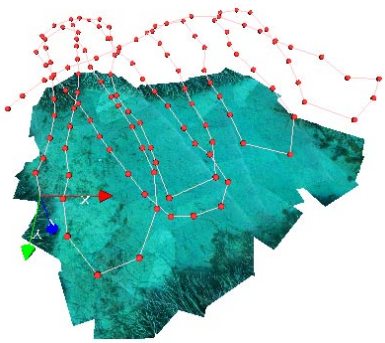
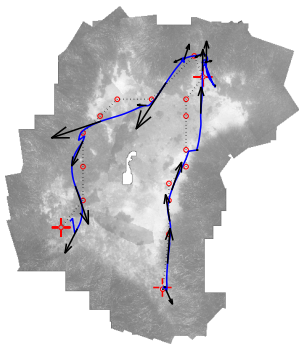
UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Mosaic–based Visual Navigation for Autonomous Underwater Vehicles

NUNO RICARDO ESTRELA GRACIAS
(Mestre)

Dissertação para obtenção do Grau de Doutor em Engenharia Electrotécnica e de Computadores

Orientador: Doutor José Alberto Rosado dos Santos Victor

Júri:

Presidente: Reitor da Universidade Técnica de Lisboa

Vogais: Doutor João José dos Santos Sentieiro
Doutor Shahriar Negahdaripour
Doutor Emanuele Trucco
Doutor José Alberto Rosado dos Santos Victor
Doutor João Paulo Salgado Arriscado Costeira
Doutor Paulo Jorge Coelho Ramalho Oliveira

Dezembro de 2002

Resumo

O meio submarino constitui um desafio à navegação autónoma de veículos robóticos. Uma solução comum para o problema da esimação de posição consiste no uso de faróis acústicos posicionados com grande precisão, o que implica elevados custos de operação. Recentemente, o sensoriamento baseado em Visão tem vindo a ser encarado como uma alternativa de baixo custo. Apesar de apresentar um alcance limitado, devido a restrições de visibilidade e iluminação, o posicionamento baseado em Visão pode ser usado para navegação se estiver associado a uma representação adequada do meio ambiente. Neste contexto, os mosaicos vídeo constituem uma solução natural para o problema do campo de visão limitado, podendo ser usada como representação do fundo marinho.

Esta tese aborda o problema da construção de mosaicos video capazes de servir de suporte à navegação de veículos autónomos, operando perto do fundo marinho. Os métodos desenvolvidos estão vocacionadas para missões nas quais um veículo é comandado para mapear uma região de interesse aproximadamente plana e navegar posteriormente sobre a mesma.

Na primeira parte do trabalho apresenta-se uma metodologia para a criação automática de mosaicos vídeo, que possibilita a estimação tridimensional da trajectória da câmara. O processo de estimação tira partido explícito de trajectórias em *loop*, onde se visita a mesma área em instantes distintos, de modo a obter mosaicos com elevada coerência espacial.

De seguida aborda-se o tema da estimação de pose a partir de um mosaico previamente criado. Uma soluções algébrica é apresentada para o caso de conhecimento total dos parâmetros intrínsecos das câmaras. Esta solução é posteriormente refinada com estimadores de máxima verosimilhança. A incerteza associada é calculada através da estimação da covariância dos parâmetros de pose como função das observações de imagem.

A última parte do trabalho ilustra o uso de mosaicos como mapas, para a navegação autónoma de uma plataforma robótica. Um conjunto de rotinas eficientes é usado para a localização precisa de um veículo em relação ao mosaico, levando em conta os requisitos operacionais de posicionamento em tempo–real, erros limitados e baixo peso computacional. Um módulo de geração de trajectórias é usado para guiar a navegação sobre zonas onde o posicionamento visual é mais fiável. A geração dos sinais de controlo dos actuadores assenta numa estratégia de *servoing* visual.

De forma a validar a abordagem e caracterizar o desempenho dos vários métodos, um veículo submarino operado remotamente foi usado em condições reais de operação. Apresentam-se resultados de testes realizados no mar, nos quais foi possível efectuar navegação autónoma durante extensos períodos de tempo. Nesta tese mostra-se que, sem recurso a outro tipo de sensores, a informação visual pode ser usada para criar representações do fundo marinho e suportar navegação.

Palavras Chave: Visão por Computador, Mosaicos Sub–aquáticos, Navegação baseada em Mosaicos, Controlo Visual, Estimação de Pose, Estimação Robusta de Movimento

Abstract

The underwater environment poses a difficult challenge for autonomous vehicle navigation. Common positioning solutions require the deployment of precisely located acoustic beacons, which typically implies high operating costs. Vision sensing is increasingly being regarded as a low cost alternative, but is limited to short range due to visibility and lighting factors. However, it can provide precise positioning if an adequate representation of the environment is found. Video mosaicing presents itself as a suitable technique to overcome the limited underwater field–of–view.

This thesis addresses the problem of creating accurate video mosaics, capable of serving as navigation maps for autonomous vehicles operating close to the sea–floor. It is devised for mission scenarios where a robotic platform is required to map an approximately flat area of interest and to navigate upon it afterwards.

The first part of this work presents a methodology for the simultaneous creation of mosaics and the estimation of the camera trajectory. Mosaicing is performed in a fully automatic manner and attains full spatial coherence by exploring time-distant superpositions, such as the ones arising from loop trajectories or zig-zag scanning patterns.

Next, the problem of the pose estimation using a previously constructed mosaic is addressed. A direct algebraic solution is presented the case of known camera intrinsics, which is refined with a maximum likelihood estimator. The associated uncertainty is computed by propagating the covariance from image measurements to the pose estimates.

The last part illustrates the use of mosaic maps for autonomous navigation. A set of efficient routines is required for the accurate localization of the vehicle with respect to the mosaic, taking into account the operating requirements of real-time position sensing, error bounding and low computational load. A trajectory generation module is used to guide the navigation over well defined areas of the mosaic, where the visual based positioning is most reliable. The control signals are generated using a visual servoing strategy.

In order to assess the performance of the overall system, a Remotely Operated Vehicle was used under real operating conditions. Extensive testing was performed at sea, where the vehicle was able to autonomously navigate over previously created mosaics for large periods of time. This work demonstrates that, without resorting to additional sensors, visual information alone can be used to create environment representations of the sea bottom and support long runs of navigation.

Key Words: Computer Vision, Underwater Mosaics, Mosaic–based Navigation, Visual Servoing, Pose Estimation, Robust Motion Estimation

Agradecimentos

A concretização desta dissertação beneficiou da ajuda e empenho de um grupo de pessoas, às quais gostaria aqui de agradecer.

Em primeiro lugar, quero expressar o meu mais sentido agradecimento ao Prof. José Alberto Santos–Victor, pelo empenho com que orientou este trabalho, pela objectividade e rigor nas discussões, por todo o convívio e amizade.

A todos os membros do Laboratório de Visão com quem tive o prazer de partilhar a convivência diária. Ao Sjoerd e ao Alexandre, que contribuiram com grande parte do trabalho que está por detrás do comando e controlo do ROV. Sem o seu empenho e competência, o *servoing* com mosaicos não teria sido uma realidade. Ao Etienne pela infinita paciência nas inúmeras questões matemáticas. Ao César, João Maciel, José Gaspar, Jonas e a todos restantes membros do Vislab pelas discussões acesas, ajuda pronta e espírito de equipa.

Ao Prof. João Sentieiro pela criação e manutenção de condições de excelência para a investigação no Instituto de Sistemas e Robótica – Pólo de Lisboa, onde este trabalho foi realizado. Dentro do ISR beneficiei de um ambiente pluridisciplinar onde o espírito de entreajuda permitiu a troca de ideias, críticas e sugestões a todos os níveis.

Ao Stefan (herr Flick!) pela incansável ajuda nos testes com o ROV e por partilhar as alegrias de ver o trabalho funcionar. Os teste em Villefranche-sur-Mer serão sempre momentos inesquecíveis (mesmo quando se teve que retirar o veiculo da água a dois!). Agradeço à Prof^a Maria João Rendas as condições de trabalho reunidas, e ao Matteo por todo o apoio e boa disposição.

À Fundação para a Ciência e Tecnologia e Fundo Social Europeu no âmbito do III Quadro Comunitário de Apoio, pelo apoio financeiro através da bolsa PRAXIS XXI / BD / 13772 / 97. Este trabalho beneficiou ainda do apoio do projectos NARVAL ESPRIT–LTR 30185 e SUMARE IST 1999–10836.

Ao Henrique e Inês pela amizade, passeios, visitas aos Estados Unidos e férias bem passadas. Ao Vasco, Pedro Chico e Zé Miguel pela amizade de muito anos. A todo o pessoal das "Setas" pelos momentos de descontração. Ao Miguel pelos desabafos da hora de almoço.

Aos meus pais pelas condições criadas, sem as quais eu não teria concretizado este trabalho, e por todo o apoio e conselhos que só os pais podem dar. À Maria João pelas conversas e companhia do dia–a–dia. Ao Rudolfo pelas turras!

Finalmente, quero agradecer à Patrícia todo o carinho, dedicação, paciência e amizade para os quais não tenho palavras.

Contents

List of Figures

Chapter 1

Introduction

This thesis addresses the problem of creating accurate video mosaics, capable of serving as navigation maps for autonomous vehicles operating close to the sea floor. It is devised for mission scenarios where a robotic platform is required to map an approximately flat area of interest and to navigate upon it afterwards.

1.1 Motivation

The autonomous navigation of underwater vehicles is a growing research and application field. A contributing factor is the increasing need of underwater activities such as environmental and industrial monitoring or geological surveying. Applications that require data acquisition at precise locations usually resort to the use of unmanned underwater vehicles, either in the form of human–piloted Remotely Operated Vehicles (ROVs), or unthetered Autonomous Underwater Vehicles (AUVs).

Recent interest has been devoted to the development of smart sensors, where the data acquisition and navigation are intertwined. These systems aim at releasing the human operation from low-level requirements, such as the path planning, obstacle avoidance and homing. By providing the platforms with such level of human independence, these systems allow for the reduction of operating costs while broadening the potential end-users group. The user main tasks are in the definition of mission primitives to be carried out and higher level mission control.

1.1.1 Underwater position sensing

The underwater environment poses a difficult challenge for precise vehicle positioning. The severe absorption of electromagnetic radiation prevents the use of long range radio transponders. Aerial or land robot navigation can rely upon the Global Positioning System to provide real-time updates with errors of just a few centimeters, anywhere around the world. The underwater acoustic equivalent is limited both in range and accuracy. It requires the previous deployment of carefully located beacons, and restricts the vehicle operating range to the area in between. Sonar equipment provides range data and is increasingly being used in topographic matching for navigation, but the resolution is too low for precise, sub–metric navigation.

