 6.9: *Two sample trajectories, performed with delayed monocular SLAM, red line, and the multiple view collaborative monocular SLAM, blue line.* **Left:** *A sample 90º turn, one of the most common features to be found in large building dedicated to human activities.* **Right:** *Sample full U-turn (180º). Notice how the trajectory estimation fails for the pure delayed MonoSLAM approach as it is unable to initialize enough features in time.*

FIGURE 6.9 shows two experiments focused on turning. The robotic platform is traveling at 0.8m/s and performs a 90º turn and a full 180º respectively, with the human following approximately the dashed blue line. In FIGURE 6.9 *left*, the red trajectory shows how a pure monocular SLAM approach cannot really deal with a close turn, and the turn is overestimated. The trajectory estimation is further disrupted by the inability of the non-collaborative approach to fully deal with the forward camera depth movements. The final result overstates the turn by almost 80º and is not even able to keep the position estimation inside the corridors/observable environment.

[31] From a mathematical point of view, in an ideal situation 4 landmarks provide enough information to compute camera pose solving any ambiguity. But those assumptions mean very little when confronted with the reality of having approximately modelled uncertainties for each mathematical magnitude considered.

In FIGURE 6.9 *right* the trajectories estimated for the 180º show with clarity the difficulty of turning for EKF based monocular SLAM procedures. The purely monocular approach simply ends losing convergence (thus not being able to process the complete sequence in a meaningful way) after losing the orientation estimation and turning sense. As before, the forward movements are shown to be especially unsuitable for monocular SLAM approaches. The collaborative approach (blue trajectory) is able to estimate most of the trajectory done in the sequence. It is worth noting that the position error, at 0.94m, is almost as big as the case shown FIGURE 6.9 *left*, while the distance travelled is much shorter (about 6.65m). Introducing the turn, even when the orientation can be considered as correctly estimated, with a final orientation of 21.4º, has increased the drift error, with a final position error proportionally more than twice bigger than in a straight trajectory.

6.4.4 General trajectories and performance

In order to further evaluate the gains and effectivity of the proposed technique, and specifically, the impact of the measurements with the pseudo stereo procedure, a series of metrics have been developed. These metrics allow studying the effect of the periods where the overlap is available, taking into consideration factors such as the duration of the overlaps and their distribution. To test them and obtain relevant numbers, a more general sequence set, with both straight sections and turns has been captured.

The main interest is to study the interaction of the overlap periods with the gain in accuracy in the odometry estimation. With that end, two different metrics are used to study the overlap periods distribution and duration, the *τ overlap time regularity*, equation (6.31), and the *κ non-overlap time deviation*, equation (6.32). In these expressions, *N* and *M* are the number of intervals with and without overlap respectively, with η_i being the duration of i^{th} interval with overlap, and μ_j the duration of the j^{th} interval without overlap. These expressions are only useful for cases with more than a single field of view overlap period, as they measure the relation between them, trying to identify whether certain overlap distributions provide more advantages.

$$\tau = N\sum_{i=1}^{N}\left(\left|\eta_i - \frac{\eta_{total}}{N}\right|\right), \text{ where } \eta_{total} = \sum_{i=1}^{N}\left(\eta_i\right) \tag{6.31}$$

$$\kappa = \frac{1}{M}\sum_{j=1}^{M}\left(\left(\mu_j - \bar{\mu}\right)^2\right), \text{ where } \bar{\mu} = \frac{1}{M}\sum_{i=1}^{M}\left(\mu_i\right) \tag{6.32}$$

The two coefficients represent the regularity of the separation between overlap periods (*κ*), and the similarity between the duration of these overlapping periods (*τ*). In both metrics, the lower values, tending to zero, represent what is considered a better distribution of the overlap time (with the requisite that both M and N are greater than 1). A low κ value means that the intervals where overlap is present are distributed uniformly; while a lower τ value implies that these intervals of overlap are of similar duration, and that the overlap time is not concentrated mostly in a reduced number of periods.

$$U = \frac{1}{\left\| \mathbf{r}_k^{WC} - \mathbf{r}_1^{WC} \right\|} \cdot \sum_{i=1}^{k} \left(\left(\mathbf{r}_i^{WC} - \hat{\mathbf{r}}_i^{WC} \right)^2 \right) \cdot \eta_{total} \tag{6.33}$$

An additional metric has been designed to evaluate the return rate of the computational overhead (U) supposed by actively following the proposed collaborative SLAM strategy. Equation (6.33) describes this value, which is based on the cumulative squared error of the position, but considering also the length of the trajectories and the duration of the overlap periods. The duration of the overlap periods is introduced as a penalizing factor: if the squared errors are lowered by the use of the collaborative perception approach, they can offset the penalization, but if the improvements are low, U will grow. The inverse of the length of the trajectory is used as a normalizing factor: as the drift grows faster the longer the local trajectory is extended, the growth of the quadratic error and overlap penalization must be distributed along the whole trajectory.

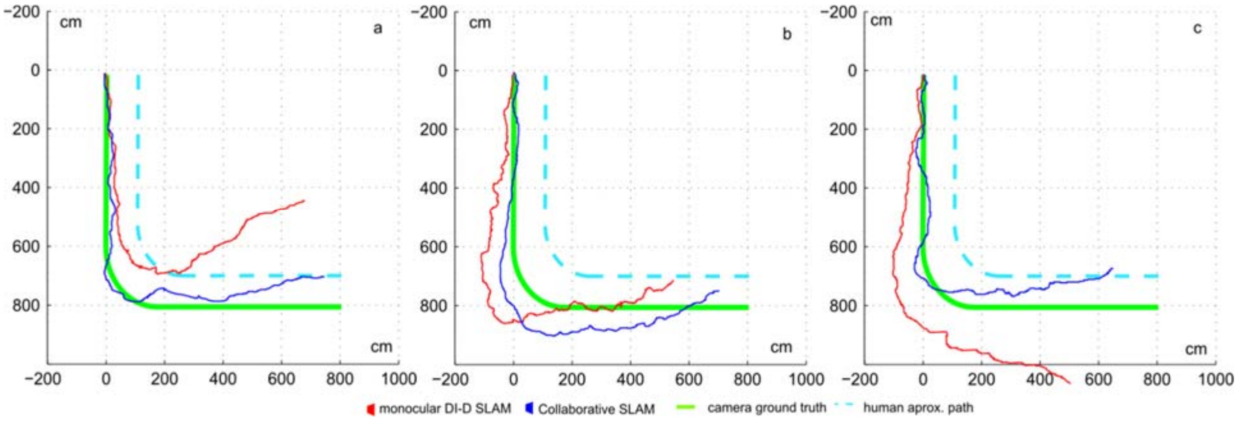


FIGURE 6.10: *Trajectories for DI-D and collaborative SLAM for cases* **a** *through* **c**.

The error and proposed metrics of the general set of sequences are shown in TABLE 6.2. Three examples of trajectories are shown in FIGURE 6.10. The introduction of the collaborative measurement into the state augmentation and estimation update processes leads to a consistent improvement into the odometry estimation. In several cases at TABLE 6.2, like FIGURE 6.10 *c*, it is observed how pure monocular SLAM cannot make locally long trajectories without further help, but the proposed approach helps improve the results notably. It is also worth noting that there may be correlation between the time where the multiple view measurement is available (noted as overlap time), and a decrease in the odometry error.

TABLE 6.2: METRICS FOR COLLABORATIVE MONOSLAM OVERLAP TIME EVALUATION.

Sequence	**DI-D SLAM errors (m)**		**Collaborative MonoSLAM err.(m)**		**Overlap time ratio (s/s)**	***τ***	***κ***	***U***
	Final position	**Avg. instant**	**Final error**	**Avg. instant error**				
a	3.88	3.17	1.24	0.82	0.23	1.9	1.4	33.8
b	2.84	2.00	1.19	0.69	0.32	1.1	0.7	36.9
c	4.12	3.04	2.51	1.72	0.22	1.4	1.2	126.2
d	3.19	2.26	1.46	0.98	0.27	2.2	2.9	54.3
e	5.32	4.12	2.43	1.73	0.38	3.4	2.5	223.1
f	3.45	2.15	1.68	1.28	0.37	2.8	3.3	115.6
g	4.83	3.34	1.74	1.12	0.58	4.1	1.8	159.1
h	3.96	2.87	1.97	1.34	0.33	1.7	2.3	122.4
i	5.91	4.43	2.73	2.26	0.49	1.6	0.8	316.7
j	4.73	3.92	2.35	1.54	0.44	3.8	3.2	232.5

FIGURE 6.11 plots the final position error for each sequence, with and without the introduction of *virtual sensor* measurements, linking the errors of each sequence, against the overlap time rate. In this figure it can be observed how the distance between the errors for the classic approach and the proposed approach grow as the overlap time ratio grows.

