Universitat de Girona

LARGE-SCALE SURFACE REGISTRATION

ELISABET BATLLE SUBIRÓS

ISBN: 978-84-692-1409-1
Dipòsit legal: GI-132-2009

Universitat de Girona

Departament d'Arquitectura i Tecnologia de Computadors

PhD Thesis

Large-scale surface registration

Thesis presented by:
Elisabet Batlle Subirós

to obtain the degree of:
PhD in Computer Engineering

Supervisor:
Dr. Joaquim Salvi

Girona, November 2008

Agraïments

En primer lloc m'agradaria agrair especialment a la persona que ha fet possible aquesta tesi, el Dr. Quim Salvi, per la confiança i el suport que m'ha donat aquests últims anys. A en Quim li he d'agrair l'oportunitat que m'ha donat de formar-me com a doctora, i el fet d'introduir-me en el món de la visió per computador i ajudar-me i aconsellar-me a cada pas de la tesi, però sobretot el seu bon humor i optimisme, sempre present.

Je voudrais remercier spécialement le Dr. Mustapha Mouaddib, Dr. Pascal Vasseur, Dr. Claude Pégard, Dr. Cédric Demonceaux, Sang Ly et en général tous les membres du Centre de Robotique, d'Electronique et d'Automatique de l'Universite-Jules Verne, Amiens, pour l'occasion qu'ils m'ont donnée de m'unir à leur groupe pendant quelque mois. Merci pour votre aide et pour les bons moments que j'ai passés pendant mon séjour en France.

També voldria agrair els consells i l'ajuda de'n Pere Ridao durant la meva primera etapa com a estudiant de doctorat dins del grup de robòtica submarina. En la mateixa línia, moltes gràcies a tota la gent del laboratori de robòtica submarina, Marc, Andres, David, Emili, Narcís i companyia.

Un agraïment molt especial a tots els membres del departament d'EIA i del grup VICOROB, professors, secretàries, PAS, projectistes, etc. Em sento molt afortunada d'haver pogut formar part d'aquest equip. Gràcies a tots, Xevi Cufí, Josep Forest, Rafa Garcia, Joan Martí, Xavier Lladó, Jordi Freixenet, Lluis Pacheco, Xavier Muñoz, Robert, Bianca, Pous, Silvana, Peplluís, Montse, Anna, Toni i companyia, per la vostra ajuda, alegria i el vostre sentit de l'humor. A tots us he d'agrair aquests darrers anys que m'han format no tan sols com a doctora i professora sino també com a persona. Gràcies als companys del laboratori de visió, Arnau, David i molt especialment a l'Anna. Als meus companys de laboratori: Carles, Tudor, Ela, Ricard, Arman, Pio, Josep, Nuno, Olivier i François, moltíssimes gràcies per l'ajuda i per les bones estones passades. A en Ricard gràcies pel suport tècnic a distància durant aquest últim any a Chicago. Vull agrair molt especialment a en Carles Matabosc no tant sols la seva impagable ajuda durant tota aquesta tesi sinó també la amistat que m'ha demostrat dia a dia.

Als meus amics gràcies pels ànims i el suport durant aquesta etapa. Gràcies a tota la meva colla de Girona i molt especialment a les "nenes" Ana, Cris i Uau. Gràcies també a l'Anna, l'Imma i en general a tota la colla "Esquí de muntanya", per les tardes de bici a Sant Miquel i les matinals d'esquí de muntanya als Pirineus que m'han permès evadir-me i passar unes estones fantàstiques en les èpoques de més estrès. Gràcies també a la gent de Chicago pels ànims en aquesta ultima etapa de la tesi. Agraïments també a la gent de la Autònoma, molt especialment a en Ramon, no només per la seva ajuda incondicional sinó per haver confiat en mi des del començament i haver-me animat en els moments més importants.

A en Jordi, moltes gràcies per fer-me riure i estar sempre al meu costat. Gràcies per la paciència que has tingut, per cuidar-me, per ajudar-me i confiar en mi dia rere dia. Gràcies per aquests ultims anys.

Per acabar, m'agradaria agrair a tota la meva família el seu suport incondicional. Gràcies especialment a la meva mare Anna, i a la meva germana Nuri per la seva paciència infinita. Als meus avis Pere i Maria i a la meva avia Roser, per animar-me sempre. Agraïments també a la Francina, la Geni i en Benoît. Finalment, m'agradaria agrair i dedicar aquesta tesi al meu pare, Joan Batlle, perque sense les seves continues mostres de suport i els seus "tocs d'humor" en els moments crucials tot això no hauria estat possible.

Abstract

In the recent years 3D reconstruction has become a very important research topic in computer vision community, involving a wide range of applications such quality control in industry, modeling of large scenarios in robot navigation or reconstruction of buildings, statues and cultural heritage among others. All these tasks rely on surface mapping processes, which imply several steps, including image acquisition, 3D registration and view integration, to obtain the final model of the object/scene. An unsolved problem when registering large objects or scenes resides in the propagation of the error during the registration process, leading to inaccuracies in the alignment of the partial views. This thesis presents a discussion of the 3D registration techniques and proposes a new approach to deal with the misalignment problems, reducing the accumulated error by means of a multiview registration strategy.

The first part of this work presents an accurate analysis of the most relevant 3D registration techniques, including initial pose estimation, pairwise registration and multiview registration strategies. A new classification has been proposed, based on both the applications and the approach of the methods that have been discussed. This analysis is the basis of the main contribution of this thesis, since it has allowed to identify the strong and weak points of the existing methods in order to propose a novel contribution that improves the performance of all the previous ones.

The second and main contribution of this thesis is the proposal of a new 3D multiview registration strategy. The proposed approach detects revisited regions obtaining cycles of views that are used to reduce the inaccuracies that may exist in the final model due to error propagation. The method takes advantage of both global and local information of the registration process, using graph theory techniques in order correlate multiple views and minimize the propagated error by registering the views in an optimal way.

The proposed method has been tested using both synthetic and real data, in order to show and study its behavior and demonstrate its reliability. In addition, some of the most relevant 3D multiview registration methods have also been tested under the same conditions. From the obtained results it can be extracted that the techniques based on graphs provide a good solution to the problem of the propagated error, improving considerably the results obtained by metaview approaches and being more efficient than simultaneous minimization in large scenes. Besides, detecting cycles does not only help improving the process in terms of error propagation, but the use of information related to visited scenes improve significantly the global results.

Resum

Durant els últims anys, la reconstrucció 3D s'ha convertit en un camp de recerca d'alta rellevància dins de la visió per computador, amb aplicacions com el control de qualitat, la modelització d'escenes de grans dimensions en la navegació robòtica o la reconstrucció d'edificis, estàtues i llegat cultural entre d'altres. Totes aquestes tasques depenen de processos de mapeig de superfícies, incloent diferents passos com l'adquisició d'imatges, el registre 3D i la integració de les vistes, per tal d'obtenir el model final de l'objecte/escena. Un problema encara no resolt es la propagació de l'error en el registre d'objectes de grans dimensions, que pot comportar a desalineaments entre les vistes parcials. Aquesta tesis presenta un estudi de les tècniques de registre 3D i proposa un nou mètode per tal de tractar amb els problemes de desalineament, reduint l'error acumulat mitjançant el registre de múltiples vistes.

La primera part d'aquest treball presenta una anàlisi acurada de les tècniques de registre 3D mes rellevants, incloent tècniques d'estimació de la posició inicial, registre pairwise i registre entre múltiples vistes. S'ha proposat una nova classificació de les tècniques, depenent de les seves aplicacions i de l'estratègia utilitzada. Aquesta anàlisi i classificació representa la base de la principal aportació de la tesi ja que ha permès identificar els punts forts i febles de cada mètode per tal de proposar una nova contribució que millora el rendiment de les tècniques ja existents.

La contribució mes important d'aquesta tesi és la proposta d'un nou mètode de registre 3D utilitzant múltiples vistes. El mètode proposat detecta regions ja visitades prèviament, obtenint cicles de vistes que s'utilitzen per tal de reduir els desalineaments en el model final deguts principalment a la propagació del error durant el procés de registre. Aquest mètode utilitza tant informació global com local, correlacionant les vistes mitjançant tècniques de grafs que permeten minimitzar l'error propagat i registrar les vistes de forma òptima.

El mètode proposat ha estat provat utilitzant dades sintètiques i reals, per tal de mostrar i analitzar el seu comportament i demostrar la seva eficàcia. Per comparar el rendiment, també s'han provat en les mateixes condicions alguns dels mètodes de registre 3D mes rellevants. A partir dels resultats es pot extreure que les tècniques basades en grafs presenten bones solucions al problema de la propagació de l'error, millorant considerablement els resultats obtinguts per les tècniques de metaview i mostrant una eficiència mes elevada que la minimització simultània en escenes de grans dimensions. Cal afegir que el fet de detectar cicles utilitzant informació de àrees prèviament visitades no nomes contribueix a millorar el procés en termes de propagació de l'error sinó que millora significativament els resultats globals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the origin of this work and how it is related with the work currently under development in the Computer Vision and Robotics Group of the University of Girona. An overview of the 3D registration problem is described with the aim of focusing the objectives of this thesis and the potential applications of its contributions. In addition, an outline of the thesis is presented at the end of the chapter.

1.1 Research background

There has been considerable progress on computer vision in the last decade. We are at the edge of understanding how the eye works and how it perceives what it is seen, as a previous step for translating such information to computers in the most accurate way. However, there is no general solution to recover the geometry of space and objects that works in any case and environment condition. A long way has been done since Aristotle, who believed the eye emitted rays that reached out and brought back the visible information to the viewer. Nowadays, although enormous progress has been made towards simulating the real world using computers, we are still far away from obtaining an efficient system dealing with the understanding of the environment and reconstructing the real world.

From autonomous navigation to quality control applications, scene reconstruction appears to be one of the most relevant research fields in computer vision. Since 1992, our research group (VICOROB) has been working in several projects related to image process-

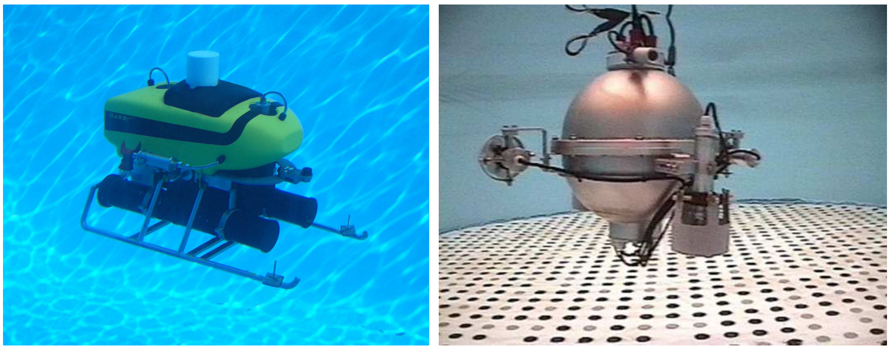


Figure 1.1: Autonomous Underwater vehicle prototypes GARBI (left) and URIS (right).

ing and computer vision with applications to active surveillance [66], medical image [65], 3D reconstruction [54] and robotics [7] among others.

In the robotics field, one of our main research lines has been related to underwater robotics navigation. The group has developed its own autonomous vehicles, including Uris and Garbi, which have been used as a valuable testing platform to perform research in computer vision (see Figure 1.1). In this context, several approaches for 2D and 3D scene reconstruction have been presented as a key to safety navigation. In [29] Garcia proposed the construction of 2D mosaics (visual maps of the ocean floor), with the aim of recovering the pose and motion of the vehicle. Recently, the research of the group has been focused in Simultaneous Localization and Mapping (SLAM) techniques using predictive filters with the aim of estimating the robot pose and building a 3D map of the scene simultaneously. In this context, some relevant contributions has been published [80] [70]. More specifically a new technique to reconstruct large 3D scenes from a sequence of video images by combining 3D computer vision and Bayesian filtering was presented. The approach performs the alignment of a sequence of 3D partial reconstructions of the seafloor thanks to the re-observations of passive landmarks by means of a Kalman filter-based SLAM approach.

Early works related with the present doctoral thesis were associated with these robotic platforms. The developed work concerned the simulation of Multi-AUV (Autonomous Undewater Vehicles) in underwater environments [74] [75] [76], and was a fundamental step for understanding the associated problems with navigation and environment perception [6]. In this context 3D scene understanding and reconstruction techniques were proved to be fundamental for outdoor scenes and underwater navigation applications.

Aspects such as sensoring, camera modelling and calibration, structured light projection and 3D registration can be considered the background of the present work and future research proposals [25]. Related to theses subjects, several thesis have been presented in the group during the last years. Armangué presented a deep study about stereovision systems, with the aim of recovering 3D geometric information of the environment, including topics such as camera calibration, epipolar geometry and egomotion estimation among others [2]. Two years latter Pages proposed a new coded pattern projection technique and its application in assisted visual servoing focused to robot positioning and simplifying the problem of dealing with featureless environments (see Figure 1.2) [67].

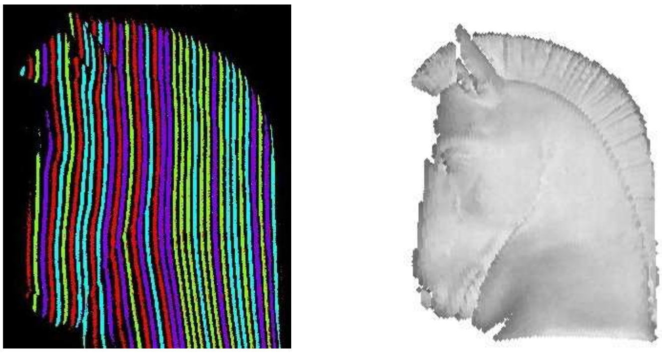
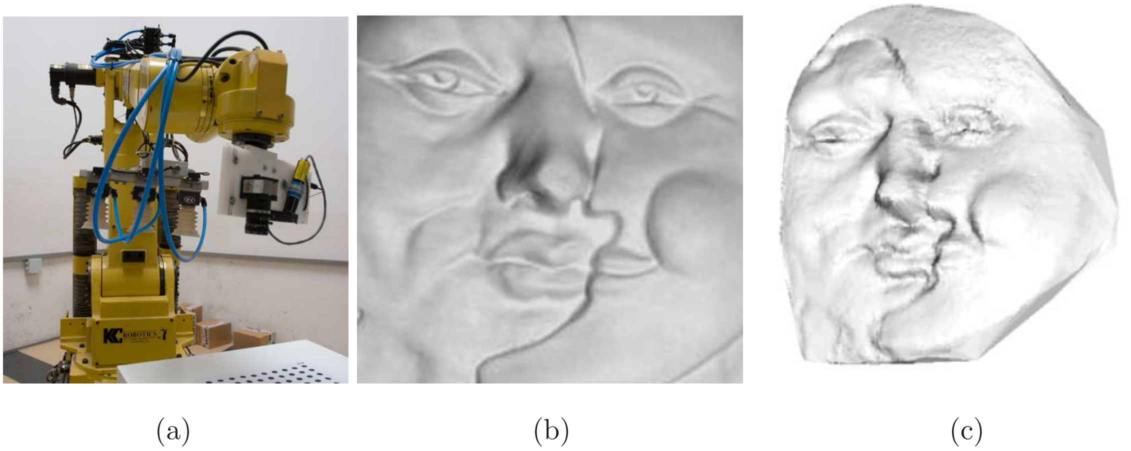


Figure 1.2: Example of coded structured light for 3D surface reconstruction.

Matabosch focused his work in data acquisition and 3D registration of small objects [53]. One of the main contributions of this work resides on the design of a 3D hand-held multi-slit laser scanner to acquire 3D views without any motion restriction. In addition, the author proposed a new approach for 3D registration of small objects. Widening the line of research, the present thesis faces the problem of 3D registration of large-scale objects, without loosing small details and features and obtaining a high resolution model.

1.2 Motivations and objectives

The present thesis has been mainly developed in the context of three research projects funded by the Spanish government: a) DAFSEC-VIC (TIC2003-08106-C02-02), focused on active vision systems for surveillance tasks in large scenarios such as airports and train stations; b) AQUAVISION (DPI2007-66796-C03-02) that deals with the development of 3D computer vision algorithms for underwater cartography; and c) AIRSUB (DPI2005-09001-C03-01), that faced visual inspection of hard accessible scenarios such as harbors or dams with the aim of detecting problems such as fissures and holes.



(a) (b) (c)

Figure 1.3: Example of 3D reconstruction developed by Matabosch. (a) Industrial manipulator used in the experiments. (b) Real plaster object. (c) 3D reconstruction.

These projects addressed to industrial and research developments need of 3D computer vision and 3D image understanding. Historically, computer vision and image processing have proved to be fundamental to improve visual inspection in industrial quality control. Early applications concerned the assessment of manufactured parts correctness with respect to a known model by 2D imaging, using wide range of techniques starting from image morphology up to complex image filters. Nevertheless, depending on the complexity of the objects, it is still difficult to introduce a reliable quality control in terms of 3D information. Nowadays, recent contributions in 3D registration permit quality assessment tasks based on the comparison of a 3D scanned manufactured part and its corresponding 3D model. These techniques are not only focused on the detection of object imperfections (such as surface bumps, cracks and fissures) in tiny or medium size objects, but also in large scenarios such as the inspection of automotive and avionics manufactured parts, submerged parts of harbors and dams, and even the complete acquisition of ancient remains and large sculptures.

Focusing on what is the main subject of this thesis, that is the 3D acquisition of large-scale objects, several chained processes are involved from previous data acquisition to a ultimate view integration procedure. Note that the complete acquisition of large-scale objects requires to consider the imaging of a sequence of partial views while the object or the acquisition system is moving. Hence, surface registration is an intermediate step consisting in transforming all the partial views of that object/surface to a common reference. Several techniques have been proposed by different communities (computer vision, robotics, photogrammetry), differing considerably depending on the application. Computer vision and Photogrammetry fields of research usually focused on obtaining accurate reconstruction of the 3D objects for applications such as quality control and statue digitalization, among others. Here it is not so important to work in real time but to obtain an accurate result. The main problem of these approaches is the propagation error present in the registration process of large surfaces. This problem has also been faced in the robotics community, where scene reconstruction plays an important role in safe autonomous navigation. Several techniques have been presented, focused mainly in the localization of the vehicle and the mapping the environment simultaneously (SLAM), providing reliable solutions to the problem of error propagation. Here, the resolution of the map is not as important as the accurate localization of the obstacles and the identification of the correct path.

Although the efforts of the scientific community to provide reliable methods to reconstruct 3D objects and surfaces, the problem of remove (reduce) the accumulated error in the registration process has not been solved. This problem worsens when the size of the object and the resolution desired increase. Therefore, a reliable approach to deal with this problem presents a real challenge that has motivate the development of this work.

The purpose of this thesis is to develop a new approach with the aim of solving the issues of the 3D registration of large-scale objects. Two main problems need to be solved: (1) The registration of the 3D partial views and (2) the minimization of the accumulated error during the alignment process. After analyzing the most important registration techniques available in the literature, a new algorithm based on multi-view graph-based techniques is proposed. The algorithm takes advantage of the information provided when a region is revisited and uses this information to correct the movement between the registered views, reducing the misalignment in the final model.

1.3 Thesis outline

This thesis is structured in 6 chapters, including this introduction detailed in Chapter 1.

In chapter 2, a survey of surface registration for 6DOF (Degrees of Freedom) robot/camera pose estimation is presented. The study is divided in three main sections. The first is related with methods for obtaining an initial coarse pose estimation, including a detailed description of different feature extraction and matching techniques. The second describes fine registration methods that refine the alignment departing from a close solution given by the coarse method. Finally, the third and main part of the survey concerns the presentation of the most representative error minimization techniques, including graph-based techniques, simultaneous minimization and SLAM approaches. A detailed list of works and several bibliographical references are briefly explained and exhaustively classified according to their applications and the methodology used.

Chapter 3 describes some fundamentals on graph theory with the aim of presenting the basics that are used in the succeeding chapters. The basic definitions of graphs are provided together with the description of several algorithms focused on finding paths and cycles that are important in 3D registration. This chapter provides a theoretical framework for the correct understanding of the new contributions described in this thesis.

Chapter 4 describes a new technique for 3D registration and error minimization of large-scale objects. The method detects revisited regions obtaining cycles of views that are used to reduce the misalignments in the final model.

Validation and testing experiments are presented in chapter 5. Results of the new proposed approach using both synthetic and real data are analyzed. With the aim of proving the rightness of the new method, a subset of the most representative techniques of the survey has been implemented and executed with the same data set. The chapter includes a comparison of the new approach with the programmed techniques and ends with a discussion of the obtained results.

Finally, chapter 6 presents the conclusions of this work, the contributions published in journals and conferences and the further work that still remains to be done.

Chapter 2

State of the art

In recent years, 6 Degrees Of Freedom (DOF) Pose Estimation and 3D Registration and Mapping are becoming more important not only in the robotics community for applications such as robot navigation but also in computer vision for the registration of large surfaces such as buildings and statues. In both situations, the robot/camera position and orientation must be estimated in order to be used for further alignment of the 3D map/surface. Although the techniques differ slightly depending on the application, both communities tend to solve similar problems by means of different approaches. In this chapter a survey of pose estimation and 3D registration methods is presented. The surveyed techniques have been compared pointing out their pros and cons and their potential applications.

2.1 Introduction

6DOF Localization and Mapping is a growing research field among scientists involved in a wide range of 3D applications covering from registration of large surfaces in photogrammetry to mapping of complex scenarios in robot navigation. The process of surface mapping implies several steps, including image acquisition, 3D registration and view integration, to obtain the final model of the object/scene. This work will be centered mainly on registration techniques, focusing in error minimization methods. Registration is the process of determining the Euclidian motion between range images. That is, given a set of partial views of an object, the main goal is to represent them in a common coordinate system (see Figure 2.1). Registration process implies several steps, from initial pose estimation to multiview error minimization. In this process, robot/camera pose estimation is crucial in order to obtain accurate results in the final model.

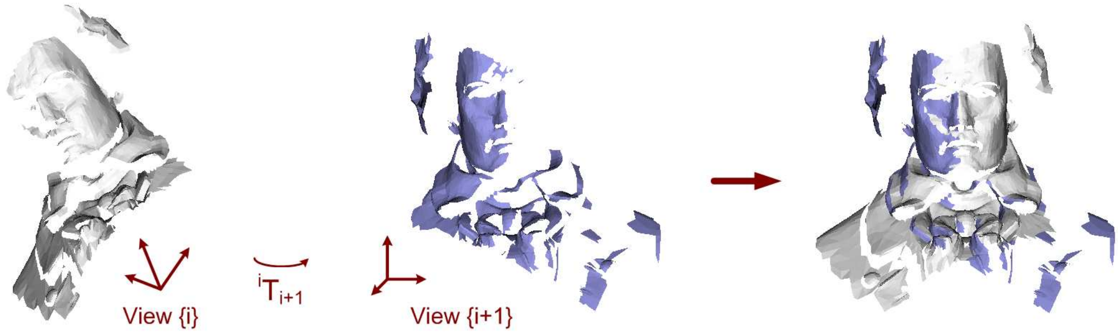


Figure 2.1: Pairwise registration process.

Until now, robot navigation has been focused on 2D mapping in flat terrains and usually restricted to indoor structured scenarios [87]. Recently, the need to explore complex and unstructured environments has increased [59], requiring six-degree-of-freedom (6DOF) movement for dealing with the unevenness of the terrains and the environmental complexity. Therefore, 6DOF localization and 3D mapping have become an important research field in the robotics community. Moreover, in the field of computer vision, the growing interest in the 3D modeling of large objects and scenes has forced the scientific community to face new challenges, with the aim of reducing the propagation error present in the registration process [85]. In both situations, the robot/camera pose is estimated in order to be used in a further alignment of the 3D map/surface. Although techniques differ slightly depending on the application, both communities tend to solve similar problems by means of different approaches [26] [73].

This chapter presents a survey of the most significant 3D registration techniques, analyzing their pros and cons and potential applications. In general, a good estimation of the initial position is always required independently of the approach or technique used. Hence, Section 2.2 provides a classification of the most important methods used to obtain a coarse pose estimation, including inertial navigation, visual odometry and surface-to-surface based matching, among others. Once initial pose estimation is obtained, pair-wise registration approaches such as the Iterative Closest Point are used to refine the alignment between two clouds of points. These techniques are described in Section 2.3. Finally, any error accumulated between correlated views is minimized by means of cycles and overlapping regions common among the acquired views. Hence, Section 2.4 discusses a new classification of these techniques including metaview registration, simultaneous minimization approaches, techniques based on graphs and statistic techniques (in the context of Simultaneous Localization and Mapping). These techniques are compared and discussed analyzing their pros and cons. The chapter ends with a summary of the most relevant conclusions.

2.2 Coarse one-to-one pose estimation

The first step in 3D image registration process consists in obtaining an initial estimation of the Euclidian motion between two views as a previous step for further refinements. The coarse pose registration techniques used can differ considerably depending on several factors such as the vision system used and the camera movement. The initial pose can be obtained using two well-known approaches, as shown in Table 2.1. The former is based on the use of some sort of device: a) sensors, such as odometers, compasses or inertial systems; or b) mechanisms, such as rotating tables, robot arms or conveyors. The latter is based on direct analysis of visual images (given by cameras) or surface acquisitions (given by scanners) looking for correspondences which are used to solve the alignment and consequently, the pose.

2.2.1 Initial pose estimation by mechanical devices

In the field of mobile robotics, odometry is the most extended positioning system. Odometry is based on the use of motion sensors/encoders that provide information such as the rotation of the wheels to determine the distance traveled by the vehicle from a known position. However, sometimes the motion of the wheels does not correspond to robot motion, due to some typical problems such as wheel differences and irregularities of the ground (bumps, holes, etc). Nevertheless when the environment is nearly planar and the distance traveled by the vehicle is small, odometry provides good accuracy. However, when the environment is rough and unstructured several problems such as glides can cause some errors that are accumulated and increased with the distance traveled. Usually, unstructured terrains imply 6 Degrees Of Freedom (6DOF) motion of the vehicle, that is, three linear degrees of freedom (x,y,z) for the position and three rotational degrees of fredoom

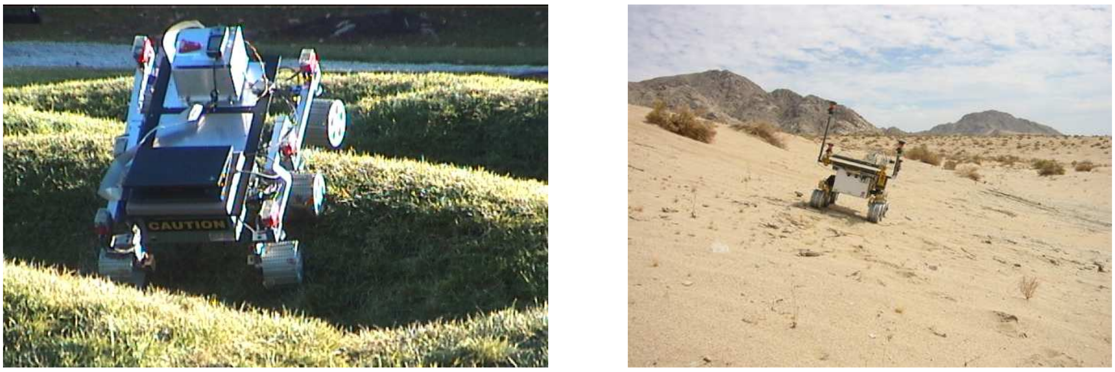


Figure 2.2: Odometry over rough terrains

(roll, pitch and yaw) for the orientation (see Figure 2.2). Therefore, when dealing with 6DOF movement, inertial data must be added in order to provide the attitude variations and compensate the errors caused by navigating through rough terrains [26]. Several inertial navigation systems such as Inertial Measurement Unit (IMU) are usually mounted on the mobile vehicles and, combined with odometry, provide better accuracy for navigation. Usually, these kind of systems have been used in order to make a coarse approximation of the camera/vehicle pose. Some authors proposed to estimate the egomotion of the mobile robot by using a feature-matching algorithm in which odometry is used to predict the region where the search for features starts [59]. In addition, as will be seen in Section 2.4, data provided by sensors can also be used for Kalman Filter initialization in many SLAM (Simultaneous Localization and Mapping) approaches [26].

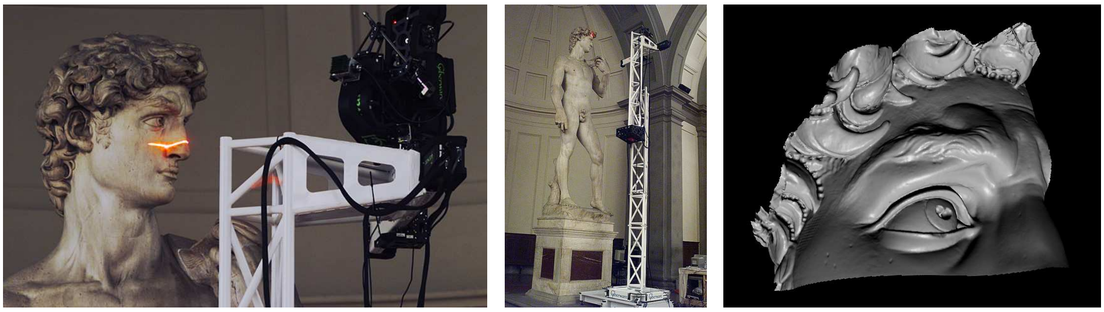


Figure 2.3: Scanning process from the ”Digital Michelangelo Project”. Left and center: Scanner mounted in a mechanical structure. Right: 3D Model of Michelangelo’s David head.

In the computer vision field and when the 3D reconstruction of small objects is considered, cameras and scanning lasers are mounted on rotating tables or mechanical structures in order to scan the whole object. In this case, the initial pose can be estimated by means of calibration of these mechanisms and the use of encoders. In many cases, the position given by the mechanics is combined with some sort of range image processing. Figure 2.3 shows the scanning process of David of Michelangelo in the context of Michelangelo's project, where a combination of mechanical instrumentation and interactive alignment is used for obtaining the motion among views [73]. Other mechanical instruments such as a magnetic tracker, used to capture the translation and rotation coordinates of the camera, have also been considered to provide a starting point for a further alignment [9]. The problems with using this kind of mechanisms is the area restrictions and movement constraints. That is, objects must be placed in precise positions, even though sometimes it is not possible to move the object from its original position or to install the mechanisms in a given environment.

Finally, as will be shown in next section, the use of range or visual image processing becomes crucial when dealing with large objects or environments due to the problems associated to odometry and the high cost of using specific mechanisms to cover the whole scanning surface.

2.2.2 Initial pose estimation by computer vision

When sensors or mechanical devices can not be used or when their measure is rough or inaccurate, an estimation of the initial position by means of computer vision may be a good choice. The main objective is to compute the motion between two views by determining the point correspondences between them. Two main groups of techniques have been proposed: (1) image-to-image correspondences, dealing with 2D images directly acquired by a stereohead or a moving camera; and (2) surface-to-surface correspondences, dealing with 3D features or clouds of points acquired by any 3D acquisition technique such as stereo, laser triangulation or time-of-flight lasers, among others.

All these methods process the 2D/3D points of the given images/surfaces to extract significant points which are used in the matching process [34] [85] [20]. Hence the techniques are classified according to: a) *feature-to-point* approach when the significant points are only those that satisfy a given feature and b) *point-to-feature* approach when an arbitrary

Table 2.1: Classification of coarse one-to-one pose estimation techniques.

Technique				Author	DOF	Sensor	Scene
Coarse one-to-one pose estimation	Mechanical devices	Sensors		Folkesson, 2003 [26]	6R	TOF	outdoor
				Nüchter, 2004 [59]	6	TOF	outdoor
				Kohlhepp, 2004 [44]	6R	TOF	indoor
		Mechanisms		Pulli, 1999 [73]	6	LT	object
				Bernardini, 2002 [9]	6	LT	object
	Computer vision	Image to image	Feature to point	Huang, 1989 [37]	6	monocular	indoor
				Shang, 1998 [101]	6	binocular	indoor
				Davison, 2003 [18]	6	monocular	indoor
				Burschka, 2004 [13]	6	monocular	outdoor
			Point to feature	Ashbrook, 1995 [4]	6R	database	object
				Lowe, 1999 [48]	6	binocular	indoor
				Se, 2002 [81]	6R	trinocular	indoor
		Surface to surface	Feature to point	Stamos, 2003 [85]	6	TOF	outdoor
				Wyngaerd, 2003 [99]	6	DLP	object
				Nister, 2004 [64]	6	monocular	outdoor
				Triebel, 2005 [92]	6R	TOF	outdoor
			Point to feature	Chua, 1997 [17]	6	database	object
				Chen, 1998 [15]	6	DLP	object
				Johnson, 1999 [40]	6	DLP	object
				Carmichael, 1999 [14]	6	DLP	object
				Kim, 2002 [43]	6	database	object
				Huber, 2003 [39]	6	LT	object

R: Restricted (some DOF are constrained in a limited range); TOF: Time-of-flight; LT: Laser Triangulation; DLP: Digital Light Projector.

group of points are characterized obtaining a set of features that differ one from another depending on point neighbourhood. Some of the most common *feature-to-point* methods are the corner detector proposed by Harris for 2D images [34] and the straight line-based method proposed by Stamos [85] for 3D views. In these methods, points with similar features are potential matchings in the registration process. Referring to *point-to-feature* approaches, some of the most used methods are the Scale Invariant Feature Transform (SIFT) algorithm proposed by Lowe [48] for 2D images and the point signature proposed by Chua [17] for 3D views. All mentioned methods are explained in detain within the following sections and classified in Table 2.1.