Vision can provide precise positioning if an adequate representation of the environment exists, but is limited to short distances to the floor due to visibility and lighting factors. However, for the mission scenarios where the working locations change often and are restricted to relatively small areas, vision–based positioning appears as an inexpensive and promising alternative.

There is a number of commercially available technologies and products capable of providing position or velocity information to underwater vehicles. These can be grouped into two categories depending on whether the sensing relies upon the active emission and propagation of acoustic waves in the water. The first group comprises long baseline transponder networks, sonar-based altimeters and Doppler velocity logs. The second encompasses the non-acoustic sensors such as gyroscopes, accelerometers, magnetic compasses and inclinometers.

In order to illustrate comparatively the advantages and limits of the vision–based approach of this thesis, some of the distinctive features of these commonly used sensors are now briefly discussed.

Acoustic Transponder Networks

Long Baseline (LBL) acoustic positioning constitute the standard technique for 3D navigation in AUV applications [124]. It relies upon a network of 3 or more transponders which need to be placed in fixed and accurately known locations.

Under typical operation, the position determination process is triggered by the vehicle emitting a short duration pulse at a given interrogation frequency. Upon detection of

such pulse, each transponder replies by emitting at a distinctive frequency, after a known constant time delay. The vehicle senses the replies and measures the response time intervals. Given the knowledge of the sound propagation velocity, the response intervals are converted into distances. The vehicle position is obtained by triangulation, with respect to the transponder locations. The position update rate is thus dependant on the distance to the furthest transponder.

Long baseline transponder networks have been successfully used for several decades and can be considered a mature technology in terms of reliability and commercial availability. However, in order to provide absolute 3D localization, they require the previous deployment and calibration of the network. This bears high operating costs and is not suitable for applications where the working area changes frequently.

The accuracy of the position measurements strongly depends upon the pulse frequency and the geometric arrangement of the networks. These factors define the trade-off between the size of the work area and measurement resolution. A typical 12kHz LBL has a range of 5 to 10 km with 10 meters accuracy and an update rate of 0.1 Hz [124]. Using higher frequency pulses attains better resolutions at the expenses of much lower ranges due to the higher propagation attenuation. A shorter range 300kHz LBL can be used up to 100 meters and provide centimeter level accuracy under 1Hz update rates [124]. High frequency LBL are suited for vehicle docking maneuvers into docking stations to which the transponders are permanently attached.

Inertial Navigation Systems and other passive internal sensors

Inertial Navigation Systems (INS) are self–contained devices, comprising accelerometers and gyroscopes.

Traditional mechanical gyroscopes rely on the momentum conservation of a fast rotating mass, which gives rise to reaction forces when the orientation is changed. Such forces are measured and converted into angular velocity readings. High quality mechanical gyroscopes attain small angular drifts of less than 0.2 degrees/hour). However their large size, cost and maintenance requirements precludes their use in AUVs and small ROVs.

Recently, lower cost gyros using technology that does not require rotating parts, have become commercially available. The Ring Laser Gyros and Fiber Optics Gyros measure the phase shift between two light beams travelling in opposite directions, from which the angular velocity can be computed [20]. Such devices typically exhibit bias errors of 0.1 to

0.01 degrees/hour and are commonly found in low cost vehicles.

Accelerometers use masses and springs to measure small mass displacements along perpendicular axes, when the devices are subject to external forces. The displacements are converted into acceleration readings.

For control purposes it is often required to have velocity (both linear and angular) or position measurements. These can be obtained by single or double integration of the gyro and accelerometers output. As such, the use of INS suffers from unbounded error growths, typical of the dead reckoning operation, which requires periodic error resetting by means of other sensors. However they can provide readings with fast updates and low latency when compared to acoustic based position sensing.

Compasses obtain heading information by measuring the Earth's magnetic field. This sensor has the advantage of being drift free (i.e. does not accumulate errors over time) and to provide information in a fixed world frame (the orientation of the Earth's magnetic field at that place). However, it can be severely affected by magnetic perturbations caused either by on–board equipment or local environment variations [44].

Inclinometers provide roll and pitch information by taking angular measurements of the gravity vector orientation with respect to the sensor frame. Depth sensors determine the distance to the surface by measuring the hydrostatic pressure caused by the water column. Compasses, inclinometers and depth sensors are inexpensive, passive sensors, that constitute standard equipment even in low cost underwater vehicles.

Doppler Velocity Logs

Doppler Velocity Logs (DVL) are sonar based devices that measure the vehicle velocity with respect to the sea bottom by taking advantage of the Doppler effect. A sonar pulse is emitted at a known frequency. The frequency of the returned signal is measured and compared to that of the original pulse. The rationale behind the method is that the frequency shift is proportional to the vehicle velocity across the direction of the sonar beam.

A typical DVL configuration uses 3 sonar beams with 120 degrees separation, or 4 beams at 90 degrees, which allow for determining the 3D velocity components in the vehicle reference frame. These components can be integrated over time in order to obtain displacement values.

The precision of the velocity measurements depends on factors such as the knowledge

of the local sound propagation speed, the distance to the floor, and the pulse frequency [61]. Errors in the assumed sound propagation speed (which varies with depth, salinity and temperature) induce bias in the measured velocity. These can have large effects if the DVL is used as position sensor due to the rapid growth of dead-reckoning accumulated errors.

Vision

For several decades, the use of vision for navigation has been a topic of extensive research for the land robotics community. This fact somehow contrasts with research in marine robotics, where only recently has optical vision been regarded as a promising positioning modality. The most important contributing factors are:

- Optical cameras are now standard equipment in underwater vehicles. Most ROVs rely upon image feedback for human–assisted motion control and manipulation.
- Optical sensing requires inexpensive equipment. The most common configuration comprises an analog video camera, a frame grabber board and a general purpose host computer.
- It provides low level information (images) at a fast rate (25 or 30 Hz). The update rate for higher level information, such as 3D position, is only limited by the computer processing power.
- It allows for effective man–machine interfaces based on visual content. Here video mosaics assume a predominant role as an intuitive representation of the environment, that can be very efficiently interpreted by human operators.
- Optical sensing operates in a passive mode. It is not intrusive, apart from possible artificial illumination requirements.
- It can provide 3D position and orientation information, in a fixed world coordinate frame, without requiring the deployment of artificial landmarks or transponders. This topic is addressed and illustrated in Chapters 4 and 5.
- It can be used for autonomous navigation, without requiring further sensors. This is the core topic of the thesis. Navigation is illustrated in Chapter 6.

However there are two main limiting factors, related with optical observability:

- The sensor range is limited by visibility conditions. Even under favorable conditions, ocean water turbidity and absorption limit the sensor range to 7 to 10 meters. Also, artificial illumination is required for deep waters, which can be very power demanding.

- Optical sensing requires the presence of distinctive visual features (or cues). Video processing for position sensing or mapping applications is based upon the analysis of the image content. Therefore a minimal amount of texture must be observable.

The performance of a vision based positioning system is dependant on a large number of factors which complicate the sensor characterization. Apart from optical–related factors, such as visibility and texture content, a number of processing–related factors influence the accuracy of position estimate. These include the choice of visual cues that are extracted at low level, the geometric models underlying the position estimation process and, if applicable, the way the spatial representation of the environment is created.

This topic constitutes an area of present–day active research. Rapid progress in computer vision algorithms and underwater camera technology will eventually push optical positioning solutions to a commercial level.

1.2 Overview of the Approach

This thesis describes a methodology for solving the problem of autonomous navigation for underwater vehicles operating close to the sea floor. It is devised for mission scenarios where an autonomous platform is first required to map an area of interest, and to navigate upon it afterwards. This type of mission is common in salvaging operations, natural habitat monitoring [89] or marine archeology [4].

The methodology is divided into two operation modes that are schematically represented in Fig. 1.1.

The first corresponds to the creation of extended visual representations of the sea floor. For this, a high-quality video-mosaic is automatically built from a set of images that cover the area of interest. The resulting mosaic will cover a much wider region than what is covered by each of the single images.

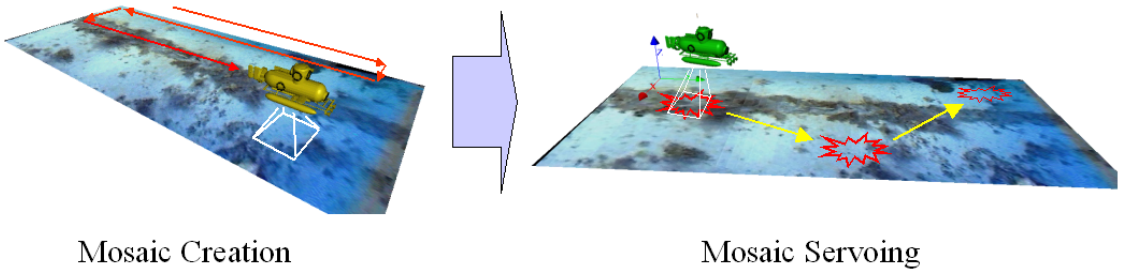


Figure 1.1: The operation modes for the proposed mosaic-based navigation system.

In the second part the mosaics are used as spatial representations to support autonomous navigation. A visual servoing strategy is used to drive the vehicle along a computed trajectory that avoid undefined regions of the mosaic. Position errors are computed by comparing (registering) the instantaneous views acquired by the vehicle with the mosaic. These errors are used to generate motion commands to the vehicle. The proposed approach was tested at sea with a computer controlled ROV.

At a user level, this work provides extended navigation capabilities that are illustrated in Fig 1.2. This image contains part of the man-machine interface that was built for issuing the X-Y position commands during the field experiments. The estimated position and orientation of the ROV with respect to the mosaic map is represented by the rectangular frame in the lower part of the map. This frame corresponds to the area that is currently being imaged by the live camera, on the right. After an initial mosaic lock–down procedure, the operator can specify a location for the vehicle to move to, by clicking on the desired position. A suitable trajectory is automatically generated which simultaneously minimizes the total travel distance and avoids unmapped regions. The motor commands are then issued.

This figure also illustrates one of the advantages, mentioned above, of optical vision in underwater navigation. By providing the actual floor appearance, video mosaics allow for man–machine interfaces based on visual content, that can easily be interpreted by human operators.