The proposed metrics, κ, τ and U produced mixed results. While τ showed no appreciable correlation between the regularity of the overlap periods and the different error metrics, κ exhibits some more relation between the results. Although intuitively, splitting the overlap time in several periods in a spaced manner should be more convenient, as it reduces the covariance between the observed features and the camera to that uncertainty of the multiple view measurement, the data obtained is not conclusive enough to infer a correlation.

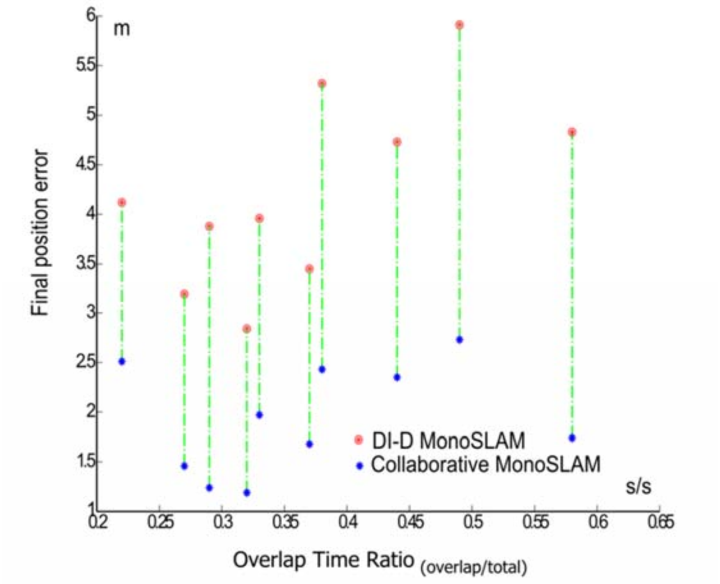


FIGURE 6.11: *Final position error versus overlap time rate for sequences* **a** *through* **j**.

On the other side, the *U* value offered insight and helped provide an analysis less focussed on accuracy and centred on the costs of the multiple view measurements. The computational costs of the DI-D monocular SLAM have been already discussed in (Guerra et al., 2013), and given a fixed maximum on the number of features, it can be assumed to be bound by an upper limit. Then it is logical to observe the other process with great computational costs associated, which is the introduction of the *virtual sensor*. The costs are incurred because the proposed technique requires to search for points of interest at one image and to compute SURF descriptors of two frames at each EKF iteration where is applied. Thus, the *U* value helps to keep in perspective the trade-off between accuracy and cost. On average, the additional overhead introduced by the multiple view measurement procedure supposed less of a quarter of the total computational cost (about 23% of time) in the simple MATLAB implementation. Still, this cost could increase as this overhead was only incurred in 37% of the frames on average. In a worst case run, where the *virtual sensor* cost penalty is incurred for each frame (even when there is no overlap between fields of view), this penalty becomes almost the 45% of the time consumed. This increased computational cost would probably make the approach unmanageable in real time, unless deep work at optimization was performed.

6.5 Conclusions

This chapter describes a completed approach to the monocular SLAM problem, where data obtained from a human-deployed sensor is fully fused into the EKF SLAM methodology. The data produced by the secondary sensor allows converting the standard monocular measurements (detailing heading and attitude)[32] into full-observability measurements, which also include the depth. These augmented measurements are used in all the steps of the EKF, including the measurement and update step of the extended Kalman filter and the feature initialization, building upon previous chapter, where only the feature initialization task (Guerra et al., 2014) used the multiple view *virtual sensor* depth estimation. This implies that the multiple view measurement procedure has to be accounted for both in the direct and inverse observation models, as it is used in both steps. While the C_f camera can move freely, a combination of data from the robotic sensors and the wearable devices allows estimating its pose with respect to the robotic camera C_s. It was discussed in Chapter 5, although it is possible to perform a full stereo process based on epipolar geometry with the available data, the epipolar stereo estimation was reject based on the image processing required warping images according to the relevant homographies, according (Fusiello et al., 2000) or any related approaches. Thus, matching points with SIFT/SURF descriptors proved to be the most convenient approach.

One of the shortcomings in the work described in chapter 5 and presented in (Guerra et al., 2014) is the utilization of a standard undelayed inverse observation model to compute the update of the covariance matrix once a new feature was introduced into the EKF. As the complete approach requires the formulation of new Jacobians to compute the Kalman gain and innovation covariance during the update step, the initialization process has been updated to use a more accurate representation of the process covariance, although its impact is thought to be small. The update step has been deeply modified, introducing measurements with full depth obtained without delay, instead of only measuring features in terms of pixel coordinates. In order to support these measurements, the classical Kalman innovation formulation for pixel-based features has been updated. The described procedure builds the Jacobian ∇H once all the features have been correctly measured, in order to know if any given feature will be treated as pixel in camera frame coordinates or as fully measured point. While delaying the construction of the Jacobian produces a slower approach than building it along the measurement process, as typically done in monocular approaches (Munguía and Grau, 2012), it avoids the dynamic matrix resizing penalization incurred by having to refit a partially built Jacobian matrix.

An initial study in simulations allowed characterizing the gains and advantages of the approach with respect to the uncertainty in the feature measurements. The results of these

[32] Pixel coordinates are essentially the description of a ray w.r.t. the camera optical center, thus they can be interpreted as spherical coordinates.

simulations showed a high correlation between the uncertainty in the depth measurement and that of the state of the system, especially in terms of the camera position estimation.

The experimental sequences captured have allowed testing the proposed methodology with real data. The main focus has been evaluating the strengths of the proposed technique, both as a general approach, and specifically against the most troublesome scenarios for classical monocular SLAM, be it delayed or undelayed. Thus, multiple sets of sequences were captured: on one hand those looking like a general trajectory, and on the other hand specific sequences with singular movements in mind, like those aligned with the depth axis of the camera, and close turns. For processing the sequences, no large map management technique was used, thus all the drift was accumulated over. These sequences show how the proposed approach has much more accuracy and resilience than ordinary monocular EKF SLAM. The forward advance sequences show clearly how monocular EKF SLAM has many troubles estimating the forward movement, while the proposed approach estimates the trajectory with greater accuracy. On the other side, the turning sequences showed that close turns are probably one of the hardest movements for monocular SLAM to estimate, to the point of completely losing convergence if quick enough. These claims have been further proved by the computed error metrics.

During the experiments the collaborative SLAM approach was executed offline in a MATLAB implementation, thus time performance data would be unreliable. Still, previous works based on the same monocular SLAM methodology performed robustly on real-time, as seen in Chapter 4 and (Guerra et al., 2013). Additional computational overhead introduced by processing two images per frame when overlap is found and matching the SURF descriptors could be dealt using parallel processing of the images within a strong implementation from a computer science point of view.

Part IV

Concluding Remarks

Though my eyes could see I still was a blind man,

though my mind could think I still was a mad man.

Masquerading as a man with a reason,
my charade is the event of the season.

And if I claim to be a wise man, well, it surely means that I don't know

Carry on my wayward son -Kansas

Excerpt from *Carry on my wayward son*, from Kansas. Many researchers have faced the struggle, tribulations and self-doubt of the wayward son, with its lyrics being curiously resounding in the field of robotics, especially for perception and AI.

This thesis has presented efforts of some years, both with successes and failures, researching the field of monocular SLAM. In a certain sense, after all this research, I feel more like an expert in how you should not try to solve the SLAM problem, possessing a clear sight over the vast void we still have to fill to actually solve it, than someone who knows the actual solution. It makes me wonder how far we are from a generalist solution with human-like performance[33] within assailable computational requirements, and many other questions: Are we even on the right path? Which is the penalization we are paying in terms of computational efforts by working with points, lines, and other accurate mathematical entities instead of uncertain generalizations like the human brain? Will we

[33] Measured in results. Though we are far from solving the SLAM problem, I am pretty confident that neuroscience is several orders of magnitude farther away of comprehending how the human mind works.

even be able to emulate the performance and adaptability of the human sensory system without knowing its inner workings?

From the vantage point of having studied the SLAM problem and its vision based solutions, it can be seen that many of the ‘solved challenges’ still are in its infancy, and the general SLAM solution looks far away. Still, as it will be presented and discussed in the following chapter, I am confident that the research developed can prove itself useful to the field, maybe not by becoming a universally accepted standard, but by providing and proving certain insights otherwise untested or unfounded. As it has been discussed, under certain circumstances our developments for data association validation can suppose a reduction in computational efforts of several orders of magnitude. At the same while filter based visual SLAM is starting to feel dated, our work in collaborative perception focused on joining the data through a *virtual sensor* strategy, being an uncommon and untested approach, which was proven successful, presenting new opportunities and challenges applicable to any strategy of visual SLAM in the right circumstances.