Image-to-image correspondences

Some 3D pose estimation approaches are based on 2D image-to-image matching, which also concerns camera egomotion [35] [100]. Since 1980s, methods based on both discrete and differential epipolar constraint have been proposed. The discrete case is used mainly in self-calibration of stereo-heads (both monocular and binocular), whereas the differential case deals with a unique moving camera at high image rate. The discrete case is based on the so-called Essential Matrix, when the intrinsic camera parameters are known, or the Fundamental Matrix in the uncalibrated case [2]. The differential case is based on the optical flow and the differential epipolar constraint [3]. An early work was developed by Huang, who proposed a linear matching algorithm based on the Essential Matrix for determining 3D motion by using eight point correspondences in two views [37]. Another method for the motion estimation of a moving uncalibrated stereo ring was proposed by Zhang in 1996 [101]. Even using stereo pairs, the fact that they are not calibrated makes it comparable to the motion estimation of a unique moving camera. Zhang's proposal was based on computing the fundamental matrix and then estimating the motion up to a scale factor by solving the well-known Kruppa equations computing a perspective reconstruction. The Euclidean reconstruction was obtained by taking any metric measure from the scene that allows the determination of the scale factor, usually a distance between two 3D features [18] [86].

In summary, techniques based on the discrete epipolar geometry have been widely studied and nowadays robust solutions are available even in 6DOF [13]. Besides, the differential movement estimators are quite sensitive to noise. Therefore, these methods are usually adapted to the application constraining the number of DOF with the aim of reducing the error in the estimation.

Note that image-to-image methods are commonly based on *feature-to-point* approaches. However, in 1995 Ashbrook et. al. [4] proposed a robust *feature-to-point* technique called Pairwise Geometric Histogram that used histograms to recognize 2D rigid shapes. The method starts performing edge extraction and then approximating detected edges by line segments. The next step of the algorithm consists in characterizing each edge (feature) by means of the histograms of their corresponding segments. Finally, matches between histograms can be performed. Some years latter, in 1999, Lowe proposed a new feature extraction algorithm called SIFT (Scale Invariant Feature Transform), which has been widely used in the last years especially dealing with 2D images. The method is classified as a *point-to-feature* approach since the main contribution of the method is the characterization of significant points according to scale invariant features. However, the method first selects the significant points by using a *feature-to-point* approach, such as Harris Corner Detector, so both approaches are combined in the SIFT technique [48].

Surface-to-surface correspondences

Many authors proposed techniques that process directly the clouds of points provided by any 3D acquisition system. Here, the main difference is found in the way of selecting the matching points.

As explained before, some approaches, which have been classified as *feature-to-point*, are based on searching points that satisfy a given feature and use then these points to solve the matching. Here, some authors propose to extract 2D features from the images, such as corners or straight lines. Subsequently, features are tracked over time to solve the matching with their corresponding 3D points already acquired, obtaining an initial estimate of the movement. Following this idea a new concept named Visual Odometry was introduced by Nister et al. in 2004 [64]. The authors proposed a new algorithm for stereo camera pose estimation based on the well known preemptive RANSAC algorithm. Other techniques deal directly with 3D features extracted from the clouds of points, such as the straight line-based method proposed by Stamos [85], who used a range segmentation algorithm in order to extract planar regions and linear features, and the curved line-based method presented by Wyngaerd [99], who proposed a coarse motion estimation between views by matching bitangent curves (see Figure 2.4). Other methods such as the Algebraic Surface Models and the Principal Curvature, explained in detail in the survey presented by Salvi et al. [79], can also be included in the same group.

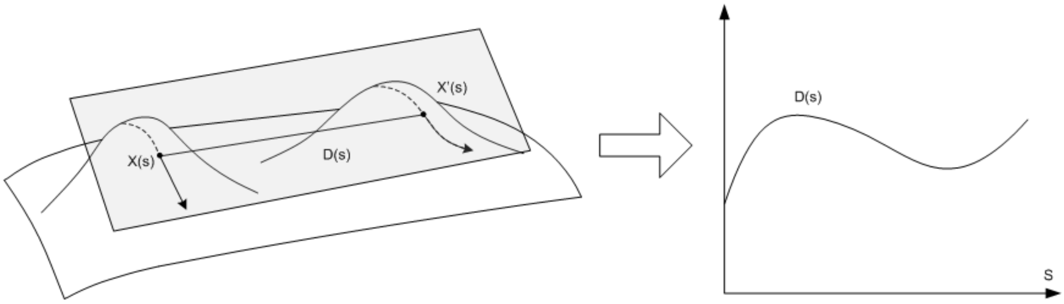


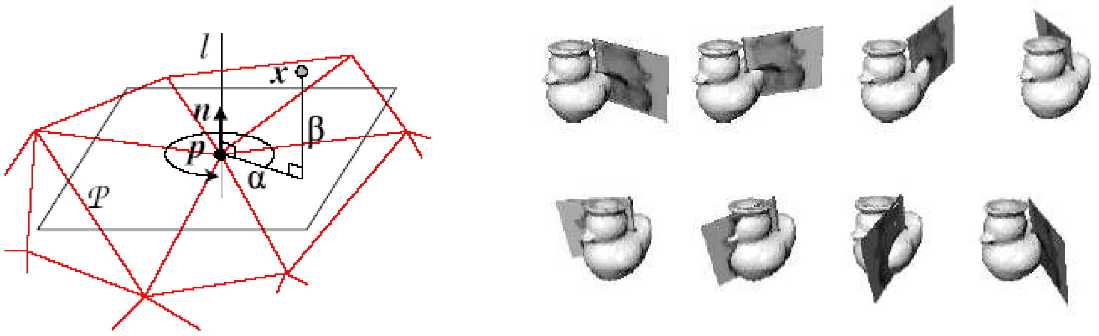
Figure 2.4: Bitangent curves. The distance between a two bitangent points (X, $X^{'}$) is expressed by a invariant signature of bitangent curve.

Other approaches are based on characterizing the points by using their neighborhood information to obtain a set of features. Once obtained, these features are sought within the cloud of points in subsequent views to solve the matching. That is the reason why they have been classified as *point-to-feature*. Two of the most used approaches of this group are the Spin Image and the Point Signature.

Spin Image is a point descriptor based on projecting neighboring 3D points onto a 2D plane tangent at a given point, obtaining as a result a 2D Image (feature) for that point. Considering a region around a given point, and computing the distance between the normal vector of that point to a point p in the range image and the distance between a point p and the 2D tangent, the spin image is determined [40]. That is, given a point p, the first step of the method is to create an "oriented point" using the position of p and the surface normal n at this point p. A tangent plane P at this given point p is computed using the position of its neighboring points. Then, a set of points $(x_1, \ldots, x_n)$ that belong to a region around the given point p are considered to characterize such point. That is, two distances from each neighbor point x are computed to determine what the author called the spin-map S_O: the distance α between each point x to the normal vector through the tangent plane P and the perpendicular distance β from such point to the plane P (see Figure 2.5).

$$\begin{aligned} \alpha &= \sqrt{\|x-p\|^2 - (n \cdot (x-p))^2} \\ \beta &= n \cdot (x-p) \end{aligned} \tag{2.1}$$

where p is the given 3D point that we want to characterize, n is the normal vector at this point and $(x_1, \ldots, x_n)$ the set of neighboring points used to generate the spin image.



(a)

(b)

Figure 2.5: (a) Spin image axis. Representation of the oriented point and the surface normal. (b) Spin image generation process, where a plane spins around oriented point, accumulating points at each step.

Once all the distances are obtained, a nxn table is built, where α represents the x-axis and β the y-axis. Each cell of the table contains the number of points that belong to

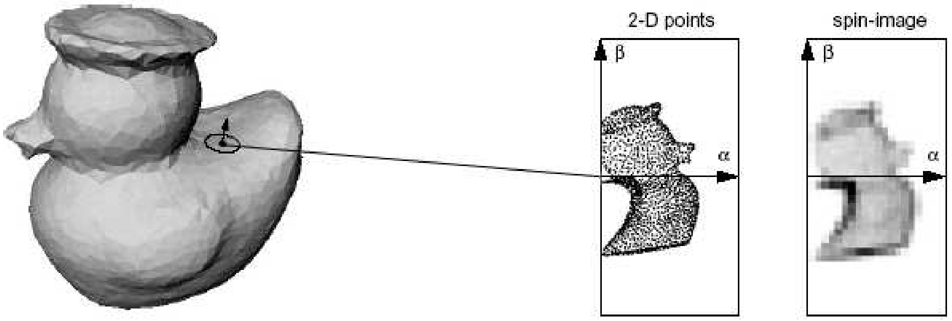
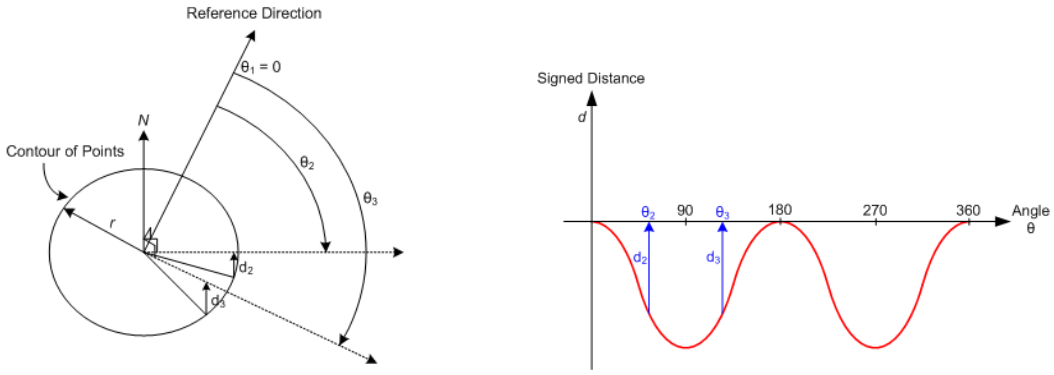


Figure 2.6: Spin image point characterization. Spin image is represented by a matrix where dark parts reflect regions with a huge amount of points.

the local region, forming the spin image (see Figure 2.6). When several spin images from different views are computed, a matching process can determine the best correspondences between them. A variant of this method was proposed by Carmichael, carrying out an interpolation of a set of points inside every triangular mesh with the aim of normalizing the number of points in every spin image [14].

In 1997 Chua proposed a new *point-to-feature* method named Point Signature, that consists in a new 3D point representation with the aim of describing 3D form surfaces by using the structural neihbourhood of a point [17]. Given a point p, the author placed a sphere of radius r centered on p. The intersection of this sphere with the surface gives the contour of points C, whose orientation is defined by the normal vector n_1, a reference vector n_2 and the vector obtained by the cross-product of the other vectors. The point signature method defines the distance between the 3D points that compose this contour to the reference plane where they are projected (see Figure 2.7). Therefore, each point of C is characterized by the distance from the contour to its projection to the plane and the angle from the reference vector n_2. The description of a 3D point is given by the characterization of all the points of the contour. Once the points of the two views are characterized using point signatures, they can be compared in order to find the correspondences.

Other authors proposed the use of the Principal Component Analysis in order to estimate the main axis of the whole cloud of points. Subsequent views are aligned assuming that the main axis in two consecutive acquisitions do not vary significantly [43]. Other methods propose to estimate the motion between surfaces by using algebraic surface models. That is, all the points of the surface are represented (characterized) by polynomial models. Therefore, these techniques can also be considered as *point-to-feature* [96].



(a) (b)

Figure 2.7: Point signature definition process.

In general the main problem of most *point-to-feature* algorithms is the time involved in computing the point descriptors. Although the matching process is very fast and efficient, the points of both surfaces must be characterized before searching correspondences. For example, Point signature method needs to search for the neighbors and interpolate the surface in order to find the contour. This usually implies a huge amount of time in comparison with the segmentation methods used to search corners or lines.

In this section we have presented the most representative coarse pose estimation techniques. Although initial pose estimation methods based on mechanical devices provide good results in flat terrains, a combination of both mechanical and computer vision methods is usually required in the presence of rough and unstructured environments. Techniques based on the discrete epipolar geometry have been widely studied and nowadays robust solutions are available even in 6DOF. Besides, the differential movement estimators are quite sensitive to noise. Hence, these methods are, in general, adapted to the application constraining the number of DOF with the aim of reducing the error in the estimation. Therefore, surface-to-surface alignment is more adequate for complex 3D scenarios, but then we have to avoid symmetries in the views to obtain accurate registrations.

2.3 Fine One-to-One Pose Estimation

Once an initial 3D pose is estimated by any coarse registration technique, an iterative minimization should be applied to obtain a refined pose and hence a better alignment between both views improving the quality of the registration. In this section, the methods are classified according to the minimization function, which is usually the distance between corresponding points (*point-to-point*) or the distance between points and their corresponding planes (*point-to-plane*) as shown in Table 2.2 and discussed in the following paragraphs.

2.3.1 Point-to-point

Point-to-point alignment such as the Iterative Closest Point (ICP) focuses on minimizing the distance between point correspondences [10]. Once an initial estimation of the motion is known, all points are transformed to the same reference frame applying the Euclidian motion. Then, every point in the first image is taken into consideration to search for its closest point in the second image. The distance between these correspondences is minimized, and the process is iterated until convergence is achieved.

Some modifications of ICP have been presented in recent years to improve the efficiency of the algorithm [31] [92] and also to decrease the computing time [41]. In addition, other authors proposed some improvements to increase the robustness of ICP. For instance, Trucco [Trucco et al., 1999] implemented the RICP method making use of the Least Median of Squares approach [94].

Overall, ICP is the most common fine registration method and the results provided so far are satisfactory. However, this method usually presents problems with convergence, requiring many iterations and in some cases converging to a local minima [30] [77]. Moreover, unless a robust implementation is used, the algorithm can only be used in *surface-to-model* registration [54].

2.3.2 Point-to-plane

The problem with *point-to-point* distance is that the correspondence of a given point in the first view may not exist in the second view because of the limited number of points

Table 2.2: Classification of fine one-to-one pose estimation techniques.

Technique		Author	DOF	Sensor	Scene
Fine one-to-one pose estimation (Pair-wise)	Point to point	Besl, 1992 [10]	6	LT	outdoor
		Trucco, 1999 [94]	6	database	object
		Greenspan, 2001 [31]	6	DLP	object
		Jost, 2002 [41]	6	database	object
		Guidi, 2004 [32]	6	DLP	object
		Triebel, 2005 [92]	6R	TOF	outdoor
	Point to plane	Chen, 1991 [16]	6	DLP	object
		Pulli, 1999 [73]	6	LT	object
		Gagnon, 1994 [28]	6	monocular	object
		Park, 2003 [68]	6	database	object

R: Restricted (some DOF are constrained in a limited range); TOF: Time-of-flight; LT: Laser Triangulation; DLP: Digital Light Projector.

acquired by the sensor, especially considering low resolution surfaces. To address this problem, some authors use the *point-to-plane* distance. The Point-to-plane algorithm proposed by Chen [16] is an alternative to ICP. The algorithm is based on distance minimization between points and planes. Given a point in the first image, the intersection of the normal vector at this point with the second surface determines a second point in which the tangent plane is computed (see Figure 2.8). The distance between this plane and the initial point is the function to minimize.

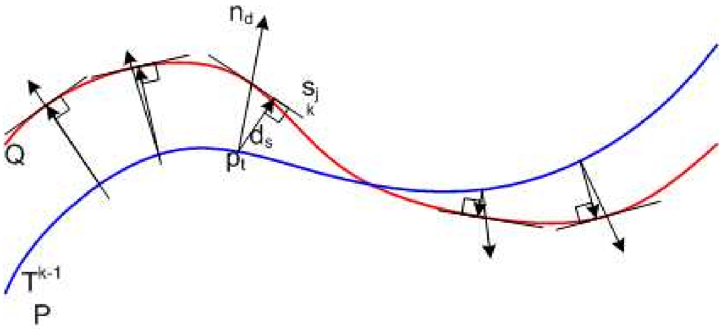


Figure 2.8: Distances between points in surface P and planes in surface Q

Different algorithms have recently been presented with the aim of speeding up the computation of the difference between points and planes. Gagnon et al. [28] presented a new method based on Chen's idea but using a different approach. The authors proposed the use of a grid where the surface of the second image and each normal vector at a given point in the first image are projected. In 2003, Park [Park and Subbarao, 2003] proposed a fast variant based on recursively computed the intersection between the line and the 3D surface. This technique is based on a iterative point-to-projection, obtaining fast and accurately results.

This method is more robust to local minima and, in general, better results are obtained. Moreover, the method is less influenced by the presence of regions with few overlap since only the control points whose normal vector intersects the second view are considered in the matching, differing from ICP, where all points in the first cloud are used in the registration. Moreover, *point-to-plane* approaches usually require fewer iterations compared to *point-to-point* ones.

2.4 Multiview Minimization

In the previous sections, this survey was focused on techniques based on the *one-to-one* alignment of two views (pairwise). However, the full reconstruction of a 3D object/scene usually implies the use of multiple views that must be registered to obtain a complete model. The sequential pairwise alignment of several views causes a drift that is propagated throughout the sequence. Therefore, some authors have improved their algorithms by adding a final step that uses the information of all the acquired views in the registration process. These approaches, known as multi-view registration, spread *one-to-one* pair-wise registration errors throughout the sequence of views.

Some techniques try to minimize the error of all the views at the same time. Other methods try to reduce the propagating error benefiting from the existence of loops and re-visited regions. Therefore, Multiview techniques are classified into: (1) metaview, (2) graph-based, (3) statistic and (4) simultaneous minimization methods, depending on the methodology used, as shown in Table 2.3. These techniques will be described in detail in the following sections.

2.4.1 Metaview techniques

Early approaches proposed the aggregation of subsequent views in a single metaview that was progressively enlarged each time another view was registered [16]. The main constraint

Table 2.3: Classification of Multiview minimization techniques.

Technique		Author	DOF	Sensor	Scene
Multiview minimization	Metaview	Chen, 1991 [16]	6	DLP	object
		Pulli, 1999 [73]	6	LT	object
		Nüchter, 2004 [59]	6	TOF	outdoor
	Simultaneous minimization	Triggs, 2000 [93]	6	database	object
		Pollefeys, 2000 [72]	6	monocular	outdoor
		Fitzgibbon, 2001 [24]	6	database	object
		Masuda, 2002 [51]	6	LT	object
		Silva, 2003 [83]	6	database	object
		Mouragnon, 2007 [58]	6	monocular	outdoor
	Graph analysis	Bergevin, 1996 [8]	6	monocular	object
		Neugebauer, 1997 [62]	6	LT	object
		Huber,2003 [39]	6	LT	object
		Sharp, 2004 [82]	6	DLP	indoor
	Statistic techniques	Guivant, 2000 [33]	6	TOF	outdoor
		Martinelli,2005 [49]	6R	TOF	indoor
		Liu, 2003 [47]	6R	TOF	outdoor
		Bosse, 2003 [12]	6	TOF	outdoor
		Estrada, 2003 [23]	6R	TOF	outdoor
		Montemerlo, 2002 [57]	6R	TOF	outdoor

R: Restricted (some DOF are constrained in a limited range); TOF: Time-of-flight; LT: Laser Triangulation; DLP: Digital Light Projector.

of this strategy is the lack of flexibility to re-register views already merged in the metaview.

Method of Chen

In 1991, Chen and Medioni [16] proposed an early approach for registering and matching multiple 3D scans. Their method started by registering two views and merging them in a single metaview, using a previously obtained coarse pose estimation. Afterwards, the new views were sequentially registered and merged within the metaview, using the point-to-plane pairwise method explained in the previous section. As an initial step of the multiview registration process, the author converts the acquired views into cylindrical or spherical coordinate systems, using the data provided by the pairwise registration, obtaining an intermediate representation of the object. In this way, the elevation and angular coordinates of the points can also be used to determine the correspondences in overlapping areas. Once the registration process ends, the views are represented again in the Cartesian coordinates.

The main advantage of this method, when it is compared to the ones based only on sequentially pairwise alignments is that the whole information from the integrated surface can be used in each new registration step. Although the views are not registered simultaneously, this method can be considered as the beginning of multiview registration. However, this method does not allow to re-register views already merged in the metaview. The success depends on the accuracy of the movement estimation between views, since there is no way back to modify already merged sequences. Therefore, bad movement estimations lead to registration errors which will be accumulated through the subsequent registrations. This problem increases with the size of the objects, decreasing the accuracy of the method. This approach has been followed for several authors such as Masuda [52], who proposed a similar method with the main difference that outliers were detected and removed before the global registration leading to a more robust solution.

Method of Pulli

In 1999, Pulli [73] proposed an ICP relaxation method based on the previous metaview approach but considering all the potential alignments between views before proceeding with the multi-view registration. The main idea of the method is to use the constraints imposed by pairwise registration for global alignment in multiview, avoiding to keep all scan data in memory. In addition this method takes into account the information of all the overlapping areas allowing that the already registered regions can be analyzed again for further re-registrations. In order to obtain the set of constraints for the later multi-view step, the approach starts by performing a pairwise registration using the previously explained point-to-plane technique (or method of Chen). Regarding the alignment step, the concept of the virtual mate approach has been introduced. This approach uses the relative transformations between the scans provided by pairwise registration in order to predict were a point from the first scan should be situated on the second one, creating a virtual mate for each point (correspondences between points).

Once all pairwise constraints are obtained, multi-view registration is carried out. The algorithm follows the idea of the metaview approach, adding into a set of consistent views one scan each time, spreading the error uniformly between pairwise registration (see algorithm 2.4.1). That is, we have an initial list of related scans, what the author called *dormant_set*. The goal is to create a list, *active_set*, containing a final set of accurate aligned views. The multiview algorithm analyses the scans contained in the *dormant_set* with the already classified views in the *active_set*. This procedure selects the view from the *dormant_set* which has more connection with the ones contained in the *active_set* and memorizes it into both a temporal queue and *active_set*. The next step is to remove the first scan of the queue and align it with the overlapping views of the *active_set* using the pairwise constraints. If the error is reduced enough this set of neighbours are merged into the queue, repeating this step until it becomes empty. This process is repeated iteratively for each component of the dormant_set, spreading the error among all views. Notice that in order to initialize the *active_set* the procedure starts analysing all the views of the *dormant_set* choosing the best correlated one. To refine the final model, the set of views obtained at the end of the algorithm can be considered a new *dormant_set*, restarting the global process again.

Algorithm 2.4.1: MULTIVIEW REGISTRATION (*views*)

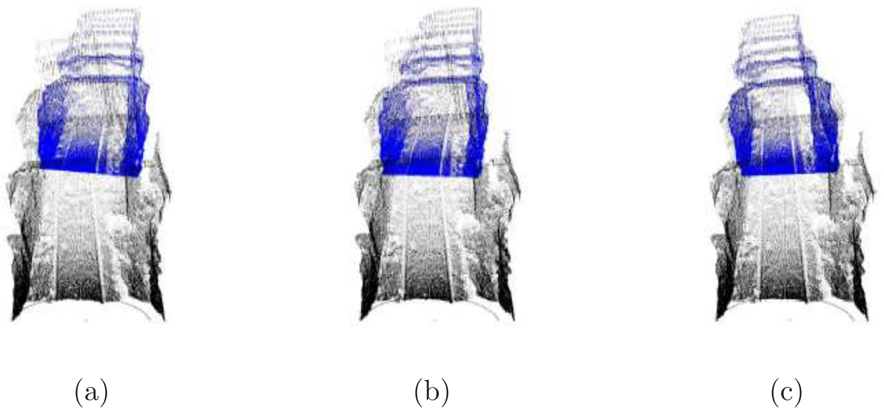
```
$dormant\_set = views$
$curr = most\_links(dormant\_set, dormant\_set)$
$active\_set.\text{ADD}(curr)$
$dormant\_set.\text{REMOVE}(curr)$

while not empty dormant_set
  do
    $curr = most\_links(dormant\_set, active\_set)$
    $active\_set.\text{ADD}(curr)$
    $dormant\_set.\text{REMOVE}(curr)$
    $queue.\text{PUSCH}(curr)$
    while not empty queue
      do
        $curr = queue.\text{POP}()$
        $nbors = active\_set.\text{NEIGHBORS}(curr)$
        $relative\_change = align.(curr, nbors)$
        if $relative\_change > tolerance$
          then $queue.merge(nbors)$

return
```

Pulli's multiview approach is independent of the initial alignment chosen, so that the most suitable pairwise can be used in every case. The author proposed a point-to-plane pairwise that provides more robustness to local minima and requires fewer iterations compared to point-to-point ICP. However, a main constrain of this multiview relaxation method is the need of preserving the pairwise alignments of the overlapping regions as much as possible. Therefore, good initial pose estimation plays an important role for obtaining accurate 3D models. On the other hand, since registration pairwise is used before multiview alignment, the high cost that entails to align and merge range views simultaneously can be avoided. In addition, the use of constraints instead of the full set of views reduces the memory space needed, allowing large data sets to be used.

Method of Nüchter

In 2004 a similar approach was developed by Nüchter [59]. The method provides a new solution to the simultaneous 6DOF localization and mapping oriented to Autonomous Mine Mapping (see Figure 2.9). The author presented a global relaxation method called *simultaneous matching* based on Pulli's proposal but with the main difference that no iterative pair-wise alignment is required. At the beginning stage of the algorithm, the reference system is determined by the first obtained scan (master scan). Then, using the initial pose estimation provided by odometry or other coarse pose estimation technique, a simple pairwise alignment is carried out to obtain a start motion estimation for further alignments. When a new scan is obtained, a queue is initialized with it and starts the iterative process. The first scan of the queue is removed and a set of scans overlapping with it is calculated (neighbors of the current scan). The current scan is aligned with all its neighbours by using the well known ICP method. If the scan has changed its location, all its neighbours are added to the queue. This process is repeated iteratively for each component of the queue until it is empty, spreading the error among all views. Note that in order to reduce the data for ICP algorithm the author uses a combination between a median and a reduction filter. In addition, kD-trees are used to speed up the data access for matching process.

The approach proposed by Nuchter does not strongly require loop detections since the error can be diffused appropriately over all the 3D scans. However, if a view of a revisited area is obtained, the result of the multiview registration can be considerably improved. If a loop is detected after the coarse matching, the error can be spread over the 3D scans as a previous step for posterior refinement. That is, there is a transformation matrix (R, t)



(a) (b) (c)

Figure 2.9: 3D reconstruction project of Mathias mine. (a) Pose estimation provided by Odometry (b) Pose estimation after 5 iterations of the algorithm (c) Final alignment.

that describes where the scan that closes the loop should be. The author proposes to distribute the total error transformation (R, t) proportionally to each view of the path $(R_i,\ t_i)$, were R_i and t_i are the rotation and translation of each view i. A weight c_i is assigned to each scan depending on its distance to the first one.

$$c_i = \frac{distance(initial\_scan, current\_scan)}{total\ distance\ of\ the\ path} \tag{2.2}$$

Therefore, the translation t_i that should be applied to each corresponding scan can be calculated as: $t_i = c_i t$

The rotation matrix R_i is represented as:

$$R_i = \begin{pmatrix} cos(c_i\theta) + a_x^2(1 - cos(c_i\theta)) & a_z sin(c_i\theta) + a_x a_y(1 - cos(c_i\theta)) & -a_y sin(c_i\theta) + a_x a_z(1 - cos(c_i\theta)) \\ -a_z sin(c_i\theta) + a_x a_y(1 - cos(c_i\theta)) & cos(c_i\theta) + a_y^2(1 - cos(c_i\theta)) & -a_x sin(c_i\theta) + a_y a_z(1 - cos(c_i\theta)) \\ a_y sin(c_i\theta) + a_x a_z(1 - cos(c_i\theta)) & -a_x sin(c_i\theta) + a_y a_z(1 - cos(c_i\theta)) & cos(c_i\theta) + a_z^2(1 - cos(c_i\theta)) \end{pmatrix} \tag{2.3}$$

Where $\mathbf{a}$ and θ are computed by:

$$a = \begin{pmatrix} \frac{q_x}{\sqrt{1-q_0^2}} \\ \\ \frac{q_y}{\sqrt{1-q_0^2}} \\ \\ \frac{q_z}{\sqrt{1-q_0^2}} \end{pmatrix}; \qquad \theta = 2arcos(q_0) \tag{2.4}$$

Note that the rotation by an axis **a** and the angle θ, are described by the quaternion q.

$$q = \begin{pmatrix} q_0 \\ q_x \\ q_y \\ q_z \end{pmatrix} = \begin{pmatrix} \frac{1}{2}\sqrt{trace(R)} \\ \\ \frac{1}{2}\frac{r_{3,3}-r_{3,2}}{\sqrt{trace(R)}} \\ \\ \frac{1}{2}\frac{r_{2,1}-r_{2,3}}{\sqrt{trace(R)}} \\ \\ \frac{1}{2}\frac{r_{1,2}-r_{1,1}}{\sqrt{trace(R)}} \end{pmatrix}, where\ \boldsymbol{R}\ is\ the\ total\ error\ rotation\ matrix \tag{2.5}$$

When compared to Pulli's approach, this method does not require to acquire and register all the views previously, allowing to simultaneously localize de vehicle and mapping the environment in real time. In addition, Nuchter approach turns out to be more reliable in large environments since loop constrains allow to reduce the accumulated error. However the success of this method depends on a correct initial pose estimation of the vehicle. Another challenge appears when the robot has to deal with featureless environments, requiring the use of specific methods such as Kalman Filter extensively explained in Section 2.4.4.

2.4.2 Simultaneous Minimization

The main problem of metaview approaches is that the already merged surfaces can not be modified when a new view is registered. Therefore the information provided by the new acquired views can not be used for decreasing the error of the previous registrations.

Some authors have proposed several techniques in order to register multiples views and minimize the accumulated error simultaneously. These techniques make use of all the acquired views at the same time and have been classified as *simultaneous minimization techniques.*

Signed distance Field of Masuda

In 2002 Masuda proposed a new method that registers and integrates all views simultaneously, rejecting outliers in the iterative process. The method is based on Matching Signed Distance Fields and works as follows: Initially a coarse registration is carried out in order to transform all views to a common reference frame, where the data shapes are integrated. In the next step, data shapes are alternatively registered and integrated in an iterative process until convergence is achieved. The main idea is to generate a grid of arbitrary 3D points that are used as key points for the Signed Distance Field approach. The goal is to establish correspondences (the closest point) between the key points and the 3D points of the object surface, computing the distance between them. Once correspondences have been obtained the new motion parameters are calculated and the process is repeated again until residual error converges. Note that points are weighted depending on the computed distances in order to detect and reject outliers.

The main advantage of this method is that all views are registered simultaneously and consequently the error does not accumulate among them. In addition outliers are automatically removed leading to a robust method. On the other hand, the need of having in advance the complete set of 3D views of the object/scene leads to high memory requirements when dealing with large objects, together with the impossibility of working in real time.

Genetic Algorithms

Other authors have proposed the use of methods based on Genetic Algorithms (GA) for multiple view registration, as an alternative to the ICP-based approaches. In 2003 Silva et al. [83] proposed an algorithm to register multiple range images based on GA that allow to deal with low-overlapping views. The main contribution of their method is the introduction of a novel robust measure called the Surface Interpenetration Measure (SIM) that quantifies the visual registration error in order to determine the overlapping area between two partial views. Following this idea, they use the SIM along with a robust estimator in order to implement a Genetic Algorithm in the transformation space, where the results of the SIM of each chromosome are used in each generation to choose the ones that yield the most accurate transformations. Note that each chromosome is composed by 6 parameters (three components for the translation vector and three components for the rotation angles).

Registration methods based on GAs provide accurate solutions in multiple view registration, presenting interesting solutions to problems such as local minima or error propagation. In addition, this kind of algorithms presents good results in the presence of outliers and low-overlaping views. However, all this advantages are subject to the size of the population used. That is, the most important drawback of methods based on GA is the high computational time required to converge to a good solution.

Bundle Adjustment

Other simultaneous minimization methods has been recently proposed. In the last few years, a photogrammetric technique called Bundle Adjustment has increased popularity in the computer vision community and it is also growing in robotics. Bundle adjustment deals with the problem of refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameters (camera pose and/or calibration) estimates [93]. Therefore, bundle adjustment techniques have been used in both robot/camera localization and 3D mapping in many fields such as camera calibration, robot navigation and scene reconstruction providing reliable solutions to the error accumulation in the registration process. Since bundle adjustment is a non-linear minimization problem, it is solved by means of iterative non-linear least squares or total squares methods such as Levenberg-Marquardt or M-estimator techniques [24] [53]. Recently, a new improvement of the bundle adjustment was proposed by Mouragnon et al. [58]. The authors proposed a generic real-time method based on bundle adjustment approach that permits an incremental 3D registration and reconstruction of the scene minimizing the angular error simultaneously.

Although bundle adjustment is commonly classified as a multiview technique, some authors have used it in consecutive pairwise alignment as a technique to reduce error propagation [72].

2.4.3 Graph Analysis

Another alternative are the registration techniques based on graphs. Here the main idea is to use the relation between views in multiview registration. For instance, several methods try to find the optimal path between views to obtain an accurate registration [39], while other methods use the information obtained when a part of the object/scene is revisited to detect a cycle and spread the error among the views that compose it [82].

Method of Bergevin-Gagnon

Bergevin et al. [8] can be considered the precursors of this kind of methods. The authors presented a multi-view registration technique based on graph theory, that considers all views as a whole and treats them simultaneously. The graph is created as follows: views are associated to nodes while edges represent the transformations T among them. A path between two nodes is represented by a sequence of transformation matrices, one for each link. Different graph topologies can be obtained depending on the way that range views have been acquired. When using mechanisms such as a rotating table the graph acquires a lineal shape (Figure 2.10a). On the other hand, when the images have been acquired by a free moving camera one node can have more than two nodes linked to it (Figure 2.10b). In such a case the graph acquires a more general topology.