1.3 Contributions

The work in this thesis contributes to the field of visual underwater navigation in several ways:

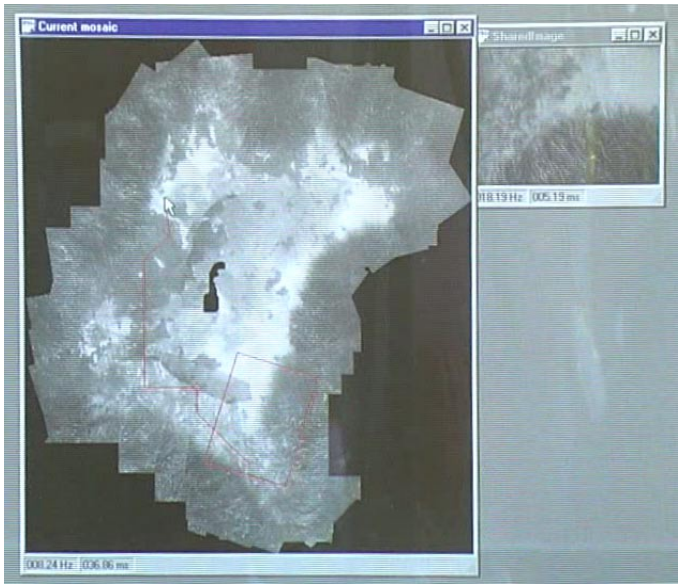


Figure 1.2: Screen capture of the man–machine interface used for navigation. It contains the mosaic map (left) with the estimated position (red rectagular frame), specified end position (arrow) and trajectory. The live camera feed is shown in the upper right window.

- A novel parameterization is devised for the *accurate global registration* of the images. *All the degrees of freedom* arising from the mosaic geometry are taken into account and parameterized as geometrically meaningful entities – pose parameters and world plane description. As a result, we recover the 3D vehicle trajectory undertaken during the mosaic image acquisition.

- The mosaic creation is approached in a *fully automated and integrated* way where *global spatial consistency* is imposed by estimating the image neighboring topology.

- The mosaics are used as maps for localization. A maximum likelihood solution is presented for estimating the *full 3D pose with the associated uncertainty.*

- An *efficient real-time mosaic tracking* strategy is proposed for the navigation. Efficiency is achieved by devising different techniques of inter-image motion and image-to-mosaic matching. These techniques make use of robust estimation methods to attain the degree of accuracy and robustness required for long periods of operation.

- A *new capability of mosaic servoing* for ROVs is proposed and demonstrated by successful experimental testing in the challenging, real–world conditions of the underwater environment at sea.

Part of this work was carried out under the European Project NARVAL [73], whose main scientific goal was the design and implementation of reliable navigation systems for mobile robots in unstructured environments. A strong emphasis was put on the ability to navigate without resorting to global positioning methods. The algorithms and results described in this thesis, where large mosaics are created and used for posterior navigation, constitute a major achievement regarding this goal.

1.4 Thesis Organization

Chapter 2 introduces the notation used in the thesis. It reviews some specific background in geometry and estimation. The first section describes the image formation basics and details the relation between image projections of the same 3D planar surface. The second presents robust sampling algorithms used in motion analysis.

Chapter 3 overviews the related work in motion estimation and mosaic creation. The topic of mosaic–based navigation is also addressed.

Chapter 4 details the mosaicing approach proposed for the creation of visual maps. The topics of sequential motion estimation, path topology and accurate global registration are discussed in detail. Selected results are presented.

Chapter 5 addresses the use of a previously created mosaic for the localization of a camera equipped vehicle. Both algebraic and maximum likelihood methods are presented along with the covariance propagation of the produced estimated. Results are presented and compared.

Chapter 6 deals with mosaic–based navigation. The topics of real–time visual tracking, trajectory generation and visual servoing are detailed. The chapter concludes with results from fully integrated mosaic navigation runs at sea, where the performance of the overall positioning system is tested and discussed.

Finally, Chapter 7 summarizes the achievements of the work presented in this thesis, and points out some directions for short–term interesting developments.

Chapter 2

Background on Geometry and Estimation

This chapter reviews some theoretical background in geometry and estimation that is used in the following chapters.

Section 2.1 presents the geometric foundations of mosaic creation methods, and introduces the notation that will be used throughout the thesis. Namely, it introduces the collineation in the 2-D projective space, which is the backbone model for image motion in the mosaicing process. Section 2.2 reviews a commonly used class of robust model-based estimation techniques using random sampling.

2.1 Projective Geometry

This section is dedicated to well established geometry considerations. Some of the key properties of projective geometry will be described, such as the notions of projective space and collineation, followed by the perspective camera model. We then move on to the study of the planar transformations. This class of transformations relates the image projections of 3-D points lying on planes and constitutes the central model for the image motion estimation. For an in–depth introduction to this subject the reader is referred to Faugeras [21] and Hartley and Zisserman [42].

2.1.1 Basic Properties of the Projective Space

Definition 1 (Affine Space and Projective Space) *The set of points parameterized by the set of all real valued n-vector* $(x_1, \ldots, x_n)^T \in \mathbb{R}^n$ *is called* Affine Space.

The set of points represented by a $n+1$ *vector* $(x_1, \ldots, x_n, x_{n+1})^T \in \mathbb{R}^{n+1}$ *is called a* Projective Space $\mathbb{P}^n$ *if the following condition and property are considered:*

1. *At least one of the* $n+1$ *vector coordinates is different from zero.*
2. *Two vectors* $(x_1, \ldots, x_n, x_{n+1})^T$ *and* $(\lambda x_1, \ldots, \lambda x_n, \lambda x_{n+1})^T$ *represent the same point for any* $\lambda \neq 0$.

The elements x_i of a projective space vector are usually called *homogeneous coordinates* or *projective coordinates*. The affine space $\mathbb{R}^n$ can be considered to be embedded in $\mathbb{P}^n$ by the use of the canonical injection $(x_1, \ldots, x_n)^T \rightarrow (x_1, \ldots, x_n, 1)^T$. Conversely, one can recover the affine coordinates of a point from its homogeneous ones by the mapping,

$$(x_1, \ldots, x_{n+1})^T \doteq \left(\frac{x_1}{x_{n+1}}, \ldots, \frac{x_n}{x_{n+1}}, 1\right)^T \rightarrow \left(\frac{x_1}{x_{n+1}}, \ldots, \frac{x_n}{x_{n+1}}\right)^T \text{ for } x_{n+1} \neq 0 \ ,$$

where $\doteq$ denotes the equality-up-to-scale property of the projective coordinates. From this chapter on, we will be using the tilde symbol on top of a vector letter (ex. $\widetilde{\mathbf{x}}$) to denote the projective coordinates of a given point. This notation will not be used if there is no risk of confusion with the affine counterparts.

If the last coordinate of a point $\widetilde{\mathbf{x}} \in \mathbb{P}^n$ is null, i.e., $x_{n+1} = 0$, then $\widetilde{\mathbf{x}}$ is called *point at infinity*. The direction of such point is given in the affine space by $(x_1, \ldots, x_n)^T$. Under the framework of projective geometry, the set of all points at infinity behaves like any other hyperplane, thus called hyperplane at infinity.

Definition 2 (Collineation) *A linear transformation or* collineation *of a projective space* $\mathbb{P}^n$ *is defined by a non-singular* $(n+1) \times (n+1)$ *matrix* A.

The matrix A performs an invertible mapping of $\mathbb{P}^n$ onto itself, and is defined up to a non zero scale factor. The usual representations for a collineation are $\lambda \mathbf{y} = A\mathbf{x}$ or $\mathbf{x} \overline{\wedge} \mathbf{y}$.

2.1.2 Image Formation

The most commonly used camera model in computer vision is the pinhole model, depicted on Figure 2.1. This is a simple and effective way of modelling most of the modern CCD cameras by considering the projection of rays of light passing through a small hole and being projected on a flat surface.

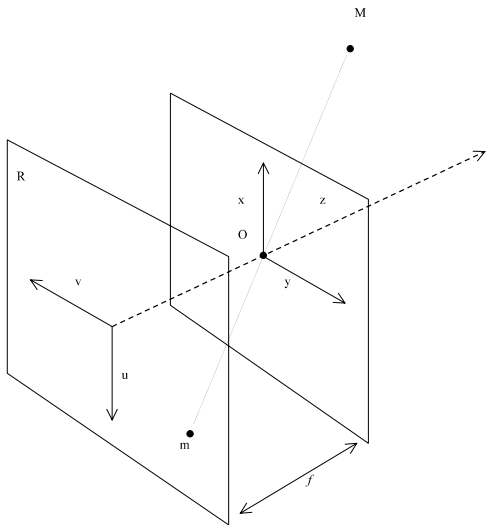


Figure 2.1: Perpective Camera Projection.

The image of the 3-D point M undergoes a *perspective projection*, passing through the *optical center* O, and is projected on the *image plane* R. The distance f of the optical center to the image plane is called the *focal distance*. The line passing through the optical center and orthogonal to the retinal plane is called *optical axis*. The optical axis intersects the image plane in the *principal point*. The use of projective geometry allows for the perspective projection model to be described by a linear equation, which makes the model much easier to deal with. A camera can be considered to perform a linear projective mapping from the projective space $\mathbb{P}^3$ to the projective plane $\mathbb{P}^2$.

The Perspective Projection Matrix

The general form of the perspective camera, that maps 3–D world points $\left({}^{W}x,{}^{W}y,{}^{W}z\right)$ expressed in a world coordinate frame, into 2–D image points (u, v) is

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \cdot \begin{bmatrix} {}^{C}R_{W} & {}^{C}\mathbf{t}_{W} \end{bmatrix} \cdot \begin{bmatrix} {}^{W}x \\ {}^{W}y \\ {}^{W}z \\ 1 \end{bmatrix} . \tag{2.1}$$

The (3×3) matrix ${}^{C}R_{W}$ descries the rotation between the 3–D world and camera frames. The (3×1) vector ${}^{C}\mathbf{t}_{W}$ contains the coordinates of the origin of the 3–D world frame expressed in the camera frame. The (3×3) matrix K is upper triangular and depends only

on parameters which are internal to the camera, thus called *intrinsic parameter matrix:*

$$K \doteq \begin{bmatrix} f \cdot k_u & f \cdot k_\theta & u_0 \\ 0 & f \cdot k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

This matrix accounts for the fact that the origin of the image coordinates is not usually located at the principal point, but at the upper left corner of the image. Moreover, the scaling along the u and v axis is not necessarily the same. The parameters k_u and k_v are scaling factors (along u and v), (u_0, v_0) is the location of the principal point in the image coordinate frame, and f is the focal distance. The additional parameter k_θ gives the skew between axes. For most CCD cameras k_θ can be considered zero, on applications not relying on high accuracy calibration.