Chapter 7

Conclusions and future work

Flyability Elios UAV vs ascience fiction mapping drone (courtesy of Flyability and 20th Century Fox, respectively). The german word *weltschmerz* was coined by Jean Paul to denote the pain and anxiety produced by the comparison between how it is and how we think should be. Which word should we use to denote the deception between what was promised by visionaries, salesmen and tech gurus, and what was actually delivered?

7.1 Introduction

The relevance of the SLAM problem has been commented several times, and cannot be overstated. The research in this problem has led to several publications as well as the work presented in this thesis. To conclude this thesis, this chapter lists the publications produced in the research field, with commentary on the contribution of each publication, the work done, the achievements, and also the frustrations and failures. This means that the list includes even those works and results which, although pertaining to the SLAM field, ended largely unrelated to the visual SLAM focus of the thesis. The final section is devoted to present the general conclusions of my research, briefly discussing the results achieved, the future opportunities that remain opened in the lines of research I worked, and commenting what I expect of the research in the SLAM problem for the future.

7.2 Publications and contributions

The work and results presented in this dissertation are the fruit of an initial will to melt several areas of research and interest of the members of the Vision and Intelligent Systems research group into a unique crucible. The initial preliminary research pointed towards studying the viability of producing new bearing-only mapping approaches, not necessary based on vision. Thus, the first publication on the SLAM area of research proposed an approach based on sound mapping, and discussed some of the early results obtained:

- Edmundo Guerra, Yolanda Bolea, Antoni Grau, Rodrigo Munguía (2011). **New approach on bearing-only SLAM for indoor environments**, in *Proceedings of the 16th IEEE Conference on Emerging Technologies Factory Automation (ETFA).*
  DOI:**10**.1109/ETFA.2011.6059227.

As sound probed being too unreliable as a way of measuring the environment with the available resources, the research was eventually redirected into studying the viability and expected gains of introducing modelling techniques generally associated with the field of automatic control, like linear parameter varying (LPV) EKF, and pseudo-measurement methodologies.

- Edmundo Guerra, Yolanda Bolea, Antoni Grau (2012). **Pseudo-measured LPV Kalman filter for SLAM**, in *Proceedings of the 10th IEEE International Conference on Industrial Informatics (INDIN).*
  DOI:10.1109/INDIN.2012.6301358

Though in terms of mathematical theoretical development and 2D simulations the method appeared to be advantageous, this did not translate into actual gains into the real vision-based SLAM methodology: the mathematical development enabling the pseudo-measurement method in 2D was not applicable in the 3D scenario.

Tests with the delayed I-D monocular SLAM developed in parallel with the commented works, revealed that it still presented weaknesses and issues that could be addressed. The most urgent need detected was a way to address robustly the data association problem, which led to Part II of this dissertation. An initial proposal for the HOHCT method for data validation was presented in:

- Edmundo Guerra, Rodrigo Munguía, Yolanda Bolea, Antoni Grau (2013). **New validation algorithm for data association in SLAM**, in *ISA Transactions*. Vol 52(2013): 662-671. DOI:10.1016/j.isatra.2013.04.008. (2013 IF: 2.256 Ranking: 9/85 - Q1)

In addition to the results presented in paper commented above, several more tests were performed, and the profiling implementation of the algorithm was polished to obtain more real-time-like statistics. All these improvements were published in:

- Edmundo Guerra, Rodrigo Munguía, Yolanda Bolea, Antoni Grau (2013). **Validation of Data Association for Monocular SLAM**, in *Mathematical Problems in Engineering*. Volume 2013, Article ID 671376, 11 pages. DOI:10.1155/2013/671376. (2013 IF: 1.082 Ranking: 33/87 Q2)

Additional work with the HOHCT was developed, trying to reduce the penalization produced removal of "good features" due to a single failure of the SMD test. A multiple strike-based policy was tested, delaying removal of landmarks until they tested as jointly incompatible multiple times. The results probed that the trade-off was neutral in the 'best case' scenarios; but in the average cases it penalized performance, as it increased the possibilities of multiple incompatible pairings, which are inconvenient, as discussed in Chapter 4.

- Edmundo Guerra, Yolanda Bolea, Antoni Grau (2014). **Policy-based optimization for matching validation algorithm in monocular robotics**, in *Proceedings of the 2014 Complexity in Engineering (COMPENG)*. Barcelona, Spain, 16-18 June 2014. DOI:10.1109/CompEng.2014.6994678.

The approach to producing a collaborative virtual sensor was focused initially in dealing with the metric scale initialization problem, as discussed in Chapter 5. The initial works with the collaborative sensing, including description of the hardware system, the theoretical framework to work the multiple view geometry and the first results solving the feature depth initialization problem in a multimodal system were published in:

- Edmundo Guerra, Rodrigo Munguía, Antoni Grau (2014). **Monocular SLAM for Autonomous Robots with Enhanced Features Initialization**. *Sensors*, Vol. 14, pages 6317–6337. DOI:10.3390/s140406317. (IF: 2.245 Rank 10/53 Q1).

A follow-up to this work, introducing an accurate Jacobian for the inverse observation model, as described in Chapter 5, and additional experimentation focused in industrial like corridors was presented in:

- Edmundo Guerra, Rodrigo Munguía, Antoni Grau (2015). **Human-Robot SLAM in industrial environments**. In *Proceedings of the IEEE*

International Conference on Industrial Informatics 2015 (INDIN). Pages 390-395.

This line of research also produced an invited chapter in a book, focusing in an expanded state of the art review, and shifting the work's discussion towards a more theoretical approach.

- Edmundo Guerra, Yolanda Bolea, Rodrigo Munguía, Antoni Grau (2016). **Recent Development in Monocular SLAM within an HRI Framework**. Published in *Recent Advances in Robotics Systems*, pages 87-105. Digital version: ISBN 978-953-51-2571-6; Printed version: ISBN 978-953-51-2570-9. DOI: 10.5772/63820

The method was further expanded to include the full total of the works presented in Part III, including the hybridized monocular SLAM with the virtual sensor to enable multiple view estimation fully integrated in the EKF methodology (including the measurement and update steps). This work, with a small set of results was presented as an invited presentation in a workshop at the IROS congress:

- Edmundo Guerra, Yolanda Bolea, Antoni Grau (2015) **Human-assisted mapping of urban environments in a robotic framework/with bearing-only cameras**. In *Urban Robotics Applications Workshop of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany, September 28 - October 02, 2015

Once the fully integrated multiple view estimation was thoroughly tested, and the new metrics developed to evaluate the behaviour of the secondary monocular sensor and the effects of the availability of the collaborative measurements in short local trajectories were completed, the results were collected in a paper:

- Edmundo Guerra, Rodrigo Munguía, Yolanda Bolea, Antoni Grau (2016). **Human collaborative localization and mapping in indoor environments with non-continuous stereo** *Sensors* Vol. 16, num. 3, p. 1-23. DOI: 10.3390/s16030275. (2015* IF:2.033 12/56 Q1).

In addition to the original research performed with the VIS group, the author also collaborated in other works related to the SLAM problem:

- David Gómez, Rodrigo Munguia, Edmundo Guerra, Antoni Grau (2014). **Full autonomous navigation for an aerial robot using behavior-based control motion and SLAM**. *In Proceedings of the 19th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA).* DOI: 10.1109/ETFA.2014.7005240

7.3 Conclusions and future work

The SLAM problem is probably the most important challenge to be solved in order to have truly autonomous robots. This problem constitutes the essential perception task to understand the environment, learn it, and comprehend the spatial relations between the elements present and the robot itself. The work have been developed in a feature point based visual framework: as we are willing to create autonomous robots that shall operate in spaces developed for human beings, it is only natural that we try to match human senses used for the tasks. Thus, the utilization of visual perception, as we commented earlier during the thesis, provides enormous quantities of data, introducing the problem of processing it. This problem, closely related to the map representation, can be solved in different ways depending on which level we want to work: most of the approaches deal it using salient point features as landmarks, based on point detectors and descriptor; but there are also works dealing higher level primitives, complex object and relation detection based approaches.

Keeping a point feature-based strategy to process the measurements and represent the environment may probe unsuitable for achieving a comprehensive spatial mapping and producing semantically rich maps: there are plenty of works which consider additional data, for example, higher level features like lines, or including object recognition and plane estimation. Still, these techniques generally provide little gains in terms of accuracy in the localization[34] problem. Moreover, especial care is required when dealing with recognition of high level patterns and objects that are commonly repeated in human inhabited environment. Thus, point feature based SLAM is still the standard monocular SLAM strategy.