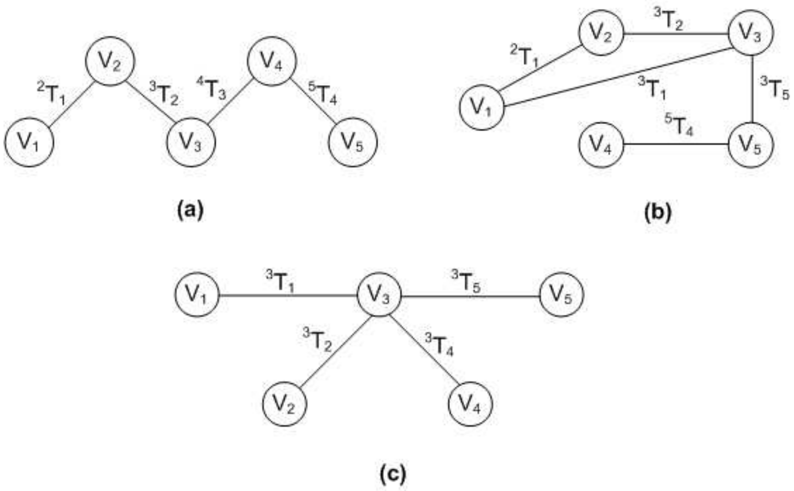


Figure 2.10: Example of different network topologies. (a) Lineal network topology, (b) General topology, (c) Star-shaped topology.

The main goal of Bergevin approach is to obtain what the author calls a *well-balanced network* , which is a network where (1) all transformations between views have a similar error, and (2) there is a unique transformation matrix between two different views, regardless the path used between them. This last condition is automatically accomplished when using star topology, where all nodes have only one link that joins them to a central node and therefore there is a unique path between two views. Therefore, the first step of the Bergevin approach is to transform these obtained graphs to star-shape topology (see Figure 2.11). A central node is selected and only one transformation matrix separates it from the rest of the nodes. In this way the path between two nodes includes at most two transformation matrices. The range view represented by the central node enforces the reference system and its transformation matrix remains fixed during all the process. The view selected as the central node is the one for which all other views can be transformed using the smallest number of matrix multiplications.

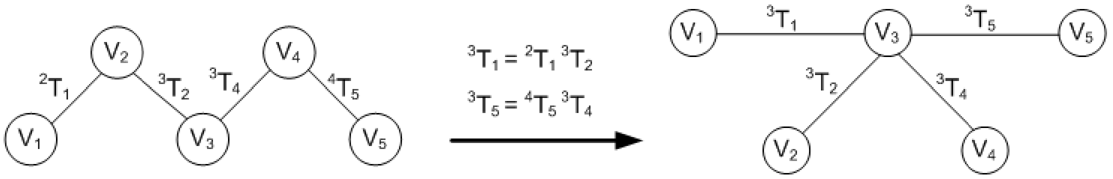


Figure 2.11: Example of transformation from a linear network topology to a star-shape.

Once the central view V_c is defined, the algorithm computes the motion between each view V_i and all the other non-central views, using two transformation matrices each time. The first matrix $M_{i,c}$ defines the motion of V_i to the central reference frame V_c, and the second $M_{j,c}^{-1}$, the motion from V_c to the other non-central view V_j. Finally, the transformation errors are minimized using the version of the point-to-plane algorithm explained in Section 2.3. That is, correspondences between points of V_i and tangent planes of each view V_j are used to minimize the error and compute the new transformation matrices using least-square technique. In this way the set of views is considered as a whole and the error in transformation matrices is uniformly distributed obtaining what the author calls a ”well-balanced network”. The algorithm converges when the transformation matrix between two non-central views tends to the Identity matrix (see Equation (2.6)).

$$C = \sum(\Delta T_{i,j} - I_{i,j})^2 \tag{2.6}$$

A similar approach was proposed by Neugebauer [62] also based on the star-shape graph topology. The main difference with the previous method is that the author readjusted the transformation matrices by minimizing the distances between correspondences using Levenberg-Marquardt, adding an statistical termination criteria instead of using a predefined threshold to indicate when the algorithm converged. Another important aspect is the reduction of the computational cost in the matching process thanks to what the author called "resolution hierarchically". In order to speed up the refinement process only few points are used in the initial iterations while the rest of the points are incrementally added to the system in further iterations.

Notice that performing matching and alignment at each iteration leads to a high computational cost and large memory requirements. Moreover, the star-shape topology can present a constraint working with high dimensional objects/environments, when the registered view is far from the one used as a reference system. In addition, if two views are close to each other but far away from the central node it is obvious that the star-shape topology deteriorate the computation of the best transformation between both views. Working with this kind of scenes the method can hardly converge since usually a huge overlapping between regions is required. When two views are strongly bad-registered, the error is spread over the other ones providing poor results, since the transformation matrix obtained by least squares is an average of all the matches. In this cases, techniques based on other graph topologies or even some metaview approaches are able to provide better accuracies.

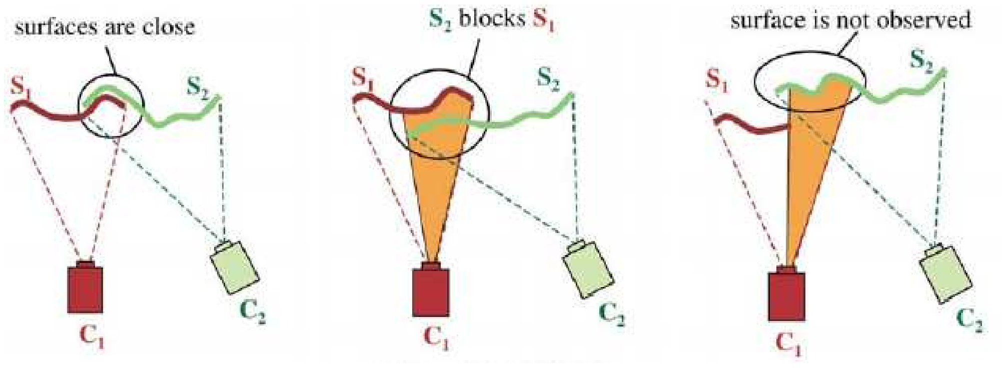
Multiview surface matching of Huber

In 2002 Huber proposed a new multiview approach that automatically registers 3D data sets without requiring any initial pose estimation of the views, what the author calls *multiview surface matching* [38]. The method is also based on graph theoretical approach, were nodes are the input views and edges represent each pairwise match between two overlapping regions. The algorithm, apart from an initial surface triangulation process, involves two main steps: (1) local registration and (2) global registration. The main goal of the local registration is to create a "model graph" with all the potential alignments between views by using a pairwise matching algorithm based on Spin Images [40]. Although refining the obtained results using a a pairwise registration algorithm (ICP) some graphs may contain huge amount of incorrect matches which have to be removed (see Figure 2.13).

In order to detect them the author proposed a local consistency test based on three consistency criteria: overlap distance, free space violation (FSV) and occupied space violation (OSV). The first one analyzes the average distance of overlapping regions trying not to exceed a defined threshold. According to the author, given two surfaces S_i and S_j, a point $p \in S_i$ overlaps the surface S_j when given a non-boundary point $q \in S_j$ close to point p the angle between the surface normals at p an q and their Euclidian distances are less that a threshold. Therefore, the average overlap distance between two surfaces S_i and S_j can be calculated as:

$$O_D(S_i, S_j) = \frac{\sum_{f \in F_0} w_f A(f)}{\sum_{f \in F_0} A(f)} \tag{2.7}$$

Where F_0 represents a set of faces (obtained by triangulation process) on S_i that overlap S_j, $A(f)$ the surface area of each face f and W_f the average distance between the three corners of f.

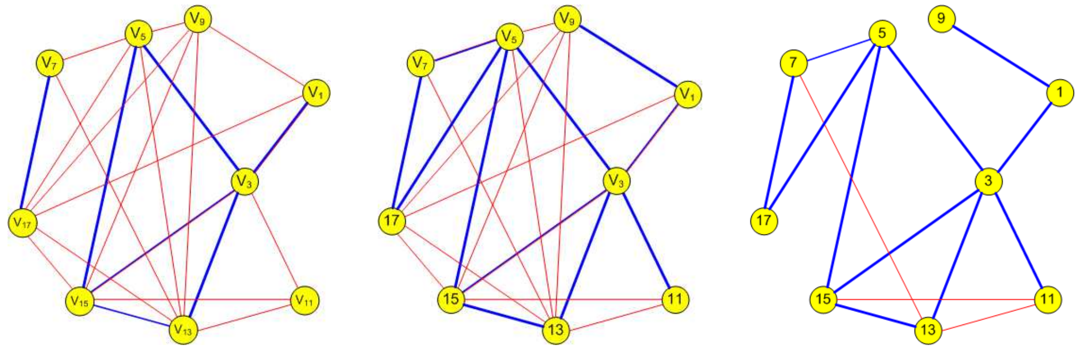


(a) (b) (c)

Figure 2.12: Visibility consistency. (a) Consistent surface. (b) Free space violation. (c) Occupied space violation.

The second and third criteria, globally named ”visibility consistence”, take advantage of the sensor viewing volume, searching for occlusions and missing surfaces in the sensor line of sight by projecting a ray from the center of the sensor C to the point p that we want to analyze. These two inconsistencies are known as: (a) free space violation (FSV), that occurs when a region blocks the visibility of another from the point of view of the sensor, and (b) occupied spaces violation (OSV), that occurs when a region is not observed form C, though it should be (see Figure 2.12).

Therefore, the local registration step involves an exhaustive pairwise alignment followed by a posterior refinement using these surface consistency restrictions. When one or more of such criteria appear in a view, it is considered incorrect and has to be removed from the graph. Then a new graph G_{LR} (Local Registration Graph) without incorrect alignments is obtained (see Figure 2.13).



(a) (b) (c)

Figure 2.13: Local registration Model Graphs. (a) Exhaustive pairwise registration though spin image. (b) Refined pairwise registration. (c) Registration after local consistency filtering.

The next step is to obtain a final subgraph with only correct matches by performing a global registration process, using the locally consistent matches of the graph G_{LR}. The process of global registration is divided in a continuous and a discrete optimization problem (error minimization). In the discrete optimization process a modified version of Kruskal's algorithm [45] is used to construct several spanning trees using the edges of G_{LR} graph with the aim of obtaining acyclic subgraphs. Here, the iterative merging method starts with a graph containing only nodes. In each iterative step of the algorithm two best-matching views are merged using pairwise information and verifying the global consistency (see Figure 2.14). In this kind of structures, there is at most one path between two views and therefore the chances of obtaining incorrect matches are reduced. In addition, this allows to directly compute absolute pose estimation to initialize the continuous optimization process, where point to plane correspondence error is minimized using Nebauer's approach, previously explained [62], spreading the pairwise errors among all the graph.

A distinctive characteristic of this algorithm is that previously knowledge of neither the scanning positions nor the scanning sequence is required. However, in case that an incorrect match is added to the final graph the global model can become irremediably inconsistent, implying restarting the full process. In addition, the visibility consistency

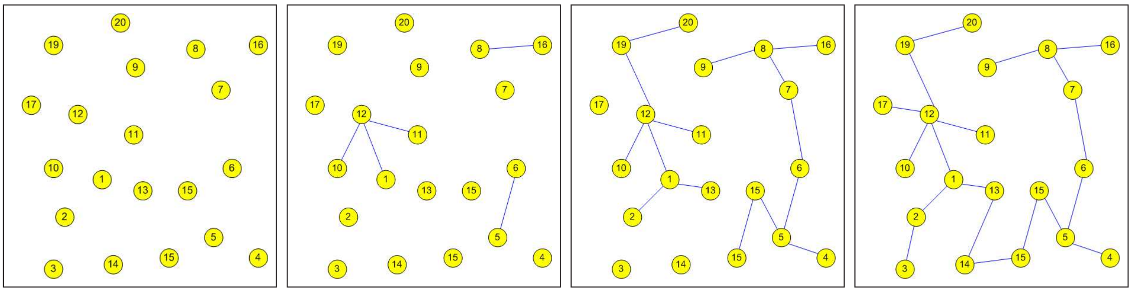


Figure 2.14: Global registration process.

techniques require the projection of a ray from the sensor to the scene, which apart from the camera requires the use of a laser. Therefore, images obtained only by cameras can not be filtered using visibility consistency techniques and overlap distance consistency method usually become insufficient.

Cycle-based detection methods

Other authors proposed to reduce the propagating error of the pairwise registration process benefiting from the existence of cycles and re-visited regions and considering the uncertainty in the alignment.

In 2004, another multi-view approach based on graph theory was presented by Sharp [82]. Similarly to the previous method the problem is treated in two main steps, starting by solving the local registration problem and finishing with a global error minimization. Initially the graph is decomposed in basic cycles, using a ring-shape topology, with the aim of first solving the local motion estimation errors. As it can be seen in Figure 2.15 a cycle is composed by a set of views aligned sequentially until the first and the last view match together. A cycle can be created by adding edges to a normal spanning tree. Transformations between views, obtained by odometry or pairwise registration results, are represented by a rotation and a translation matrix $(R_{i,j}, t_{i,j})$. A cycle can be considered ”consistent” when the rotations or rigid transformations compose to identity (see Figure 2.15).

Therefore, the main goal is to reduce the error distribution within a cycle with the aim of obtaining a consistent set of aligned views. As rotation and translation are considered independent, the local error minimization implies two main steps: (a) Translational Error Distribution and (b) Rotational Error Distribution.

Translational Error Distribution step tries to find a consistent rigid transformation set

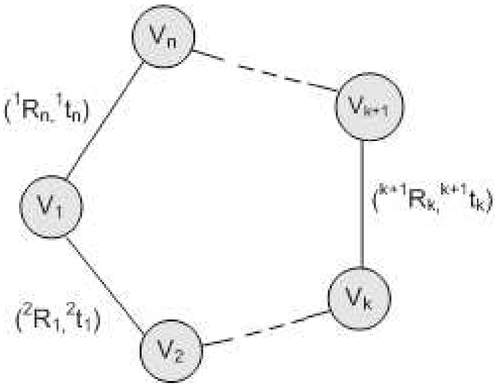


Figure 2.15: Cycle composed by several views aligned sequentially.

$t_{k,k+1}$, one for each view of the cycle, and minimize their mean square distances in order to satisfy the linear constraint $\sum_j t_{j,j+1} = 0$. This problem has been solved by least-squares estimation using Lagrange multipliers (see Equation (2.8)), finding the best translation by distributing uniformly the error among all views of the cycle:

$$t_{k,k+1} = t_{k,k+1} - \frac{1}{n}\sum_{j=1}^{n} t_{j,j+1} \tag{2.8}$$

Next step is the Rotational Error Distribution process, which attempts to minimize the rotation error among a cycle of views. First of all, the total error of the cycle is spread along all the views of the cycle (see Figure 2.16). That is, the error is divided and distributed to each rotation between views. Note that the Rotation matrix is transformed to axis-angle representation.

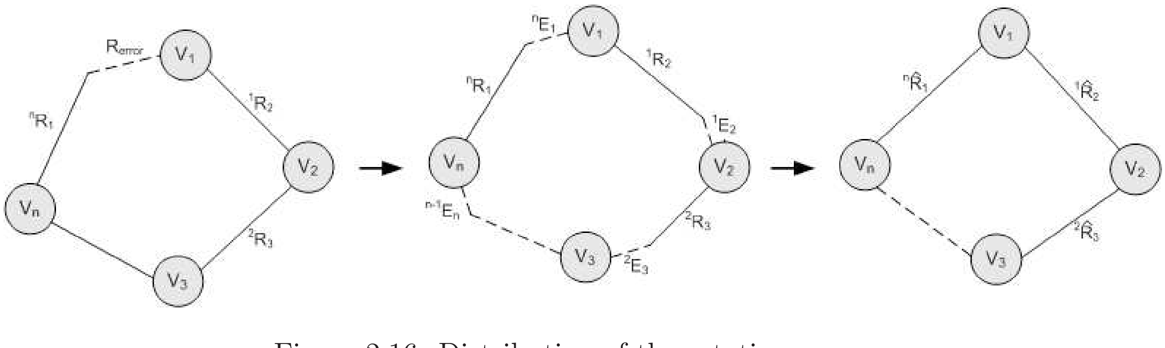


Figure 2.16: Distribution of the rotation error.

Since initial estimations of the movements are obtained by odometry or pairwise registration, the error among the alignments (angular error) might not be equally distributed. Therefore, a different weight is associated to each edge depending on the confidence in each measurement, in order to improve the accuracy of local registration.

$$R_{k,k+1} = R_{k,k+1} E_{k,k+1}^{<1/n>} \tag{2.9}$$

Once the new rotations has been obtained, the next step attempts to minimize the sum of squared angular errors among the rigid transformations, taking into account that the product of the rotations must be equal to identity matrix ($R_{1,2} * R_{2,3} * \cdots * R_{n,1} = Id$) and that the rotations must be as close as possible to the initial ones.

$$\sum_k (\angle R_{k,k+1}^{-1} \hat{R}_{k,k+1})^2 \leq \sum_k (\angle R_{k,k+1}^{-1} \tilde{R}_{k,k+1})^2 \tag{2.10}$$

Once a set of consistent cycles has been obtained, the next step of the approach consists in minimizing the error of the complete graph. Following the same idea of Translational and Rotational error distribution of the local process a global error minimization is computed taking into account the results obtained by pairwise registration and the consistency constraints of all the partial cycles. Some nodes of a basic cycle can also be members of other cycles, forming circuits. The global register process has to guarantee a global consistency, not only of the cycles but also of the circuits. The inconsistency of a complete graph can be defined as:

$$inconsistency(G, B) = \sum_{j=1}^{n} \sum_{i=1}^{m(j)} (E_j^i)^2 \tag{2.11}$$

Where G represents the spanning tree of the graph and B the basis cycles. The inconsistency value decreases in each minimization process. Therefore, a graph can be considered consistent when its inconsistency value is equal to 0.

The main characteristic that differentiates this method from previous ones is the use of information from already visited regions, detecting cycles. This procedure provides the global registration process with valuable information that can be used in order to spread the error through all views. The error is spread proportionally using the weights related to the residue obtained in the local registration process. Since only pairwise registration is required, the method becomes very fast. However the method is not reliable when partial results present misalignments, since the global process only takes into account the constraints imposed by local registration. Hence, the views are not analyzed and registered again, only rotation and translation constraints among cycles are taken into account. Therefore, the success of this method is directly related to the accuracy of the pairwise registration. In addition, all views must be acquired before starting the iteration process, constraining the use in real time applications.

2.4.4 Statistical techniques

The same problem of registering 3D views in a sequence has been also faced by means of probabilistic approaches, especially in mobile robot navigation [80]. The efforts of this community have been centered in developing new efficient techniques for safety navigation, treating the problem of Simultaneous Localization and Mapping (SLAM) where both the vehicle/camera pose and the structure of the environment are estimated at the same time. Recently, techniques have been focused in using probabilistic methods, in which the main difference from analytic multiview techniques is that the uncertainty in the measure is not neglected. Hence, two main groups of techniques have been considered depending on the way of representing such uncertainty: a) Gaussian filters and b) non-parametric filters, which are discussed in the following paragraphs.

Gaussian Filters

Both Kalman Filter (KF) for linear systems and Extended Kalman Filter (EKF) for non-linear systems are undoubtedly the most well-known Gaussian filters for treating the SLAM problem, where the belief is represented by a Gaussian distribution. This section focuses in Extended Kalman Filter rather than Kalman filter since robot and camera movements are usually non-linear. The main goal of EKF is the estimation of the current state of a dynamic system by using data provided by the sensor measurements. Extended Kalman Filter is a recursive system, that is, it only uses the information of the previous step and the actual measurements in order to estimate the current state and update the system. Whenever a landmark is observed by the on-board sensors of the robot, the system determines whether it has been already registered and updates the filter. In addition, when a part of the scene is revisited, all the gathered information from past observations is used by the system to reduce uncertainty in the whole mapping, strategy known as closing-the-loop.

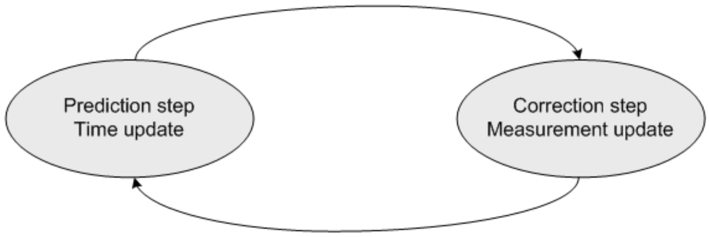


Figure 2.17: Extended Kalman Filter iterative process

In EKF-based SLAM approaches, the environment is represented by a stochastic map $M = (\hat{x}, P)$, where $\hat{x}$ is the estimated state vector (mean), containing the location of the camera/vehicle R and the features of the environment $F_1 \cdots F_n$, and P is the estimated error covariance matrix, where all the correlations between the elements of the state vector are defined (see Equations (2.12) and (2.13)). All data is represented in the same reference system. The map M is built incrementally, using the set of measurements z_k obtained by sensors such as cameras or lasers. For each new acquisition, data association process is carried out with the aim of detecting correspondences between the new acquired features and the previously perceived ones.

$$\hat{x} = E[x] = \begin{bmatrix} \hat{x}_R \\ \hat{x}_{F_1} \\ \vdots \\ \hat{x}_{F_n} \end{bmatrix} \tag{2.12}$$

$$P = E[(x - \hat{x})(x - \hat{x})] = \begin{bmatrix} P_R & \cdots & P_{RF_n} \\ \vdots & \ddots & \vdots \\ P_{F_nR} & \cdots & P_{F_n} \end{bmatrix} \tag{2.13}$$

The state x_k and the measurement z_k of the process can be represented by the non-linear stochastic equations (2.14) and (2.15), where w_k and v_k represent the process measurement noise:

$$x_k = f(x_{k-1}, u_k, w_k) \tag{2.14}$$

$$z_k = h(x_k, v_k) \tag{2.15}$$

As mentioned above, Extended Kalman Filter is a recursive filter that consist in two main steps: a) *Prediction step*, which estimates the current state by using the temporal information of previous states; and b) *Update step*, which uses the current information provided by robot on-board sensors to refine prediction. In the prediction step, the prior estimation of the state vector $\hat{x}_{k|k-1}$ and the error covariance $P_{k|k-1}$ are calculated using the following equations:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1}, u_k, 0) \tag{2.16}$$

$$P_{k|k-1} = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{2.17}$$

where $\hat{x}_k$ represents the state vector (the estimation of the state at time k), u_k a driving function, and the parameter equal to 0 represents zero mean process noise ($w_k = 0$). Q_{k-1} represents the process noise covariance matrix, which will reflect how reliable is our system. This parameter might contain the uncertainty of our system's model such as the known odometry errors or the noise concerning camera calibration parameters among others. This will cause the growing of uncertainty at each step of the SLAM process, essential in order to predict the pose of the vehicle or the elements of the environment (see Figure 2.18). Note that the accuracy introducing the uncertainty of the model can be crucial for the success of the process.

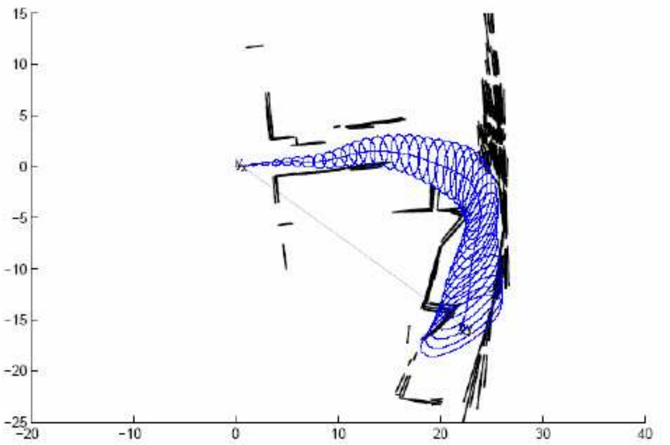


Figure 2.18: Growing uncertainty represented by eclipses.

Since f is a non-linear function, it needs to be linearized before being applied in the covariance equation (see Equation (2.17)). The linear approximation of this function can be calculed using a Jacobian Matrix of partial derivatives . Therefore, A_k and W_k are the Jacobian Matrices at step k and are computed as:

$$A_{k[i,j]} = \left.\frac{\partial f_{[i]}}{\partial x_{[j]}}\right|_{(\hat{x}_{k-1},u_k,0)} \quad ; \quad W_{k[i,j]} = \left.\frac{\partial f_{[i]}}{\partial w_{[j]}}\right|_{(\hat{x}_{k-1},u_k,0)} \tag{2.18}$$

The next step of the process is the update step, where the measurements previously acquired are used in order to obtain a posteriori estimation.

$$K_k = P_{k|k-1}H_k^T(H_k P_{k|k-1} H_K^T + V_k R_k V_k^T)^{-1} \tag{2.19}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1}, 0)) \tag{2.20}$$

$$P_k = (I - K_k H_k)P_{k|k-1} \tag{2.21}$$

where K_k is the Kalman gain that minimizes the error covariance, R_k is the measurement noise covariance (the uncertainty in our measurements), and I is the identity matrix. H_k and V_k are the jacobian matrices of partial derivatives of h respect to x and v at step k:

$$H_{[i,j]} = \left.\frac{\partial h_{[i]}}{\partial x_{[j]}}\right|_{(\hat{x}_k,0)} \quad ; \quad V_{[i,j]} = \left.\frac{\partial h_{[i]}}{\partial v_{[j]}}\right|_{(\hat{x}_k,0)} \tag{2.22}$$

Several methods based on EKF have been proposed during the last years with a variety of applications such as mobile robot localization [81], autonomous airbone navigation [42] or scene reconstruction [18] among others, working in a huge variety of indoor, outdoor and underwater scenarios. Although EKF presents good accuracy in reduced scenes, when it comes to deal with large environments in which tons of data are gathered, Gaussian filters state vectors increase considerably leading to inefficiency in terms of computational cost. Hence, some authors have proposed different techniques such as the Compressed Extended Kalman Filter CEKF to cope with computational cost and memory size [33]. Some techniques try to reduce complexity by doing an accurate selection of the landmarks and discarding the unnecessary ones, decreasing the filter size [49]. Other approaches such as Atlas [12], Graphical SLAM [27] and Hierarchical SLAM [23] use graph based techniques to reduce the complexity. Another interesting problem arises in the presence of data provided by bearing-only sensors such as omnidirectional or unique moving cameras, loading to a new challenge known as Bearing-Only SLAM.

Compressed Extended Kalman Filter

As it has been seen, the use of EKF in large environments requires a high computational cost, due to the huge amount of data that is gathered. Every time a new landmark is observed, it has to be correlated with the already existing ones in the map, establishing a relation between them. In 2001 Guivan proposed an algorithm to reduce this high computational cost by using a compressed version of the EKF, that permits to update only a part of the map at every step [33]. The author introduces the idea of the local map A, as a part of a global map G, where the vehicle is operating for a period of time (see Figure 2.19).

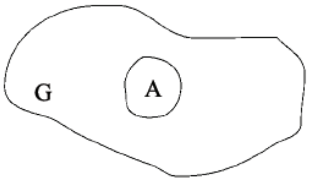


Figure 2.19: Local area A included in a global map G

The main goal of the proposed approach is to reduce the number of features to be treated by the filter, analyzing and updating only the local area of the map A, instead of working with the full map G. Therefore, there is no need to update all the map in each step of the process. Only the data from the local area is added to the filter, which reduces the computational cost of the algorithm to the square of the number of landmarks of the region. The map is divided in a grid were the size of each rectangle is usually determined by the range of the sensor used (see Figure 2.20). There is a central region r where the vehicle is navigating. The area composed by the rectangle r and all its eight neighbors are considered the local area A. This area is perfectly defined and if a landmark that overpasses the limits is observed (i.e when the range of the sensor is larger than the area of the region), it can be immediately discarded. While the vehicle remains in this region, only this part of the map is updated.

Therefore, in the update step, the state vector is divided in two parts, one with the

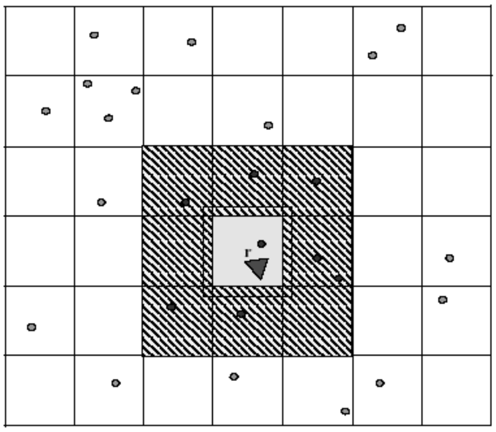


Figure 2.20: Global and local map. The lined area represents the local map A, centered in r.

state of the region A and the other with the state of the full map G:

$$X = \begin{bmatrix} X_A \\ X_G \end{bmatrix}, \qquad \begin{matrix} X_A \in R^{N_A} \\ X_G \in R^{N_G} \end{matrix} \qquad X \in R^N \tag{2.23}$$

were X_A is the state vector of the region A and X_G is the state vector of the global map. Note that the state representing the vehicle/camera pose are also included in the state vector X_A.

During the period that the robot remains in the region r, the state vector will only involve the states on vector X_A, and therefore, measurements of the environment are represented by:

$$h(X) = h(X_A) \tag{2.24}$$

Therefore, in order to linearize h, the partial derivative of h with respect to x has been computed as:

$$H_k = \left.\frac{\partial h}{\partial X}\right|_{(\hat{x}_{k-1},u_k,0)} = \left.\frac{\partial h}{\partial(X_A,X_G)}\right|_{(\hat{x}_{k-1},u_k,0)} = \left[\frac{\partial h}{\partial X_A}\frac{\partial h}{\partial X_G}\right] = \left[\begin{array}{cc} H_a & 0 \end{array}\right] \tag{2.25}$$

where H_a do not depend on X_b Following the Equation (2.26) explained above, the Kalman gain is computed as:

$$K = PH^T S^{-1} = \begin{bmatrix} P_{aa} H_a^T S^{-1} \\ P_{ga} H_a^T S^{-1} \end{bmatrix} = \begin{bmatrix} W_a \\ W_g \end{bmatrix} \tag{2.26}$$

where:

$$P = \begin{bmatrix} P_{aa} & P_{ag} \\ P_{ga} & P_{gg} \end{bmatrix} \tag{2.27}$$

$$S = H_a P_{aa} H_a^T + R \tag{2.28}$$

When the vehicle leaves region r, a new local area is defined and the full map needs to be updated. The relation between the local and global map is defined in the covariance matrix with the terms P_{ag} and P_{ga}, that together with the global map covariance matrix P_{gg} and the local one P_{aa}, will be used when a full map update is required. This process requires the complexity of a normal EKF SLAM process, but it is not need at each step. Note that when the full map is updated, some boundary landmarks associated to a region can move to another local area. Detailed equations of the process can be found in [33].

The Compress Extended Kalman Filter proposed by Guivant has supposed a great improvement in EKF-based approaches, specially in the ones where real-time is critical. The algorithm has reduced the computational time drastically from $\sim O(N^3)\ to\ \sim O(N^2)$ (being N the number of features), where the complexity is totally independent of the size of the global map. In addition, working in local areas allows to increase the frequency of the external sensors, obtaining more data per step increasing the accuracy of the final results without penalizing the global complexity cost.

Constrained Local Submap Filter

William [98], following the idea of the Compresed Extended Kalman Filter, proposed to reduce the complexity of EKF by using local maps linked among them forming a tree structure, presenting the so-called Constrained Local Submap Filter (CLSF) [98]. The local map is composed by a set of landmarks located in the surrounding navigation area of the vehicle/sensor. Each single map has its own reference system and all the landmarks are defined with respect to that reference. A relation between the coordinate system of each submap and the global map is computed. Each time a new local map is created, a new covariance matrix is defined containing only the new local landmarks and the pose of the vehicle/camera, initialized in the origin of the submap with zero uncertainty. In this way, the EKF update step of each submap does not require to deal with a huge amount of data and the partial accumulated error is relatively small. The state vector of the submap might contain not only the relative estimation of all the landmarks but also the global position of some of them and the relationship between the local and the global map (see Figure 2.29). In this way the system will detect duplicate landmarks, finding their correspondences in the global map in a data association process.

At every instant of time, data from local maps are merged with the global map taking into account all the global information and several constraints detailed in [63] to ensure the global consistency. The state vector at state k is defined as:

$$\hat{x}(k) = \begin{bmatrix} {}^{G}\hat{x}_L(k) \\ {}^{G}\hat{x}_1(k) \\ {}^{L}\hat{x}_1(k) \\ {}^{L}\hat{x}_2(k) \\ \vdots \\ {}^{L}\hat{x}_n(k) \end{bmatrix} \tag{2.29}$$

where ${}^{G}\hat{x}_L$ determine the relation between the local and the global reference frame and ${}^{G}\hat{x}_i$ and ${}^{L}\hat{x}_i$ indicate the location of a landmark i with respect to the global frame G and the local frame L. Figure 2.21 shows a detailed scheme of a local and global map.