Let us now introduce the notion of *normalized coordinates* of a 3-D point projection. Let $\widetilde{\mathbf{m}}$ be a point projection such that $\widetilde{\mathbf{m}} \doteq P\widetilde{\mathbf{M}}$ where P can be expressed in the form of Eq. (2.1) ,as

$$P = K \cdot \begin{bmatrix} {}^{C}R_W & {}^{C}\mathbf{t}_W \end{bmatrix}.$$

Let P' be a camera matrix with the same extrinsic parameters but with intrinsic parameters such that K is the identity matrix. Then $\widetilde{\mathbf{n}} \doteq P'\widetilde{\mathbf{M}}$ are the normalized coordinates of $\widetilde{\mathbf{m}}$. It is easy to see that P' corresponds to a camera with unit focal length, principal point coincident with the origin of the image frame and no scaling or skewing along the axes.

Lens Distortion

The described camera model presents the useful property of being a linear projective transformation from $\mathbb{P}^3$ into $\mathbb{P}^2$ thus allowing a simple mathematical formulation. However the pinhole model is not accurate enough for applications requiring high accuracy, such as photogrammetry and accurate metrology, as it does not model systematic non-linear image distortion, which is present on most cameras. When performing lens modelling, there are two main kinds of distortion to be taken into account [117]: radial and tangential. For each kind, an infinite series of correction terms is theoretically required. However, it has been shown that, for most off-the-shelf cameras[1] and industrial applications, the

[1] By *off-the-shelf*, we consider the normally used general purpose cameras, as opposed to professional metric cameras used in photogrametry.

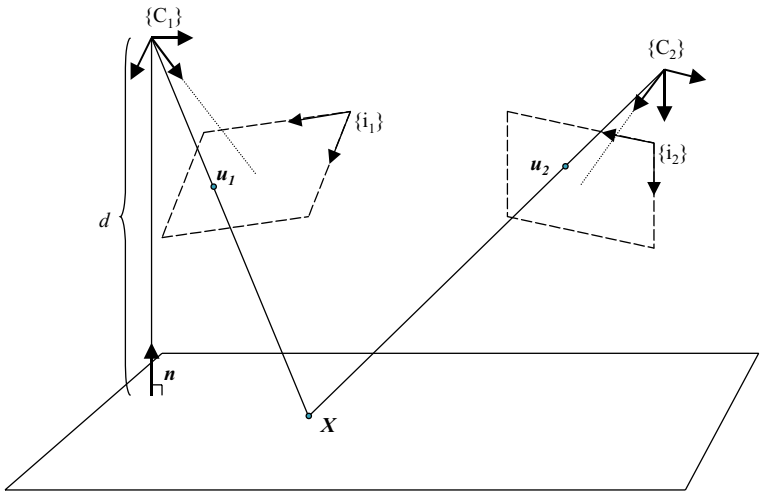


Figure 2.2: Geometry of two perspective projection cameras facing the same plane.

non-linearity can be dealt with just by using a single term of radial distortion. A four-step camera calibration procedure allowing radial correction was presented by Tsai in [117].

2.1.3 Geometry of Two Cameras Looking at a Plane

We will now show that two different views of the same planar scene in 3-D space are related by a collineation in $I\!P^2$, and how this collineation can be computed by the use of at least four pairs of matched points on the two images. This is an extensively used result (the reader is referred to [42] and the reference therein), that is here included for completeness.

Let P_1 and P_2 be two perspective projection matrices corresponding to two cameras imaging the same 3-D plane, as depicted in Figure 2.2. Without loss of generality, let us assume the 3–D world coordinate frame to be the one of the first camera. In this case the projection matrices can be written as

$$P_1 \doteq K_1 \left[\begin{array}{cc} I_3 & \mathbf{0} \end{array} \right]$$

$$P_2 \doteq K_2 \left[\begin{array}{cc} {}^2R_1 & {}^2\mathbf{t}_1 \end{array} \right]$$

where ${}^2R_1$ and ${}^2\mathbf{t}_1$ are the rotation and translation between the camera frames, expressed in the frame of the first camera. Let π be a plane not containing the cameras optical centers and defined by its normal vector $\mathbf{n}$ and perpendicular distance d_1 to camera 1 optical center. Also let $\mathbf{u}_1$ and $\mathbf{u}_2$ be coordinates of the image projections of the same

3–D point, visible in both cameras. In this case, $\mathbf{u}_1$ and $\mathbf{u}_2$ are related by a collineation in $I\!P^2$ represented by a 3×3 matrix H_{21} such that

$$\widetilde{\mathbf{u}}_2 \doteq H_{21} \cdot \widetilde{\mathbf{u}}_1$$

and

$$H_{21} \doteq K_2 \left({}^2R_1 + {}^2\mathbf{t}_1 \frac{\mathbf{n}^T}{d_1} \right) K_1^{-1} \ . \tag{2.2}$$

The above equation is described in [23], but will be here compactly derived. Let $\widetilde{X}$ be the homogeneous representation of the 3–D point living in the plane and projected on the two cameras as $\widetilde{\mathbf{u}}_1$ and $\widetilde{\mathbf{u}}_2$. Then $\widetilde{X}$ satisfies simultaneously the projection equation

$$\lambda \cdot \widetilde{\mathbf{u}}_1 = K_1 \left[\begin{array}{cc} I_3 & \mathbf{0} \end{array} \right] \cdot \widetilde{X}$$

and the plane restriction

$$\left[\begin{array}{cc} \mathbf{n}^T & -d_1 \end{array} \right] \cdot \widetilde{X} = 0 \ .$$

The two equalities can be grouped in the following system,

$$\left[\begin{array}{c} \lambda \cdot K_1^{-1} \cdot \widetilde{\mathbf{u}}_1 \\ 0 \end{array} \right] = \left[\begin{array}{cc} I_3 & \mathbf{0} \\ \mathbf{n}^T & -d_1 \end{array} \right] \cdot \widetilde{X} \ .$$

Given the assumption that the plane does not contain the camera centers, the right hand side matrix is invertible since $d \neq 0$. Therefore,

$$\widetilde{X} = \left[\begin{array}{cc} I_3 & \mathbf{0} \\ \frac{\mathbf{n}^T}{d_1} & -\frac{1}{d_1} \end{array} \right] \cdot \left[\begin{array}{c} \lambda \cdot K_1^{-1} \cdot \widetilde{\mathbf{u}}_1 \\ 0 \end{array} \right] = \lambda \cdot \left[\begin{array}{c} I_3 \\ \frac{\mathbf{n}^T}{d_1} \end{array} \right] \cdot K_1^{-1} \cdot \widetilde{\mathbf{u}}_1 \ .$$

By projecting $\widetilde{X}$ on the second camera, one gets

$$\widetilde{\mathbf{u}}_2 = \lambda \cdot K_2 \cdot \left[\begin{array}{cc} {}^2R_1 & {}^2\mathbf{t}_1 \end{array} \right] \cdot \left[\begin{array}{c} I_3 \\ \frac{\mathbf{n}^T}{d_1} \end{array} \right] \cdot K_1^{-1} \cdot \widetilde{\mathbf{u}}_1$$

from which is the collineation of equation (2.2) follows

$$\widetilde{\mathbf{u}}_2 \doteq K_2 \left({}^2R_1 + {}^2\mathbf{t}_1 \frac{\mathbf{n}^T}{d_1} \right) K_1^{-1} \cdot \widetilde{\mathbf{u}}_1 \ .$$

Linear estimation of planar transformations

The computation of a planar collineation requires at least four pairs of corresponding points. If we have more than four correspondences, then a least-squares solution can be found in the following manner. Let H be the collineation relating two image planes from

which we have a set of n correspondences such that $\widetilde{\mathbf{u}}'_i \doteq H \cdot \widetilde{\mathbf{u}}_i$, for $i = 1, \ldots, n$. For each pair we will have two linear constraints on the elements of H. An homogeneous system of equations can thus be assembled in the form

$$L.\mathbf{h}_l = 0 \tag{2.3}$$

where $\mathbf{h}_l$ is the column vector containing the elements of H in a row-wise fashion, and L is a $(2n \times 9)$ matrix

$$L = \begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u'_1u_1 & -u'_1v_1 & -u'_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -v'_1u_1 & -v'_1v_1 & -v'_1 \\ & & & & & \vdots & & & \\ u_n & v_n & 1 & 0 & 0 & 0 & -u'_nu_n & -u'_nv_n & -u'_n \\ 0 & 0 & 0 & u_n & v_n & 1 & -v'_nu_n & -v'_nv_n & -v'_n \end{bmatrix} . \tag{2.4}$$

From Eq. (2.3) it can be seen that $\mathbf{h}_l$ is the null space of L, thus defined up to scale. To avoid the trivial solution $\mathbf{h}_l = 0$, one has to impose an additional constraint, usually $\|\mathbf{h}_l\| = 1$[2]. Furthermore, real applications are prone to inaccuracies on the measurements of point locations and L will not be rank deficient. In order to find a least-squares solution for this equation, we can formulate the classical minimization problem:

$$\mathbf{p}_l = \arg\min_{\mathbf{p}_l} \|L \cdot \mathbf{p}_l\| \quad \text{subjected to} \quad \|\mathbf{p}_l\| = 1 \ . \tag{2.5}$$

By the use of the Lagrange multipliers it can be easily shown that the solution to this problem is the eigenvector associated with the smallest singular value of H. A suitable algorithm for finding the eigenvector is the Singular Value Decomposition (SVD) [87].

The most general collineation in $I\!P^2$ has eight independent parameters. As it has been shown, it accounts for the perspective mapping of a planar scene to the image plane of a camera. As explained above, it can be used to register two different images of the same plane, from point correspondence data. If the scene is static but not planar, then there will be image misalignments due to parallax, except for the case where the cameras have the same optical center. This is the same as to say that the cameras are free to rotate in any direction and to zoom, but not to translate. Table 2.1 summarizes the two cases where the general collineation captures the exact mapping between point projections.

[2] Alternatively, one can remove the scale ambiguity by setting one of the entries of H to a non–zero value (for example $H_{33} = 1$), rewrite equation 2.3, where L will be a $(n \times 8)$ matrix and solve it accordingly. However, this solution is more restrictive, as it fails to represent the case where $H_{33} = 0$.

	Scene assumptions	Camera assumptions
Case 1	Arbitrary 3-D	free to rotate on any directions and to zoom
Case 2	Planar	no restrictions on camera movement

Table 2.1: The two parallax-free cases for static scenes.

2.1.4 Restricted Collineations

If the camera motion or intrinsic parameters are constrained, then the collineation may be parameterized by less than eight independent parameters (of the general case), and still accurately describe the image motion. An example is the case of a down–looking camera attached to an underwater vehicle with stable pitch and roll motion, and observing an horizontal seabed at a constant altitude. Under the assumption of equal intrinsic scaling factors $k_u = k_v$ and small perturbations in pitch and roll, the induced image motion is simply described by a 2–D translation and rotation. If a simpler motion model can be used, then there is a clear advantage to do so, in terms of estimation sensitivity to outliers and noise in the data [113]. The use of several motion models for the same image sequence is illustrated in Chapter 4, where the advantage of using the most adequate, instead of the most general, is clear.