Notice that when discussing the fitness of point features as basis for a SLAM approach, I denoted specifically in monocular SLAM. Although when I started my research monocular sensors were the obvious choice, the current availability of inexpensive sensors, pushed by the development of MEMS for consumer electronics, makes that decision questionable as of today. Introduction of inertio-visual strategies could help enhance results, as many other sensors, and their availability would make hard to justify ignoring them from a logistics point of view. Still, introduction of said sensors is known to add complexity to the SLAM problem. I feel that even today this could disrupt research lines where the problem still have not been fully explored, hindering development of purely monocular solutions by producing earlier improvements through multimodal techniques; and also hinder developments in the emergent research trying to gap the divide between the visual SLAM problem and the novel computer vision techniques based in convolutional neural networks.

As for the work presented itself, our research into the data association problem has improved the base considered monocular SLAM technique with the introduction of the

[34] Understanding localization as pose estimation with respect to the map.

HOHCT algorithm to assess the joint compatibility. As it was reported in (Munguía and Grau, 2012), this algorithm was able to compete with state of the art approaches without using data association validation. Then, the introduction of the HOHCT improved the delayed monocular SLAM, giving it not only more accuracy in the general cases, but improving greatly its robustness against disruptive conditions. The HOHCT algorithm has also been probed to beat which used to be considered the golden standard of data validation (JCBB) in the average cases for the considered delayed monocular SLAM approach.

Work with respect to the data association problem can progress in several directions, accounting for the current state of the art in SLAM. Firstly, given the increase in computational power and the emergence of new development tools to program at GPU level, there is margin to work in the matching and measurement step keeping the EKF architecture. This could be combined with map management techniques to produce denser maps. Notice that although modern approaches, dominated by the Bundle Adjustement (BA) technique, use high level feature descriptors to solve both detection and matching, and improve the solutions to other problems like place recognition, active search strategies are still active an active field of research in high performance computation specialized works, as mentioned in (Törtei Tertei et al., 2016). Thus, the presented work could be expanded aiming towards highly specific architectures where, through parallelization and other advanced programming techniques, it could be possible to introduce several upgrades, e.g.: working with denser maps, or introduction of advanced measurement models allowing estimation at subpixellic resolution.

Within validation step itself, as current trends in visual SLAM point towards optimization based approaches, the use of a batch gating based technique may look dated. Still, as it was discussed in (Strasdat et al., 2010), below a given computational power threshold, filter-based SLAM approaches produce a better trade-off than BA in terms of accuracy against computational power required. Thus, apart from high-end purely research-based application, most of the real world applications still are based on filtering approaches. Moreover, data association (DA) is still being solved through JCBB and its derivatives are being researched in several problems, like multimodal sensing (Li et al., 2014), scan based association (Shi et al., 2014), and point cloud matching (Shen et al., 2016).

Besides the data association, the other feature identified in the delayed monocular SLAM approach which offered the most interesting opportunities was the landmark feature initialization. As it has been described, the delayed monocular SLAM generally requires an initialization process to introduce some landmarks with actual depth measurements to produce scale. The shift towards the multiple view architecture under the human collaborative sensing framework offered the chance to introduce an improved method to initialize the features. Instead of introducing a given set of known features, thus requiring calibration a measurements, the produced technique allows initialization under fully unknown scenarios. Moreover, the scale is propagated in a smother way, as the initial feature set is not limited to a small set of coplanar features, thus being more representative of the different depths observed in the environment.

As the results of introducing the multiple view sensing into the feature initialization process were successful, the logical conclusion was to complete the integration of the multiple view sensing into the SLAM methodology. For the data collected, the analysis of the impact produced by the availability of the alternative sensing process probed that the most positive outcome was when the multiple view sensing was available during the whole sequence at regular intervals. This validates the intuitive idea that the multiple view sensing, although it presents a set of errors and uncertainties due the composition of transformations estimated by several sensors, it helps bounding the uncertainties, as new landmarks and multiple view measurements present an approximately constant scale error, unlike pure delayed monocular SLAM. This improves the general accuracy of the localization, becoming especially noticeable in singular movements, as it has been studied, and in rapid turns, which constitute the worse cases for SLAM.

Still, the proposed collaborative sensing framework can be further developed to exploit the multiple view sensing capabilities and the HRI opportunities. Fully integrating the state of the different elements of the collaborative virtual into the EKF state should provide interesting results, though it would rapidly converge into an approach to collaborative mapping. The opportunities here would range from introducing additional mapping elements (like the camera performing SLAM), to add sensing capabilities through additional secondary devices, e.g., instead of a human and a robot collaborating, a small group of humans with one or more robotic devices exploring and exhaustively mapping an area. In this latter example, the HRI could become explicit: the humans can provide knowledge and object recognition capabilities to enhance the map, thus upgrading the mapping results to include clouds of points and annotations about specific areas or clusters of features. An approach like this can be directly applied into SAR situations, and would provide invaluable help.

Going back to the field of monocular SLAM, I would like to note how the most successful approaches in recent years have been more a result of masterful technical integration, than purely novelty research. This has translated into an increased threshold to produce relevant research, as any novel SLAM approach is expected to implement solutions to all the different challenges within the SLAM problem[35]. Although at first thought this should not constitute too much of a problem, it is worth noting how for example, in the current bundle adjustment architectures, although there are different threads to perform the different tasks, the coupling between said processes, not only from a technical point of view but also from a theoretical and mathematical one, has increased, and is even greater than in classical filtering approaches. This means that novel research into any of the challenges of SLAM

[35] Not necessarily all the solutions must be novel research. In fact it is becoming increasingly common to produce research where the contribution lays more into the innovation than in the novelty.

tends to represent increasing loads of technical work for diminishing returns in terms of validation of new theoretical approaches.

Because of this, and the influence that has already been exerted by the open source and free software movements in many research, I feel inclined to believe[36] that we should start seeing in a few years the emergence of modular architecture/s to solve the SLAM problem. In this architectures software design criteria will become more relevant, and different layers of interfaces will eventually become de-facto standards. These interfaces, in the same measure that they fix certain criteria that should be met, will also allow to work more freely inside the different problems they isolate, and will help sharing solutions, both for dissemination and cross testing. Still, the complexity in the design steps of these modular architectures means that though it is possible that more than one appear, I would not expect them to be common enough to fragment the research community back into a point where producing novel results require more technical effort than theoretical development.

[36] I would not go as far to say expect, but I would not consider it a vain hope.

Bibliography

Abbe, E. (1890). Measuring Equipment for Physicists (Messapparate für physiker). *Z. Für Instrumentenkunde* 10, 446–447.

Aidala, V., and Hammel, S. (1983). Utilization of modified polar coordinates for bearings-only tracking. *IEEE Trans. Autom. Control* 28, 283–294. doi:10.1109/TAC.1983.1103230.

Alahi, A., Ortiz, R., and Vandergheynst, P. (2012). FREAK: Fast Retina Keypoint. in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 510–517. doi:10.1109/CVPR.2012.6247715.

Alspach, D., and Sorenson, H. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Trans. Autom. Control* 17, 439–448. doi:10.1109/TAC.1972.1100034.

Anandan, P. (1987). Measuring Visual Motion From Image Sequences. Technical Report. Amherst, MA, USA: University of Massachusetts.

Andersson, L. A. A., and Nygards, J. (2008). C-SAM: Multi-Robot SLAM using square root information smoothing. in *2008 IEEE International Conference on Robotics and Automation*, 2798–2805. doi:10.1109/ROBOT.2008.4543634.

Armangué, X., and Salvi, J. (2003). Overall view regarding fundamental matrix estimation. *Image Vis. Comput.* 21, 205–220. doi:10.1016/S0262-8856(02)00154-3.

Armesto, L., and Tornero, J. (2004). SLAM based on Kalman filter for multi-rate fusion of laser and encoder measurements. in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings*, 1860–1865 vol.2. doi:10.1109/IROS.2004.1389668.

Atrey, P. K., Hossain, M. A., Saddik, A. E., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimed. Syst.* 16, 345–379. doi:10.1007/s00530-010-0182-0.

Bailey, T., Bryson, M., Mu, H., Vial, J., McCalman, L., and Durrant-Whyte, H. (2011). Decentralised cooperative localisation for heterogeneous teams of mobile robots. in *2011 IEEE International Conference on Robotics and Automation*, 2859–2865. doi:10.1109/ICRA.2011.5979850.

Bailey, T., and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): part II. *IEEE Robot. Autom. Mag.* 13, 108–117. doi:10.1109/MRA.2006.1678144.

Bailey, T., Nieto, J., and Nebot, E. (2006). Consistency of the FastSLAM algorithm. in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*, 424–429. doi:10.1109/ROBOT.2006.1641748.

Bar-Shalom, Y. (1987). *Tracking and Data Association*. San Diego, CA, USA: Academic Press Professional, Inc.