During the *local update* step, only the local covariance matrix is updated, what is defined as:

$$P(k) = \begin{bmatrix} {}^{G}P(k) & 0 \\ 0 & {}^{L}P(k) \end{bmatrix} \tag{2.30}$$

were ${}^{L}P(k)$ represents the covariance matrix of the local area, that is, the correlation of vehicle/sensor and all the landmarks between each other, and ${}^{G}P(k)$ represents the estimate covariance of the local map and its element respects to the global reference.

$${}^{L}P(k) = \begin{bmatrix} {}^{L}P_{vv}(k) & {}^{L}P_{vm}(k) \\ {}^{L}P_{vm}(k) & {}^{L}P_{mm}(k) \end{bmatrix} \tag{2.31}$$

$${}^{G}P(k) = \begin{bmatrix} {}^{G}P_{LL}(k) & {}^{G}P_{mL}(k) \\ {}^{G}P_{mL}(k) & {}^{G}P_{mm}(k) \end{bmatrix} \tag{2.32}$$

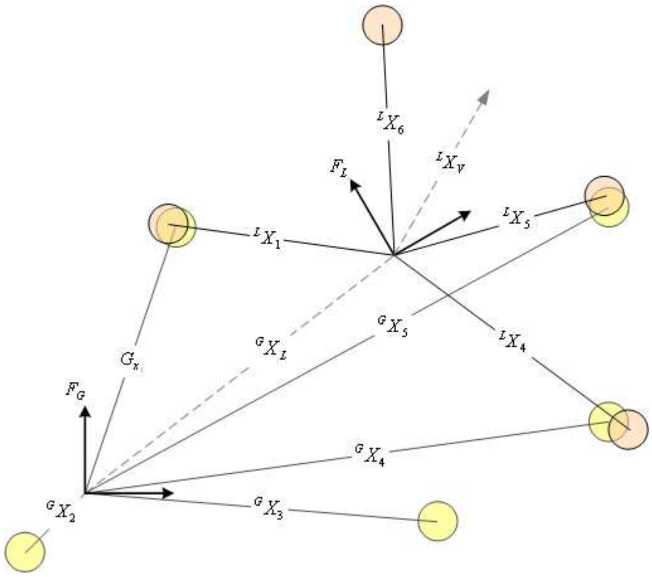


Figure 2.21: Local submap and global map estimation.

where v represents the vehicle or the sensor, and m the local landmarks.

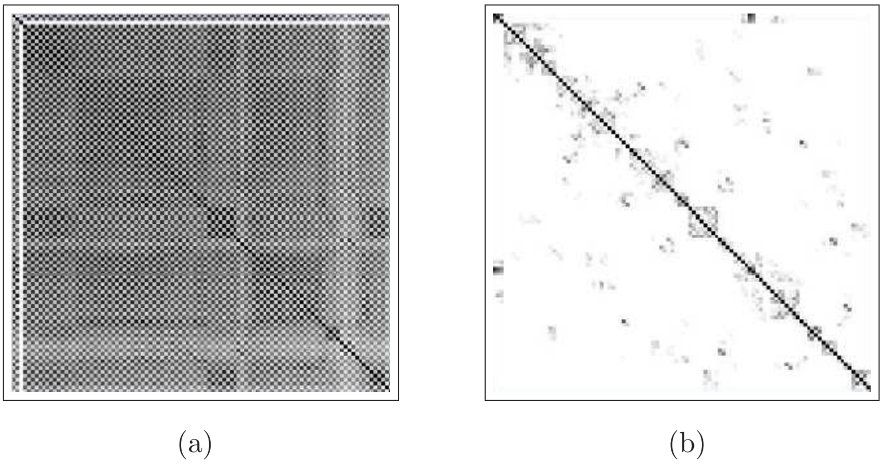
When a global update is required, the process needs to determine the global position of each feature. As can be seen in 2.29, each local map contains some features that are both related to local and global reference frame. Therefore, using this information and the relation between the landmark's local map and the global frame, the constraint detailed in Equation (2.33) must hold on in order to guarantee the stability of the global system and ensure the correctness of the landmark pose prediction.

$$^{G}\hat{x}_i(k) - (^{G}\hat{x}_L(k) \oplus {}^{L}\hat{x}_i(k)) = 0 \tag{2.33}$$

In summary, the CLSF method presented by Williams provides a reliable solution to reduce the high computational cost of EKF when dealing with a huge amount of data. Some improvements have been presented from the Compressed Extended Kalman Filter previously explained. The first one is related to the fact of not using all landmarks for aligning the local and global maps, that implies a considerably reduction of the data association process. In addition, the global constrains added at the global alignment process guarantee a more accuracy in the final map. However, when dealing with huge sensor noise, the method is restricted to small environments due to the lack of closing-the-loop constraints. That is, when the uncertainty of the measurements grows considerably, global constrains might not be enough to assure success in the global estimation process.

Sparse Extended Information Filter

This method was proposed by Thrun [90] with the aim of solving the complexity problems of EKF-based methods working in large environments. The author proposed the use of the extended information filter (EIF), the dual of Extended Kalman Filter. Both filters are based on representing the uncertainty by Gaussian, with the main difference that in the EIF the map estimations are created using pairwise constraints between landmarks, while in EKF the correlation between landmarks are defined in a normalized covariance matrix called correlation matrix. SEIF uses the inverse covariance matrix P^{-1} called information matrix. As it can be seen in Figure 2.24, only neighbor landmarks are linked, whose are represented by the dark regions, and the other elements of the matrix are 0. Therefore, since the majority of the matrix elements are equal to 0, the computational problem can be drastically reduced.



(a)

(b)

Figure 2.22: (a) Correlation Matrix (normalized covariance matrix). (b) Information Matrix (inverse covariance matrix).

The state vector and the measurements in EIF approach are represented in the same way that has been seen in previous EKF-based methods (see Equations (2.14) and (2.15)). The information filter can be represented in terms of the EKF filter, obtaining the covariance matrix $P_k = Y_k^{-1}$ and the estimated state vector $\hat{x}_k = P_k \hat{y}_k$, where Y_k^{-1} is the information matrix and $\hat{y}_k$ the information vector. In order to build the map, the author proposed three iterative steps: the Measurement update, the motion update and the sparsification. In the first step, the relation between the observed landmarks and the vehicle/sensor is defined. As it can be seen in Figure 2.23, when a landmark y_1 is observed by the sensor, a link is created between them in the information matrix Y. Therefore, and since the information matrix is simetric, the links Y_{x_1,y_1} and Y_{y_1,x_1} are defined. Note that diagonal elements always contain links since every observed landmark is related to itself. When a new landmark is observed from the same position of the vehicle, a new link is added between this landmark and the sensor.

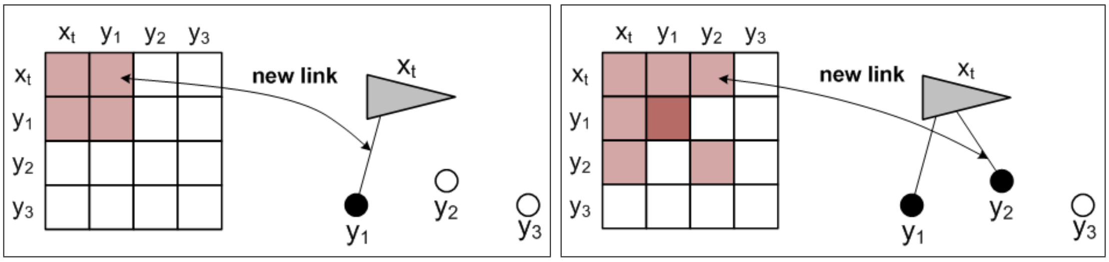


Figure 2.23: Example of information matrix generation. When a vehicle x_t observe a feature y_n, a link between them is generated.

However, links between two landmarks will not be generated until the robot/sensor change its position (motion step). During this second step, new links can be introduced between the features that where indirectly related through the robot position, that is when two landmarks were linked to the vehicle at the same time step (see Figure 2.24). The initial defined links can become less strong due to the uncertainty introduced by the sensor noise. In parallel, other links can be reinforced by using the data obtained by the new measurements.

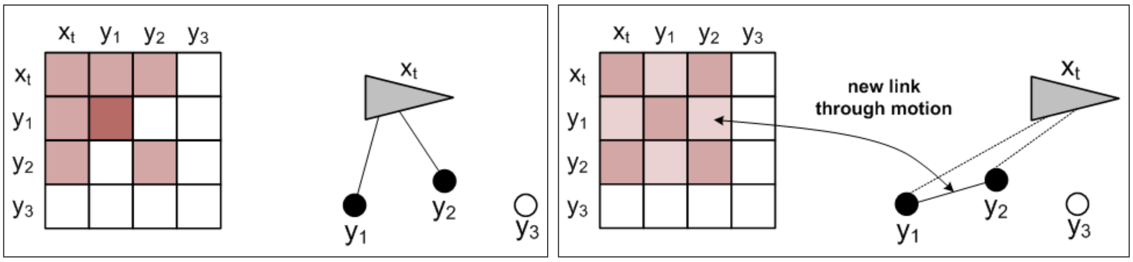


Figure 2.24: Information matrix before vehicle motion (left). Effect after motion (right). A link between features y_1 and y_2 is generated.

Finally, when the number of links increases considerably, exceeding a determinate threshold, some of them are removed in order to guarantee the low computational cost of the system. Therefore, sparsification step consists in removing the less strong and unnecessary links, reinforcing alternatively paths in order to guarantee the stability of the system.

It must be taken into account that every time a new feature is observed, data association process is usually required in order to determine whether it has already seen previously and localize it in the environment. When the position information provided by sensors is not enough, a maximum likelihood estimation is used to determine which feature of the map has more probability to be the new observed one (the best match) (see Equation (2.34)). However, if the best match probability does not exceed a determinate threshold it is considered as a new feature and is added to the map consequently.

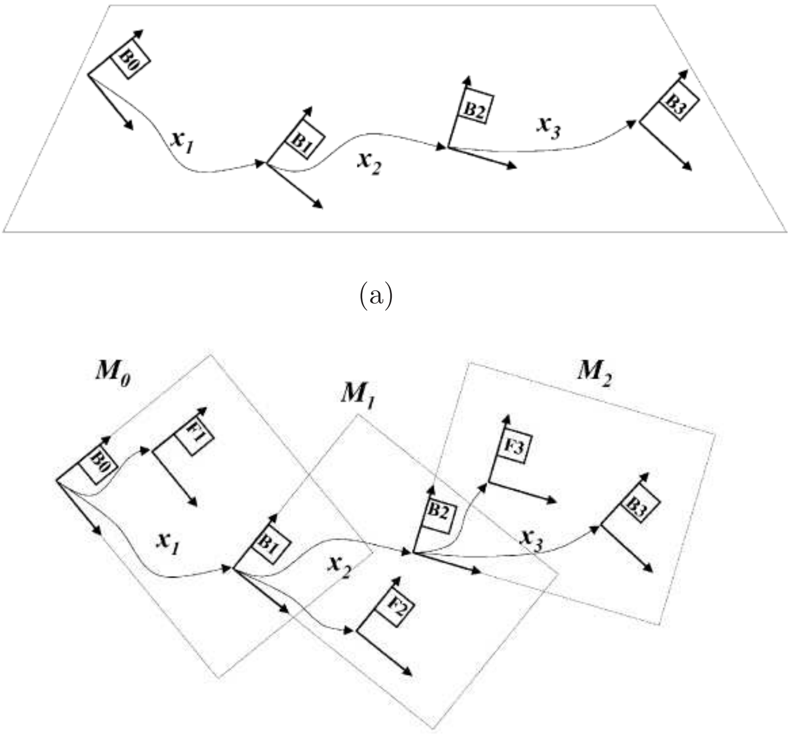
$$n_t^* = \arg\max_n p(n_t|z^t, u^t) \tag{2.34}$$

The presented method provides a reliable solution for mapping large environments. The proposed Information filter has better computational time compared to the EKF-based methods that deal with submaps ($\sim O(N)$ instead of $\sim O(N^2)$). However, when there is a lot of links between features the computational cost increases and some of these links should be removed, which might cause the lost of valuable information deteriorating scene reconstruction. Therefore, since this method offers a good solution for real-time problems such as robot navigation, the sparsification step might not be appropriated for detailed scene reconstruction.

Hierarchical SLAM

In 2005, Estrada presented a new approach focused on SLAM in the presence of large environments. Following the idea of the graph-based methods, the author proposed the creation of independent stochastic local maps with the aim of reducing the computational cost of mapping large environments. In addition, a new "Closing the loop" method was proposed in order to reduce the error propagation and ensure the consistency of the system.

Similarly to [98] two main levels, local and global, are used in order to represent the environment (see Figure 2.25). The local level contains the information of the local areas where features and vehicle are represented with respect to a local reference frame. The global level consists on a graph in which the relative locations between local maps are described. The size of local maps can be determined depending on each situation.



(a)

(b)

Figure 2.25: Hierarchical SLAM model. (a) Global level representation. (b) Local level composed by the local maps M_0, M_1 and M_2.

Each local map $M^B = (\hat{x}^B, P^B)$ is composed by: a) the state vector $\hat{x}^B$, containing the pose of the vehicle R and the features F with respect to the reference system B and

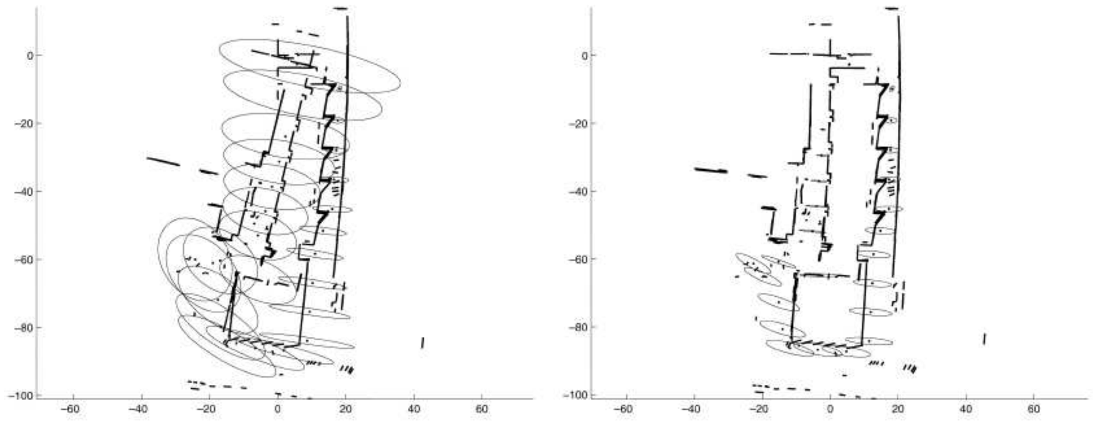
b) the covariance matrix P^B with all the relations between them (see Equations (2.12) and (2.13)). Data association process is carried out using Individual Compatibility Nearest Neighbor (ICCN) to establish correspondences and Joint Compatibility test to ensure the robustness of the matchings. Detailed explanation of these data association techniques can fe found in [60]. In addition, each local map must contain the relation between its reference frame and the one of its neighbors, in order to estimate the global position of the vehicle when it is required. The global map is composed by a graph where nodes correspond to local maps $(M_i, M_j, \ldots)$ and edges to the relative locations between their reference frames $(x_{ij} = x_{B_j}^{B_i})$, represented by an estimation value.

$$\hat{x}_u = \begin{bmatrix} \vdots \\ \hat{x}_{ij} \\ \vdots \end{bmatrix}; \qquad P_u = \begin{bmatrix} \cdot & 0 & 0 \\ 0 & P_{ij} & 0 \\ 0 & 0 & \cdot \end{bmatrix} \tag{2.35}$$

were x_u and P_u represent the state vector and covariance matrix of global stochastic map. As observed in Equation (2.35), local maps are independent and therefore the covariance matrix P only includes diagonal elements.

While the vehicle is navigating, a set of local maps are built and an estimation of the vehicle pose is computed. As seen in Figure 2.26, uncertainty grows considerably as the robot is moving within the environment (the size of the elipses increases what means that uncertainty increases). Therefore, the author proposed to reduce such uncertainty and correct the misalignments by benefiting of the information provided when the vehicle crosses already visited areas. These techniques are known as "Closing the loop". The main idea is to use the information of the robot pose, provided by the relation between the local maps, and the relocation algorithm RS detailed in [61] in order to predict whether an area has already been visited. Once a loop is detected, the next step is to fuse both maps M_i^B and $M_j^{B'}$ that are representing the same area. The author proposes a local map joining technique presented in detain in [87] (see Equation (2.36)).

$$M_{i+j}^B = (\hat{x}_{i+j}^B, P_{i+j}^B) \tag{2.36}$$



(a) (b)

Figure 2.26: Hierarchical SLAM experimentation. (a) Map before closing-the-loop restriction. (b) Map after closing-the-loop.

were $\hat{x}_{j,j}$ and $P_{i,j}$ represent the state vector and the covariance resultant of the fusion of the maps M_i and M_j.

Since the relative reference frames of both maps are known, the main goal of the algorithm is to transform one of the maps and its features into the reference system of the other one. In order to guarantee the stability of the global map, some closing-the-loop constraints are added. Similarly to some previous described analytic methods, one of the main requirements is that the composition of all the transformations $[x_1 \ldots x_n]$ between the local maps in a loop must be equal to zero:

$$h(x) \equiv x_1 \oplus x_2 \oplus \ldots \oplus x_{n-1} \oplus x_n = 0 \tag{2.37}$$

were x_i represent the set of transformations between local maps.

Therefore, the maximum likelihood estimation of the relative relations between local maps is the problem to be minimized. Since $h(x)$ is non-linear, non-linear least square is used in order to minimize the loop constraint function:

$$\begin{gathered} \min_x f(x) = \min_x \tfrac{1}{2}(x - \hat{x}_u)^T P_u^{-1}(x - \hat{x}_u) \\ h(x) = 0 \end{gathered} \tag{2.38}$$

were x is the vector of the relative transformations between local maps and $\hat{x}_u$ the state vector of the global map representing the loop.

In the same manner, the whole map can be refined using the same optimization technique, allowing uncertainty minimization of several loops at the same time. In addition, when the loop area is revisited again, the information is used to refine the pose estimation.

The huge improvement that supposes the use of closing-the-loop constraints is shown in Figure 2.26, where the uncertainty grows considerably in 2.26a while the closing-the-loop constraints forces to minimize such uncertainty in 2.26b. Techniques presented previously only propose global constrains without taken into account "closing the loop" consistency, which can cause misalignments when dealing with large scenes. Therefore, the presented approach provides a reliable solution to the problem of Localization and Mapping in large areas, specially in mobile robot navigation.

Non-Parametric Filters

In mobile robots, Simultaneous Localization and Mapping (SLAM) problem has also been tackled by using non-parametric filters such as the histogram filter or the particle filter (PF). The main difference compared to Gaussian filters is the possibility of dealing with multimodal data distribution, using multiple values (particles) to represent the belief [89]. That is, each estate X_k of the environment can be represented for multiple particles, one for each hypothesis.

$$X_k = \{x^{k,[m]}\}_m = \{x_1^{[m]}, x_2^{[m]}, \dots, x_k^{[m]}\} \tag{2.39}$$

where each particle $x_t^{[m]}$ represents m different hypotheses of the estimation of the vehicle/camera pose at a time step k, represented as:

$$x_k^{[m]} \sim p(x_k | u_k, x_{k-1}^{[m]}) \tag{2.40}$$

were x_k and u_k represent the estimated state of the vehicle and the measurements at time step k.

In comparison with EKF-based filters, PF present more robustness to periods of considerably uncertainty and sensor noise, due to its multi-modal data distribution. However, Gaussian filters usually have a polynomic computational cost, whereas the computational cost of a non-parametric filter may be exponential. During last years, several interesting approaches based on particle filters have been presented as an alternative to EKF-based techniques [44] [57] [84], with the aim of solving the SLAM problem. Stachniss proposes the use of a Rao-Blackwellized particle filter for local map representation, combined with some techniques for particle reduction and a "Closing the loop" strategy [84]. The strongest point of this approach is the possibility of dealing with periods of great uncertainty, due to its ability to recover already vanished hypotheses. This represents a considerable improvement with respect to EKF-based approaches, which do not allow to recover hypotheses that have been already vanished in the past even if these hypotheses were correct.

Alternatively, Montemerlo has proposed a new PF-based approach named FastSlam, which combines the use of particles with Kalman filters for map representation [57]. That is, each particle $x^{k,[m]}$ (composed by all the hypothesis of the robot pose estimation at time state k) has, at the same time, K Kalman filters representing each landmark pose estimation with respect to the vehicle pose.

$$S_t = \{x^{k,[m]}, \hat{x}_1^{[m]}, P_1^{[m]}, \ldots, \hat{x}_K^{[m]}, P_K^{[m]}\} \tag{2.41}$$

were $x^{k,[m]}$ represents all the hypotheses of the robot pose estimation at state k (see Equation (2.39)) and $\hat{x}_K^{[m]}$ and $P_K^{[m]}$ represents the estimate state vector (mean) and the covariance matrix of each landmark with respect to each particle. Note that update process in FastSlam is carried out in the same way as in EKF approaches. In addition, a weight is assigned to each particle depending on its reliability.

This hybrid method has provided reliable solutions to several problems of EKF-based approaches such as the high computational cost that requires to update filters containing considerable amount of data. That is, since the problem is divided into multiple small Kalman Filters containing only Gaussians of two dimensions (for 2D feature location), the computational cost can be reduced to $O(MlogK)$, where M is the number of particles and K the number of landmarks. However, if the complexity of the environment requires the use of 3D data the computational cost increases considerably, forcing the reduction of the number of features at each step, which has a direct effect on the quality of the results.

Bearing-only SLAM

Recently, some approaches have been focused on solving the problem problem that arises in the presence of data provided by bearing-only sensors such as omnidirectional or unique moving cameras. Since depth information is not provided, EKF can not be directly initialized, leading to a new challenge known as Bearing-Only SLAM. An early approach was proposed by Deans [19], who combined Kalman filter and bundle adjustment in filter initialization, obtaining accurate results at the expense of increasing filter complexity. Besides, Davison [18] proposed a top-down Bayesian framework for unique moving camera localization based on a particle filter, which benefits from the initialization of using an A4 piece of paper as a landmark to recover metric information of the scene. Then, whenever a scene landmark is observed a set of depth hypotheses are made along its direction. In subsequent steps, the same landmark is seen from different positions reducing the number of hypotheses and leading to an accurate landmark pose estimation. Recently, Lemaire [46] proposed a 3D Bearing-Only SLAM algorithm based on EKF filters, in which each feature is represented by a sum of Gaussians.

Overall, we conclude that SLAM methods offer accurate solutions for localization and mapping in reduced environments. Since PF methods can handle multiple hypotheses they present more robustness in periods of global uncertainty and sensor noise, but are less efficient than EKF in terms of computational costs. However, when dealing with large environments both methods present problems associated to the increasing uncertainty and the huge amount of data treated. This drawback can be solved by using methods based on building submaps such as CEKF, which present more robustness against uncertainty compared to methods based on a unique global map. Some methods impose global restrictions for global map joining, providing accurate solutions in the presence of short loops. However, loop consistency constraints used in methods such as Hierarchical SLAM can be essential to handle larger loops and prevent inconsistency and misalignments in the final map.

2.5 Conclusions

In this section we have presented a state of the art of the most representative techniques for 6DOF pose estimation and 3D registration of large objects and maps. The most referenced articles over the last few decades have been discussed analyzing their pros and cons as well as their potential applications.

Related to coarse pose estimation, surveyed methods have been classified into two main groups. The first one benefits from sensors or other mechanical devices while the second one focuses on computing the initial pose by solving the matching problem among surface views (computer vision techniques). Although techniques based on mechanical devices such as odometry provide good results in flat terrains, a combination of both methods is usually required in the presence of rough and unstructured environments. Computer vision techniques have been divided in two main groups: *image-to-image* (dealing with 2D views) and *surface-to-surface* (3D views). In this section a new classification of feature extraction techniques has been proposed, published in the "Journal of Electronic Imaging" [78]. The features have been classified as *Feature-to-point*, (corners, lines, etc) and *Point-to-Feature* (Spin image, Point Signature, etc). In our opinion, when dealing with 2D views, SIFT (Scale Invariant Feature Transform) is the feature extraction method that provides more accuracy, since it presents more robustness against rotation and scale changes that other methods such as Harris corner Detector or Line-based ones. However, these last methods are less complex and more adequate for real time applications. When dealing with 3D views of unstructured scenes, were usually a huge amount of points are required, Spin image is the method that has presented more accuracy. The main problem associated to this technique is the computational time required to find a solution, since point descriptors need to be computed. Furthermore, it has been observed that image-to-image (2D) alignment presents good results in the presence of nearly planar areas where depth can be neglected. Otherwise, the alignment produces artifacts ruining the registration. Therefore, surface-to-surface alignment (3D) is more adequate for 3D scenarios, but we have to avoid symmetries in the views to obtain accurate registrations.

Once a coarse pose between two views is estimated, a refinement step can be applied in order to provide a more accurate alignment. Two main methods and their variants have been discussed: *Point-to-point* and *Point-to-plane.* Although the first method is the most commonly used, a huge amount of iterations is required and the method may converge to a local minima. Besides, the *point-to-plane* method has demonstrated to work better in the presence of regions with less overlap and usually converges faster. In addition, *point-to-point* is the most used in *surface-to-model* registration, while *point-to-plane* is the most accurate in *surface-to-surface* registration.

Finally, a consistent classification of multiview registration and mapping techniques has been proposed. The techniques have been classified as Metaview, Simultaneous Minimization, Graph Analysis and Statistic techniques. From our point of view, techniques based on metaview provides good results when dealing with small objects. However, when dealing with larger object/scenes the accumulated error present in the registration process increases considerably leading to poor results. This is mainly due to the lack of flexibility to re-register views already merged in the metaview. Simultaneous minimization approaches present a reliable solution to the problem of error propagation present in multiview registration, avoiding problems such as being stuck in local minima. Moreover, they present robustness against outliers and low-overlapping views. However, the most important drawback of these methods is the high complexity in terms of computational cost, which make them poor candidates for large object registration. Besides, graph-based approaches provide better results that the ones based on metaview for large objects. Although these methods are not robust as the ones based on Simultaneous Minimization, they require less computational time and present better solution for large scene mapping. In addition, it could be observed that include cycle detection and minimization presented more accuracy than the others that do not consider information provided by revisited areas. Although multiview registration methods have demonstrated to provide accurate solutions, misalignments can appear in the presence of featureless environments, symmetries and smooth objects. Note that an accurate treatment of outliers and the removal of the less confident paths in the graph are also compulsory steps to ensure an accurate registration. Finally, statistical techniques present good results reducing error propagation specially in large objects/scenes due to the fact that the uncertainty is considered in the minimization process. These methods are mainly used in robot navigation and are globally known as SLAM approaches. The advantage of statistical methods with respect to analytical ones is their performance in the presence of less reliable sensors, complex environments and unstructured scenes with few features and landmarks. However, they are not recommended for handling tons of data since the manipulation of large state vectors derives to an inefficient computation. Depending on the way of representing the uncertainty of the environment these methods have been divided on Gaussian (Extended Kalman Filter, Information Filter) and non-parametric (Particle Filter). In the first group we would like to highlight methods such as CEKF, SEIF and Hierarchical SLAM which use local maps in order to reduce the computational cost involved in EKF map update. Although some of these techniques use global constraints, only Hierarchical SLAM imposes loop consistency, preventing from misalignments leading to an accurate mapping. Therefore we consider Hierarchical SLAM the most reliable EKF-based method for mapping large environments. Finally, it is interesting to point out that non-parametric methods such as the Fast-SLAM (based on PF) are more suitable than Gaussians when the mobile robot is in phases of global uncertainty, despite the high computational cost. This is due to the possibility of handling multiple hypotheses of the state of the environment. Although SLAM techniques have been usually faced to 2D mapping due to the high computational time required for filter updating (critical for real time operations), the presented methods can also be used in 3D applications. However, they are limited to the use of reduced number of points, which involves less accuracy on the final results.

Chapter 3

Graph Theory

In this chapter some fundamentals on graph theory with specific relevance for the present work are reviewed. Graph theory permits to relate a set of nodes by edges and provides techniques to find paths and cycles and costs throughout the pathway. Graph theory is crucial in the computation of 3D multiview registration since permits to establish relations among partial views (nodes) and their corresponding transformations (edges) in a formal way, leading to more accurate and less computing intensive algorithms. In this chapter we emphasize the importance of some algorithms of moving around graphs but focused on finding paths, cycles and the minimum spanning tree that are important in 3D registration.

3.1 Introduction

Graph theory is widely used in 3D registration techniques since provides an abstract model that permits to relate partial views (nodes) and their corresponding transformations (edges) in a formal way. Some of the contributions in 3D registration surveyed in the previous chapter take advantage of Graph theory and Graph theory is also the cornerstone of the new approaches proposed in this thesis. This chapter is divided into 5 sections. In Section 3.1 the basics of Graph theory is given. Then, Sections 3.2, 3.3 and 3.4 review the most important algorithms we use in 3D registration applications. Finally Section 3.5 presents a summary of conclusions.

Basic Definitions

A graf $G(V, E)$ is an abstract model composed by a set of vertices (nodes) $V = (v_1, v_2..., v_n)$, and a set of edges (lines) $E = (e_1, e_2..., e_n)$, where each edge joins two vertices (see Figure 3.1). A graph is used to represent some elements and their relations. A simple example could be a group of cities and the roads that connect them; in this case the vertices of a graph would depict the cities and each one of the edges can be associated to a road between two of that cities. Besides, in 3D registration, we represent every partial view by vertices and the relation (Euclidean transformation, overlapping rate, registration error, etc) among them by edges.

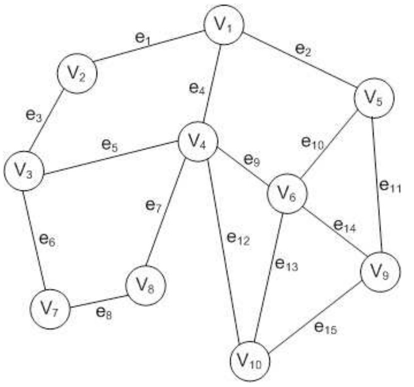


Figure 3.1: The data structure of a graph composed by a set of nodes connected by edges.

Depending on the way the edges of a graph are defined, a graph is classified according to the following criteria:

A *directed graph* is a graph where the edges have a direction, usually represented by an arrow going from a *start* vertex to an *end* vertex (Figure 3.2-b). An *undirected graph* is a graph in which none of the edges have fixed directions so they are considered bidirectional (Figure 3.2-a).

A *weighted graph* is a graph where a numerical value is associated to each edge (Figure 3.2-b,d). This value is known as the weight of the edge and represents the cost of going throughout that edge in a hypothetic path. Should we go back to the example where nodes symbolize cities and edges the roads among them, the distances between cities are represented by the weights of the edges. In 3D registration, where the partial views are represented by nodes, the weight of the edges may represent the Euclidean distance between these views.

A *simple graph* is an undirected and weightless graph that does not contain self-loops (edge that joins a vertex to itself) and where each pair of vertices are related by one edge at most (Figure 3.2-a). A *simple graph* where each pair of vertices is connected by a an edge is called *complete graph* (see Figure 3.2-c). A *pseudograph*, as opposite to the *simple graph*, is a graph where the two previous mentioned features (self-loops and multiple edges between two vertices) are permitted (Figure 3.2-d).

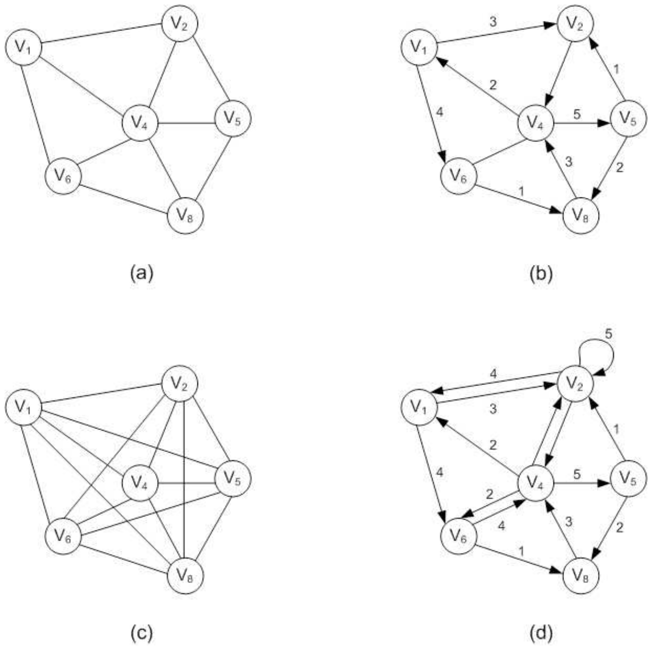


Figure 3.2: Examples of different types of graphs: (a) Simple Graph; (b) Directed weighted graph; (c) Complete graph; (d) Pseudograph.

A *subgraph* $G'(V', E')$ of a graph $G(V, E)$ is a graph where $V' \subseteq V$ and $E' \subseteq E$. That is, if G contains G', then G' is a subgraph of G. In this case, G is said to be a supergraph of G'. Figure 3.3 shows a subgraph of Figure 3.2-a.

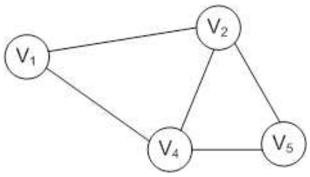


Figure 3.3: Subgraph of the graph from Figure 3.2-a

Having this definitions in mind, we here introduce several concepts that are crucial for our purposes:

A *path* between two vertices is the set of connected nodes that link them. Figure 3.4-a shows the path between v_1 and v_{10}.

A *cycle* is a special *path* where the starting vertex coincides with the ending vertex. In Figure 3.4.b a cycle composed by the vertices $C = (v_1, v_4, v_6, v_5, v_1)$ is depicted. If the path does not contain any repeated vertex then the path is called a *simple path*. Analogously, a cycle with no repeated vertices (apart from the starting vertex) is called a *simple cycle*. A *Hamiltonian cycle* is a cycle that connects all the vertices of a graph by visiting each vertex only once (see Figure 3.4-c).