The restricted image motion models considered in this thesis are presented in Table 2.2. The models where the calibration matrix K is explicitly shown, assume constant intrinsic parameters. In Chapter 4, the similarity and the 2–D rotation and translation model are employed for the early stages of mosaic creation. Next, the 3D rotation and translation is used to obtain an accurate final registration. Finally, a special case of the 3D pure rotation model is used to render a fronto–parallel view of the seabed. In Chapter 6, the similarity model is used for the real–time tracking of the live camera images on the mosaic map.

The 2–D rotation and translation, the 3–D pure rotation and the 3–D rotation and translation models require a iterative estimation algorithm due to the structure of the rotation matrices. The 3–D rotations are parameterized by the X-Y-Z fixed angle convention [12],

$$R_{3D}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}$$

Motion Model	Matrix form	p	Domain
2–D Rotation and translation	$H = \begin{bmatrix} \cos\alpha & \sin\alpha & t_u \\ \sin\alpha & \cos\alpha & t_v \\ 0 & 0 & 1 \end{bmatrix}$	3	Image plane parallel to the planar scene at constant distance. Rotation on the camera axis and translation perpendicular to it.
3–D Rotation	$H = K \cdot R \cdot K^{-1}$	3	3–D rotation but no translation. Known intrisics.
Similarity	$H = \begin{bmatrix} t_1 & t_2 & t_3 \\ -t_2 & t_1 & t_4 \\ 0 & 0 & t_5 \end{bmatrix}$	4	Image plane parallel to the planar scene. Rotation only on the camera axis but free 3–D translation.
Affine Transformation	$H = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ 0 & 0 & t_7 \end{bmatrix}$	6	Distant scene and small field of view.
3D Rotation and translation	$H = K\left(R + \mathbf{t}\frac{\mathbf{n}^T}{d}\right)K^{-1}$	6	Most general 3–D rotation and translation. Known intrisics and plane inducing the homography.
General Projective	$H = \begin{bmatrix} t_1 & t_2 & t_3 \\ t_4 & t_5 & t_6 \\ t_7 & t_8 & t_9 \end{bmatrix}$	8	Most general planar transformation.

Table 2.2: Description of the models used for image merging, ordered by the number of free motion parameters p.

where $s(.)$ and $c(.)$ represent the sine and cosine functions of the rotation angles. The estimation is initialized with values obtained from the linear estimation, using the similarity model for the 2–D rotation and translation and the full planar for the two other.

2.1.5 Recovering Camera Motion from the Collineation

One of the early references to the problem of recovering the 3–D motion from two images of a planar surface is Tsai and Huang [118], where it is shown that the solution is not unique. In the most general case there are eight different sets of solutions, as described by Faugeras and Lustman in [23]. However, only two are feasible if one considers the world plane to be non-transparent. A closed form relationship between the two solutions is given by Negahdaripour in [74]. In our work we follow the method of Triggs [115], where the two solutions are found using the SVD of $M_{21} = K^{-1}H_{21}K$.

It can easily be seen from Eq. (2.2) that, for the case where there is no translation, the dependency of the collineation on the planar structure is lost. In practical terms, this is also true for the case where the scene is distant enough for the ratio $\frac{t}{d_1}$ to be close to zero. This has motivated linear selfcalibration techniques, both for constant intrinsic cameras [41], as well as for the case of varying intrinsics [14]. Such techniques do not require any knowledge on the scene structure, nor the rotation of the camera frame between images. Therefore, they are specially suited to applications where on-line calibration is required and the camera can be rotated around its optical center.

The theory behind the method of self–calibration from cameras under pure rotation is presented by Hartley in [41]. For the case of a constant intrinsic matrix K the homography between corresponding points in two views is represented by

$$H_{21} = K \cdot {}^2R_1 \cdot K^{-1} \ . \tag{2.6}$$

This homography can be computed directly from image measurements, and depends only on the intrinsic parameter matrix and on the camera rotation ${}^2R_1$ between the two images. As noted in [41], H_{21} is only meaningfully defined up to scale, but taking into account the fact that the product $K \cdot {}^2R_1 \cdot K^{-1}$ has unit determinant, the exact equality $H_{21} = K \cdot {}^2R_1 \cdot K^{-1}$ will hold if H_{21} is scaled by an appropriate factor.

A linear system of equations, not depending on the rotation matrices, can be constructed on the elements of the symmetrical matrix $C = KK^T$, and solved using the SVD [41]. The recovery of K can be achieved by means of the Choleski decomposition[48] of

C, if C is positive-definite, which is the case for noise-free or moderately noisy data. Also, the decomposition is unique if K is assumed to have positive diagonal entries.

2.2 Robust Estimation

Model estimation (in the sense of model fitting to noisy data) is employed in computer vision on a large variety of tasks. The most commonly used method is the least-squares mainly due to the ease of implementation and fast computation.

The least-squares is optimal when the underlying error is additive and Gaussian [71]. However, in many applications the data are not only noisy, but also contain *outliers*, *i.e.* data in gross disagreement with the assumed model. Under a least-square framework, outliers can distort the fitting process to the point of making the fitted parameter arbitrary. As pointed out in [114], this can be particularly severe if the non-outlying data are degenerate or near-degenerate with respect to the expected model. In such case outliers can mask the degeneracy, making the adequacy of the postulated model hard to evaluate.

According to Meer *et al.* [71], there are three concepts usually employed in evaluation a robust regression method: the relative efficiency, the breakdown point and the time complexity. The *relative efficiency* is defined as the ratio between the lowest achievable variance for the estimated parameters and the actual variance provided by the given method. The *breakdown point* is the smallest proportion of outliers that can force the estimated parameters outside an arbitrary range. For the least-squares estimation, the breakdown point is 0 since just one outlier is required for corrupting the estimated values. The *time complexity* can be defined from the number of arithmetic operations required by the algorithm.

2.2.1 Random Sampling Algorithms

A robust regression method was proposed by Rousseeuw, called the *least-median-of-squares* [91] (LMedS). Let us consider the generic unconstrained estimation problem, where we want to obtain an estimate $\widehat{\theta}$ from noisy observations $\widehat{X}$, using the observation equation $F(\theta) = X$ that relates the noise–free parameter θ and data X vectors. Using the LMedS, and estimate is obtained by solving

$$\widehat{\theta} = \arg\min_{\theta} \operatorname*{med}_{i} \left(F_i(\theta) - \widehat{X}_i \right)^2$$

where $F_i(\theta)$ and $\widehat{X}_i$ are the i^{th} entry of vectors $F(\theta)$ and $\widehat{X}$ respectively.

As pointed out in [71], this minimization problem cannot be reduced to a least-squares based solution. The minimization on the space of all possible solutions is usually impracticable. Therefore it is common practice to use a Monte Carlo technique [54] and analyze only randomly sampled subsets of points. The number of samples to be performed may be chosen as to insure a high probability of selecting an outlier-free subset. The expression for this probability P_f is

$$P_f = 1 - (1 - (1 - \varepsilon)^p)^m \tag{2.7}$$

for m samples of size p, taken from a data set where the fraction of outliers is ε. From this expression, it can easily be seen that the number of samples is not directly linked to the absolute number of outliers, but just with its proportion. Clearly, this is also true for the time complexity of the sampling algorithm. The expression also implies that the less data point are used for instantiating the model, the less samples will be required for the same P_f.

The random sampling greatly reduces the time complexity of the basic LMedS, from $O(n^{p+1} \log n)$ to $O(nm \log n)$ [71] where n is the size of the data set, while keeping the breakdown point of 0.5. In spite of the high breakdown point, the relative efficiency of the LMedS method is low when Gaussian noise is present in addition to outliers. Therefore an association of LMedS with weighted least-squares which has high Gaussian efficiency, can be used, as proposed by Rousseeuw [91].

Another robust estimator, based on random sampling, is the *Random Sampling Consensus* (RANSAC). It was proposed by Fishler and Bolles [25] in 1981, and originally used in the context of computer vision, for automated cartography. The RANSAC is based on the following paradigm. The estimation is performed on a subset of data points sampled from all the available points, such that the subset has the minimum number of elements required for instantiating the model. All the data is then evaluated according to the instantiated model. For a given error threshold, the points are classified as being part of the consensus group of the model if they are within the threshold. This process is repeated until a sufficiently large consensus group is found (or a maximum number of iterations is reached). The final estimation is performed on the largest consensus group found. The RANSAC requires therefore, the specification of three algorithm parameters: the error threshold for evaluating the compatibility with the model, an estimate of the cardinality consensus set for checking if a sufficiently supported model has been found, and a maximum number of samples to try.

Although developed independently, LMedS and RANSAC are based on similar concepts. According to Meer *et al.* [71], the main difference lies in the fact that the LMedS generates the error measure during estimation, while RANSAC requires it beforehand. An in depth comparison of LMedS and RANSAC can be found in [71]. The two methods are also compared in [114], for the estimation of the fundamental matrix.

2.2.2 A Two-Step Variant of LMedS

Several variants to these random sampling algorithms have been proposed in the literature, in the context of Computer Vision. As an example we can point out the "empirically optimal algorithm" presented in [114], which combines LMedS and M-estimators.

In the work presented in this thesis, we have used a two-step variant of LMedS, which exhibits a similar breakdown point [71], but requires less random sampling in order to achieve the same degree of outlier rejection. The algorithm comprises two phases of random sampling LMedS. After the first phase, the data set is reduced by selecting the best data points in the sense of the chosen cost function. Next, the reduced data undergoes another random sampling LMedS phase.

Let S_{total} be a set of N matched points projections possibly containing outliers $S_{total} = \{(\mathbf{u}_1, \mathbf{u}'_1), \ldots, (\mathbf{u}_N, \mathbf{u}'_N)\}$. The computation of the homography H from S_{total} is performed by the following operations :

1. Randomly sample a set of p pairs taken from S_{total}, where p is the minimum number of matched points required to instantiate the model.