Bar-Shalom, Y., and Tse, E. (1975). Tracking in a cluttered environment with probabilistic data association. *Automatica* 11, 451–460. doi:10.1016/0005-1098(75)90021-7.

Bartoli, A., and Sturm, P. (2001). The 3D line motion matrix and alignment of line reconstructions. in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, I-287-I-292 vol.1. doi:10.1109/CVPR.2001.990488.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). “Surf: Speeded up robust features,” in *Computer Vision–ECCV 2006* (Springer), 404–417.

Besl, P. J., and McKay, N. D. (1992). A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* 14, 239–256. doi:10.1109/34.121791.

Blair, J. R. S., and Peyton, B. (1993). “An Introduction to Chordal Graphs and Clique Trees,” in *Graph Theory and Sparse Matrix Computation* The IMA Volumes in Mathematics and its Applications (Springer New York), 1–29. doi:10.1007/978-1-4613-8369-7_1.

Blanco, J. L., Gonzalez-Jiménez, J., and Fernandez-Madrigal, J. A. (2012). An Alternative to the Mahalanobis Distance for Determining Optimal Correspondences in Data Association. *IEEE Trans. Robot.* 28, 980–986. doi:10.1109/TRO.2012.2193706.

Bosse, M., Newman, P., Leonard, J., Soika, M., Feiten, W., and Teller, S. (2003). An Atlas framework for scalable mapping. in *IEEE International Conference on Robotics and Automation, 2003,* 1899–1906 vol.2. doi:10.1109/ROBOT.2003.1241872.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. in *European conference on computer vision* (Springer), 778–792.

Carlone, L., Ng, M. K., Du, J., Bona, B., and Indri, M. (2011). Simultaneous Localization and Mapping Using Rao-Blackwellized Particle Filters in Multi Robot Systems. *J. Intell. Robot. Syst.* 63, 283–307. doi:10.1007/s10846-010-9457-0.

Castellanos, J. A., Montiel, J. M. M., Neira, J., and Tardós, J. D. (1999). The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Trans. Robot. Autom.* 15, 948–952.

Censi, A., and Scaramuzza, D. (2014). Low-latency event-based visual odometry. in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 703–710. doi:10.1109/ICRA.2014.6906931.

Chatterjee, C., and Roychowdhury, V. P. (2000). Algorithms for coplanar camera calibration. *Mach. Vis. Appl.* 12, 84–97.

Chebira, A., Dragotti, P. L., Sbaiz, L., and Vetterli, M. (2003). Sampling and interpolation of the plenoptic function. in *Proceedings 2003 International Conference on Image Processing*, II-917-20 vol.3. doi:10.1109/ICIP.2003.1246832.

Checchin, P., Gérossier, F., Blanc, C., Chapuis, R., and Trassoudaine, L. (2010). "Radar Scan Matching SLAM Using the Fourier-Mellin Transform," in *Field and Service Robotics* Springer Tracts in Advanced Robotics., eds. A. Howard, K. Iagnemma, and A. Kelly (Springer Berlin Heidelberg), 151–161.

Cheein, F. A. A., Lopez, N., Soria, C. M., Sciascio, F. A. di, Pereira, F. L., and Carelli, R. (2010). SLAM algorithm applied to robotics assistance for navigation in unknown environments. *J. NeuroEngineering Rehabil.* 7, 10. doi:10.1186/1743-0003-7-10.

Chekhlov, D., Pupilli, M., Mayol-Cuevas, W., and Calway, A. (2006). "Real-time and robust monocular SLAM using predictive multi-resolution descriptors," in *Advances in visual computing* (Springer), 276–285.

Chiuso, A., Favaro, P., Jin, H., and Soatto, S. (2000). "3-d motion and structure from 2-d motion causally integrated over time: Implementation," in *Computer Vision—ECCV 2000* (Springer), 734–750.

Civera, J., Davison, A. J., and Montiel, J. M. M. (2006). Unified inverse depth parametrization for monocular slam. in *In Proceedings of Robotics: Science and Systems*.

Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M. (2009). 1-point RANSAC for EKF-based Structure from Motion. in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009*, 3498–3504. doi:10.1109/IROS.2009.5354410.

Civera, J., Grasa, O. G., Davison, A. J., and Montiel, J. M. M. (2010). 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. *J. Field Robot.* 27, 609–631.

Clemente, L. A., Davison, A. J., Reid, I., Neira, J., and Tardós, J. D. (2007). Mapping large loops with a single hand-held camera. in *Robotics: Science and Systems*.

Cummins, M., and Newman, P. (2008). Accelerated appearance-only SLAM. in *IEEE International Conference on Robotics and Automation, 2008. ICRA 2008*, 1828–1833. doi:10.1109/ROBOT.2008.4543473.

Cunningham, A., Paluri, M., and Dellaert, F. (2010). DDF-SAM: Fully distributed SLAM using Constrained Factor Graphs. in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3025–3030. doi:10.1109/IROS.2010.5652875.

Dalal, N., and Triggs, B. (2005). Histograms of oriented gradients for human detection. in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 886–893 vol. 1. doi:10.1109/CVPR.2005.177.

Dale, S. A., Kitching, I. D., and Daly, P. (1989). Position-fixing using the USSR's GLONASS C/A code. *IEEE Aerosp. Electron. Syst. Mag.* 4, 3–10. doi:10.1109/62.16990.

Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. in *IEEE International Conference on Computer Vision*, 1403–1410.

Davison, A. J., Cid, Y. G., and Kita, N. (2004). Real-Time 3D SLAM with Wide-Angle Vision. in *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*.

Davison, A. J., and Murray, D. W. (2002). Simultaneous localization and map-building using active vision. *Pattern Anal. Mach. Intell. IEEE Trans. On* 24, 865–880.

Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 1052–1067. doi:10.1109/TPAMI.2007.1049.

Dellaert, F., and Kaess, M. (2006). Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *Int. J. Robot. Res.* 25, 1181–1203. doi:10.1177/0278364906072768.

Diosi, A., Taylor, G., and Kleeman, L. (2005). Interactive SLAM using laser and advanced sonar. in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 1103–1108.

Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *Robot. Autom. IEEE Trans. On* 17, 229–241.

Durrant-Whyte, H., and Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE Robot. Autom. Mag.* 13, 99–110. doi:10.1109/MRA.2006.1638022.

Durrant-Whyte, H. F. (1988). Uncertain geometry in robotics. *IEEE J. Robot. Autom.* 4, 23–31. doi:10.1109/56.768.

Durrant-Whyte, H., Rye, D., and Nebot, E. (1996). "Localization of Autonomous Guided Vehicles," in *Robotics Research*, eds. G. Giralt and G. H. P. Dr.-Ing (Springer London), 613–625.

Durrant-Whyte, H., Stevens, M., and Nettleton, E. (2001). Data fusion in decentralised sensing networks. in *Proceedings of the 4th International Conference on Information Fusion*, 302–307.

Eade, E., and Drummond, T. (2009). Edge landmarks in monocular SLAM. *Image Vis. Comput.* 27, 588–596. doi:10.1016/j.imavis.2008.04.012.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W. (2012). An evaluation of the RGB-D SLAM system. in *Robotics and Automation (ICRA), 2012 IEEE International Conference on* (IEEE), 1691–1696.

Engel, J., Schöps, T., and Cremers, D. (2014). LSD-SLAM: Large-scale direct monocular SLAM. in *European Conference on Computer Vision* (Springer), 834–849.

Estrada, C., Neira, J., and Tardos, J. D. (2005). Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Trans. Robot.* 21, 588–596. doi:10.1109/TRO.2005.844673.

Fallon, M. F., Johannsson, H., Brookshire, J., Teller, S., and Leonard, J. J. (2012). Sensor fusion for flexible human-portable building-scale mapping. in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 4405–4412.

Fanto, P. L. (2012). Automatic Positioning and Design of a Variable Baseline Stereo Boom. (M.S. Thesis) Available at: http://scholar.lib.vt.edu/theses/available/etd-07252012-081926/.

Favaro, P., and Jin, H. (2003). A semi-direct approach to structure from motion. *Vis. Comput.* 19, 377–394.

Fenwick, J. W., Newman, P. M., and Leonard, J. J. (2002). Cooperative concurrent mapping and localization. in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on* (IEEE), 1810–1817.

Ferrer, G., Garrell, A., and Sanfeliu, A. (2013). Social-aware robot navigation in urban environments. in *Mobile Robots (ECMR), 2013 European Conference on* (IEEE), 331–336.

Fischer, C., Sukumar, P. T., and Hazas, M. (2013). Tutorial: Implementing a Pedestrian Tracker Using Inertial Sensors. *IEEE Pervasive Comput.* 12, 17–27. doi:10.1109/MPRV.2012.16.

Flint, A., Mei, C., Reid, I., and Murray, D. (2010). Growing semantically meaningful models for visual slam. in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (IEEE), 467–474.