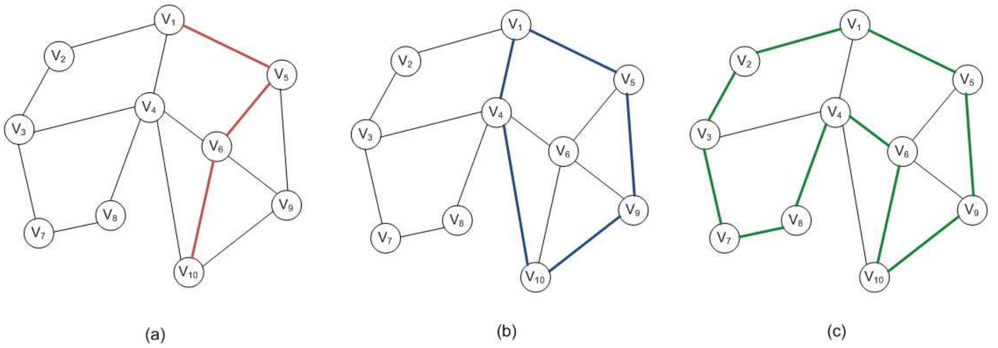


Figure 3.4: Examples of paths and cycles: (a) A simple path coloured in red; (b) A cycle in blue; and (c) a Hamiltonian cycle in green.

A *tree* is a connected, undirected graph without cycles. When a tree contains all the vertices of a graph, it is called a *spanning tree*. Note that a graph may have several different spanning trees. For example, the red path in Figure 3.5 represents a possible spanning tree of that graph. Considering $G(V, E)$ a graph in which the edges are weighted with non-negative costs, a minimum spanning tree is the tree where the sum of the costs of the edges is minimum. Now that the basics on Graph theory are reviewed, the following sections deal with the algorithms of vertex touring throughout graphs focusing on 3D Registration.

3.2 Traversing graph algorithms

One of the most fundamental problems of graph theory is how to traverse a graph ensuring that every edge and every node is visited. The most common strategies proposed in

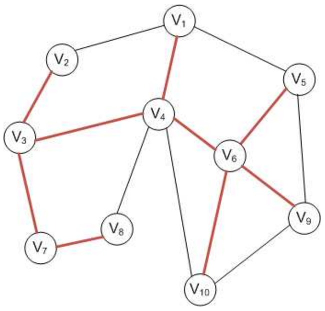


Figure 3.5: Spanning tree.

the literature are the *Breadth first search* (BFS) and the *Depth first search* (DFS). The difference between both algorithms resides in the way of exploring all the possible paths in the graph.

3.2.1 Breadth first search

The *Breadth first search* strategy prioritizes the parallel exploration of all the possible paths from the current vertex. This algorithm begins by choosing a vertex (the root in case of a tree) which is labeled with the 0. The next step consists in exploring all the neighbors of this node (adjacent vertices), which are labeled with the 1. Then, for each of the nodes of this level the algorithm explores their neighboring nodes labeling them with the corresponding label. The process ends when all the vertices have been visited.

Led $V = \{V_i\}$ be the set of vertices of a graph G:

1. Start with all the vertices unlabeled.
2. Select an starting vertex and label it with 0.
3. j = 0 (j represents the level of the visited vertices).
4. From each node labeled j, select all the non-labeled vertices adjacent to it and label them as j + 1. If there is no more unlabeled nodes the process ends.
5. j = j + 1 (increment one level). Go to step 4.

Table 3.1: Breadth first search algorithm.

The result of the *Breadth first search* algorithm is depicted in Figure 3.6, where the vertices have been visited following the path: $P = \{v_1, v_2, v_4, v_5, v_3, v_8, v_6, v_9, v_7, v_{10}\}$

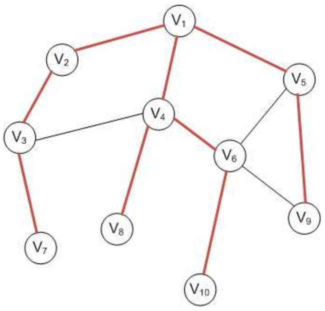


Figure 3.6: Graph traversing using breadth first search algorithm.

3.2.2 Depth first search

The *Depth first search* algorithm prioritizes the exploration of a single path from the selected node before backtracking. The first step of this algorithm is to select a vertex (the root in case of a tree), which will be considered the *current vertex*. Then, the idea is to extend the path to an unvisited vertex neighbor of the *current vertex*, and so on, going as far as possible along the branch, performing a depth search. When no unvisited vertices are found, a backtracking process is carried out and a new alternative path is considered. The algorithm ends when all the vertices have been visited. Notice that some vertices can be visited more than once, so that only the first visit is considered in the path. This process and the resultant tree is shown in Figure 3.7, where the vertices have been visited following the path: $P = \{v_1, v_2, v_3, v_7, v_4, v_8, v_6, v_5, v_9, v_{10}\}$

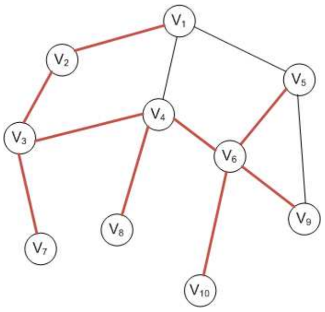


Figure 3.7: Graph traversing using depth first search algorithm.

Led $V = \{V_i\}$ be the set of nodes of a graph G:

1. Start with all the vertices labeled as unvisited
2. Select a unvisited vertex, label it as visited and consider it the current vertex
3. Select a neighbor (adjacent) vertex of the current vertex, mark it as visited and consider it the current vertex
4. If the current vertex has not any unvisited vertex then go back along the path to its parent and consider it again as the current node. If there are unvisited vertices adjacent to the current vertex, repeat steps 3 and 4.
5. If there are unvisited nodes in the graph, go to step2

Table 3.2: Depth first search algorithm.

Considerations

Both $Breadthfirstsearch$ and $Depthfirstsearch$ algorithms provide reliable solutions to the problem of traversing a graph. Besides, these algorithms can be used as well for other problems such as the generation of trees and spanning trees. Although both algorithms deal with the same kind of problems, they have several differences that must be taken into account. That is, depending on the problem we are dealing with, one algorithm may perform better than the other. One of the differences between these algorithms resides in the memory requirements. DFS strategy requires less memory than BSF, since only the path of the current explored branch is recorded. In addition, DFS results very efficient dealing with a searching problem with several solutions where each solution is in a similar depth level of the tree/graph. However, if the searching is along a branch that does not contain any solution, DFS can become very inefficient. Although BFS requires more memory, it usually requires less steps to find the solution (shortest path). For instance, should we have a graph containing a very deep branch and several short branches where the solution of our problem resides in one of the shortest branch, then BFS should be more appropriate than DFS. As it is shown in the following chapters, DFS and BFS algorithms are closely related to several 3D multiview registration techniques. Some related problems such as the order in which the partial views should be registered are determined using DFS and BFS. The graph traversing strategy used in each situation depends on several factors such as the way in which we want to register the views, the sort of object to be registered and the trajectory followed by the camera while the partial views were acquired.

3.3 Minimum spanning trees algorithms

Another important algorithm in 3D registration is the computation of the minimum spanning tree of a graph which is equivalent to determine the best way (minimum cost) of registering a set of views in a graph. Consider $G(V, E)$ a graph where the edges are weighted with non-negative costs, a minimum spanning tree is the tree where the sum of the costs of its edges is minimum (see Figure 3.8). In this section, the two most common algorithms to determine the minimum spanning tree of a graph are discussed: Prim's algorithm and Kruskal's algorithm, whose both are greedy algorithms. Note that greedy algorithms always try to find the best local solution without taking into account how this local solution contributes to the global solution.

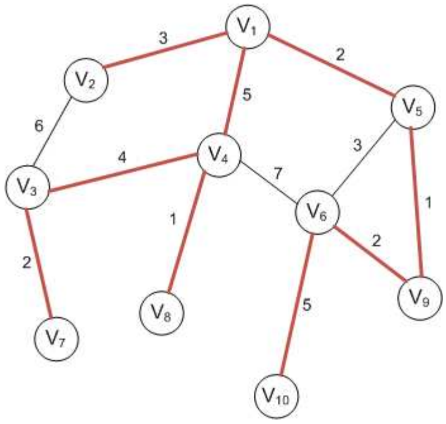


Figure 3.8: Minimum spanning tree

3.3.1 Kruskal's algorithm

Kruskal's algorithm generates a spanning tree by selecting at each step the edge with minimum cost and avoiding the selection of any edge that generates a cycle. When two or more edges have the same cost, the order in which they are added does not alter the final result and therefore can be chosen randomly. The algorithm finishes when all the nodes are connected. In order to make this algorithm clear, the method is depicted in Figure 3.9.

Let $V = \{v_i\}$ be the set of nodes and $E = \{e_i\}$ the edges of a graph $G(V, E)$. Let n be the number of edges:

1. Start with a graph containing only nodes and no edges: $G_t = (V, \phi)$.
   k = 0;
2. Arrange the edges of G in order of increasing cost, generating a list of ordered edges
3. Select the first non-used edge from the list that does not form a cycle and add the edge to G_t.
   k=k+1;
4. If k < n-1 go to step 3. Otherwise, terminate the process, being G_t the minimum spanning tree.

Table 3.3: Kruskal's algorithm.

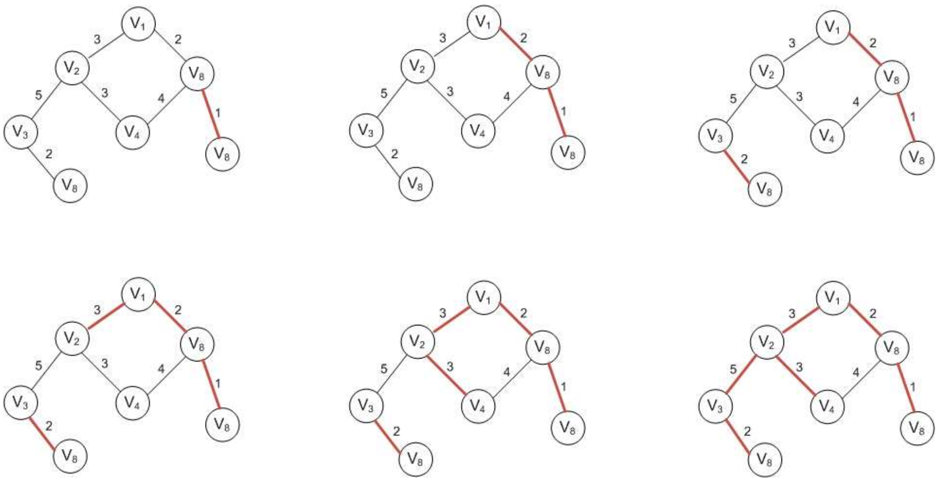


Figure 3.9: Spanning tree generation using Kruskal's algorithm. The process goes from left to right and from top to bottom

3.3.2 Prim's algorithm

Prim's algorithm generates the minimum spanning tree by adding one vertex at each step. The procedure works as follows: A vertex is selected randomly. The edge with the minimum cost connected to that vertex is selected, adding another vertex to the tree. At each step, the edge with the minimum cost that connects any of the visited vertices with an unvisited vertex is selected. The algorithm ends when all the vertices are connected. Figure 3.10 illustrates the process of generating a minimum spanning tree using Prim's algorithm.

Led $V = \{v_i\}$ be the set of nodes and $E = \{e_i\}$ the edges of a graph $G(V, E)$.

1. Select an arbitrary vertex v_i from $G(V, E)$ to start the spanning tree G_t
2. Select the edge with minimum cost between a vertex of G_t and a non-visited vertex (vertex do not included in G_t).
3. Add the new edge and the new visited vertex to G_t
4. If there are still vertices of G that are not in G_t, repeat steps 3 and 4. Otherwise, the process terminates, being G_t the minimum spanning tree.

Table 3.4: Prim's algorithm.

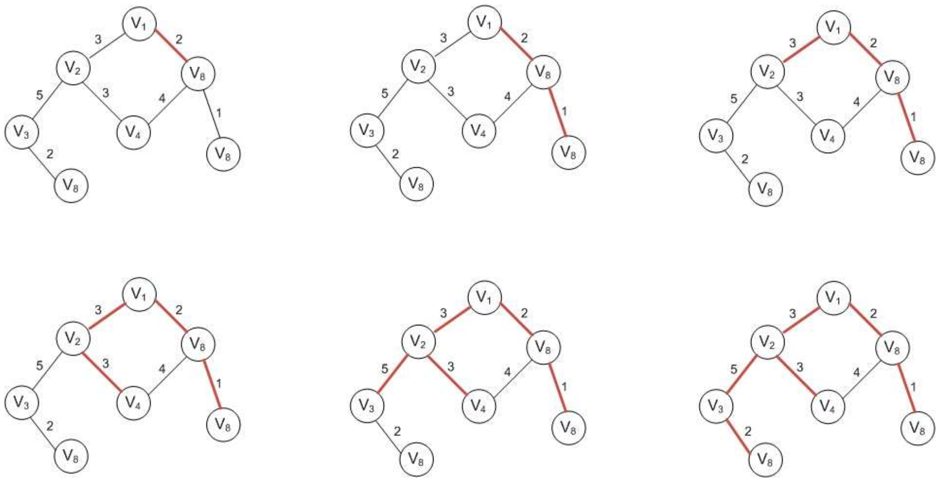


Figure 3.10: Spanning tree generation using Prim's algorithm. The process goes from left to right and from top to bottom

Considerations

Both Kruskal's and Prim's algorithms provide the minimum spanning tree of a graph. Although both algorithms return similar solutions (the same solution if every edge has a different cost), the process followed to deliver this solution differs considerably. Kruskal's algorithm selects a set of vertices (choosing the one with the minimum cost at each step) and then joins them generating the spanning tree. In Prim's algorithm a node is selected and the spanning tree is growing while adding, at each step, the "best" edge (the one with minimum cost) that connects the current node with an unvisited node. Note that Kruskal's algorithm is more complex than Prim's one due to the fact that it has to check at each step whether the added vertex generates a cycle, and discard it if it does. Therefore, the use of

Prim's algorithm is more appropriate when dealing with dense graphs. In 3D multiview registration, the partial views to be registered are the vertices and the edges can be the Euclidean distances between these partial views. It is well known that usually, the shorter the distance between two views is, the larger is the overlapping between these two views and hence the better the registration is. Therefore, finding the minimum spanning tree in 3D registration provides a way of registering all the partial view according to minimum distances. Consequently, it is a method to obtain better global registration results in the presence of multiple views.

3.4 Cycle detection

A *cycle* is a special *path* where the starting vertex coincides with the ending vertex. Concretely, a *fundamental cycle* is a cycle without repeated vertices (apart from the starting vertex) and that does not contain any cycle in it. For example, consider a graph with an 8-shape (see Figure 3.11); this graph contains a *cycle* ($C = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}\}$) composed of two *fundamental cycles* (red and blue in Figure 3.11).

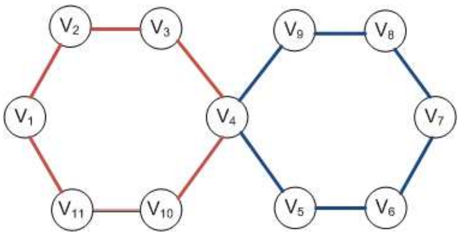


Figure 3.11: An example of a graph with two *fundamental cycles*.

Several algorithms have been proposed in order to determine cycles in a graph. This section focuses on the techniques for detecting fundamental cycles and minimum cycles in a graph. Note that a minimum cycle is the cycle in which the sum of the edge weights that compose the cycle is minimum.

A common way to determine the fundamental cycles in a graph is to adapt the algorithms presented in the last sections. As we have seen, these algorithms discard the vertices that compose cycles when generating the traversing path or the spanning tree. Therefore, several techniques take advantage of this information to detect fundamental cycles.

A widely used method was presented by Paton in 1969 [69]. The author proposes to detect the fundamental cycles of an weightless graph by using a *Breadth first search* strategy. The idea is to start by generating a spanning tree of a graph, while detecting the already visited vertices in the process. Each time a visited vertex is detected, a cycle is generated using both the new edge and some of the edges that compose the tree. The algorithm has been detailed below:

Let $V = \{v_i\}$ be the set of vertex and $E = \{e_i\}$ the set of edges of a graph $G(V, E)$. Let T be the set of vertices of the spanning tree and X the non-examined vertices. Initially, $T = \phi, X = V$.

1. Select an arbitrary vertex v from X to start the tree T. Then $T = \{v\}, X = V$
2. Select the first vertex z included in both T and X. If there are no vertices, the process ends.
3. Consider each vertex w adjacent to z.
4. If $w \subset T$, a fundamental cycle is found. The cycle is composed by the path going from z to w added to the unique path going from w to z using the edges that compose the tree T
5. Otherwise (if T does not contain z), add the edge from z to w and the vertex w to the tree T
6. When all the edges from z have been examined, remove it from X and go to step 2

Table 3.5: Paton's cycle detection algorithm

The process of this algorithm, step by step, can be followed in Figure 3.12. Note that the spanning tree has been represented in red, and the edges that compose a cycle in blue. However, it should be taken into account that the blue edges does not belong to the tree. Therefore, the algorithm has detected two fundamental cycles in this graph: $C1 = \{v_1, v_2, v_5, v_3, v_1\}$ and $C2 = \{v_1, v_3, v_6, v_4, v_1\}$.

Other approaches for detecting fundamental cycles in an non-weighted graph have been proposed using modified versions of *Breadth first search* and *Depth first search* strategies or even a combination of both [21]. Some authors have focused they efforts in detecting *fundamental cycles* in directed graphs. In 1970, Tiernan proposed an algorithm based on exploring *elementary paths*, using a modified version of *Depth first search* algorithm, looking for cycles [91]. This algorithm starts by choosing a vertex randomly and traverse the graph generating an elementary path. When no more unvisited vertices can be found throughout this path, the algorithm backtracks to an already visited vertex of the path

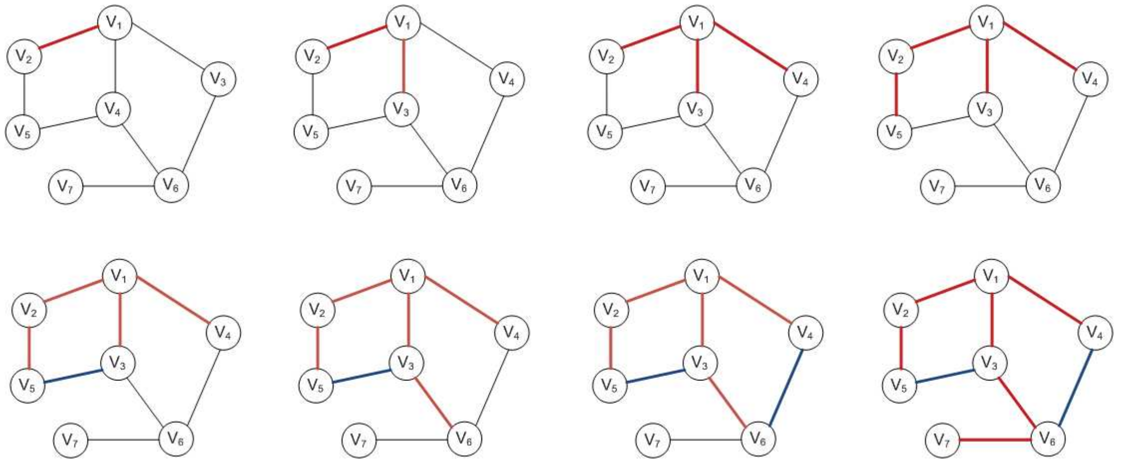


Figure 3.12: Process of detecting fundamental cycles using Paton's algorithm (from left to right and from top to bottom)

and chooses another edge to traverse to a new vertex. If this new vertex is adjacent to the first vertex of the elementary path, a cycle has been found. The process continues until al the vertices have been explored. Other techniques for searching fundamental cycles in a directed graph have been proposed by different authors [97] [88].

Other authors have faced the problem of finding the minimum cycles of a graph, that is to find a set of cycles where the sum of the edge weights that compose the cycles is minimum [71] [36]. A common way to solve this problem is to modify the fundamental cycle detection techniques studied previously. For example, Amaldi [1] proposed a new strategy using the fundamental cycle detection method proposed by Paton [69]. The algorithm starts finding the spanning tree and the fundamental cycles. Then, the idea is to perform edge swaps, removing and adding edges to the tree. Every time an edge is swapped, the cost of the new cycle basis is calculated. The goal is to determine which composition of the tree will give us the minimum cycle of the graph. This process modifies both the spanning tree and the fundamental cycles considerably. Note that minimum spanning tree algorithms can also be used for finding the minimum cycles with slightly modifications.

Considerations

Cycle detection methods take advantage of *graph traversing* and *spanning tree* algorithms. The differences between these methods reside in both the complexity and efficiency of the algorithms as well as in the final results. It is known that methods using *Breadth*

First Search (BFS) strategy (such as the one proposed by Paton [69]), generate shorter fundamental cycles that the ones using *Depth First Search* (DFS) strategy due to the differences in the shape of the obtained spanning trees. This is because shorter branches are obtained when BFS is used. The differences in complexity and efficiency of each algorithm also may depend on the used strategy. It is known that DFS strategy might be less efficient dealing with dense graphs. Consequently, cycle detection methods based on DFS result less efficient than the ones based on BFS. In addition, the way in which the algorithm explores the path for detecting the cycles can be determinant for the algorithm efficiency. For example, some algorithms such as Tiernan [91] explore more paths than needed, slowing the algorithm unnecessarily.

3.5 Conclusions

In this chapter we have reviewed the fundamental concepts of graph theory from basic definitions up to detailed description of the most common algorithms used in 3D registration. The chapter has focused mainly in three main problems: traversing a graph, generating the minimum spanning tree, and detecting cycles. The different problems and the best techniques to solve them have been analyzed, both theoretically and using practical examples.

Two main techniques for traversing graphs have been described: Breadth first search (BFS) and Depth first search (DFS). Although both techniques provide reliable results for searching and traversing graphs, BFS strategy usually provides faster and more optimal results, mainly when the problem has more than a single solution or when the solution resides in a short branch. In addition, in the presence of graphs with large branches, the DFS strategy can become very inefficient or even it can get stuck without returning a solution.

Although using *traversing graph* techniques the different spanning trees of a graph can be obtained, sometimes the minimum spanning tree is required. Here, two main techniques have been analyzed: Kruskal's and Prim's algorithm. As we have seen, both techniques provide similar results (the same if all edges have different values) but using different methodologies. However, Kruskal's algorithm has demonstrated to be less efficient, since it must verify at each step that the new added edge does not compose a cycle.

Finally, the techniques for detecting cycles in a graph have been analyzed and divided into the detection of *fundamental cycles* and *minimum cycles.* These algorithms take advantage of the searching algorithms in order to detect the cycles of a graph. Depending on the searching technique used, the complexity and efficiency of these algorithms can vary considerably. It is well known that techniques based on DFS strategy are less efficient and generates larger fundamental cycles than the ones based in BFS. However, depending on the problem we are trying to solve, some techniques may result more appropriate than others.

To conclude, graph techniques can provide reliable solutions to the problem of registering multiple 3D views, mainly when dealing with large objects or scenes, where the number views to be registered is huge. In the present work, some of these algorithms have been used to determine the relation between these 3D views and to determine the best order in which the views should be registered. Therefore, this chapter has presented a review of the fundamental concepts of graph theory, as well as the most common algorithms adequate to our purposes, with the aim of providing a base for a better understanding of the following chapters. Further details in graph theory can be found in [5] [22] [11].

Chapter 4

Multiview registration based on cycle minimization

An unsolved problem when registering large objects or scenes resides in the propagation error accumulated during the pairwise registration process, leading to inaccuracies in the alignment of the partial views. This chapter presents a new approach to deal with the 3D registration of multiple views thanks to cycle detection and cycle minimization strategies. The technique cope with the propagation error and yet deliver accurate alignments of the set of partial views.

4.1 Introduction

The process of creating a complete model of an object or scene involves several steps, including data acquisition, pairwise registration and multiview minimization (see Figure 4.1). In Chapter 2 we have discussed that pairwise (one-to-one) alignment of views accumulates an error that is propagated throughout all the sequence of views, producing a drift in the registration. This problem worsens in the presence of large scenes, since the number of views to be registered increases. As we have seen in Chapter 2, some techniques have been proposed to deal with multiview registration.

In this chapter we present a new multiview registration technique with the aim of reducing error propagation in the presence of large sequences of views and providing accurate 3D registration models. The new approach takes advantage of graph theory techniques to deal with cycle detection and cycle minimization. Besides, the approach benefits from one-to-one pairwise registration techniques as a technique to compute the correlation among the partial views, determining the most adequate way to register such views. The idea is to detect the existence of cycles and re-visited regions and use this information to reduce the propagation error by means of the Levenverg-Mardquart minimization technique.

Although the main contribution of this work is in the proposal of a new multiview minimization technique, this chapter also describes all the steps involved from the acquisition up to the pairwise registration of partial views, with the aim of providing a better understanding of the whole algorithm.

The chapter is divided as follows. Section 4.2 details briefly the process of acquiring the set of 3D partial views and the method used to perform the pairwise registration between these views. Section 4.3 is focused on the new multiview registration approach describing in the detail all the steps involved in both cycle detection and propagation error minimization. The chapter ends with a summary of conclusions detailing the pros and cons of the new method compared to literature 4.4.

4.2 Pairwise registration

Partial views acquisition

The first step in 3D modeling is to acquire the set of 3D partial views of the object to be registered. These views can be acquired by any Shape-from-X technique such as stereovision, laser triangulation, laser scanning and pattern projection (see Chapter 2). Acquisition differs from the technique used but all of the techniques tend to deliver a cloud of 3D points referenced with respect to the pose of the measuring sensor.

Since the goal of this thesis is in the registration of the views instead of in their acquisition, we have preferred to use already acquired views available in the web[1]. This decision has been also made to facilitate the comparison of the technique since these sets of views are nowadays used by several researchers. Besides, we have also simulate a virtual scanner to acquire synthetic partial views of a virtual object. Synthetic data is extremely

[1] http://www.csse.uwa.edu.au/ ajmal/3Dmodeling.html

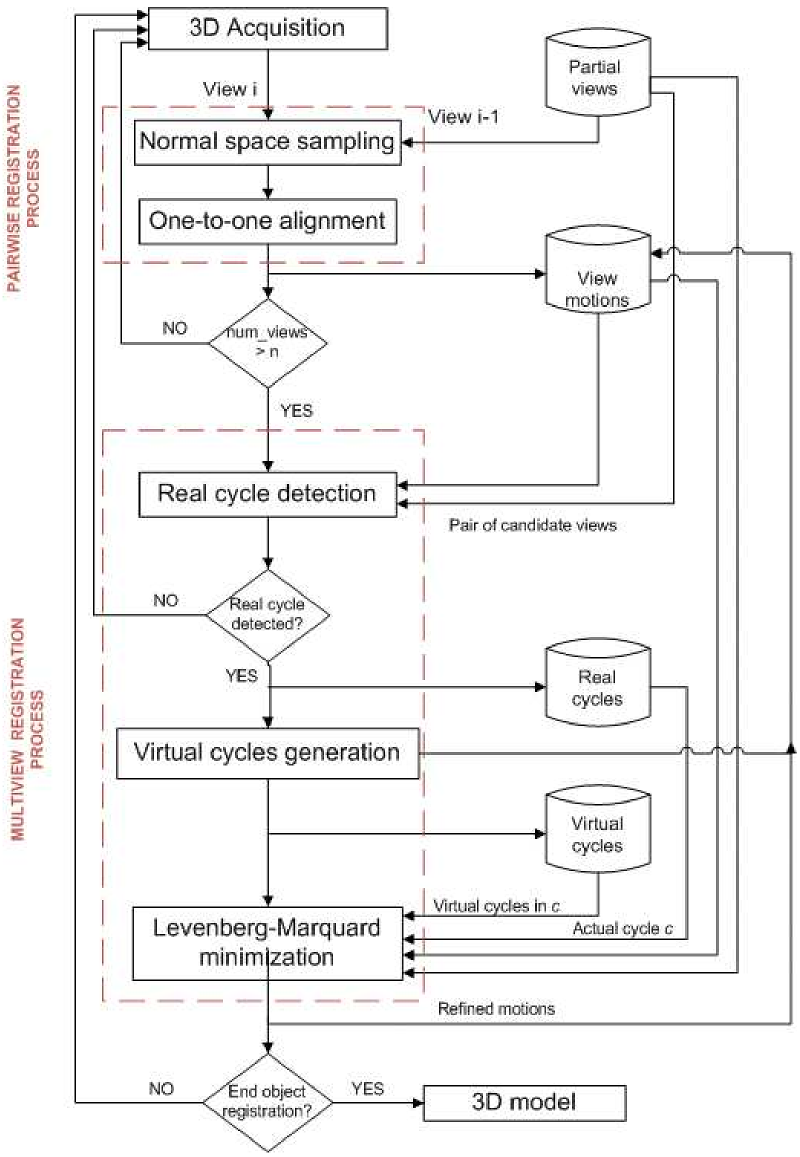


Figure 4.1: Scheme of the multi-view registration process.

important during algorithm development since is the only way to analyze the accuracy of the method compared to ground truth.

Once all the partial views have been acquired, the next step is to align them in order to obtain the final model. In general, a former coarse registration should be used to get an initial estimation of the alignment from some surface features. Then, the coarse registration is considered as an initial solution for a fine registration minimization. Note that, the initial solution is crucial to avoid local minima.

However, in the present work, views have been acquired sequentially both in synthetic and real data (see Chapter 5), ensuring a sufficient overlapping region between them. In this case, the motion between each two consecutive views is very small (a huge amount of surface is shared) and the fine pairwise registration process can be initialized considering that this motion is null. This assumption avoids the expensive computational time required to compute the initial estimations (coarse registration) of the motion, without penalizing the accuracy of the process. However, not in all the real cases this small amount of motion can be assumed. In order to apply this assumption, the motion between consecutive views should be small enough to provide a significant overlapping when possible. In cases where such condition could not be assumed, additional techniques should be previously applied to pre-align the views, such as the mentioned coarse registration. The pairwise registration method used is explained in the following section.

Partial views registration

Pairwise registration consists in aligning the views sequentially while their are acquired. The fast Point-to-plane strategy proposed by Park [68] has been used, though some modifications have been made to increase accuracy and reduce computational time [53].

A common strategy to accelerate pairwise registration is to reduce the number of treated points. The idea is to consider only a sample of significative points from each view without losing surface information. Note also that the removal of despicable points should also increase registration accuracy so that the method use to perform the sampling is extremely important.

Several sampling techniques have been presented in the literature, including uniform sampling [50] [95] and random sampling [52], among others. In 2001, Rusinkiewicz et al. presented a new approach called *Normal space sampling* [77]. The author proposed to discard the points that do not provide useful information by selecting more points from the regions where the normal is different from the other parts of the surface.

In a similar way, a variation of the method proposed by Rusinkiewicz has been used in our registration process. The strategy tries to eliminate redundant information by discarding points from the planar areas and keeping a high percentage of points on the uneven areas of the surface. Detailed information of this approach can be found in [53].

Once the new sets of points have been obtained, the next step is to perform a refined alignment of the views. The method used is a variation of the fast point-to-plane approach proposed by Park [68], from the original Chen's point-to-plane technique [16]. The idea is to compute the intersection between lines and surfaces using a recursive strategy. Initially, the points from one view (*previous view*) are projected orthographically onto the XY plane of the camera. Then, a grid composed of 50x50 square cells is generated and scaled so that it contains the projection of all the points. Next step is to project a point p_0 from the *current view* to the grid and analyze all the projected points from the *previous view* that reside in the same cell, looking for the closest one, obtaining q_{p0}. The projection of point q_{p0} to the normal vector of p_0 defines a new point p_1, which is actually an approximation of the intersection. This approximation is refined recursively by projecting new points p_i until $norm(p_i - q_{pi})$ is smaller than a threshold (see Figure 4.2). Finally, the process is repeated for all the points conforming to the current view and a set of correspondences is obtained.

Then, the motion between these views is computed by minimizing the function:

$$f = \frac{1}{N_p} \sum_{i=1}^{N_p} \|m_i - Rp_i - t\|^2 \tag{4.1}$$

where N_p is the number of correspondences; m_i is the set of points selected in the former view that have a correspondence in the present view; p_i are the correspondences of m_i in the present view; and R and t are the rotation matrix and the translation vector that align both views.

The function of Equation 4.1 is minimized by iteratively refining R and t, using quaternions [10]. The algorithm stops when the mean of the square errors (distances between correspondences) is smaller than a given threshold. See Park [68] for a extended review of this method.

The pairwise registration method here explained aligns two consecutive views delivering the motion between them; a residue or aligning error is always present due to the intrinsic noise of the technique used in point acquisition. In the registration of small sequences of views, this error can be neglected. However, if the number of views to be registered increases, the propagated error may become larger leading to inaccuracies that produce important misalignments. Error propagation can be tackled by using a multiview registration technique. As we have seen in Section 2.4, several techniques have been proposed

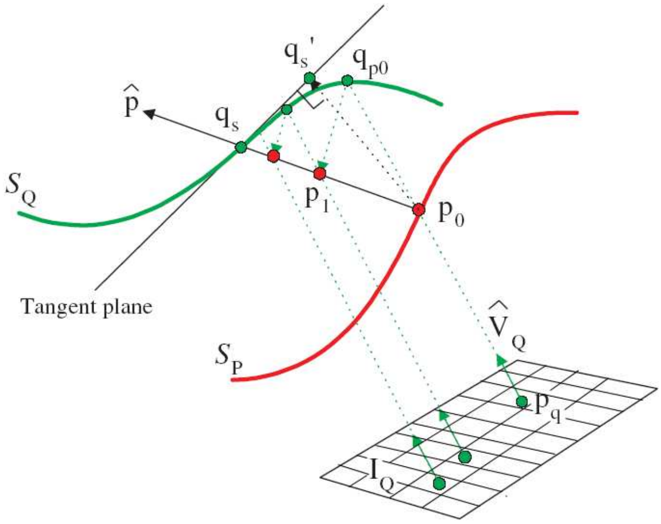


Figure 4.2: Strategy used to compute the intersection between the tangent plane and the surface S_q along the orthogonal vector $\hat{p}$.

that include *metaview*, *simultaneous minimization*, *graph* and *statistic* approaches. Next section proposes a new multiview approach with the aim of dealing with error propagation.