2. Estimate the H matrix and compute the median of the point squared distances for S_{total},
$$\underset{i}{\text{med}} \left[d^2 \left(\widetilde{\mathbf{u}}_i, H \cdot \widetilde{\mathbf{u}}'_i \right) + d^2 \left(\widetilde{\mathbf{u}}'_i, H^{-1} \cdot \widetilde{\mathbf{u}}_i \right) \right]$$
The function $d(\cdot, \cdot)$ takes two point locations in homogeneous coordinates and returns their (affine) distance on the 2–D image plane,
$$d \left(\widetilde{\mathbf{u}}, \widetilde{\mathbf{u}}' \right) = \sqrt{(u - u')^2 + (v - v')^2} \tag{2.8}$$
where $\widetilde{\mathbf{u}} = \begin{bmatrix} \lambda u & \lambda v & \lambda \end{bmatrix}^T$ and $\widetilde{\mathbf{u}}' = \begin{bmatrix} \lambda' u' & \lambda' v' & \lambda' \end{bmatrix}^T$ with $\lambda \neq 0$ and $\lambda' \neq 0$.
If the median is below a given threshold m_T, return H and exit.

3. Repeat 1. and 2. for a specified number of samples m_1.

4. Select the H_{best1} for which the minimal median was found, and sort the matched points by their sum of the point distance squares, using H_{best1}.

5. Create the set S_{best} with the elements of S_{total} whose distance is below the median.

6. Repeat 1. and 2. on S_{best} for a m_2 number of samples.

7. Select the homography H_{best2} corresponding to the lowest median.

8. For H_{best2} select the matched points whose average distance,

$$\frac{1}{2}\left[d\left(\widetilde{\mathbf{u}}_i, H_{best2}\cdot\widetilde{\mathbf{u}}'_i\right) + d\left(\widetilde{\mathbf{u}}'_i, H^{-1}_{best2}\cdot\widetilde{\mathbf{u}}_i\right)\right] \tag{2.9}$$

   is less or equal to a specified distance threshold d_T.

9. Compute and return the final H using simple least-squares with all the selected matched points above.

The required parameters are the number of samplings on each part m_1 and m_2, the median threshold, and the distance threshold. If the ratio of outliers in the data is known, then m_1 can be set according to Eq. 2.7 to ensure a specified probability of selecting a outlier–free sample, during the first random sampling step of the algorithm[3]. For the image registration application of our work, the ratio of outliers is not constant. It depends upon factors such as image texture contents and degree of overlap, that vary to a large extent. For this reason, we have opted to define m_1 and m_2 based on processing time constraints. The real–time matching implementations for the results in Chapters 4 and 6 use 100 and 50 samplings, respectively.

[3]The extention of Eq. 2.7 to deal with both m_1 and m_2 of our algorithm is not trivial, and was not carried out.

Chapter 3

Previous Work in Mosaic–based Navigation

This chapter reviews some of the most relevant Computer Vision techniques related with mosaic–based navigation. The main purpose is to provide comparative information on the contributions from other authors to this area. This allows for the better understanding of this thesis work with respect to what has been accomplished before.

The chapter is divided into three sections. The first addresses the techniques related to image registration and mosaic construction. The second specializes on the use of mosaics for robot navigation, both in land and underwater applications. Finally, a closing section discusses the most important features that are desirable on a mosaicing system intended for navigation.

3.1 Mosaic Construction

Over the last decade there has been a growing interest in video mosaicing techniques, as a way of creating useful scene representations. Current applications cover diverse areas such as video coding for low bit-rate transmission, super-resolution for forensic video enhancement, environmental representations for robot navigation and wide field-of-view panoramas for commerce and tourism over the internet.

A central problem to the mosaicing process is image matching, which can be stated as follows.

Given two images of the same scene, for each point projected in one image, find a corresponding point in the other such that they relate to the same 3–D scene point.

This problem closely relates to the *image motion estimation*, in which a parametric representation is used to model the motion of the point projections from one frame to the other. This parametric representation usually employs a small number of parameters to represent the global image motion, and is adapted to the particular camera motion and scene structure constraints. The image motion estimation constitutes an important step in mosaicing applications, as it allows for images to relate spatially over a common frame.

3.1.1 Motion Estimation

The commonly used methods for estimating the motion between frames can be categorized according to several criteria. One is the representation domain, which distinguishes the methods that operate either on frequency or on spatial representations. Another criterion separates continuous from discrete approaches [123] depending on the temporal assumption for the time instants of the image capture. This criterion is associated with the common division between *feature–based* and *optic flow* methods. Recently, a detailed taxonomy of stereo matching techniques was presented in [99], where the most widely used algorithms are evaluated and compared.

i) Frequency Domain Methods

Frequency–based methods take advantage of the properties of the 2–D Fourier Transform regarding translation, rotation and scale. As an example, for images that are simply shifted, the estimation of the translation can be attained by locating the peak of the inverse Fourier Transform of the cross–power spectrum phase [60]. This approach has been extended to deal with rotation and scaling [88], and full affine transformations [59] by aligning the images Fourier spectra over several resolution levels.

Since this class of methods use frequency information over the complete spectrum, they are well suited for applications where the images are corrupted with narrow bandwidth noise. However, for cases where the noise is spread across the spectrum, area cross-correlation methods are more adequate [7].

ii) Spatial Domain Methods

The set of techniques that explicitly use the spatial (geometric) relations are commonly referred to as *spatial domain methods*. These can be further divided into two groups,

depending on whether explicit image points are singled out in the motion estimation process.

Correlation–based methods The first group of *correlation–based* methods usually involves the extraction of a set of *features* from the images, such as object corners, line segments or curves. These features are usually sparse, when compared with the extent of the underlying images, but are representative of the objects appearance or shape.

The choice of the type of feature to use is usually dependant on the availability of such features in the images and on the reliability of their measurement. As an example, straight lines are difficult to observe in underwater scenes whereas they are very common in man–made environments. The estimation of lines is typically less sensitive to image noise when compared to points, but since lines have an extra degree of freedom, more correspondences are required to uniquely determine image motion. As an example, it has been shown in [123] that 3–D motion cannot be recovered from line correspondences of just two views.

A wide variety of corner point detectors can be found in the literature [100]. These can be divided into three categories: contour based, intensity based and parametric model based methods.

After the feature extraction process, this class of methods requires the establishment of correspondences over the images. In practical applications, this *matching* step is very prone to gross errors and has been at the core of intense research over the last decade. Successful approaches to deal with this problem include the use of robust methods over geometric constraints [114], or concave minimization over rigidity [63]. A commonly used match metric for point features is the normalized cross-correlation [52], which serves as a similarity measure between image patches around the selected points.

This group of correlation–based methods also includes the case of *dense correlation*, for applications where a very large number of points need to be matched, such as dense tri–dimensional reconstruction [2].

Optical flow methods The second class of methods, commonly referred to as the *optical flow* approach [47, 46, 5], is based on the computation of the velocity field of brightness patterns in the image plane.

As opposed to the feature-based approach, the optical flow does not require a matching

process, but suffers from the generalized aperture problem [3, 6, 53]. According to Black and Anandan [6], most of the current techniques for the estimation of the optical flow are based on two image motion constraints: data conservation and spatial coherence. The first arises from the observation that the intensity patterns of the surfaces of the world objects remain constant over short intervals of time, although their image position may change. The second assumes that the surfaces have spatial extent, thus making neighboring pixels likely to belong to the same surface. This is usually implemented in the form of a smoothness constraint on the motion of spatially close pixels.

The generalized aperture problem refers to the dilemma of choosing the appropriate size for the area of analysis (aperture) R. In order for the motion estimation to present some insensitivity to noise and be constrained [6], a large R is desirable. However, the larger the aperture is, the less realistic the data conservation and the spatial coherence become. One other problem with optical flow techniques lies on the fact that it is only possible to determine the flow in the direction of the image brightness gradient. The optical flow along this direction is therefore perpendicular to the image contour, hence called normal flow. The flow component along the contour cannot be established directly from the brightness patterns, without resorting to additional constraints such as smoothness or second order derivatives. This condition is referred to as the aperture problem [47, 46, 5].

The original formulation of optical flow as defined by Horn and Shunck in [47] was extended by Negahdaripour [75] to take into account both geometric and radiometric models.

For some applications, the choice of the approach, either feature-based or optical flow, is not trivial. This statement is supported by the large amount of research in the last few years using the two approaches as a starting point for higher level image interpretation. Optical flow has successfully been used on tasks such as egomotion estimation [103, 104], motion segmentation [3] and image registration [96, 109], whereas feature-based approaches have proven adequate for 3-D reconstruction [22, 66] and image registration as well [128].

3.1.2 Motion Models

As detailed in Chapter 2, the most general model for image motion of planes has eight degrees of freedom (dof). However, under certain assumptions such as simplified image motion or unitary pixel aspect ratio in the cameras, a restricted model may be sufficient.

In such cases, the motion estimation with a restricted model is less sensitive to noise. Also, it may lend itself to a faster implementation by using, for instance, an FFT-based correlation [7]. Examples of restricted model applications are a 4 dof model to register high altitude aerial images [128] and document scanning.

Different motion models with 8 or more parameters have been used in the mosaicing literature. An example of such are the bilinear and biquadratic models [65, 69]. The later has 12 dof and can be obtained by taking the second order polynomial approximation of the most general projective mapping. It can be written as

$$u' = q_1 \cdot u^2 + q_2 \cdot uv + q_3 \cdot v^2 + q_4 \cdot u + q_5 \cdot v + q_6$$
$$v' = q_7 \cdot u^2 + q_8 \cdot uv + q_9 \cdot v^2 + q_{10} \cdot u + q_{11} \cdot v + q_{12}$$

where (u', v') and (u, v) are corresponding point projections. Although the biquadratic has more degrees of freedom than the full projective, it fails to correctly model the most general projective mapping. The use of the biquadratic has been justified as being computationally less demanding when used with optical flow techniques, while being able to model (to some extent) the effect of converging lines and chirping [65].

3.1.3 Global Registration

Some applications require the registration of a large set of views of the same scene. This is the case of the underwater video mapping described in this thesis. Most commonly the image registration is performed by pair–wise image registration in chronological order. This is motivated by typical high overlap between time-adjacent image frames, which provides a large region of support for the motion estimation. The estimates are then concatenated to infer the relation between any pair of images. However, even small amounts of noise in the estimation process may result in large accumulated error. This is most noticeable if the image sequence contains regions of the scene that have been captured some time before, such as loop camera trajectories.

A number of authors have tackled the problem of registration for camera loop trajectories in order to create spatially coherent mosaics.

Topology Estimation

Sawhney *et al.* [97] proposed an end–to–end solution for image mosaicing where the image topology (*i.e.* the spatial relations between overlapping frames) is iteratively estimated.

Spatial consistency is improved by identifying and registering images with large superposition. Their formulation allows for the creation of planar and spherical mosaics, given the appropriate parametric mapping models from the mosaic surface to the image surface.