Folkesson, J., and Christensen, H. (2004). Graphical SLAM - a self-correcting map. in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, 383–390 Vol.1. doi:10.1109/ROBOT.2004.1307180.

Fox, D., Burgard, W., Kruppa, H., and Thrun, S. (2000). A probabilistic approach to collaborative multi-robot localization. *Auton. Robots* 8, 325–344.

Fusiello, A., and Irsara, L. (2008). Quasi-euclidean uncalibrated epipolar rectification. in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, 1–4.

Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Mach. Vis. Appl.* 12, 16–22.

Gallup, D., Frahm, J.-M., Mordohai, P., and Pollefeys, M. (2008). Variable baseline/resolution stereo. in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8.

Galvez-López, D., and Tardos, J. D. (2012). Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. Robot.* 28, 1188–1197. doi:10.1109/TRO.2012.2197158.

Ganapathy, S. (1984). Decomposition of transformation matrices for robot vision. in *1984 IEEE International Conference on Robotics and Automation Proceedings*, 130–139. doi:10.1109/ROBOT.1984.1087163.

Gee, A. P., and Mayol-Cuevas, W. (2006). "Real-Time Model-Based SLAM Using Line Segments," in *Advances in Visual Computing* Lecture Notes in Computer Science (Springer Berlin Heidelberg), 354–363. doi:10.1007/11919629_37.

Grasa, O. G., Civera, J., and Montiel, J. M. M. (2011). EKF monocular SLAM with relocalization for laparoscopic sequences. in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 4816–4821.

Guerra, E., Munguia, R., Bolea, Y., and Grau, A. (2013). New validation algorithm for data association in SLAM. *ISA Trans.* doi:10.1016/j.isatra.2013.04.008.

Guerra, E., Munguia, R., and Grau, A. (2014). Monocular SLAM for Autonomous Robots with Enhanced Features Initialization. *Sensors* 14, 6317–6337. doi:10.3390/s140406317.

Guivant, J. E., and Nebot, E. M. (2001). Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Trans. Robot. Autom.* 17, 242–257. doi:10.1109/70.938382.

Hamilton, W. (1844). On Quaternions; or on a new System of Imaginaries in Algebra. *Z Hist. Lineární Algebry*.

Hargrave, P. J. (1989). A tutorial introduction to Kalman filtering. in *IEE Colloquium on Kalman Filters: Introduction, Applications and Future Developments*, 1/1-1/6.

Harris, C., and Stephens, M. (1988). A Combined Corner and Edge Detector. in *Proceedings of the 4th Alvey Vision Conference* (Alvey Vision Club), 147–151. doi:10.5244/C.2.23.

Harville, D. A. (1998). *Matrix Algebra From a Statistician's Perspective*. Springer-Verlag

Henrique Brito, J., Angst, R., Koser, K., and Pollefeys, M. (2013). Radial distortion self-calibration. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1368–1375.

Hermann, R., and Krener, A. J. (1977). Nonlinear controllability and observability. *IEEE Trans. Autom. Control* 22, 728–740.

Howard, A. (2004). Multi-robot mapping using manifold representations. in *2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, 4198–4203 Vol.4. doi:10.1109/ROBOT.2004.1308933.

Howard, A. (2008). Real-time stereo visual odometry for autonomous ground vehicles. in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 3946–3952.

Howell, S. B., Everett, M. E., Horch, E. P., Winters, J. G., Hirsch, L., Nusdeo, D., et al. (2016). Speckle Imaging Excludes Low-mass Companions Orbiting the Exoplanet Host Star TRAPPIST-1. *Astrophys. J. Lett.* 829, L2. doi:10.3847/2041-8205/829/1/L2.

Huang, G. P., Mourikis, A. I., and Roumeliotis, S. I. (2009). On the Complexity and Consistency of UKF-based SLAM. in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 4401–4408.

Ila, V., Porta, J. M., and Andrade-Cetto, J. (2010). Information-Based Compact Pose SLAM. *IEEE Trans. Robot.* 26, 78–93. doi:10.1109/TRO.2009.2034435.

Ives, H. E. (1930). Parallax panoramagrams made with a large diameter lens. *J. Opt. Soc. Am. 1917-1983* 20, 332.

Joerger, M., and Pervan, B. (2006). Autonomous ground vehicle navigation using integrated GPS and laser-scanner measurements. in *San Diego: Position, Location, and Navigation Symposium.*

Juan, L., and Gwun, O. (2009). A comparison of sift, pca-sift and surf. *Int. J. Image Process. IJIP* 3, 143–152.

Julier, S. J., and Uhlmann, J. K. (1997). A New Extension of the Kalman Filter to Nonlinear Systems. in *AeroSense: The 11th International Symposium on Aer ospace/Defence Sensing*, 182–193.

Julier, S. J., and Uhlmann, J. K. (2004). Unscented filtering and nonlinear estimation. *Proc. IEEE* 92, 401–422. doi:10.1109/JPROC.2003.823141.

Kaess, M., Ila, V., Roberts, R., and Dellaert, F. (2010). "The Bayes Tree: An Algorithmic Foundation for Probabilistic Robot Mapping," in *Algorithmic Foundations of Robotics IX* Springer Tracts in Advanced Robotics (Springer Berlin Heidelberg), 157–173. doi:10.1007/978-3-642-17452-0_10.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree. *Int. J. Robot. Res.* 31, 216–235. doi:10.1177/0278364911430419.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *J. Basic Eng.* 82, 35–45.

Kalman, R. E., and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *J. Basic Eng.* 83, 95–108.

Kang, S. B., Webb, J. A., Zitnick, C. L., and Kanade, T. (1995). A multibaseline stereo system with active illumination and real-time image acquisition. in *Computer Vision, 1995. Proceedings., Fifth International Conference on*, 88–93.

Kashif, M., Deserno, T. M., Haak, D., and Jonas, S. (2016). Feature description with SIFT, SURF, BRIEF, BRISK, or FREAK? A general question answered for bone age assessment. *Comput. Biol. Med.* 68, 67–75. doi:10.1016/j.compbiomed.2015.11.006.

Kerl, C., Sturm, J., and Cremers, D. (2013). Dense visual SLAM for RGB-D cameras. in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2100–2106. doi:10.1109/IROS.2013.6696650.

Klein, G., and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, 225–234.

Klein, G., and Murray, D. (2008). “Improving the agility of keyframe-based SLAM,” in *Computer Vision–ECCV 2008* (Springer), 802–815.

Kleiner, A., Dornhege, C., and Dali, S. (2007). Mapping disaster areas jointly: RFID-Coordinated SLAM by Hurnans and Robots. in *IEEE International Workshop on Safety, Security and Rescue Robotics, 2007. SSRR 2007*, 1–6. doi:10.1109/SSRR.2007.4381263.

Kotani, S., Kaneko, K., Shinoda, T., and Mori, H. (1998). Mobile robot navigation based on vision and DGPS information. in *1998 IEEE International Conference on Robotics and Automation, 1998. Proceedings*, 2524–2529 vol.3. doi:10.1109/ROBOT.1998.680721.

Krig, S. (2014). “Interest point detector and feature descriptor survey,” in *Computer Vision Metrics* (Springer), 217–282.

Kumar, S., Micheloni, C., Piciarelli, C., and Foresti, G. L. (2010). Stereo rectification of uncalibrated and heterogeneous images. *Pattern Recognit. Lett.* 31, 1445–1452. doi:10.1016/j.patrec.2010.03.019.

Kurazume, R., Nagata, S., and Hirose, S. (1994). Cooperative positioning with multiple robots. in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1250–1257 vol.2. doi:10.1109/ROBOT.1994.351315.

Leonard, J., and Newman, P. (2003). Consistent, convergent, and constant-time SLAM. in *International Joint Conference on Artificial Intelligence*, 1143–1150.

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). BRISK: Binary robust invariant scalable keypoints. in *Computer Vision (ICCV), 2011 IEEE International Conference on* (IEEE), 2548–2555.

Li, M., and Mourikis, A. I. (2013). High-precision, consistent EKF-based visual–inertial odometry. *Int. J. Robot. Res.* 32, 690–711. doi:10.1177/0278364913481251.

Li, Y., Li, S., Song, Q., Liu, H., and Meng, M. Q. H. (2014). Fast and Robust Data Association Using Posterior Based Approximate Joint Compatibility Test. *IEEE Trans. Ind. Inform.* 10, 331–339. doi:10.1109/TII.2013.2271506.

Li, Y., and Olson, E. B. (2012). IPJC: The Incremental Posterior Joint Compatibility test for fast feature cloud matching. in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3467–3474. doi:10.1109/IROS.2012.6385470.