4.3 Multiview registration

In this section a new multiview registration technique is presented. The method used is classified in the so-called graph technique since it takes advantage of the detection of cycles due to revisited regions to minimize the propagated error. The technique is composed of three main parts: *cycle detection*, *virtual cycle generation* and *error minimization.*

The proposed method uses the information provided by the pairwise registration explained in Section 4.2 to correlate the partial views. That is, when two views are registered, the Euclidian transformation between them is obtained. As we have seen in Chapter 3, we can construct a graph using this information, where the vertices are the acquired views and the edges the transformation T between these views ($T = [R\ t]$ where R is the rotation matrix and t the translation vector; see Figure 4.3). Here, the motion between two non-consecutive views can be estimated from the product of all the consecutive motions throughout the path that links that two views. For example, the motion between views V_1 and V_7 depicted in Figure 4.3 is: ${}^1T_2 * {}^2T_3 * {}^3T_4 * {}^4T_5 * {}^5T_6 * {}^6T_7$.

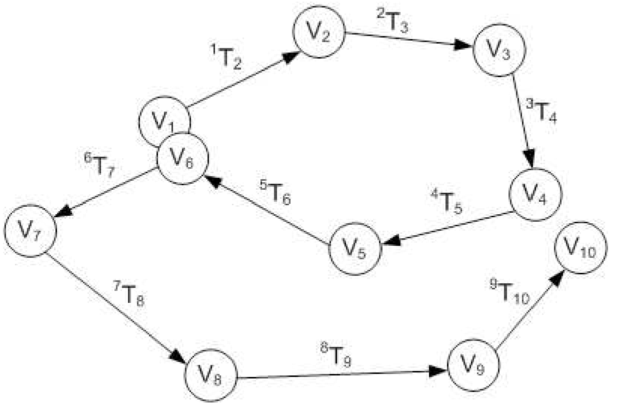


Figure 4.3: Trajectory followed by a camera: The nodes represent the views and the edges the motion (transformation matrix T) between them.

During partial views acquisition, it is possible that some object regions may be visited (acquired) more than once depending on the trajectory followed by the sensor. Therefore, two partial views obtained at different periods of time may belong to the same object region. If the registration process is accurate, the position of the two views should be quite similar. However, since an error is propagated during the process, the final position may vary considerably. For example, the views V_1 and V_6 in Figure 4.3 represent the same region but they are located in a sightly different position due to the propagation error. If we are able to detect that the two non-consecutive views share to the same object region (re-visitation), then we have detected a cycle and we can use this information to recalculate the motion between the views throughout the sequence and obtain a better registration of the object. In addition, the views composing the cycle may share some object regions among them, providing useful information for the minimization process.

However, first two important questions must be answered: (1) How can we detect that an area has been revisited? and (2) How can we use this information to minimize the propagation error in the registration process?. Following sections gives an answer to these questions and detail the three main steps of the new multiview minimization approach: *cycle detection*, *virtual cycle generation* and *cycle error minimization*.

4.3.1 Cycle detection

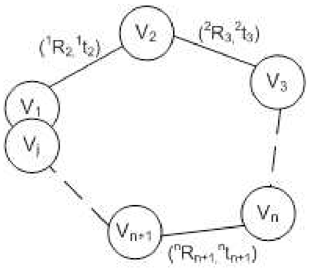
It is agreed that the problem of detecting revisited regions (cycles) of an object/scene is not an easy task. Some authors proposed the use of odometry and inertial systems in order to determine the motion of the acquisition sensor. Others used computer vision techniques to extract image/surface features of the scene to use a sort of correlation.

The cycle detection method we proposed is based on both the transformations between views, provided by the pairwise registration step, and the characteristics of these partial views. Remember that the position of one view with respect to another is the product of the Euclidian transformations throughout the path that links those two views. In the proposed method, two views compose a cycle if the following conditions are accomplished: (1) the accumulated translation between views is close to zero; (2) the views share a common object region; (3) the mean of the normals of both views are similar; and (4) the motion between the two end-views is small.

Translation vector

The basic principle for detecting a cycle is to find two non-consecutive views where the motion between them is close to zero. When an scene is revisited, the motion between the views that compose the cycle can be obtained by the product of all the consecutive transformation matrices throughout the path that links them. In an ideal situation, the translation vector of this motion must be zero. Hence, if the translation is less than a threshold and the views are not consecutive, then we can consider that a potential cycle has been detected. In order to choose the appropriate threshold several considerations have been taken into account: If the threshold is too large, views close to the one that composes the cycle may be considered as candidates before the ideal view is analyzed, slowing down the process or even causing a wrong cycle detection: If the chosen threshold is too small, this lead to the misdetection of potential cycles, mainly at later stages of the process, where the translation error is larger. Therefore, it is important to fix a balance between the amount of cycles detected and their confidence. After several empirical tests a threshold of 17% of the width of the view was fixed, since it has been demonstrated to perform optimally according to our needs (see chapter5). Be aware that we can also add an additional constraint to force the cycle to have a minimum number of views.

This method has an important constraint that must be taken into account: If the



$$t_{i,j} = \sum_{k=i}^{j-1} {}^{\mathrm{i}}R_k * {}^{\mathrm{k}}t_{k+1}$$

Figure 4.4: Graph representing a potential cycle. If $t_{1,j} < threshold$, the cycle is considered.

camera acquires two views from the same position but with a different orientation (different rotation angle), the translation between these two views is zero, but the analyzed region is completely different. Figure 4.5 shows the problem. In conclusion, this condition is necessary but not sufficient and more tests are required.

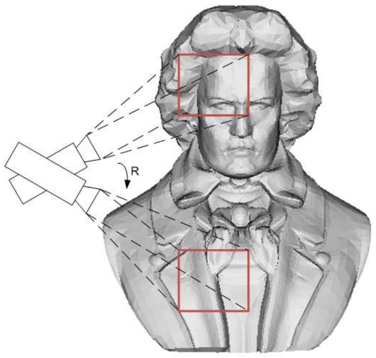


Figure 4.5: Acquisition from the same camera position but different orientation.

Overlapping area

If two views are candidates to compose a cycle, then they must share a common surface. Therefore, once a potential cycle is detected the next step is to verify that the overlapping area between these views is large enough. Otherwise they may not represent the same region.

To compute this overlapping region, the idea is to generate two bounding boxes containing the 3D points of each candidate view. The intersection of the bounding boxes gives an approximation of the overlapping area between them. In order to reduce the complexity of a 3D intersection, the points of each bounding box have been projected onto the three

2D planes (XY, XZ and YZ) computing three overlapping areas (see Figure 4.6). When the maximum of these three overlapping areas is larger than a determined threshold, then it could be considered that the two candidate views form a cycle.

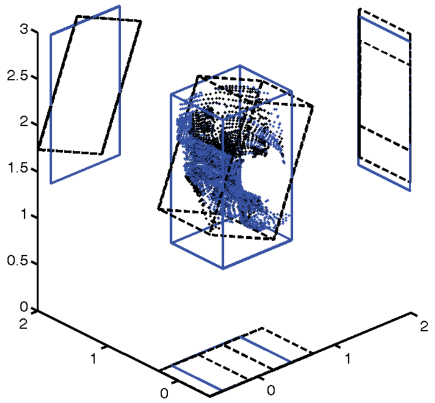


Figure 4.6: Projection of the bounding boxes of two different views in the XY , XZ and YZ planes.

The main problem of this technique resides in the presence of outliers. That is, if one of the views contains an outlier, the size of its bounding box can increase considerably, depending on the position of the outlier (see Figure 4.7). In this case, the intersecting area between two bounding boxes may not reflect the real overlapping area between the two views. As we will see in Chapter 5, using synthetic data this situation is unlikely to happen, because the model does not contain outliers. However, when dealing with real data, this technique should be combined with the others proposed in this section to ensure a correct detection of the cycle.

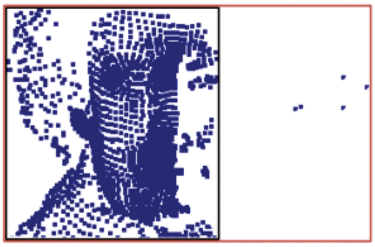


Figure 4.7: Bounding box of set of points (2D representation). In black the ideal bounding box. In red, the bounding box containing the outliers

Another situation that may cause the failure of this method is when the two views are really close in space but represent two completely different regions (for example the front and the back of a thin object). Then, some of the 2D bounding boxes may have a huge overlapping area, but the views are completely different. This problem can be solved by comparing the mean of the normals of the two views.

Mean of surface normals

If two views represent the same region, the mean of their normals should be similar, so that if they differ less than a threshold, the cycle is considered.

This strategy solves the problem mentioned in the previous section in which two views belonging to the front and the back regions of a thin object were considered. Even though they may share a huge overlapping area and the distance between them is quite small, their normals are oriented to a completely opposite direction.

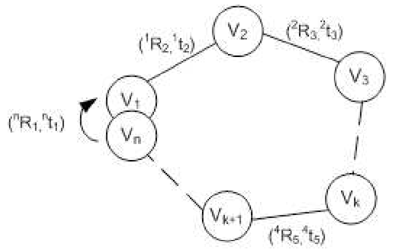
Besides, note that this technique is again necessary but not sufficient since two completely different regions may be similarly oriented and hence may have similar means of normals.

Closing the cycle: Motion between the two end-views

The last step in the cycle detection process is to compute the pairwise between the two end-views that close the cycle, that is the transformation matrix $T = [R\ t]$ that aligns both end views. Therefore we can say that two views compose a cycle if there is almost no motion between them. In order to verify that the rotation between the two views is small enough, the axis-angle representation has been used. Here, the rotation is represented by a unit vector (direction of directed axis) and an angle representing the rotation about the axis. After several tests it has been determined that in order to detect the cycle properly, the angle (from axis-angle representation) between two views should be smaller than 0.025 radians (1.4 degrees).

The pairwise registration also gives the error accumulated during the process when both end views coincide. That is, the *rotation error* is the difference between the rotation and the identity matrices, and the *translation error* is the translation vector (see Figure 4.8).

The main constraint of this step is its high computational cost. Therefore, the previous steps explained in former sections are crucial to ensure that only views with a high probability of composing a cycle are registered.



$$R_{error} = {}^{\mathrm{n}}R_1 - Id$$
$$t_{error} = {}^{\mathrm{n}}t_1 - Id$$

Figure 4.8: Rotation and translation errors between the two views closing the cycle.

Once a cycle has been detected, this information can be used to minimize the error accumulated in the registration process, following the strategy presented in Section 4.3.3. However, with the aim of improving the accuracy of the final result, an intermediate step is proposed: *virtual cycles generation.* A description of the *virtual cycles generation* strategy and the advantages of including it in the multiview minimization process are provided in next section.

4.3.2 Virtual cycles generation

Once a cycle (*real cycle*) has been detected, the cycle is analyzed to detect views within the cycle that share common object regions with the aim of generating *virtual cycles.* The main difference between a *real cycle* and a *virtual cycle* is that the first one is originated as the result of revisiting an area during the acquisition process, while the last one is generated artificially, using information that correlates the views within the cycle.

Until now, views have been acquired and registered sequentially. However, some of the non-sequential views composing the *real cycle* may be close to each other and could be related using several parameters such as the distance between them or the registration error. The proposed method takes advantage of the information that can provide the registration of these neighbor views within the cycle, with the aim of adding new constrains in the minimization process.

The first step for *virtual cycle* generation is to determine which views of the *real cycle* are close to each other. Then, a weighted graph is generated in which views are the nodes of the graph and the edges are the distances between the related views. Finally the fundamental cycles of this graph are detected. The process concerning *virtual cycle generation* includes several steps: *neighbor view detection*, *graph generation* and *fundamental cycle* *detection* steps. This process is detailed in the following subsections.

Neighbor view detection and graph generation

An accurate way of detecting neighbors within a cycle is to register each view with all the others composing the cycle and determine the closest ones depending on the registration results. The views with lower distances between point correspondences are chosen. However, as it has been mentioned in previous sections, this method has a high computational cost. An alternative is to use a variation of the *cycle detection technique*, proposed in Section 4.3.1, adapted to the new purpose. That is, it must be taken into account that two neighbor views do not represent the same region and, in consequence, the thresholds used in the *cycle detection strategy* must be relaxed. Figure 4.9 shows an example of the obtained graph after determining the views that are close to each other within a *real cycle*. The trajectory followed by the camera is showed in black, and the edges that relate neighbor views in red. Note that the cycle is closed by the views V_1 and V_{15}

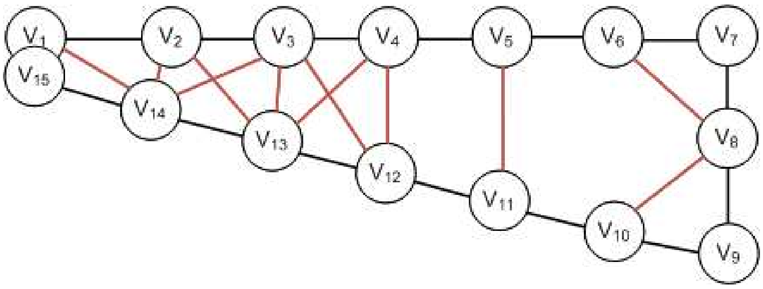


Figure 4.9: Graph representing the correlations between acquired views. Black edges show the sequential path followed by the camera and red edges the neighbor views.

Once the graph relating the closest views is generated, weights could be assigned to each edge. The weights can represent several parameters such as the distances between the neighbor views or the registration error between them. Depending on the characteristics of the object/scene to be registered and the trajectory followed by the camera, one parameter may be more appropriate than the other. For example, if a cycle contains a huge number of views, then the propagation error along the sequence can lead to a poor calculation of the translation between the two views, mainly if one of the views resides in the last part of the cycle. In that case, the use of the registration error between views may offer a better solution. Be aware that the discrepancy between view distances are more significative than the discrepancy between registration errors and therefore, when the situation allows it, it is more recommendable to weight the graph using such distances between views.

The distances between two views can be calculated by computing the norm of the accumulated translation motion between them, as showed in Equation 4.2.

$$e_{i,j} = \|t_{i,j}\| = \| \sum_{k=i}^{j-1} {}^{\mathrm{i}}R_k * {}^{\mathrm{k}}t_{k+1} \| \tag{4.2}$$

were $e_{i,j}$ is the edge that joins the partial views i and j, ${}^{\mathrm{i}}R_k$ is the 3×3 rotation matrix that transforms the views k to the view i, and ${}^{\mathrm{k}}t_{k+1}$ is the translation between views k and $k+1$.

Figure 4.10 shows the weighted graph where edges represent the distances between views. In the chosen example it can be verified that some of the non-consecutive views are really close to each other, sometimes even closer than two consecutive views. Notice that for example views V_2 and V_{14} have a distance of 0.14 (norm of the translation vector between them) while the distance between the consecutive views V_2 and V_3 is 0.54. This demonstrates that the registration of two non-consecutive neighbor views can provide a valuable additional information of the views and contributes to a more accurate error minimization process.

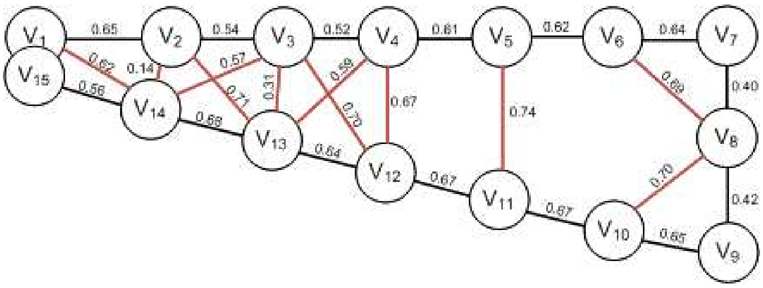


Figure 4.10: Weighted Graph where edges represent the distances between acquired views.

Virtual cycles

Once the weighted graph is obtained, the following step is to detect all the *virtual cycles* within it. Remember that these cycles are considered *virtual* because the path between the views that compose the cycle do not follow the real trajectory of the acquisition system. The idea is to use the correlations between views in order to determine the fundamental cycles of the graph. If the graph is weighted, then a *minimum cycle* can be obtained. As shown in chapter 3, several techniques have been proposed in order to detect the fundamental cycles of a graph [69] [21]. In this work, a variation of the strategy proposed by Paton has been used [69]. First, the minimum spanning tree of the graph is generated using Kruskal's algorithm [22]. Then, the tree is traversed and edges composing fundamental cycles are progressively added.

An example of *virtual cycles* within a graph can be observed in Figure 4.11. Edge weights have not been represented to facilitate figure understanding.

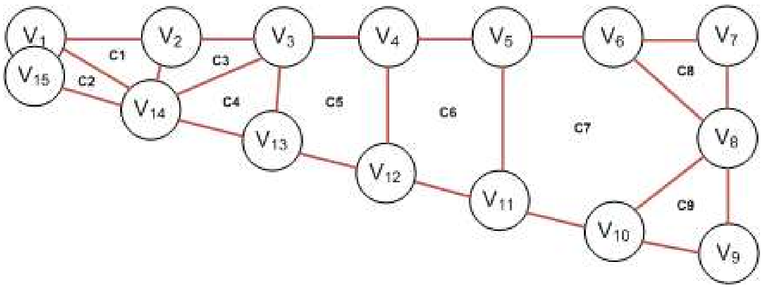


Figure 4.11: Virtual cycles within a real cycle.

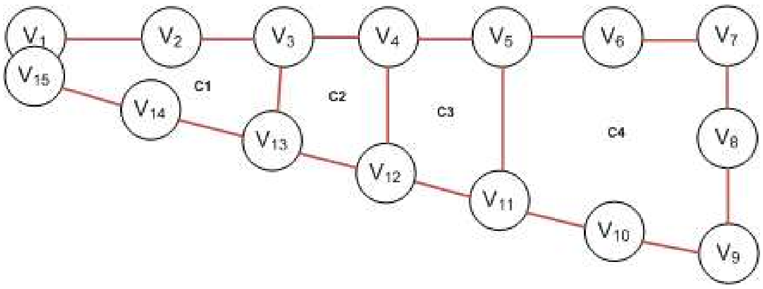


Figure 4.12: New virtual cycles obtained by joining several cycles of the graph depicted in Figure 4.11

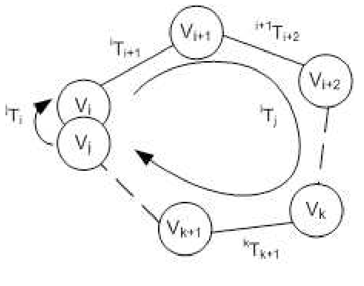
Once the *real* and all the *virtual* cycles have been detected, the next step is to minimize the registration error. As it is described in the following section, each cycle includes a constraint in the minimization process. Hence, the more *virtual cycles* we have, the more information can be used to minimize the error. However, every time a new constraint is added to the minimization process, the efficiency of the algorithm decreases. Therefore, it is important to find a balance between efficiency and accuracy and sometimes the number of *virtual cycles* should be limited. A way to reduce the number of virtual cycles is to detect and merge the cycles that have an edge in common, that is to merge adjacent cycles depending on the desired size (number of nodes) of a cycle (see Figure 4.12).

4.3.3 Cycle error minimization

Once a set of partial views conforming a cycle is detected, the next step consists in cycle minimization, that is the minimization of the propagation error within the cycle. This step tries to minimize simultaneously the distances between correspondence points in all the views that conform the *real cycle* and the *virtual cycles* using a Levenberg-Marquardt approach. Besides, some minimization constraints have been specified, both in *real* and *virtual* cycles.

In the *real cycle* minimization, the minimization constraint used forces that the overall motion in the cycle is null and hence the position of the initial and the final views coincide. Note that as shown in Figure 4.13, the propagated error along the sequence cause a drift, which is pretended to be minimized with such constraint.

Besides, depending on the acquisition system used, the two views closing the cycle may not represent exactly the same region but have slight variations. For example, when using a hand-held sensor, the trajectory followed may not be as precise as when using a sensor coupled to a moving mechanical device. Therefore, with the aim of dealing with this situation, an edge between the last and the first view has been added to the path of the cycle. This edge is the result of the registration between the first and the last view of the cycle. Hence, the overall motion within the cycle is ensured to be null.



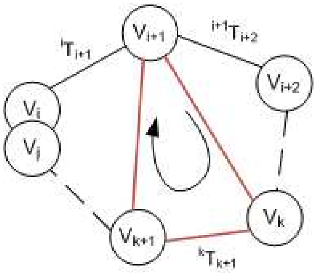
$$
{}^{\mathrm{i}}T_j = \prod_{k=i+1}^{j} {}^{\mathrm{k\text{-}1}}T_k
$$

$$
{}^{\mathrm{i}}T_j * ({}^{\mathrm{j}}T_i)^{-1} = Id
$$

Figure 4.13: *Real cycle* motion constraint.

Note that this constraint states that the motion between the two end-views j and i (${}^{\mathrm{j}}T_i$) should be equal to the motion between these two views throughout the whole path (${}^{\mathrm{i}}T_j = \prod_{k=i+1}^{j} {}^{\mathrm{k\text{-}1}}T_k$). This constraint adds robustness to the minimization process.

In the case of the *virtual cycle* minimization strategy, the cycles have been generated artificially joining neighbor views. In these type of cycles, the initial and the final view is the same. Therefore, there is no need to add an additional edge between the first and the last view. As in the *real cycle* strategy, the overall motion along the cycle is assumed to be null (see Figure 4.14)



$$^{\mathrm{i+1}}T_k * {}^{\mathrm{k}}T_{k+1} * {}^{\mathrm{k+1}}T_{i+1} = Id$$

Figure 4.14: *Virtual cycle* motion constraint. *Virtual cycle* has been represented in red.

Once all the constraints corresponding to the detected cycles have been defined, the algorithm minimizes iteratively the distances between that correspondences. With the aim of reducing the computational cost of the algorithm, the correspondence points used in the minimization are those obtained from the results of the pairwise registration. In the same way, the point correspondences between non-sequential views in *virtual cycles* have been obtained from the pairwise registration performed during the *virtual cycle* generation. A Levenberg-Marquardt minimization is applied to determine the most accurate motion among views in the cycle. The minimizing parameters are the rotation matrices (represented as quaternion vectors) and the translation vectors of the Euclidean transformations between views. The minimization function minimizes the sum of distances between point correspondences. In an ideal situation the residue of the minimization should be zero; in practice the residue depends on the noise intrinsic to the point acquisition system. The function to be minimized, for each *real cycle*, is:

$$min\{\sum_{i=1}^{N-1}\sum_{k=1}^{N_p}(P_i(k) - {}^{\mathrm{i}}T_{i+1} \times P_{i+1}(k) + {}^{\mathrm{j+1}}T_i \times P_i(k) - P_{i+1}(k))\} \tag{4.3}$$

where $P_i(k)$ and $P_j(k)$ are the points that configure the k correspondence between views i and j; N_p is the number of points correspondences; N is the number of views; and

${}^{\mathrm{i}}T_j$ and ${}^{\mathrm{j}}T_i$ are the Euclidean motions (see Fig. 4.13) that transform points from i to j and from j to i, respectively.

Note that equation 4.4 minimizes the distances between correspondences from i to j and from j to i. That is, the path between two views is considered in both senses of the cycle to improve the robustness of the minimization process.

In the case of *virtual cycles*, only one sense of the trajectory is considered. The reason is that a *real cycle* may contain an significative number of *virtual cycles* and the consideration of just one of both senses decreases the computational time considerably. Besides, since *virtual cycles* usually contain a reduced number of views, the minimization of the correspondences in both senses may not apport significant additional information. Therefore, the function to be minimized for each *virtual cycle* is:

$$min\{\sum_{i=1}^{N-1}\sum_{k=1}^{N_p}(P_i(k) - {}^{\mathrm{i}}T_{i+1} \times P_{i+1}(k)\} \tag{4.4}$$

where $P_i(k)$ and $P_j(k)$ are the points that configure the k correspondence between views i and j; N_p is the number of points correspondences; N is the number of views; and ${}^{\mathrm{i}}T_j$ is the Euclidean motion (see Fig. 4.14) that transform points from i to j.

As mentioned above, a motion constraint is added to the minimization process forcing that the product of all the matrices throughout the cycle is null. This constraint takes into account the rotation and translation and it is expressed as follows:

$$\varepsilon_{cr} = \varepsilon_R + s_f \varepsilon_t \tag{4.5}$$

where ε_R is the rotation constraint, ε_t is the translation constraint, and s_f is the scale factor that express the translation in the same range of the rotation parameters.

$$\varepsilon_R = sum(abs(R_{accum} - I_{3x3}))$$
$$\varepsilon_t = \|t\|$$

where R_{accum} is the product of all partial rotation matrices, I_{3x3} is the identity matrix and $\|t\|$ is the norm of the translation vector obtained as a multiplication of all partial motions.

As mentioned before, every time a real cycle is detected, it is minimized simultaneously with the *virtual cycles* within it. However, there is another situation that must be taken into account: the possibility that a *real cycle* contains another *real cycle* within it. For example, a 8-shape trajectory contains a big cycle englobing two small *real cycles* as it is depicted in Figure 4.15.

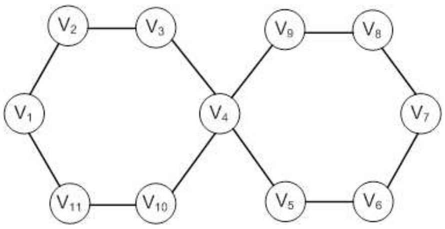


Figure 4.15: 8 Shape trajectory containing a big cycle and two small cycles within it.

To take advantage of the information that can provide such a situation, a new constraint has been added to the minimization process. That is, if a *real cycle* contains another *real cycle* within it, the new cycle is minimized taken into account the constraints of the old one. That is, the error between correspondences in the small cycle should also be minimum. This avoids that the new cycle minimization process causes misalignments in an already minimized cycle.

In addition, it should be mentioned that depending on the object to be registered and the trajectory followed, different weights can be assigned to the different constraints, giving more or less priority to the minimization of every cycle.

4.4 Conclusions

In this chapter, a new multiview error minimization technique has been presented with the aim of minimizing the error accumulated during the pairwise registration process. As shown in chapter 2.4, several multiview registration techniques have been proposed in the literature. The techniques have been classified in: *metaview registration*, *simultaneous minimization*, *graph-based* and *statistical techniques.* The proposed method takes advantage of the graph theory techniques in order to detect path and cycles among the views, minimizing the propagated error by registering the views in an optimal way. Therefore, the technique belongs to the group of multiview registration techniques based on graph.

In contrast to most of existing multiview registration strategies, the proposed approach uses the information provided by revisiting regions in order improve the final alignment. First, a pairwise registration between the consecutive views is performed, with the aim of obtaining an accurate alignment between views. During this alignment process, the method searches for revisited regions taking into account the distances between views, their overlapping surface and the orientation of their normals. Once a revisited region is detected, a graph composing a cycle is generated, where views are nodes and edges the motion between these nodes. Here, the two views belonging to the same region are the two end-views of the cycle, which motion is computed by means of pairwise registration and added to the path to close the cycle. Once a cycle is detected, neighbor views within the cycle are determined and *virtual cycles* are generated. Finally, the accumulated error of the cycle is minimized by using the information provided by both the *real cycles* and the *virtual cycles*. The Levenberg-Marquardt minimization algorithm is used.

The multiview registration techniques presented in chapter 2.4 have several drawbacks that has been solved with the new approach here proposed. One of the main problems of the metaview techniques is the propagation error when dealing with large objects. The main problem of these kind of techniques resides in the lack of flexibility to re-register already merged views. Therefore, when the number of views to be registered increases, the accumulated error causes huge misalignments in the final result. Although this methods provide good local solutions, metaview is unable to deal with the global error propagation problem. Besides, *simultaneous minimization* minimizes all the views together so it is quite inefficient especially when the overlapping between views is limited and the number of views increases drastically. The method proposed solves these problems by detecting revisited regions, detecting and minimizing cycles and always re-registering views with the aim of keeping the overall residue minimal. In this sense, *statistic strategies* follow more or less the same strategy in the so-called ”closing the loop” approach. However, the main drawback of *statistic strategies* compared to the method proposed is the memory requirements needed for the stochastic maps and convergence problems due to uncertainty specially when dealing with large sequence of views and huge amount of data.

The proposed method also presents several advantages in comparison with similar graph based approaches. For example, Huber [39] proposed a new graph-based strategy taking advantage of the minimum spanning tree algorithms to find the best way in which views should be registered, following the path in where the accumulated error is minimum. Apart from the fact that all the views must be acquired previously, this method requires an exhaustive pairwise registration to relate the views and detect the closest neighbors, leading to a high computational cost. In addition, once the error is propagated in a branch of the tree or an erroneous path is followed, the results are totally inaccurate. Comparing to this approach, the method proposed in this thesis presents several improvements. The first one is that it is not required to have all the partial views acquired before starting the registration process. That is, views can be acquired and registered sequentially, allowing to deal with real-time problems. In addition, it is not needed to perform an exhaustive pairwise for detecting neighboring views, since the new cycle detection strategy proposed uses the results of the previous pairwise registration.

The method presented it is more accurate than the cycle minimization method proposed by Sharp [82]. In Sharp a global multiview process refines point correspondences spreading the transformation error (rotation and translation) given by pairwise registration throughout the whole cycle [82]. Views within the cycle are not minimized iteratively with the purpose of keeping a reduced computational cost. Hence, the success of this method is directly related to the accuracy of the pairwise registration. In our approach, point correspondences are iteratively minimized in every cycle so that initial coarse pairwise registration can be refined and refined every time a cycle is detected.

To conclude, the new multiview approach proposed in this chapter provides a reliable solution to the problem of reducing the propagation error produced during pairwise registration especially dealing with large sequences of views. However, it is required that the trajectory followed by the sensor visits two or more times the same region of the object/scene to ensure the detection of cycles. In chapter 5, experimental results show the performance of the new approach in the presence of both synthetic and real data.

Chapter 5

Experimental Results

This chapter presents the experimental results obtained from the testing of the multiview error minimization technique proposed in this thesis. The technique has been compared to some of the most significative methods, which have been implemented and executed with the same data set to evaluate their performance. Experiments have been obtained using both synthetic and real data under quantitative and qualitative evaluation.

5.1 Introduction

The main goal of this chapter is to determine the performance of the proposing technique being compared to some of the most well-known and used multi-view registration techniques present in the literature. Every technique has been programmed and executed with the same data set. In the presence of synthetic data, obtained results are compared to ground truth obtaining a quantitative evaluation of the accuracy of every method. Besides, although experiments with real data might provide a better evaluation of the algorithms, accuracy can only be evaluated qualitatively since in most situations ground truth is not available.

Synthetic results are obtained testing the set of techniques in the presence of an increasing Gaussian noise and comparing registration results to ground truth. In this situation, registration inaccuracies are directly related to the performance of the technique and permit to obtain a quantitative evaluation. Be aware that in order to follow a standard and general camera model, we assume that Gaussian noise may be present in our data. Other noise models could be considered, but Gaussian noise is mostly used in the literature while other types of noise could be applied in not so general cases.

However, synthetic data does not model reality accurately and hence some of the essential characteristics in 3D acquisition such as noise and outliers are not modelled formally. Besides, the trajectory performed by an acquisition system such as a hand-held or a rotating table laser scanner is only approximately known. Hence, we assume ground truth is unavailable and only qualitative results are reported. With the aim of testing the new method in real situations, two different datasets have been used. The first dataset has been obtained from the 3D models used by Mian et.al. in [56] [55]. The second dataset has been acquired in our lab by using a one-shot hand-held scanner especially developed for this purpose.

This chapter provides a detailed description of the experiments carried out and the methodology followed for each type of data. Section 5.2 describes the experiments performed using synthetic data, including a description of the methodology used to simulate a real camera. In section 5.3 real data experiments are presented. Finally, a discussion of the obtained results and a summary of conclusions is detailed in Section 5.4.

5.2 Synthetic Data

To carry out the synthetic experiments, and with the aim of using a common and known comparative dataset, a 3D synthetic model of Beethoven courtesy of INRIA[1] has been used. The complete 3D model was represented by a set of 3D points and the triangles that compose the surface. The next step was to acquire a set of partial views of the 3D object by simulating a real camera. As can be seen in Figure 5.1 the trajectory of the sensor was based on a set of consecutive cycles.

Since one of the objectives is to analyze the performance of the methods working with both small and large objects, two different set of views were acquired. The main difference between these sets was the dimension of the views. In the first set (*set A*), the virtual camera acquired high dimension views while in the second one (*set B*), the size of each view was considerably reduced (see Figure 5.3).

[1] http://www-c.inria.fr/gamma/download/STATUE/index0.php

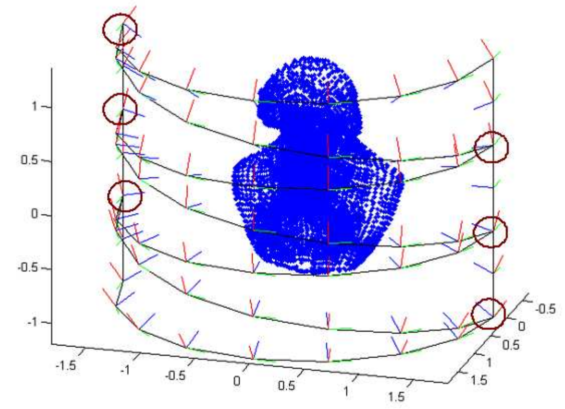


Figure 5.1: 3D partial views acquisition. Trajectory composed by a total of 6 cycles with 17 views each.