A simpler approach is followed by Davis [13] for registering images captured with no translation. Under small rotation and some assumptions on the camera intrinsic parameters, phase-correlation methods are used for pair-wise registration. A system of linear equations is defined for the elements of all the homographies relating each image with a reference image for which a least-squares solution is obtained. However, no adequate parameterization is used on these elements to take advantage of the special structure of the rotation-induced homography. Duffin and Barrett [16] use a homography parameterization for global registration that imposes constant camera skew and aspect ratio. Other constraints on the camera and scene geometry are not taken into account.

Recently Unnikrishnan and Kelly [120, 119] addressed the problem of efficiently distorting strip mosaics in order to close loops in a smooth way. The proposed solution has low computational complexity and is best suited for the case where the number of temporally distant overlaps is small compared to the adjacent ones. The problem of finding correspondences at the extremities of mosaic segments, required for imposing end–pose constraints, was not addressed.

Bundle Adjustment

Bundle adjustment techniques from the photogrammetry literature have been successfully adapted to image registering applications. In the context of camera self–calibration, Hartley [41] used a bundle adjustment technique to simultaneously estimate the homographies arising from several views. The homographies were meaningfully parameterized for pure rotation which required a non-linear estimation algorithm. McLaughlan and Jaenicke [70] illustrated the use of these optimization techniques for mosaicing, using both matched points and lines in a semi-automatic system. No topology is performed, which limits applicability to small, high overlapping mosaic. A related approach is followed by Capel [8] who presents a complete mosaicing system with topology estimation and global registration, by extending a Maximum Likelihood estimation for the 2-view homographies to the case of multiple views.

3.1.4 Notable Mosaicing Approaches and Applications

Early applications of image mosaicing were primarily interested in developing visualization tools, capable of providing extended views. An example is the composition of aerial images [128]. Recently these techniques started being regarded as efficient representation, on applications such as video compression, enhancement and search. In [51, 107] and more recently in [83], the idea of using mosaics for complete scene representation is addressed with the intent of fully recovering the video sequence from a *dynamic mosaic*. This dynamic mosaic is an extension to the usual static mosaic, comprising three elements:

- a (static) background mosaic. Static mosaics have also been called *salient stills* [109, 69].
- the set of frame transformations relating each frame to the mosaic frame.
- the image residuals containing the brightness differences of each frame to background mosaic.

Further details on mosaic classification can be found in [51], where a detailed taxonomy is proposed in the context of video applications.

High compression video coding can be attained by creating and coding the dynamic mosaic. Most video sequences tend to have a large amount of image overlap, and much of the image redundancy is due to a static background. In such cases, the dynamic mosaic residuals are small, when compared to the residuals between consecutive frames, even if motion compensation is performed [51]. Mosaic-based video coding requires, however, that whole sequence be available for the estimation of the background. For this reason it is suited for off-line coding and storage.

Mosaicing techniques can be helpful in creating compact visual representations of the complete surroundings of a particular viewpoint. Hemispherical mosaics have been used by Kang and Szeliski [57] to represent the views in any direction in order to perform wide–baseline 3–D reconstruction. A similar application was addressed by Coorg and Teller [11]. Shum and Szeliski [102] present an end–to–end solution for creating panoramas, where global alignment is imposed and the camera focal length is estimated.

A general Bayesian framework was used by Dellaert *et al.* [15], capable of producing a maximum a posteriori estimate of the mosaic image. During the mosaicing process the camera intrinsic parameters are estimated together with the globally–optimized motion

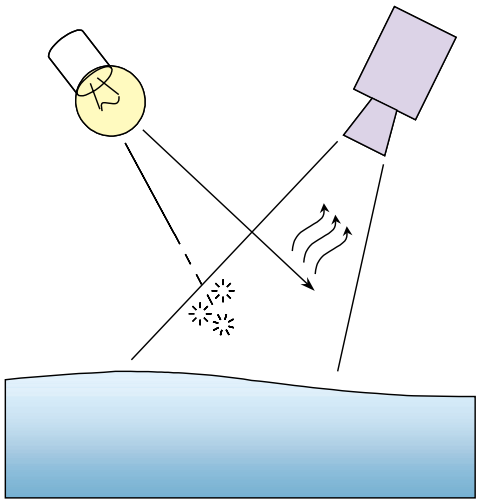


Figure 3.1: Two common sources of underwater image degradation: Scattering and Absortion.

parameters. The used observation equation takes into account the camera lens distortion and permanently occluded image regions.

3.1.5 Underwater Mosaicing

The subsea medium constitutes a challenging environment for computer vision. When compared with land and aerial applications, the light underwater is subjected to intense scattering and attenuation. Scattering refers to the angular spread of the light rays due suspended particles in the water, while attenuation is the power loss in the medium. These phenomena are schematically illustrated in Figure 3.1.

Another contributing difficulty is illumination. Most often, artificial lighting is required for depths above 10 to 20 meters, depending on the water turbidity. The light source is usually assembled close to the imaging devices. This condition creates strongly non-uniform lighting and affects the visual appearance of the scene as the camera moves around [81].

These factors severely limit underwater imagery in terms of contrast, definition and range. Under such conditions, video mosaicing presents itself as the natural solution to obtaining large visual representations of the sea floor. This is achieved by registering many close–range images.

Intense research on automatic mosaic creation for underwater applications has been

conducted in the last few years. One of the early references is the work of Haywood [43] where a setup is described for the creation of sea–floor mosaics. The problem of image registration is completely avoided by capturing images at precisely known locations. Image composition can thus be performed straightforwardly, since image motion is computed directly from the camera positions.

A real–time system for the creation of ocean floor mosaics was jointly developed by the Stanford University and the Monterey Bay Aquarium Research Institute. The problem of non-uniform lighting and marine snow was addressed by an image pre–processing step prior to image correlation. The processing consists in computing the image of the sign of the Laplacian of the original images, after being low–pass filtered by a Gaussian kernel [67]. This results in a binary images which can be registered at frame rates using special–purpose correlation hardware. Real-time operation was achieved for the creation of "single column" mosaics [68]. A very restrictive motion model is assumed, accounting just for image translation. This work is extend by Fleischer [26] to deal with loop trajectories, by detecting trajectory crossover in previous mosaiced areas. In such cases the mosaic is re–aligned using an augmented–state Kalman filtering. However, the same registering method is used, thus implicitly restricting the visually sensed motion to 2 dof, without rotation nor scale change.

The topic of motion estimation has been extensively addressed at the Underwater Vision and Imaging Laboratory of the University of Miami. Among other capabilities such as automatic station keeping [80], a mosaicing approach was developed using direct methods. These methods allow for the incremental estimation of 3D motion directly using the spatio–temporal derivatives of images captured closely in time [77]. Also a generalized dynamic image motion model [75] is used which accounts for variations in the scene radiance due to lighting and medium conditions. This is of particular importance when using flow–based methods in underwater imagery. Results on mosaicing are reported which real–time operation is achieved using standard computing hardware for motion models of 3 dof [75] and 4 dof [125]. The need for assessing the performance of the most common mosaicing approaches was addressed by Negahdaripour and Firoozfam in [76]. In here, comparative results are reported for both feature–based and direct methods for long image sequences with ground–truth.

Research in underwater mosaicing has been conducted at the Woods–Hole Oceanographic Institution, primarily as a visualization tool for the oceanic floor [105]. In [19],

Eustice *et al.* perform image registration by searching for the motion parameters that minimize the intensity differences between warped version of the images. This method mirrors the approach of Sawhney and Kumar [98], but uses some additional photometric processing such as histogram equalization.

Work on feature tracking with applications on underwater image registration and mosaicing, has been conducted at Heriot-Watt University. In [28], Fusiello *et al.* extended the feature tracker proposed by Tomasi and Kanade [101], by introducing an automatic scheme for detecting and discarding spurious features. An illustrative application to mosaicing is presented in [86].

A feature based approach has been developed at Girona University, in which texture cues are used to improve the matching efficiency [30]. The problem of looping trajectories is dealt with using a Kalman Filter approach with an augmented state vector [31].

Frequency domain methods have been applied to underwater image registration at the University of New Hampshire [93]. In order to increase the lighting homogeneity in the captured image frames, a filtering step is used to enhance the high frequency content, thus equalizing the background pixel intensity. The Fourier–Mellin transform [92] is used to register image frames, assuming no perspective distortion effects.

3.2 Mosaic Navigation

3.2.1 Land applications

One of the early references to the idea of using mosaics as visual maps is the work of Zheng and Tsuji [127], where panoramic representations were applied to route recognition and outdoor navigation. However the visual representations do not preserve geometric characteristics nor correspond to visually correct mosaics. This constitutes a drawback as the representation is not fit for human perception, which is important for mission definition.

A tour–guide robot is described by Thrun *et al.* [110, 111] which combines the use of previously created occupancy maps and ceiling mosaics for localization. The occupancy maps are used to measure and compensate for the error in the odometry, and provide a global (coarse) position estimate. Such estimate is refined by the up–looking vision sensor. The map building and localization are addressed under the probabilistic framework of the concurrent mapping & localization [112] approaches. Details on the mosaicing algorithm

are given by Dellaert *et al.* in [15].

Recently, Kelly [58] has addressed the feasibility and implementation issues of using large mosaics for robot guidance, predicting a large impact of these techniques on industrial sites. These environments are usually structured enough to allow for restricted motion models to be used. Experiments are reported for long strips of linear mosaics where it is assumed that the image plane is parallel to the mosaic areas and the motion of the vehicles is restricted to the ground plane.

3.2.2 Underwater applications

In the context of real-time concurrent mapping and localization, Xu [125] investigated the use of seafloor mosaics, constructed using temporal image gradients. Although there is a careful compensation of systematic errors [78], possible loops in the camera path are not exploited for reducing the accumulated error, which limits the use for covering large areas. Mosaic navigation as an extension of station–keeping is presented in [79] for a 3 dof floating robot.

Huster [49] described a navigation interface using live-updated mosaics, and illustrated the advantages of using it as a visual representation for human operation. However, as the mosaic is not used in the navigation control loop, there is no guaranty the vehicle is driven to the desired position.

Fleischer [26] combined spatially consistent mosaic with underwater ROV navigation. However, in their approach, the navigation system requires additional sensors to provide heading, pitch, yaw and altitude information, whereas our work relies solely upon vision to provide information for all the relevant degrees of freedom.

Regarding navigation, our approach differs from the concurrent mapping and localization approaches (CM&L [112], SLAM) in the sense that the map is totally created prior to its use. Some authors have successfully implemented (and extended) CM&L for the underwater navigation with mosaics [125, 26], which has the advantage of using the mosaic while it is being constructed. However this leads to less accurate mosaics due to the simpler motion models and algorithms, motivated by the real–time constraints.

3.3 Discussion

From what is presented above, it can be seen that motion estimation and image mosaicing constitute two topics of intense and increasing research interest.