Liang, C. K., Chang, L. W., and Chen, H. H. (2008). Analysis and Compensation of Rolling Shutter Effect. *IEEE Trans. Image Process.* 17, 1323–1330. doi:10.1109/TIP.2008.925384.

Loop, C., and Zhang, Z. (1999). Computing rectifying homographies for stereo vision. in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, 131 Vol. 1. doi:10.1109/CVPR.1999.786928.

Lovegrove, S., and Davison, A. J. (2010). Real-time spherical mosaicing using whole image alignment. in *European Conference on Computer Vision* (Springer), 73–86.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 60, 91–110.

Lukierski, R., Leutenegger, S., and Davison, A. J. (2015). Rapid free-space mapping from a single omnidirectional camera. in *Mobile Robots (ECMR), 2015 European Conference on* (IEEE), 1–8.

Luo, J., and Konofagou, E. E. (2010). A fast normalized cross-correlation calculation method for motion estimation. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* 57, 1347–1357. doi:10.1109/TUFFC.2010.1554.

Lupton, T., and Sukkarieh, S. (2012). Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions. *IEEE Trans. Robot.* 28, 61–76. doi:10.1109/TRO.2011.2170332.

Martin, J., and Crowley, J. L. (1995). Comparison of correlation techniques. in *International Conference on Intelligent Autonmous Systems, Karlsruhe (Germany)*, 86–93.

Marzorati, D., Matteucci, M., Migliore, D., and Sorrenti, D. G. (2008). Monocular SLAM with Inverse Scaling Parametrization. in *BMVC*, 1–10.

Mazo, M., Speranzon, A., Johansson, K. H., and Hu, X. (2004). Multi-robot tracking of a moving object using directional sensors. in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on* (IEEE), 1103–1108.

McElhoe, B. A. (1966). An Assessment of the Navigation and Course Corrections for a Manned Flyby of Mars or Venus. *IEEE Trans. Aerosp. Electron. Syst.* AES-2, 613–623. doi:10.1109/TAES.1966.4501892.

Mei, C., Sibley, G., Cummins, M., Newman, P., and Reid, I. (2009). A constant time efficient stereo SLAM system. in *Proceedings of the British Machine Vision Conference (BMVC).*

Metropolis, N., and Ulam, S. (1949). The Monte Carlo Method. *J. Am. Stat. Assoc.* 44, 335–341. doi:10.2307/2280232.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. in *Proceedings of the National conference on Artificial Intelligence*, 593–598.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. in *International Joint Conference on Artificial Intelligence*, 1151–1156.

Montiel, J. M. M., and Montano, L. (1998). Efficient validation of matching hypotheses using mahalanobis distance. *Eng. Appl. Artif. Intell.* 11, 439–448.

Munguia, R., and Grau, A. (2007a). Camera localization and mapping using delayed feature initialization and inverse depth parametrization. in *IEEE Conference on Emerging Technologies and Factory Automation, 2007. ETFA*, 981–988. doi:10.1109/EFTA.2007.4416890.

Munguia, R., and Grau, A. (2007b). Monocular SLAM for Visual Odometry. in *IEEE International Symposium on Intelligent Signal Processing, 2007. WISP 2007*, 1–6. doi:10.1109/WISP.2007.4447564.

Munguia, R., and Grau, A. (2009). Closing Loops With a Virtual Sensor Based on Monocular SLAM. *IEEE Trans. Instrum. Meas.* 58, 2377–2384. doi:10.1109/TIM.2009.2016377.

Munguía, R., and Grau, A. (2012). Monocular SLAM for visual odometry: A full approach to the delayed inverse-depth feature initialization method. *Math. Probl. Eng.* 2012. doi:10.1155/2012/676385.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: a Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* 31, 1147–1163. doi:10.1109/TRO.2015.2463671.

Neira, J., and Tardós, J. D. (2001). Data association in stochastic mapping using the joint compatibility test. *Robot. Autom. IEEE Trans. On* 17, 890–897.

Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. in *2011 IEEE International Conference on Computer Vision (ICCV)*, 2320–2327. doi:10.1109/ICCV.2011.6126513.

Paskin, M. A. (2003). Thin Junction Tree Filters for Simultaneous Localization and Mapping. in *IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15, 2003* (Morgan Kaufmann), 1157–1166.

Paz, L. M., Tardos, J. D., and Neira, J. (2008). Divide and Conquer: EKF SLAM in $O(n)$. *IEEE Trans. Robot.* 24, 1107–1120. doi:10.1109/TRO.2008.2004639.

Piniés, P., Lupton, T., Sukkarieh, S., and Tardós, J. D. (2007). Inertial aiding of inverse depth SLAM using a monocular camera. in *Robotics and Automation, 2007 IEEE International Conference on*, 2797–2802.

Pomerleau, F., Breitenmoser, A., Liu, M., Colas, F., and Siegwart, R. (2012). Noise characterization of depth sensors for surface inspections. in *2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, 16–21. doi:10.1109/CARPI.2012.6473358.

Rizzini, D. L., and Caselli, S. (2011). A multi-hypothesis constraint network optimizer for maximum likelihood mapping. in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, 2485–2490. doi:10.1109/ICRA.2011.5979946.

Rosten, E., Reitmayr, G., and Drummond, T. (2005). Real-time video annotations for augmented reality. in *International Symposium on Visual Computing* (Springer), 294–302.

Roumeliotis, S. I., and Bekey, G. A. (2002). Distributed multirobot localization. *IEEE Trans. Robot. Autom.* 18, 781–795. doi:10.1109/TRA.2002.803461.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. in *2011 International Conference on Computer Vision*, 2564–2571. doi:10.1109/ICCV.2011.6126544.

Sanfeliu, A., Andrade-Cetto, J., Barbosa, M., Bowden, R., Capitán, J., Corominas, A., et al. (2010). Decentralized Sensor Fusion for Ubiquitous Networking Robotics in Urban Areas. *Sensors* 10, 2274–2314. doi:10.3390/s100302274.

Scharstein, D., and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vis.* 47, 7–42.

Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. in *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on* (IEEE), 519–528.

Shen, X., Frazzoli, E., Rus, D., and Ang, M. H. (2016). Fast Joint Compatibility Branch and Bound for feature cloud matching. in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1757–1764. doi:10.1109/IROS.2016.7759281.

Shi, X., Zhao, C., and Chen, T. (2014). Data association technology based on multi algorithm matching for SLAM. in *2014 11th World Congress on Intelligent Control and Automation (WCICA)*, 934–939. doi:10.1109/WCICA.2014.7052841.

Slotine, J.-J. E., and Li, W. (1991). *Applied nonlinear control*. Englewood Cliffs, N.J: Prentice Hall.

Smith, R. C., and Cheeseman, P. (1986). On the Representation and Estimation of Spatial Uncertainty. *Int. J. Robot. Res.* 5, 56–68. doi:10.1177/027836498600500404.

Smith, R., Self, M., and Cheeseman, P. (1987). Estimating uncertain spatial relationships in robotics. in *1987 IEEE International Conference on Robotics and Automation. Proceedings*, 850–850. doi:10.1109/ROBOT.1987.1087846.

Smith, S. M., and Brady, J. M. (1995). SUSAN - A New Approach to Low Level Image Processing. *Int. J. Comput. Vis.* 23, 45–78.

Sola, J., Monin, A., Devy, M., and Lemaire, T. (2005). Undelayed initialization in bearing only SLAM. in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2499–2504.

Sola, J., Vidal-Calleja, T., Civera, J., and Montiel, J. M. M. (2012). Impact of landmark parametrization on monocular EKF-SLAM with points and lines. *Int. J. Comput. Vis.* 97, 339–368.

Spletzer, J., Das, A. K., Fierro, R., Taylor, C. J., Kumar, V., and Ostrowski, J. P. (2001). Cooperative localization and control for multi-robot manipulation. in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium*, 631–636 vol.2. doi:10.1109/IROS.2001.976240.

Strasdat, H., Montiel, J. M. M., and Davison, A. J. (2010). Real-time monocular SLAM: Why filter? in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2657–2664. doi:10.1109/ROBOT.2010.5509636.

Suzuki, T., Amano, Y., and Hashizume, T. (2011). Development of a SIFT based monocular EKF-SLAM algorithm for a small unmanned aerial vehicle. in *2011 Proceedings of SICE Annual Conference (SICE)*, 1656–1659.

Swerling, P (1959). First-Order Error Propagation in a Stagewise Smoothing Procedure for Satellite Observations. Available at: http://www.rand.org/pubs/research_memoranda/RM2329.html.

Tardif, J.-P., Pavlidis, Y., and Daniilidis, K. (2008). Monocular visual odometry in urban environments using an omnidirectional camera. in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2008. IROS 2008*, 2531–2538. doi:10.1109/IROS.2008.4651205.