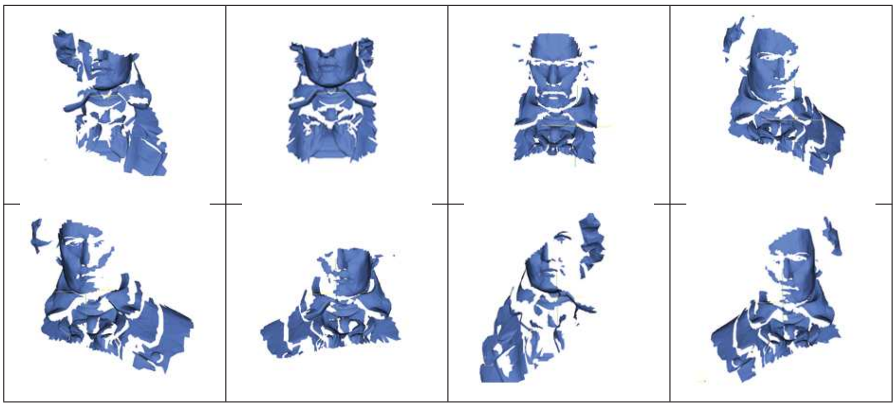
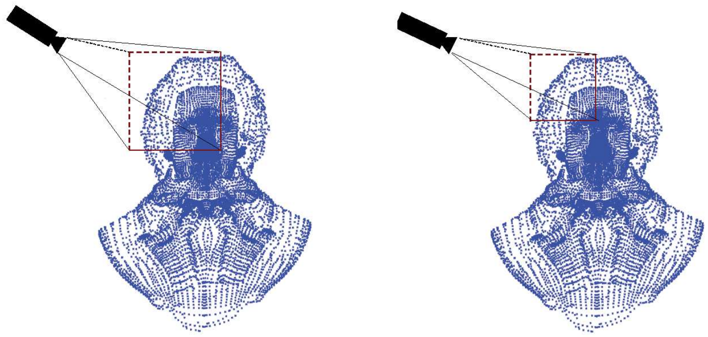


Figure 5.2: Example of several partial views of the beethoven model.

In addition, *dataset B* contains a higher number of views than *dataset A*. That is, the acquisition shots have been performed with a higher frequency to ensure that the consecutive views overlap, even if their size is small.

In *dataset A*, each partial view embrace a large part of the object to be registered. This causes all the partial views to share an important overlapping area among them. In *dataset B*, the views have a small size and hence represent a limited area of the object. As a result, only subsequent views overlap. Therefore, the first set of views share large overlapping while in the second set only some views overlap and the overlapping is quite limited. Hence, *dataset B* permits to analyze the performance of the algorithm in the presence of a drift produced by a propagating error and how the presence of cycles permits to reduce such a drift.



(a)

(b)

Figure 5.3: Camera acquisition simulation. (a) Large view size (0.5X0.5 units) simulating small object acquisition. (b) Small view size (0.3X0.3) simulating large object acquisition.

A virtual scanning was programmed in Matlab so that a virtual camera is moved as required around a measuring virtual object. Only the 3D points observed by the camera were acquired at every camera pose simulating a virtual acquisition. Note that 3D points were transformed to be related with respect to that camera pose with the aim of simulating a real acquisition system. Figure 5.4 shows the differences in the acquisition of the two datasets (A and B). Dataset A contains 6 cycles of 19 views per cycle, while dataset B contains 6 cycles of 35 views per cycle.

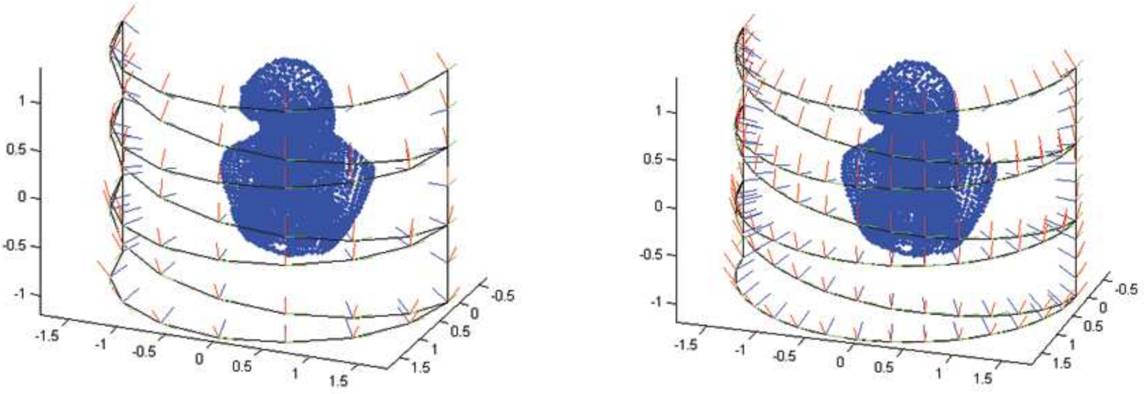


Figure 5.4: Virtual scanning showing camera trajectory and acquisition: Left: *dataset A*. Right: *dataset B*.

The performance of the cycle-detection strategy has been presented in Table 5.1. As can be observed, the cycles are detected optimally when no noise is present in the data.

Table 5.1: Cycle detection strategy performance. Theoretical cycles versus detected cycles.

Dataset	Theoretical cycle	Detected cycle (for each different Gaussian noise)				
		0%	0.5%	1%	1.5%	2%
A	$V_1 - V_{19}$	$V_1 - V_{19}$	$V_1 - V_{19}$	$V_1 - V_{19}$	$V_1 - V_{19}$	$V_1 - V_{19}$
	$V_{11} - V_{29}$	$V_{11} - V_{29}$	$V_{11} - V_{29}$	$V_{11} - V_{29}$	$V_{11} - V_{29}$	$V_{11} - V_{29}$
	$V_{21} - V_{39}$	$V_{21} - V_{39}$	$V_{21} - V_{39}$	$V_{21} - V_{39}$	$V_{21} - V_{39}$	-
	$V_{31} - V_{49}$	$V_{31} - V_{49}$	$V_{31} - V_{49}$	$V_{31} - V_{49}$	-	-
	$V_{41} - V_{59}$	$V_{41} - V_{59}$	$V_{41} - V_{59}$	$V_{41} - V_{59}$	-	-
	$V_{51} - V_{69}$	$V_{51} - V_{69}$	$V_{51} - V_{69}$	-	-	-
B	$V_1 - V_{35}$	$V_1 - V_{35}$	$V_1 - V_{35}$	$V_2 - V_{34}$	$V_2 - V_{34}$	$V_2 - V_{34}$
	$V_{20} - V_{54}$	$V_{20} - V_{54}$	$V_{20} - V_{53}$	$V_{20} - V_{53}$	$V_{20} - V_{52}$	-
	$V_{39} - V_{73}$	$V_{39} - V_{73}$	$V_{39} - V_{73}$	$V_{39} - V_{72}$	$V_{40} - V_{71}$	-
	$V_{58} - V_{92}$	$V_{58} - V_{92}$	$V_{58} - V_{91}$	$V_{58} - V_{92}$	-	-
	$V_{77} - V_{111}$	$V_{77} - V_{111}$	$V_{78} - V_{109}$	-	-	-
	$V_{96} - V_{130}$	$V_{96} - V_{129}$	$V_{96} - V_{128}$	$V_{95} - V_{129}$	-	-

However, as the amount of noise increases, less amount of cycles are detected and some of them are not composed by the optimal views (theoretical cycle). This situation worsens when dealing with cycles that contain a huge amount of views (dataset B). That is, the fact that the cycles of dataset B contain higher number of views (35 views/cycle instead of 19 views/cycle as in dataset A) causes the propagation error to be larger at the moment when the cycle is closed. This have a direct effect on the position of the views, affecting the translation motion between the two candidates (first step in cycle detection algorithm) and the overlapping area between them (second step in cycle detection algorithm), causing the non-optimal selection of the views that compose a cycle. For example, in dataset B, for an error of 0.5%, the algorithm has detected a cycle between views 78 and 109, when the views that theoretically compose this cycle are the 77 and 111. Besides, as the noise increases, the accumulated error also increases and fewer cycles are considered (see column of 2% noise). In some cases cycles are not detected because the algorithm exceeds the fixed thresholds (see Chapter 4 Section 4.3.1 for threshold value). In other cases the algorithm discards the detected cycles that could affect negatively the performance of the registration process (i.e. discarding a cycle between views 6 and 30, when the theoretical is between views 1 and 35).

Experimental results are shown in Table 5.2 and Table 5.3. The proposed method is compared to some of the most representative multiview registration techniques in terms of both translation and rotation accuracies and in the presence of both *dataset A* and *dataset B*. Note that zero-mean Gaussian noise was added to point correspondences and incremented progressively in each experiment (sigma values from $\sigma = 0$ to 0.01 (0% to 2%)). Translation errors were calculated as norm of the discrepancy between the estimated translation given by registration and the ground truth obtained from the synthetic data, as expressed in the following equation.

$$t_{error} = \|t_{est} - t_{real}\| \tag{5.1}$$

In order to calculate the rotation error, all rotation matrices given by the registration were transformed to axis-angle representation. Thus, rotation was represented by a vector (in the direction of the rotation axis) which norm is the rotation angle around this vector. Therefore, the registration error was determined as the norm of the differences between

Table 5.2: Experimental results obtained by the multiview registration methods using the synthetic *dataset A*.

Noise	Method	error t	error R(rad)	MSE
0%	Chen	0.0170	0.0186	0.0058
	Pulli	0.0125	0.0172	0.0049
	Sharp	0.0232	0.0267	0.0063
	Our approach	0.0162	0.0182	0.0056
0.5%	Chen	0.0209	0.0097	0.0094
	Pulli	0.0172	0.0084	0.0069
	Sharp	0.0388	0.0129	0.0255
	Our approach	0.0393	0.0149	0.0153
1%	Chen	0.0285	0.0287	0.0019
	Pulli	0.0279	0.0138	0.0020
	Sharp	0.0874	0.0473	0.0165
	Our approach	0.0553	0.0341	0.0135
1.5%	Chen	0.0646	0.0310	0.0196
	Pulli	0.0529	0.0262	0.0089
	Sharp	0.0902	0.0559	0.0378
	Our approach	0.0748	0.0367	0.0361
2%	Chen	0.1842	0.0401	0.0295
	Pulli	0.0718	0.0296	0.0178
	Sharp	0.1674	0.0673	0.0713
	Our approach	0.1327	0.0435	0.0659

Table 5.3: Experimental results obtained by the multiview registration methods using the synthetic *dataset B*.

Noise	Method	error t	error R(rad)	MSE
	Chen	0.0917	0.0362	0.0513
0%	Pulli	0.0808	0.0232	0.0455
	Sharp	0.0572	0.0298	0.0157
	Our approach	0.0433	0.0164	0.0192
	Chen	0.1354	0.0771	0.0626
0.5%	Pulli	0.1226	0.0609	0.0561
	Sharp	0.0607	0.0543	0.0383
	Our approach	0.0596	0.0277	0.0281
	Chen	0.3516	0.0796	0.0933
1%	Pulli	0.2061	0.0751	0.0623
	Sharp	0.1597	0.0679	0.0687
	Our approach	0.1351	0.0482	0.0324
	Chen	0.5064	0.2952	0.2570
1.5%	Pulli	0.5527	0.1813	0.1580
	Sharp	0.2452	0.0942	0.0937
	Our approach	0.1967	0.0797	0.0766
	Chen	0.6979	0.8532	0.7731
2%	Pulli	0.6534	0.7913	0.7359
	Sharp	0.3411	0.1502	0.0959
	Our approach	0.2854	0.1319	0.0881

these two vectors, as detailed in Equation (5.2).

$$R_{error} = \|\vec{n}_{est} e^{\theta_{est}} - \vec{n}_{real} e^{\theta_{real}}\| \tag{5.2}$$

where $\vec{n}_{est}$, $\vec{n}_{real}$ represent the estimated and ground truth axis of rotation and θ_{est} and θ_{real} are the estimated and ground truth angle of rotation, respectively.

The comparison is completed analyzing the discrepancy between the 3D points aligned by every registration technique with respect to the ground truth given by the synthetic model of the object. This comparison was done by computing the Mean Squared Error (MSE) from the distances between every point of the synthetic object to the closest point in the registered object, as it is shown in the last column of Tables 5.2 and 5.3

The main objective of all these experiments was to validate the proposed multiview technique compared to similar techniques available in the literature. For the comparison, we have considered the widely used Metaview techniques proposed by Chen [16] and Pully [73] and the recent Graph-based technique proposed by Sharp [82].

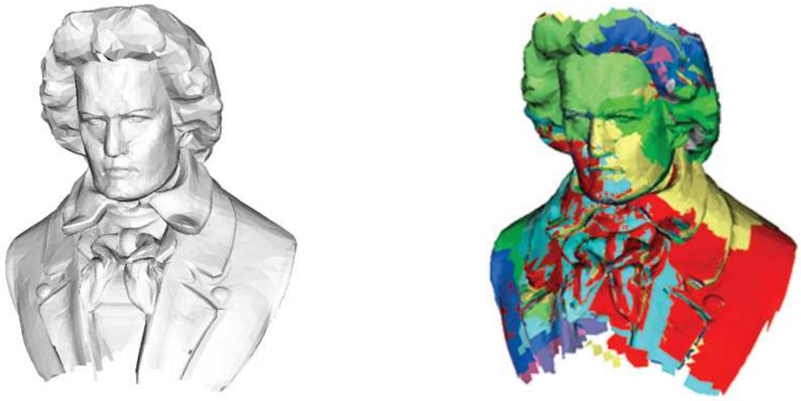


Figure 5.5: Beethoven model: Left: Synthetic model. Right: Registration of a set of partial views.

As it is shown in Table 5.2, Metaview approaches provided good results when dealing with views that presented huge overlapping areas among them (simulating registration of small objects). That is, situations where all the partial views composing the objects are not far apart from each other. Since the success of these techniques is closely related to the results of the pairwise registration (when the overlapping region between views is large), these methods provided accurate results, sometimes even better than those obtained by using methods based on cycle detection. Nevertheless, when the size of the views is reduced (decreasing the overlapping area among views), Metaview approaches provided poor results (see Table 5.3). In this case, the method of Pulli provided better results than the method proposed by Chen, since the former is more flexible and permits to spread the error uniformly along the views. Finally, techniques based on Metaview strategies become less accurate when the Gaussian noise increases (see Table 5.3).

Besides, methods based on graphs and cycle detection techniques were more accurate than Metaview techniques dealing with large objects where the overlapping is quite limited. This is mainly due to the advantage of taking information of all the already registered views in order to proportionally spread the error among the views of the cycle. Moreover, the technique proposed in this thesis considers also the generation of *virtual cycles* within detected cycles obtaining even more constraints in the minimization and yet better results. That is, the residue given by the pairwise registration of views in cycles have been taken into account and it is simultaneously minimized throughout all the views of the cycle and the *virtual cycles* within it. So, our method is more robust against Gaussian noise and limited overlapping areas that provokes poor registration alignments and, hence, it is the one obtaining better results in *dataset B* (see Table 5.3).

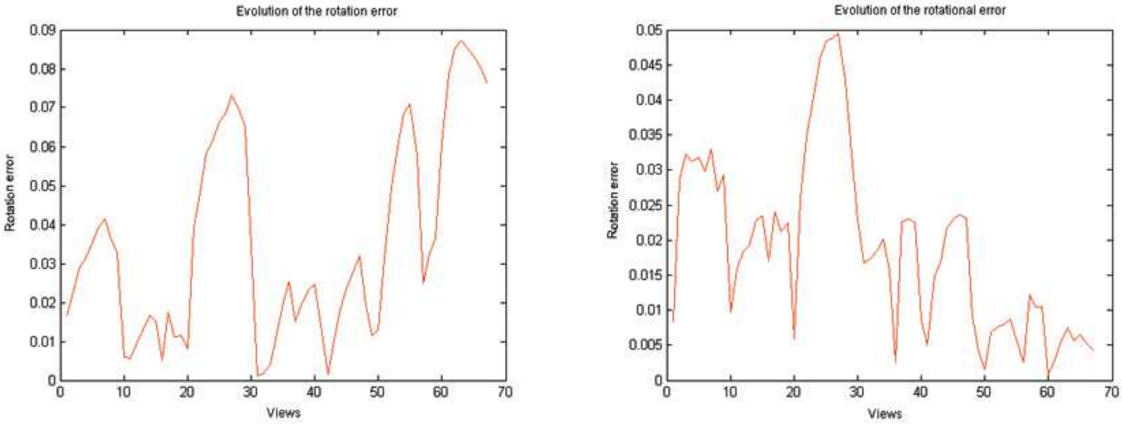


Figure 5.6: Rotation error evolution: Left: Metaview-based method. Right: Cycle error minimization method. Note that these results have been obtained using *dataset B*

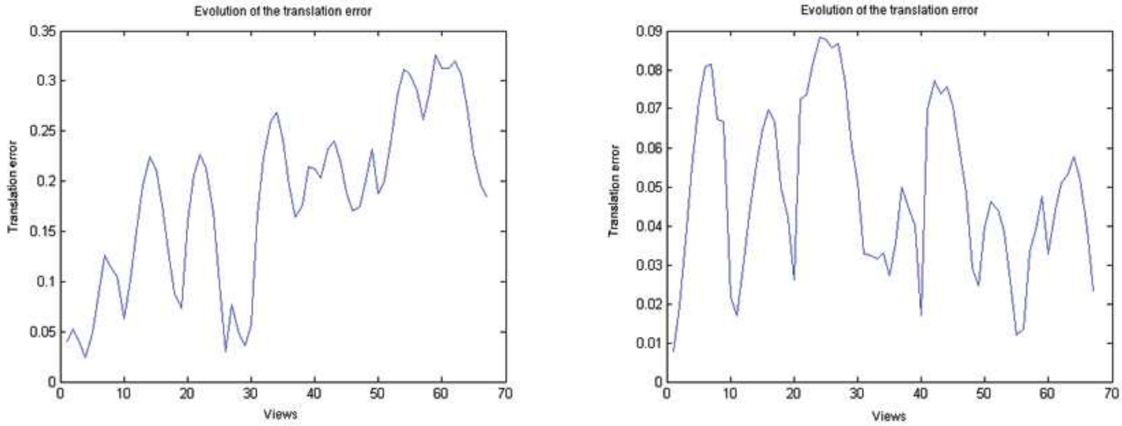


Figure 5.7: Translation error evolution: Left: Metaview-based method. Right: Cycle error minimization method. Note that these results have been obtained using *dataset B*

Finally, the comparison is completed comparing our technique to the method of Sharp. It can be observed that both methods provide accurate results when Gaussian noise is low and, hence, pairwise registration alignments are good. That is, when the propagated error is not very large. However, when Gaussian noise increases so that the pairwise registration alignments are less accurate, Sharp's method distributes the propagated error throughout the cycle, while our technique thanks to the iterative refinement obtains better results, though we need more computing time.

In order to present graphically the problem of the accumulated error throughout the registration process, several graphs showing the evolution of the rotation and translation error per view are presented in Figures 5.6 and 5.7. The observation of these figures demonstrates that Metaview techniques accumulate the error so that increases directly proportional to the number of views. This increment is even worst in the presence of *dataset B*, where views had less overlapping areas (see Figure 5.6). Besides, the methods based on cycle minimization permits to decrease (to nearly 0 in same cases) such propagation error every time a cycle is detected. That's the reason why in large objects where the drift is considerably, cycle minimization techniques provide better results.

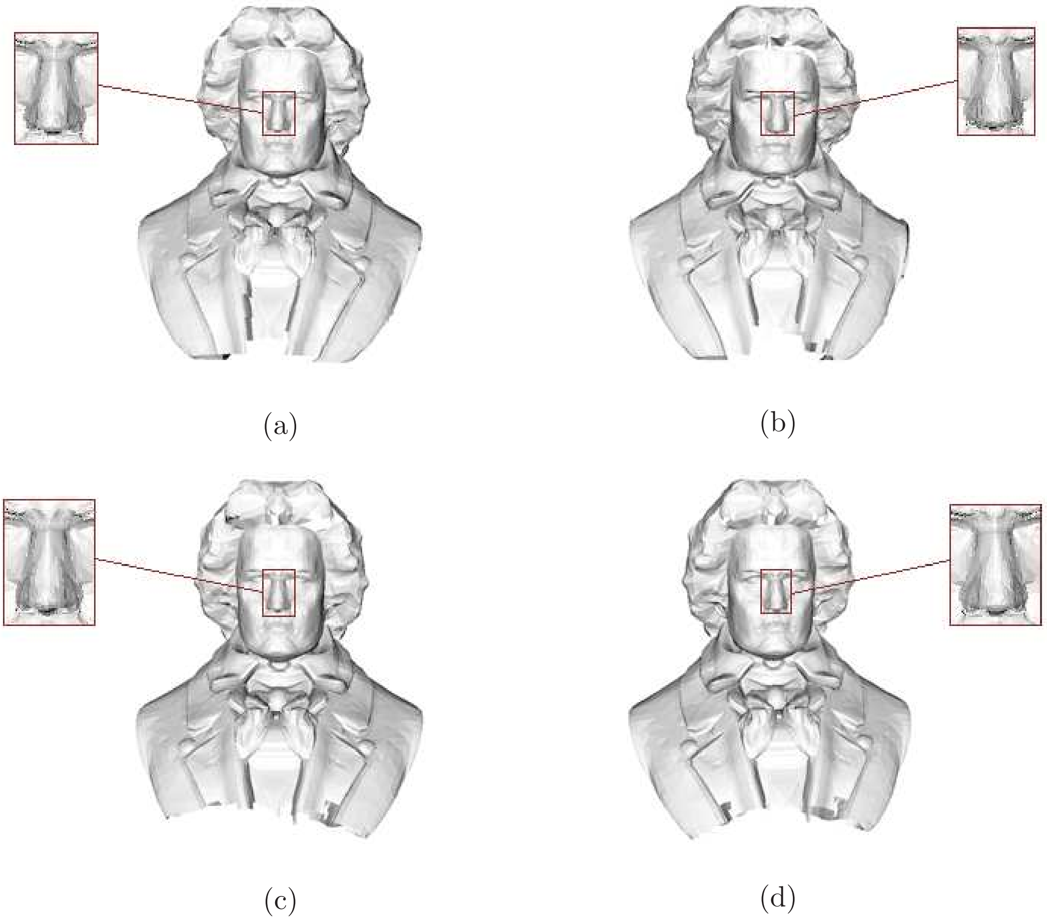


Figure 5.8: Multiview qualitative results using synthetic data (*Set B*): (a) Chen and Medioni's method. (b) Pulli's method. (c) Sharp et al.'s method. (d) Our method.

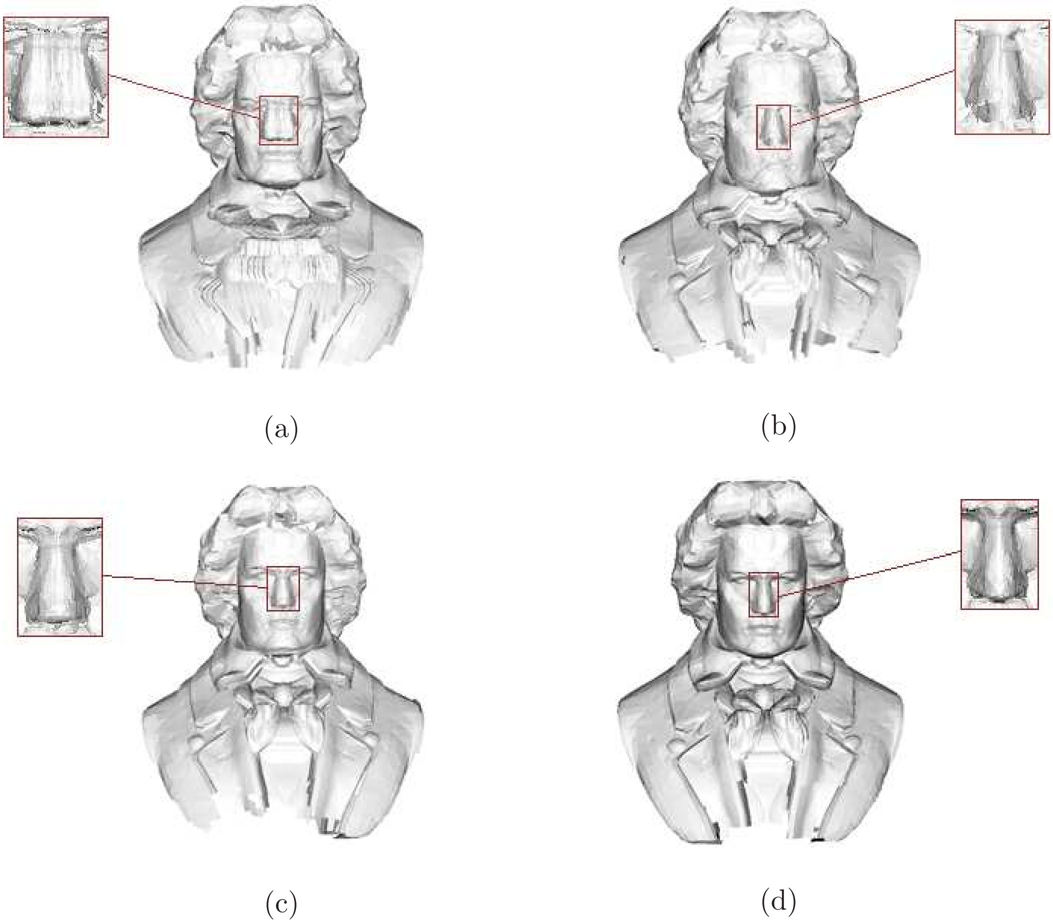


Figure 5.9: Multiview qualitative results using synthetic data (*Set B*): (a) Chen and Medioni's method. (b) Pulli's method. (c) Sharp et al.'s method. (d) Our method.

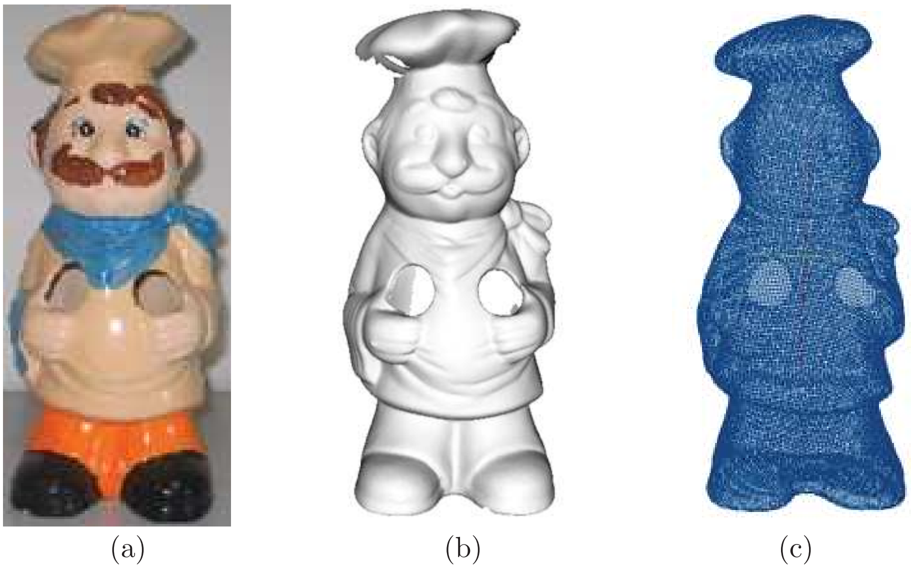
Finally, quantitative results are complemented with qualitative results. Figures 5.8 and 5.9 shows the registered object obtained for every technique implemented by means of *dataset A* and *dataset B*. Looking at Figure 5.8, it can be observed that all the results of *dataset A* are similar. As we have seen before, all methods perform well when dealing with small objects, were the overlaping area among all the views is huge. However, when the sice of the viws is reduced (*dataset B*) the graph-based methods based on cycles present less misalignments than the paraview-based ones (see Figure 5.9). This results can be compared to the synthetic model of the object depicted in figure 5.5

5.3 Real Data

With the aim of testing the same techniques with real data, two new datasets have been used. The first dataset was obtained from the complete collection of scanned objects available at the available at the webpage of the School of Computer Science and Software Engineering, courtesy of Mian et.al. [2]. The second dataset is acquired from a hand-held multi-slit laser scanner developed in the 3D Perception Lab of the University of Girona.

5.3.1 Dataset obtained from known web collections

Although full collections of partial views are provided in the webpage of the School of Computer Science and Software Engineering, courtesy of Mian et.al.[2], a new set of partial views have been generated from the set of 3D points that compose the 3D model with the aim of simulating cycles and limited overlapping but with real data. Figure 5.10 shows the 3D model used. In this model a virtual camera has acquired a set of partial views forming 6 cycles with 35 views per cycle.



(a) (b) (c)

Figure 5.10: Real dataset courtesy of [56]. (a) real object used, in particular an object representing a chef. (b) 3D model of the object without texture. (c) 3D points composing the object.

[2] http://www.csse.uwa.edu.au/ ajmal/3Dmodeling.html

Table 5.4: Experimental results obtained using real data by the multiview registration methods.

Method	error t	error R(rad)	MSE
Chen	0.1761	0.0561	0.0677
Pulli	0.0994	0.0359	0.0591
Sharp	0.0683	0.0274	0.0394
Our approach	0.0557	0.0203	0.0201

Note that, since real data is used, there is no need to add Gaussian noise and outliers artificially. As it is depicted in Figure 5.10-b, the model already contains noisy points and outliers.

Table 5.3 provides quantitative results computed using the same methodology explained in the previous section. Note that, although we are using real data, ground truth is available because partial views have been obtained using the simulated camera and, hence, real positions between views are known. Again, our technique is compared to the Metaview techniques of Pully [73] and Chen [16] and the cycle minimization technique of Sharp [82]. Qualitative results are depicted in Figure 5.11. This results can be compared to the synthetic model of the object in Figure 5.10b.

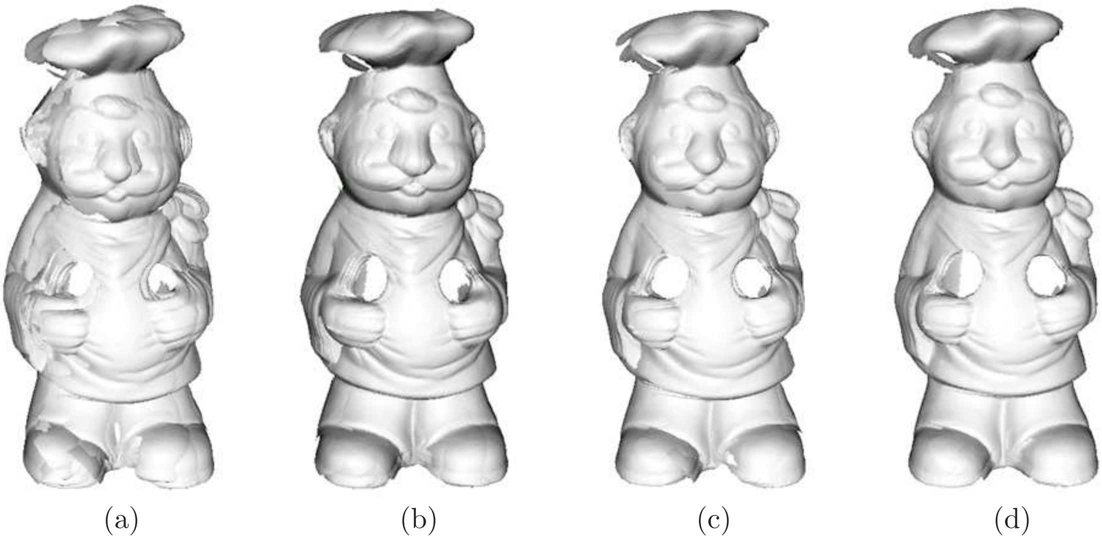


Figure 5.11: Results of the registration of the real data (chef model): a) Chen's proposal; b) Pulli's proposal; c) Sharp's proposal; d) Our proposal.

As can be observed in Table 5.3, the two methods based on graphs provide better results than the ones based on Metaview. Although Sharp's proposal obtained good results at the end of the cycle, the problem of this method resides in the inaccurate distribution of the error along the cycle. To improve this situation, the proposed approach re-registers point correspondence in the minimization process if needed, leading to a more accurate distribution and minimization of the error. Figure 5.11 depicts the differences on accuracy obtained when using both Graph-based methods. This performance is also reported computing the MSE error between both methods that it is shown Table 5.3.

5.3.2 Dataset acquired in our lab

A hand-held multi-slit 3D scanner was developed to take partial views of an object. The sensor was composed by a camera and a 635 nm laser emitter (see Figure 5.12). A complete description of the scanner is available at [53]. Since the scanner was hand-held during the acquisition, the exact pose of the scanner at every partial view was unknown and, therefore, the ground truth for this experiment was not available. Thus, in this section only qualitative results are presented.

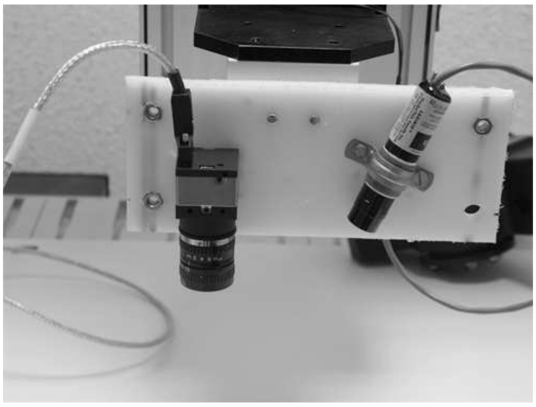


Figure 5.12: 3D hand-held prototype composed by a camera and a 635 nm laser emitter

During the acquisition process, 40 views were acquired from a ceramic object representing the Greek mythology god Zeus (see Figure 5.13.a). Each view was composed by a set of about 4000 points. Here, the trajectory of the scanner formed two cycles describing an 8-shape.

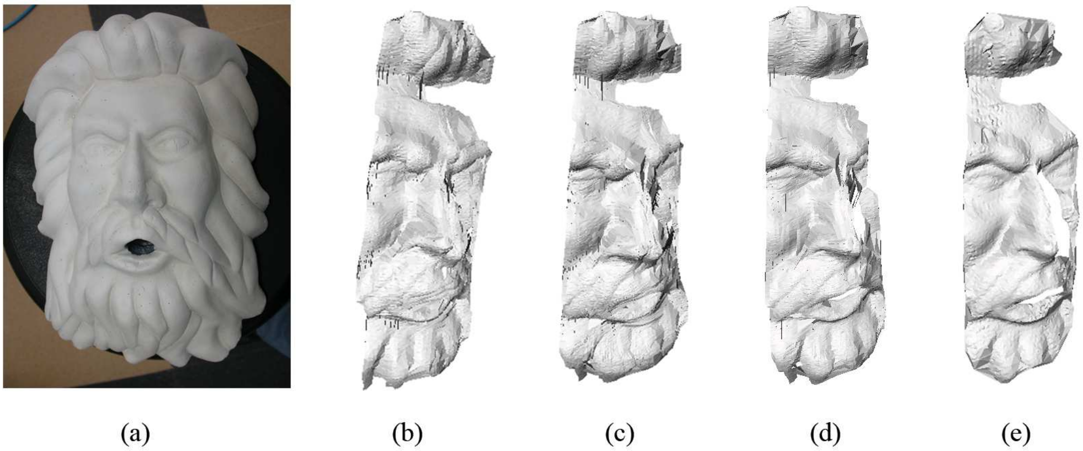


Figure 5.13: Results of the registration of a real object (Zeus object): a) Chen's proposal; b) Pulli's proposal; c) Sharp's proposal; c) Our proposal

Figure 5.13 illustrates the qualitative results. Observe that methods based on Metaview clearly provided worse results than those based on graph with cycle detection. The least accurate result was obtained by the method of Chen, as can be observed in Figure 5.13a. Although Pully improves these results, misalignments in the final model are clearly observed in Figure 5.13b. Both Sharp's technique and the proposed approach present good final alignments. However, a more accurate solution is generated by the proposed approach, as can be observed in Figures 5.13c and 5.13d, demonstrating the robustness against noise and outliers.

5.4 Conclusions

This chapter has presented experimental results that demonstrate the accuracy of our multiview registration technique compared to some of the most used registration techniques in the literature.

The first experiment was obtained from synthetic data so that ground truth was exactly known. In this case, the motions obtained by the multiview registration process were compared to the ground truth motions, providing a reliable way of analyzing the performance of every method. In particular, motion errors (rotation and translation) and MSE have been analyzed. Qualitative results have also been presented. In order to simulate real problems, Gaussian noise is added. Besides, with the aim of analyzing the performance of each method when registering small and large objects, two different sets of synthetic views have been generated. The differences between these two sets reside in the size of the views and the distance between acquisition shots.

As it can be observed looking at both quantitative and qualitative results, all multiview strategies provide good alignments when the Gaussian noise is small and partial views have large overlapping areas. However, when the size of the object increases and more Gaussian noise is added to the partial views, the Metaview-based techniques present severe misalignments. Although Metaview techniques are good solving local problems, they do not provide a good strategy to deal with propagating error due to their greedy approach. Better alignments were obtained by using cycle-based techniques, which provide better strategies to detect and minimize the drift propagated throughout the acquisition path. Despite results obtained with both cycle-based algorithms are very similar when noise is small, our multiview approach provides better results when noise increases thanks to the minimization of cycles. That is, while Sharp's approach only considers the motion between views in the minimization process, our approach considers also the distances between point correspondences and the re-registration of views within a cycle.

Additionally, experiments with real data have been performed with the aim of demonstrating the performance of our method in non-ideal conditions (presence of acquisition noise and outliers). Here, two new datasets have been used. The former were obtained from a collection of partial views available in the web [3], while the second was acquired from a hand-held multi-slit scanner developed in our lab. In both cases, our technique achieved better results compared to the surveyed techniques.

To conclude, this chapter has demonstrated the rightness of our multiview registration technique in both synthetic and real data, especially when the acquisition trajectory performs cycles that are detected and minimized by the technique.

[3] http://www.csse.uwa.edu.au/ ajmal/3Dmodeling.html

Chapter 6

Conclusions

This chapter presents the conclusions of this work which includes a summary and the list of the main contributions. The chapter lists the articles published in journals and international and national conferences during the progressing of this thesis. Finally, further work and new research lines that lie open are discussed.

6.1 Conclusions

This thesis is focused on the 3D registration of large-scale objects and scenes. Chapter 2.1 defined registration as the process of aligning a set of 3D partial views of a given object/scene by determining the Euclidian motion between these views with respect to a reference. The main problem when registering a set of partial views resides in the residue accumulated during the registration process, leading to a propagation error that is quite significant especially with large-scale objects/scenes. This thesis proposed a new multiview registration approach based on graphs and cycle detection strategies to cope with such propagation error.

The thesis surveyed the most representative 3D registration techniques. 3D registration involves three main steps: 1) initial pose estimation; 2) pairwise registration; and 3) multiview registration. The most representative techniques have been analyzed and compared, discussing their pros and cons and their potential applications. An updated and detailed classification of the surveyed techniques is presented. This survey can be considered as a useful tool for any reader interested in 3D registration to determine the best technique available for any given application.

The first step in a registration process is to obtain an initial estimation of the motion between views (coarse registration). As detailed in Section 2.2, the initial pose estimation techniques have been classified in two main groups: 1) initial pose estimation by mechanical devices; and 2) initial pose estimation by computer vision (correspondences matching). The reliability of these techniques depend on each situation. For example, some techniques based on mechanisms such as odometry provide good solutions dealing with planar areas but results are inaccurate in rough and unstructured environments. Computer vision techniques are based on searching and matching of correspondences between views and they have demonstrated to provide accurate solutions. However, the main drawbacks of these techniques are the necessity to get differentiate features between the images and the exhaustive search needed for the matching, which require high computational cost.

Next step in the registration process is the refinement of the initial coarse registration thanks to a pairwise registration. Refinement techniques are based on minimizing the distances between correspondence points. Here, the techniques have been classified in two main groups: point-to-point and point-to-plane. One of the main weaknesses of the point-to-point strategy is that a huge amount of iterations are required and it may converge to a local minima. Point-to-plane strategy presents more robustness against outliers and converges faster in the presence of regions with less overlapping area. In addition, point-to-point performs better in surface-to-model registrations, while point-to-plane is the most accurate in surface-to-surface registrations.

Finally, the most important multiview registration techniques have been analyzed an compared among them. These techniques have been classified as: Metaview, Simultaneous Minimization, Graph-based approaches and Statistic techniques. The main goal of these techniques is to reduce the error propagated during the registration process.

Metaview provides good solutions registering small objects and are accurate solving local problems. However, metaview presents significant misalignments when the number of views to be registered increases. This is mainly due to the lack of flexibility to re-register already merged views, being impossible to reduce the propagation error in the metaview. Simultaneous minimization solves this problem by minimizing all the views at the same time. This strategy presents robustness against outliers and limited overlapping between the views and avoids convergence to local minima. However, simultaneous minimization requires to have all the views already acquired before the registration process starts, being not suitable for real-time applications. Besides, the computational time of treating all the views simultaneously is huge, becoming a real problem when the size of the object increases, and inefficient, especially when the overlapping between views is rather limited.

The techniques based on graphs provide a good solution to the problem of the propagated error, improving considerably the results obtained by metaview approaches and being more efficient than simultaneous minimization in the presence of sequences of views with limited overlapping. Graph-based techniques permit to determine the best way in which the views should be registered to obtain the optimal solution. In addition, some graph-based approaches have added a cycle detection step that permits to minimize the propagated error when an object area is revisited. Although these methods are not as robust as the ones based on simultaneous minimization, they require less computational time and they are more efficient in large-scale objects and scenes.

One of the main problems of any multi-view technique resides in the presence of featureless objects/scenes which may contain many symmetries. This problem may be treated using statistical techniques, which consider the uncertainty in the measurement to predict and correct robot/camera pose. These techniques provide reliable solutions to the registration of large-scale scenes but they need high memory requirements and they are quite inefficient dealing with huge amount of data (high resolution models).

Fundamentals on graph theory applied to registration algorithms is provided in chapter 3. The chapter includes the basics on graph definition and detailed descriptions of the algorithms used in this thesis. Graph theory provides an conceptual model that permits to relate points and views among them. Therefore, the use of traversing graph strategies and minimal spanning trees algorithms provide a useful tool to determine the best order in which views should be registered. Besides, several techniques for detecting fundamental cycles and minimum spanning trees in a graph have been analyzed. The complexity and efficiency of the algorithms may vary considerably depending on the technique used. For example, cycle detection based on Depth First Search (DFS) is less efficient and generates larger fundamental cycles than Breadth First Search(BFS). Besides, DFS usually requires less memory than BFS. Therefore, depending on the problem and the situation we are treating, some methods may be more appropriate than others.

The main contribution of this thesis is the proposal of a new multiview registration technique based on cycle detection and minimization. This approach takes advantage of the results obtained by the sequential pairwise registration of the partial views (while they are acquired) to detect fundamental cycles and revisited regions. Once a cycle is detected, the views composing the cycle are analyzed searching for correspondences and leading to a set of virtual cycles. Finally, the accumulated error within the cycle is minimized taking into account the views of the fundamental and virtual cycles. Additionally, motion constraints are added to increase the accuracy of the minimization. The proposed technique provides better accuracy than similar techniques surveyed in this thesis. One reason is that the proposed technique takes into account the correspondences between views in the minimization, while for instance Sharp's [82] strategy only considers the motion. Besides, the proposed technique considers the information provided by virtual cycles within the fundamental cycle to provide more robustness in the minimization process. Although Sharp's approach reduces the global error along the cycle, this error is not always well distributed. Besides, the success of the Sharp's strategy depends on the accuracy of the pairwise registration process. As a conclusion, the proposed strategy provides more robustness in the multiview error minimization problem, and the obtained results are more accurate in the presence of noise and outliers. However, the technique proposed is restricted to the presence of revisited regions. Besides, the virtual cycle generation requires that some of the views of the cycle are close to each other. Otherwise, only the information provided by the consecutive views composing the cycle can be used.

Finally, experimental results have been performed to validate the proposed approach and compare it with similar techniques available in the literature. Both synthetic and real data have been used. Synthetic data permitted to analyze the robustness of the surveyed techniques in the presence of Gaussian noise comparing registration results to ground truth. Besides, synthetic data permitted to analyze the efficiency of the techniques dealing with small and large objects since partial views and overlapping areas where simulated by using the same 3D model. Two experiments with real data completes the chapter and demonstrates the accuracy of the technique.

To summarize, the main contributions of this thesis are:

- A state-of-the-art on 3D registration techniques, including initial pose estimation, pairwise registration and multiview error minimization approaches.

- A new classification of the surveyed methods have been proposed, which, to the best of our knowledge has not been published before. The classification has been published at the Journal of Electronic Imaging [78].

- A robust multiview error minimization approach based of graph theory techniques and cycle detection strategy has been presented. The main contributions reside in the cycle detection strategy and the minimization of the error using multiple virtual cycles within the cycle detected.

- Experimental tests comparing some of the most representative multiview registration techniques with the proposed method have been performed. Here it has been demonstrated the robustness of the proposed approach in comparison to other mehods, mainly in the presence of noise and outliers

6.2 Publications and scientific collaborations

The work developed during this thesis has lead to the publications of two journal articles, several contributions to international conferences and one contribution in a book chapter. These contributions are detailed below:

Articles published in international journals

- J. Salvi, E. Batlle, C. Matabosch, X. Lladó. Overview of Surface Registration Techniques Including Loop Minimization for 3D Modeling and Visual Inspection. Journal of Electronic Imaging, 17(3):1-16.

- C. Matabosch, D. Fofi, J. Salvi, E. Batlle. Registration of Surfaces Minimizing Error Propagation for a One-Shot Multi-Slit Hand-Held Scanner. Pattern Recognition 41(6), pp 2055-2067, 2008.

International conferences

- J. Salvi, Y. Petillot, E. Batlle. Visual SLAM for 3D Large-Scale Seabed Acquisition Employing Underwater Vehicles. IEEE International Conference on Intelligent Robots and Systems, IROS'08, Nice (France) September 22-26, 2008.

- Y. Petillot, J. Salvi, E. Batlle. 3D Large-Scale Seabed Reconstruction for UUV Simultaneous Localization and Mapping. IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, NGCUV'08, Killaloe (Ireland) April 8-10, 2008

- E. Batlle, C. Matabosch, J. Salvi. Summarizing image/surface registration for 6DOF robot/camera pose estimation. 3rd Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA'07, Girona, Spain, June 2007. Published in Lecture Notes in Compter Science, volume 4478, pages 105-112.

- E. Batlle, C. Matabosch, J. Salvi. Overview of 3D registration techniques including loop minimization for the complete acquisition of large manufactured parts and complex environments. 8th International Conference on Quality Control by Artificial Vision, QCAV'07, Le Creusot, France, May 2007. (Best paper prize)

- C. Matabosch, E. Batlle, D. Fofi and J. Salvi. A variant of point-to-plane registration including cycle minimization. Photogrammetric Computer Vision, PCV'06 , pages 61-66. Bonn, Germany, September 2006.

- E. Batlle, C. Matabosch, J. Salvi. Overview of Pose Estimation and 3D Registration techniques. Recerca en automàtica, visió i robòtica. Editorial: Barcelona Digital, pages 153-160. ISBN: 84-7653-885-5, 2006.

- E. Batlle, P. Ridao and N. Palomeras. A survey of Graphical Simulators for UUV Development. International Workshop on Underwater Robotics IWUR, pages 185-194. Genoa, Italy, November, 2005.

- P. Ridao, E. Batlle and N. Palomeras. First steps in remote Experimentation with UUVs. Workshop Internacional en telerobótica y realidad aumentada para teleoperación, June 2005. ISBN: 84-7484-173-9

- P. Ridao, E. Batlle, D. Ribas, and M. Carreras. NEPTUNE: A HIL simulator for multiple UUVs. In Oceans MTS/IEEE, volume 1, pages 524-531. Kobe, Japan, November 2004.

- P. Ridao, D. Ribas, E. Batlle, and E. Hernandez. Simulation of physical agents. An application to underwater robots. V Workshop on Physical Agents, pages 119-129. Girona, Spain, march 2004.

Book chapters

- J. Batlle, P. Ridao, R. Garcia, M. Carreras, X. Cufí, A. El-Fakdi, D. Ribas, T. Nicosevici, E. Batlle. Automation for the Maritime Industries. Editorial: Aranda

Armada De la Cruz, pages 177-203, ISBN: 84-609-3315-6, 2004.

Scientific collaborations

This thesis has been mainly developed in the Computer Vision and Robotics group of the University of Girona. In addition, two research stays has been done in the groups detailed above:

- 2-month stay in the Department of Computer Science, Illinois Institute of Technology, Chicago, USA. Supervisor: Dr.Gady Agam.
- 4-month stay in the Laboratoire Modélisation, Information et Systèmes, University of Picardy - Jules Verne, Amiens, France. Supervisor: El Mustapha Mouaddib.

These research stays have been supported by the *BE* Mobility grant from the Generalitat de Catalunya.

The work carried out during each research stay has contributed to the development of this thesis, acquiring better knowledge in topics such as feature extraction in scenes and 3D acquisition systems. Besides, an important part of the ultimate experiments of this thesis were carried out during the research stay in the Laboratoire Modélisation, Information et Systèmes (MIS), Amiens.

6.3 Future work

The work done in this thesis can be extended so that from my point of view the following issues are still opened:

First, further work should be done in order to improve the robustness of the cycle detection strategy. A weaknesses of the technique proposed may appear when the cycle is composed by a large number of views. In this situation, the accumulated error becomes really important causing a failure in the detection of a cycle. Note that in the proposed technique, the first constraint for a cycle to be detected is that the accumulated translation between the two end-views is small. Besides, a relaxation of the constraint may detect false cycles. A solution to this problem may reside in the use of statistic approaches such as SLAM (Simultaneous localization and Mapping) that considers uncertainty measuring and predicting the motion between views. Besides, the pairwise between the two end-views of the cycle will need an initial solution in the minimization that can be provided by a coarse registration technique based on 3D feature extraction such as point signatures [17] or spin images [40].

Second, I propose to apply multiview registration to surfaces apart from objects, which is a crucial part in 3D mapping for underwater and aerial robotics. 3D mapping can be used for both obtain the cartography of an area and localize the vehicle with respect to that area, and it has many applications such as environment monitoring (rainforest, coral reefs), archeology (ancient settlements and shipwrecks), and forensic applications (plane crashes, shipwrecks), among many others. The obtaining of the 3D cartography of an underwater environment is a key goal of one of the funded projects that started in the group in late 2007[1]. Although surface registration is similar somehow to object reconstruction, some modifications should be done to ensure the reliability of the technique. For example, computation time becomes crucial when the mapping is required as an input in the navigation of the vehicle. In the technique proposed in this thesis we have emphasized accuracy, while in real-time applications we should emphasize time constraints even though that means a relaxation in the accuracy.

Third, the multiview registration technique here proposed could be transferred to AQSense SL (spinout of the University of Girona) . AQSense commercializes 3D surface-to-model pairwise software for quality control applications[2]. So, it may be interested to perform a multiview registration of partial views before comparing the registration to the model. This technique may extend the field of applications of AQSense technology to the quality control of large objects such as the ones present in automotive and aeronautics.

Fourth, the proposed method only minimizes the error every time a cycle is detected. The proposed method does not perform a final global minimization once all the views are already acquired. Although in the experimental tests the registration residue is negligible and the final alignment is accurate enough, many other techniques consider this final alignment and further tests should be done to check if the residue is really minimized.

Finally, 3D registration should be completed, depending on application, laying the texture and color of the object on the 3D surface. This requires to research on image registration especially to smoothing the artifacts produced by inconstant lightning.

[1] Project AQUAVISION (DPI2007-66796-C03-02)

[2] www.aqsense.com

Bibliography

[1] E. Amaldi, L. Liberti, F. Maffioli, and N. Maculan. Algorithms for finding minimum fundamental cycle bases in graphs. *Electronic Notes in Discrete Mathematics*, 17:29–33, October 2004.

[2] X. Armangué. *Modelling Stereocopic Vision Systems for Robotic Applications*. PhD thesis, University of Girona, September 2003.

[3] X. Armangué, H. Araújo, and J. Salvi. A review on egomotion by means of differential epipolar geomety applied to the movement of a mobile robot. In *Pattern Recognition*, volume 12, pages 2927–2944, December 2003.

[4] A. Ashbrook, P. Rockett, and N. Thacker. Multiple shape recognition using pairwise geometric histogram based algortihms. In *Image Processing*, 1995.

[5] J.M. Basart. *Grafs: Fonaments i Algorismes*. Manuals de la Universitat Autonoma de Barcelona, 1993.

[6] E. Batlle, P. Ridao, and N. Palomeras. A survey of graphical simulators for uuv development. In *International Workshop on Underwater Robotics IWUR*, pages 185–194, Genoa, Italy, November 2005.

[7] J. Batlle, P. Ridao, R. Garcia, M. Carreras, X. Cufí, A. El-Fakdi, D. Ribas, T. Nicosevici, and E. Batlle. *Automation for the Maritime Industries*, chapter 11, pages 177–203. 2004.

[8] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–547, 1996.

[9] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin. Building a digital model of michelangelo's florentine pietà. In *IEEE Computer Graphics and Applications*, volume 22, pages 59–67, November 2002.

[10] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14:239–256, February 1992.

[11] L. Biggs, E.K. Lloyd, and R.J. Wilson. *Graph Theory.* Oxford University Press, 1986.

[12] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1899– 1906, Amherst, MA, USA, Sept 2003.

[13] D. Burschka and G.D. Hager. V-gps(slam): Vision-based inertial system for mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 409 – 415, 2004.

[14] O. Carmichael, D. Huber, and M. Hebert. Large data sets and confusing scenes in 3-d surface matching and recognition. In *International Conference on 3-D Digital Imaging and Modeling*, pages 258–367, Ottawa, Ont. , Canada, October 1999.

[15] C-S. Chen, Y-P. Hung, and J-B. Cheng. A fast automatic method for registration of partially overlapping range images. In *International Conference on Computer Vision*, pages 242–248, Bombay, January 1998.

[16] G. Chen, Y. ad Medioni. Object modeling by registration of multiple range images. In *IEEE Int. Conference on Robotics and Automation*, volume 3, pages 2724 –2729, April 1991.

[17] C.S. Chua and R. Jarvis. Point signatures: A new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, October 1997.

[18] A.J. Davison, W.W. Mayol, and D.W. Murray. Real-time localization and mapping with wearable active vision. In *Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reallity*, pages 18–27, 2003.

[19] M Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *Seventh International Symposium on*

Experimental Robotics, volume 271, pages 395 – 404, Honolulu, Hawaii, December 2000.

[20] C. Demonceaux, P. Vasseur, and Pégard. C. Uav attitude computation by omnidirectional vision in urban environment. In *IEEE International Conference on Robotics and Automation*, pages 2017–2022, April 2007.

[21] N. Deo, G. Prabhu, and M.S. Krishnamoorthy. Algorithms for generating fundamental cycles in a graph. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):26–42, March 1982.

[22] R. Diestel. *Graph Theory*, volume 173. Graduate Tests in Mathematics, New York, 1997.

[23] C. Estrada, J. Neira, and J.D. Tardos. Hierarchical slam: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.

[24] A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *The British Machine Vision Conference*, pages 662–670, 2001.

[25] D. Fofi, J. Salvi, and E. Mouaddib. Uncalibrated vision based on structured light. In *International Conference on Robotics and Automation*, volume 4, pages 3548–3553, Séoul, Corée, May 2001.

[26] J. Folkesson and H. Christensen. Outdoor exploration and slam using a compressed filter. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 419–426, 2003.

[27] J. Folkesson, P Jensfelt, and H.I. Christensen. Graphical slam using vision and the measurement subspace. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3383–3390, Edmundton, Canada, August 2005.

[28] H. Gagnon, M. Soucy, R. Bergevin, and D. Laurendeau. Registration of multiple range views for automatic 3-d model building. In *Computer Vision and Pattern Recognition*, pages 581–586, June 1994.

[29] R. Garcia, X. Cufi, P. Ridao, and M. Carreras. *Constructing Photo-mosaics to Assist UUV Navigation and Station-keeping*, chapter 9, pages 195–234. Robotics and Automation in the Maritime Industries, iIAI-CSIC, 2006.

[30] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy. Geometrically stable sampling for the icp algorithm. In *International Conference 3-D Digital Imaging and Modeling*, pages 260 – 267. 3DIM, 2003.

[31] M. Greenspan and G. Godin. A nearest neighbor method for efficient icp. In *Third International Conference on 3-D Digital Imaging and Modeling*, pages 161–168, Quebec, Canada, May-June 2001.

[32] G. Guidi, J.-A. Beraldin, and C. Atzeni. High-accuracy 3-d modeling of cultural heritage: The digitizing of donatello's "maddalena". In *IEEE Transactions on Image Processing*, volume 3, pages 370–380, 2004.

[33] J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics*, 3(17):242–257, 2000.

[34] C. J. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conferences*, pages 147–151, 1988.

[35] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2000.

[36] J.D. Horton. A polynomial-time algorithm to find the shortest cycle basis of a graph. *SIAM Journal on Computing*, 16(2):358–366, April 1987.

[37] T.S. Huang and O.D. Faugeras. Some properties of the e matrix in two-view motion estimation. *Pattern Analysis and Machine Intelligence*, 11(12):1310 1312, December 1989.

[38] D. Huber. Automatic three-dimensional modeling from reality. In *Ph.D.*, 2002.

[39] D. Huber and M. Hebert. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing*, 21(7):637–650, July 2003.

[40] A.E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, May 1999.

[41] T. Jost and H. Hugli. A multi-resolution scheme icp algorithm for fast shape registration. In *First International Symposium on 3D Data Processing Visualization and Transmission*, pages 540– 543, 2002.

[42] J. Kim and S. Sukkarieh. Autonomous airborne navigation in unknown terrain environments. In *IEEE Transactions on Aerospace and Electronic Systems*, pages 1031–1045, 2004.

[43] S.H. Kim, J.K. Seo, Hong H.K, C.W. Jho, and M.H Choi. Iterative registration of multiple 3d data sets using covariance matrix. In *The 8th International Conference on Virtual Systems and Multimedia*, September 2002.

[44] P. Kohlhepp, P. Pozzo, M. Walther, and R Dillmann. Sequential 3d-slam for mobile action planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 722 – 729, September 2004.

[45] J. B. Kruskal. On the shortest spanning subtree and the traveling salesman problem. In *Proceedings of the American Mathematical Society*, volume 7, pages 48–50, 1956.

[46] T. Lemaire, S. Lacroix, and J. Sola. A practical 3d bearing-only slam algorithm. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 2449– 2454, Aug 2005.

[47] Y Liu and S. Thrun. Results for outdoor slam using sparse extended information filters. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 1227– 1233, September 2003.

[48] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision ICCV*, pages 1150–1157, Corfu, Greece, September 1999.

[49] A. Martinelli, N. Tomatis, and R. Siegwart. Some results on slam and the closing the loop problem. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2917– 2922, Lausanne, Switzerland, Aug 2005.

[50] T. Masuda. Generation of geometric model by registration and integration of multiple range images. In *Third International Conference on 3-D Digital Imaging and Modeling*, pages 254 –261, May 2001.

[51] T. Masuda. Object shape modelling from multiple range images by matching signed distance fields. In *First International Symposium on 3D Data Processing Visualization and Transmission*, pages 439–448, 2002.

[52] T. Masuda, K. Sakaue, and N. Yokoya. Registration and integration of multiple range images for 3-d model construction. In *International Conference on Pattern Recognition*, volume 1, pages 879–883, Vienna, Austria, August 1996.

[53] C. Matabosch, E. Batlle, D Fofi, and J. Salvi. A variant of point-to-plane registration including cycle minimization. In *Photogrammetric Computer Vision, PCV'06*, pages 61–66, Bonn, Germany, September 2006.

[54] C. Matabosch, J. Salvi, X. Pinsach, and R. Garcia. Surface registration from range image fusion. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 678 – 683, New Orleans, USA, April 2004.

[55] A.S. Mian, M. Bennamoun, and R. Owens. 3d model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions in Pattern Analysis and Machine Intelligence (PAMI)*, 28(10):1584–1601, October 2006.

[56] A.S. Mian, M. Bennamoun, and R. Owens. A novel representation and feature matching algorithm for automatic pairwise registration of range images. *International Journal of Computer Vision (IJCV)*, 66(1):19–40, 2006.

[57] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to the simultaneous localization and mapping problem. In *National Conference on Artificial Intelligence*, pages 593–598, Vancouver, BC, July 2002.

[58] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion. In *British Machine Vision Conference*, Warwick, United Kingdom, September 2007.

[59] A. N*ü*chter, H. Surmann, K. Lingemann, J. Hertzberg, and S. Thrun. 6d slam with an application in autonomous mine mapping. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1998 – 2003, May 2004.

[60] J. Neira and J. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. on Robotics and Automation,*, 17(6):890–897, December 2001.

[61] J. Neira, J. D. Tardós, and J. A. Castellanos. Linear time vehicle relocation in slam. In *International Conference Robotics and Automation*, pages 427–433, Taipei, Taiwan, September 2003.

[62] P.J. Neugebauer. Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images. *International Journal of Shape Modeling*, (3):71–90, 1997.

[63] P. M. Newman. *On the Structure and Solution of the Simultaneous Localization and Map Building Problem.* PhD thesis, University of Sydney, Australia, March 1999.

[64] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 652–659, July 2004.

[65] A. Oliver, Martí, R. J. Martí, A. Bosch, and J. Freixenet. A new approach to the classification of mammographic masses and normal breast tissue. In *International Conference on Pattern Recognition*, volume 4, pages 707–710, Hong Kong, August 2006.

[66] R. Orghidan, J. Salvi, and E. Mouaddib. Modelling and accuracy estimation of a new omnidirectional depth computation sensor. *Pattern Recognition Letters*, 27(7):843–853, 2006.

[67] J. Pagès. *Assisted visual servoing by means of structured light.* European thesis for the university of girona and french thesis for the university of rennes, November 2005.

[68] S.-Y. Park and M. Subbarao. A fast point-to-tangent plane technique for multi-view registration. In *Int. Conference on 3-D Digital Imaging and Modeling*, pages 276 – 283, 2003.

[69] K. Paton. An algorithm for finding a fundamental set of cycles of a graph. *Communications of the ACM*, 12(9):514–518, September 1969.

[70] Y. Petillot, J. Salvi, and E. Batlle. 3d large-scale seabed reconstruction for uuv simultaneous localization and mapping. In *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, (NGCUV)*, Killaloe (Ireland), April 2008.

[71] J. Pina. *Applications of Shortest Path Methods.* PhD thesis, University of Amsterdam, Netherlands, 1995.

[72] M. Pollefeys, M.R. Koch, M. Vergauwen, and Van Gool L. Automated reconstruction of 3d scenes from sequences of images. *Photogrammetry and Remote Sensing*, (55):251–267, 2000.

[73] K. Pulli. Multiview registration for large data sets. In *International Conference on 3-D Digital Imaging and Modeling*, pages 160 – 168, October 1999.

[74] P. Ridao, E. Batlle, and N. Palomeras. Firts steps in remote experimentation with uuvs. In *Workshop Internacional en telerobótica y realidad aumentada para teleoperación*, Madrid, Spain, June 2005.

[75] P. Ridao, E. Batlle, D. Ribas, and M. Carreras. Neptune: A hil simulator for multiple uuvs. In *Oceans*, volume 1, pages 524– 531, Kobe, Japan, November 2004.

[76] P Ridao, D. Ribas, E. Batlle, and E. Hernandez. Simulation of physical agents. an application to underwater robots. In *V Workshop on Physical Agents*, Girona, Spain,, March 2004.

[77] S. Rusinkiewicz and M. Levoy. Efficient variant of the icp algorithm. In *3rd International Conference on 3-D Digital Imaging and Modeling,*, pages 145 – 152, 2001.

[78] J. Salvi, E. Batlle, C. Matabosch, and X. Lladó. Overview of surface registration techniques including loop minimization for 3d modelling and visual inspection. *Journal of Electronic Imaging*, 17(3):1–16.

[79] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing, In Press*, 25(5):578–596, May 2007.

[80] J. Salvi, Y. Petillot, and E. Batlle. Visual slam for 3d large-scale seabed acquisition employing underwater vehicles. In *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Nice (France), September 2008.

[81] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The International Journal of Robotics Research*, 21(8):735–758, 2002.

[82] G. Sharp, S. Lee, and D. Wehe. Multiview registration of 3d scenes by minimizing error between coordinate frames. In *Pattern Analysis and Machine Intelligence*, pages 1037 – 1050, 2004.

[83] L. Silva, O.R.P. Bellon, and K.L. Boyer. Enhanced, robust genetic algorithms for multiview range image registration. In *3DIM03. Fourth International Conference on 3-D Digital Imaging and Modeling*, pages 268–275, 2003.

[84] C. Stachniss, G. Grisetti, and W. Burgard. Recovering particle diversity in a rao-blackwellized particle filter for slam after actively closing loops. In *IEEE International Conference on Robotics and Automation*, pages 667–672, 2005.

[85] I. Stamos and M. Leordeanu. Automated feature-based range registration of urban scenes of large scale. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 555–561, 2003.

[86] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *Computer Vision and Pattern Recognition.*, pages 1100 – 1105, June 1997.

[87] D. Tardos, J. Neira, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research.*, 21(4):311–330, December 2002.

[88] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–169, 1972.

[89] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* The MIT Press, September 2005.

[90] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A.Y. Ng. Simultaneous mapping and localization with sparse extended information filters. *International Journal of Robotics Research*, 23(7-8):693–716, 2002.

[91] J.C. Tiernan. An efficient search algorithm to find the elementary circuits of a graph. *Communications of the ACM*, 13(12):722–726, December 1970.

[92] R. Triebel and W. Burgard. Improving simultaneous mapping and localization in 3d using global constraints. In *Proceedings of the National Conference on Artificial Intelligence*, volume 3, pages 1330–1335, 2005.

[93] B. Triggs, P. McLauchlan, and A. Hartley, R. Fitzgibbon. Bundle adjustment - a modern synthesis. In *International Workshop on Vision Algorithms*, pages 298 – 372, London, UK, 2000.

[94] E. Trucco, A. Fusiello, and V. Roberto. Robust motion and correspondences of noisy 3-d point sets with missing data. *Pattern Recognition Letters*, 20(9):889–898, September 1999.

[95] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 311–318, Orlando, Florida.

[96] M. Umasuthan and A. M. Wallace. Model indexing and object recognition using 3d viewpoint invariance. *Pattern Recognition*, 30(9):1415–1434, September 1997.

[97] H. Weinblatt. A new search algorithm for finding the simple cycles of a finite directed graph. *Journal of the ACM (JACM)*, 19(1):43–56, January 1972.

[98] S.B. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 406– 411, Washington, USA, May 2002.

[99] J.V. Wyngaerd. Combining texture and shape for automatic crude patch registration. In *Int. Conference on3-D Digital Imaging and Modeling*, pages 179 – 186, 2003.

[100] Z. Zhang and O. D. Faugeras. *3D Dynamic Scene Analysis: A Stereo Based Approach*. Springer-Verlag, 1992.

[101] Z. Zhang, Q.-T. Luong, and O. Faugcras. Motion of an uncalibrated stereo ring: Self-calibration and metric reconstruction. In *IEEE Transactions on Robotics and Automation*, pages 103–113, 1996.