Tardós, J. D., Neira, J., Newman, P. M., and Leonard, J. J. (2002). Robust mapping and localization in indoor environments using sonar data. *Int J Robot. Res.* 21, 311–330.

Thrun, S., and Liu, Y. (2003). Multi-robot slam with sparse extended information filters. in *Proceedings of the 11th International Symposium of Robotics Research* (Springer).

Thrun, S., Martin, C., Liu, Y., Hahnel, D., Emery-Montemerlo, R., Chakrabarti, D., et al. (2004). A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *IEEE Trans. Robot. Autom.* 20, 433–443. doi:10.1109/TRA.2004.825520.

Törtei Tertei, D., Piat, J., and Devy, M. (2016). FPGA design of EKF block accelerator for 3D visual SLAM. *Comput. Electr. Eng.* 55, 123–137. doi:10.1016/j.compeleceng.2016.05.003.

Walls, J. M., Cunningham, A. G., and Eustice, R. M. (2015). Cooperative localization by factor composition over a faulty low-bandwidth communication channel. in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 401–408. doi:10.1109/ICRA.2015.7139030.

Walls, J. M., and Eustice, R. M. (2013). An Exact Decentralized Cooperative Navigation Algorithm for Acoustically Networked Underwater Vehicles with Robustness to Faulty Communication: Theory and Experiment. in *Robotics: Science and Systems.*

Wang, Z., and Dissanayake, G. (2010). Efficient Monocular SLAM using sparse information filters. in *Information and Automation for Sustainability (ICIAFs), 2010 5th International Conference on*, 311–316.

Weikersdorfer, D., Adrian, D. B., Cremers, D., and Conradt, J. (2014). Event-based 3D SLAM with a depth-augmented dynamic vision sensor. in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 359–364. doi:10.1109/ICRA.2014.6906882.

Williams, B., Cummins, M., Neira, J., Newman, P., Reid, I., and Tardós, J. (2009). A comparison of loop closing techniques in monocular SLAM. *Robot. Auton. Syst.* 57, 1188–1197. doi:10.1016/j.robot.2009.06.010.

Williams, B., Klein, G., and Reid, I. (2007). Real-Time SLAM Relocalisation. in *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, 1–8. doi:10.1109/ICCV.2007.4409115.

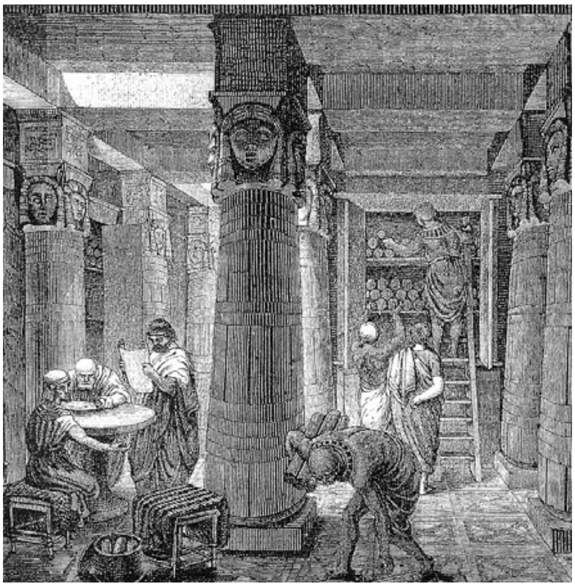
Williams, S. B., Dissanayake, G., and Durrant-Whyte, H. (2002). An efficient approach to the simultaneous localisation and mapping problem. in *IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02*, 406–411 vol.1. doi:10.1109/ROBOT.2002.1013394.

Zeller, N., Quint, F., and Stilla, U. (2016). Depth estimation and camera calibration of a focused plenoptic camera for visual odometry. *ISPRS J. Photogramm. Remote Sens.* 118, 83–100. doi:10.1016/j.isprsjprs.2016.04.010.

Zhang, S., He, B., Feng, X., and Yuan, G. (2012). ICM: An efficient data association for SLAM in stochastic mapping. in *2012 12th International Conference on Control Automation Robotics Vision (ICARCV)*, 1042–1047. doi:10.1109/ICARCV.2012.6485301.

Zhou, X. S., and Roumeliotis, S. I. (2006). Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case. in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1785–1792. doi:10.1109/IROS.2006.282219.

Part V
Annexes



The Great Library of Alexandria, impression by Otto Von Corven, around the 19th century.

V.A List of Abbreviations

BA	Bundle Adjustment
CCD	Charge-Coupled Device
CLSF	Constrained Local Submap Filter
CML	Concurrent Mapping and Localization Problem (a.k.a. SLAM)
CMOS	Complementary Metal-Oxide Semiconductor
C-SAM	Collaborative Smoothing and Mapping

CV	Computer Vision
DA	Data Association
DDF	Distributed Data Framework
DGPS	Differential Global Positioning System
DoF	Degrees of Freedom
DI-D	Delayed Inverse-Depth
DSLR	Digital Single-Lens Reflex (camera)
DTAM	Dense Tracking and Mapping
EIF	Extended Information Filter
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test (feature detector and descriptor)
fps	frames-per-second
GLONASS	Globalnaya Navigatsionnaya Sputnikovaya Sistema
GNSS	Global Navigation Satellite System
GPGPU	General-Purpose computing on Graphics Processing Units
GPS	Global Positioning System
GPU	Graphical Processing Unit
GRV	Gaussian Random Variable
GSF	Gaussian Sum Filter
HOHCT	Highest Order Hypotheses Compatibility Test
HRI	Human-Robot Interaction
ICNN	Individual Compatibility Nearest Neighbour
ICP	Iterative Closest Point
I-D	Inverse-Depth (feature parametrization)
IDP	Inverse Depth Points
IEEE1394	High Performance Serial Bus specification (used in cameras)
IEKF	Iterated Extended Kalman Filter
IF	Information Filter
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infrared (light spectrum)

JC	Joint Compatibility
JCBB	Joint Compatibility Branch & Bound
KF	Kalman Filter
Laser	Light Amplification by Simulated Emission of Radiation
LIDAR	Laser Imaging Detection And Ranging
LRF	Laser Range Finders
MCI	Muscle-computer Interface
MEMS	Microeletromechanical systems
NCC	Normalized Cross-Correlation
NN	Nearest Neighbour
P4P	Perspective of 4 Points
pdf/s	Probability Distribution Function/s
PF	Particle Filter
PnP	Perspective of *n* Points
PTAM	Parallel Tracking and Mapping
Radar	Radio Detection And Ranging
RGB-D	Red Green Blue Depth
ROI	Region of Interest
ROS	Robot Operating System
RPF	Repeated Pulse Frequency
SAD	Sum of Absolute Differences
SAM	Smoothing and Mapping
SAR	Search and Rescue
SfM	Structure from Motion
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization And Mapping
SMC	Sequential Monte-Carlo
SMD	Squared Mahalanobis Distance
SSD	Sum of Squared Differences
SURF	Speeded-Up Robust Features
TJTF	Thin Junction Tree Filter
ToF	Time of Flight (camera)

UKF	Unscented Kalman Filter
USAR	Urban Search and Rescue
UV	Ultraviolet (light spectrum)
w.r.t.	*with respect to*
ZNCC	Zero-Mean Normalized Cross-Correlation
ZSAD	Zero-Mean Sum of Absolute Differences

V.B Orientation notation and conversion

Compute a quaternion from a directional vector:

$$\mathbf{q}(\omega)=\begin{bmatrix}\cos\left(\dfrac{\|\omega\|}{2}\right)\\ \sin\left(\dfrac{\|\omega\|}{2}\right)\\ \left[\dfrac{\omega}{\|\omega\|}\right]\end{bmatrix} \tag{V.1}$$

Compute a quaternion from a rotation matrix:

$$q(R)=\begin{bmatrix} q_1\\ \dfrac{R(3,2)-R(2,3)}{4q_1}\\ \dfrac{R(1,3)-R(3,1)}{4q_1}\\ \dfrac{R(2,1)-R(1,2)}{4q_1}\end{bmatrix},\ where\ q_1=\sqrt{1+R(1,1)+R(2,2)+R(3,3)} \tag{V.2}$$

Compute a rotation matrix form a quaternion:

$$R=\begin{bmatrix}\left(q_1^2+q_2^2-q_3^2-q_4^2\right) & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_1q_0+q_2q_3\right)\\ 2\left(q_1q_2+q_0q_3\right) & \left(q_1^2-q_2^2+q_3^2-q_4^2\right) & 2\left(q_2q_3-q_0q_1\right)\\ 2\left(q_1q_0-q_2q_3\right) & 2\left(q_2q_3+q_0q_1\right) & \left(q_1^2+q_2^2-q_3^2+q_4^2\right)\end{bmatrix} \tag{V.3}$$