In the last few years, the methodologies associated with image mosaicing have been maturing rapidly, both in terms of robustness and speed. This is allowing the application domain to expand from simple proof–of–concept to the larger scale of real–world problems. In this thesis we are interested in exploring mosaics as environment representations capable of supporting autonomous navigation. This creates a set of requirements for the image mosaicing algorithms, that needs to be met in the overall system. The important features for a mosaicing system for navigation can be pointed out:

- *Fully automatic* – Early applications of image mosaics required the human–assisted selection of control points in correspondence. This has been effectively addressed, with high rates of success, using automated image registering and matching techniques both for feature-based approaches (e.g. [128]) as well as flow–based (e.g. [3]).

- *Capable of dealing with the visual information content of the application scenario and extracting reliable information* – Challenging environments such as underwater require the use of adequate visual cues. As an example, line extraction has been extensively used in indoor navigation but its use underwater is limited to man-made structures such as cable following [84]. Image texture also contains useful information that helps the matching process [30].

- *Robust to limited departures from the assumptions* – A standard underlying assumption is on the planar structure of the scene. In practical applications this is hardly the case, specially for natural environments where there is 3–D structure with non–rigid and non–static objects. Again, techniques from robust statistics can help overcoming such effects in order to accurately recover the most representative motion and identify outlying features or regions.

- *Capable of handling large areas* – Useful practical application scenarios require mobility over large areas, such as the debris area of a shipwreck. Current underwater mosaicing research publications report on much smaller results, with the exception of [19].

- *Capable of producing accurate results in useful time* – The accuracy on the mosaicing process is strongly dependant on two main factors: the appropriateness of the selected motion models and the computational resources available.

- *Capable of producing a statistical characterization of the results and detecting failures* – In order to use vision as a position sensor in an integrated, multi-sensor autonomous robot, the motion or pose estimation algorithms must also provide information on the associated uncertainty. Typically, this is presented in the form of a covariance estimate.

Almost all of the above requirements have been addressed separately in the literature. However, to the best of our knowledge, no attempt has yet been made in using large[1] mosaics for 3–D underwater navigation at sea. Previous reports on sea–bed testing have been restricted to illustrative proof–of–concepts on very small mosaics [26, 125]. More importantly, the algorithms employed for the mosaic creation seriously hinder their application to larger mosaics. The accuracy of Fleischer's approach [26] is strongly dependant on external sensors, since the vision–related processing is only capable of 2 dof translations. The method by Xu [125] cannot take advantage of looping trajectories with are essential to creating large, spatially coherent mosaics. Furthermore, the special structure of the image collineations that arise from having the same camera imaging one plane has not been used.

It is worth noting that the integration of several different sensors can be of great benefit for the robustness of the overall navigation system. This implies realistic measurement and modeling of the uncertainty associated with each sensor. However, it is of scientific relevance to know how far can underwater vision systems go when used alone, and have ways of computing the uncertainty of the pose. The work presented in this thesis is also directed towards this goal.

[1] by large we refer to larger than 10 square meters

Chapter 4

Mosaic Map Creation

This chapter addresses the problem of constructing high quality mosaics of the sea bed. An algorithm is presented for the simultaneous creation of mosaics and the estimation of the camera trajectory. Special attention is taken to the processing of long image sequences with time-distant superpositions, such as the ones arising from loop trajectories, or zig-zag scanning patterns.

The main novel aspect of this approach is the use of an adequate parameterization for the homographies that takes into account all the geometric degrees of freedom of the problem. Another aspect is the separation of the mosaicing algorithm in different stages, with distinct objectives.

4.1 Overview and Application Domain

The method for mosaic creation is here summarized, and detailed the next sections. The complete algorithmic flow of the process is shown in Figure 4.1.

The method comprises four major stages:

1. *Image motion between consecutive frames* is computed by robustly matching point features across pairs of images. This results in a set of sequentially ordered homographies. These homographies are cascaded in order to infer the approximate topology of the camera movement. The topology information will be used to predict the areas where there is image overlap resulting from non-consecutive images. This overlap is valuable in the sense it allows to further refine the motion estimation and the spatial correctness of the final mosaic.

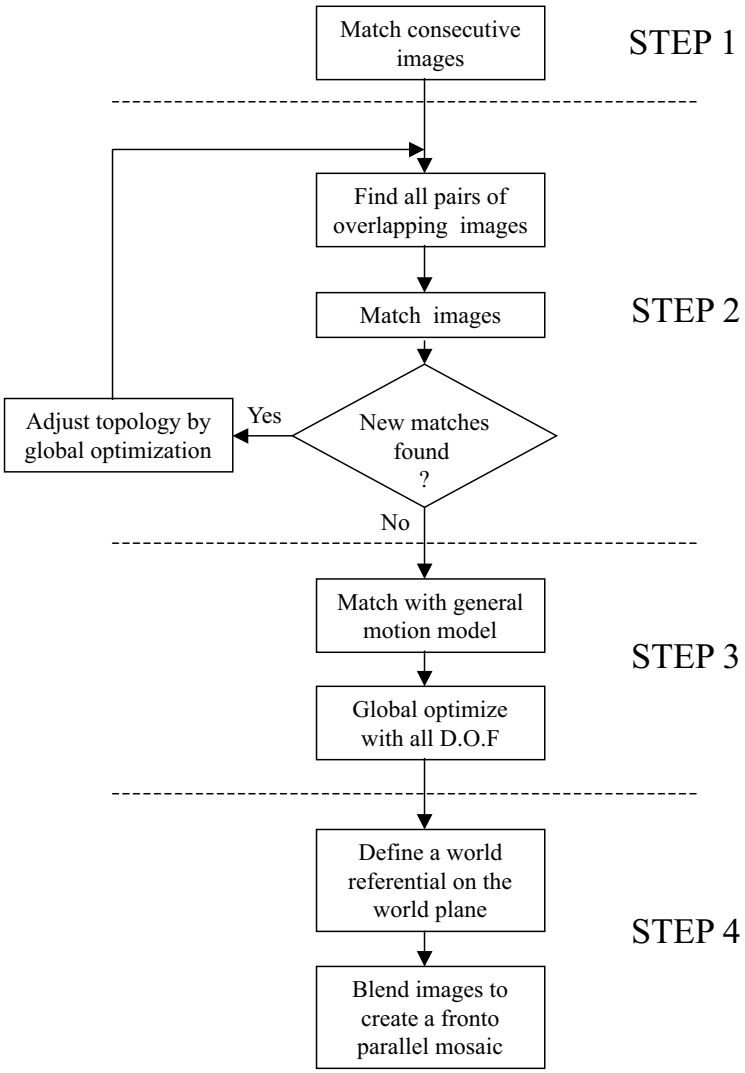


Figure 4.1: Flow–chart for the complete mosaic creation algorithm.

2. *Topology estimation of the neighboring relations between all images* in the sequence is performed by iteratively executing the following two main steps.

   (a) Point correspondences are established between non-adjacent pairs of images that present enough overlap. This is a time consuming operation but is alleviated by the fact that prior information exists on the location of the image correspondences, computed at the first stage.

   (b) The topology is refined, by searching for the set of homographies that minimize the overall sum of distances in the point matches.

3. *Trajectory estimation is carried out using all degrees of freedom* involved. An optimization problem is defined to search for the best set of pose parameters (describing the 3D positions and orientations of the camera) and for the best fitting description of the world plane.

4. Finally, a *mosaic rendering* step creates a fronto–parallel mosaic image. A 3–D world coordinate frame is associated with it.

The underlying assumptions for the method are those typical of single–plane mosaicing, *i.e.* the sea bottom is essentially flat, static and subject to small changes in the illumination. This is seldom the case in underwater mapping applications. However, the use of robust estimation over point feature matching greatly alleviates the damages of violating these assumptions and allows for the consistent recovery of image motion.

An example where some of the mosaicing assumptions are violated is given in Figure 4.2. The two images were captured in shallow waters under a short time interval, and cover approximately the same region of the sea bed. The scene is not planar and has different brightness patterns due to fast illumination changes. The image matching methods described in this chapter were able to successfully cope with this image pair.

4.2 Initial Motion Estimation

The first part of the algorithm consists on the sequential estimation of inter-frame homographies [32]. This is achieved by performing pair–wise image registration. For each pair of images, a set of highly textured feature points is extracted from one image and correlated over the other. The position where the correlation attains the maximum is taken as the match location. A robust estimation technique is used to discard false matches.

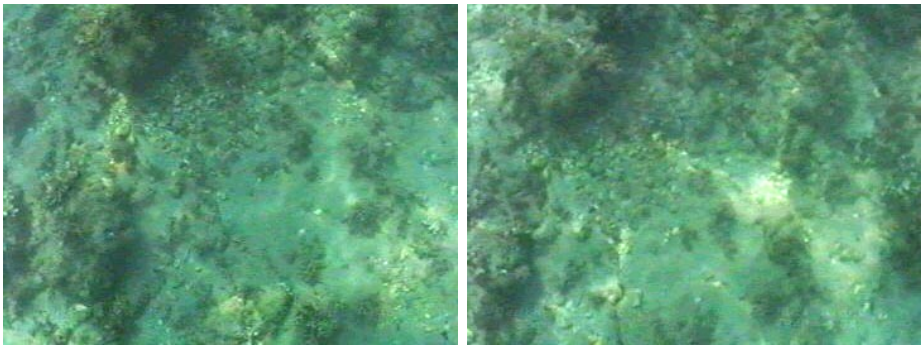


Figure 4.2: Two sequential frames, illustrating fast changes in the illumination conditions in shallow waters.

The implemented algorithm *attains real–time operation* and is executed during image acquisition. Therefore, the live video frames can be selected or discarded, based on a superposition criteria. This promotes large memory savings.

4.2.1 Feature selection

The image registration procedure evolves from the analysis of point projections and their correspondence between image frames. In order to improve the correspondence finding, a number of points are selected corresponding to image corners or highly textured patches. The selection of image points is based on a simplified version of the well-known corner detector proposed by Harris and Stephens [39, 116]. This detector finds corners in step edges by using only first order image derivative approximations. Further details on the implemented detector are presented in [37].

The extracted features will be matched over two images, and used for motion estimation. Since motion estimation is more noise sensitive to location errors when the features are close to each other [126], it is convenient to select features not just on the 'amount of texture', but also using some inter-feature distance criterion. Bearing this in mind, the implemented algorithm selects the features by finding the peaks of the 'texture' image and *excluding the subsequent selection on a circular neighborhood*. This process is repeated iteratively, up to the point where no peaks can be found, above a defined texture level. This texture level is set so that

Figure 4.3: Search area selection: image I_1(left) with selected feature, search area on I_2(center) and cross-correlation image(right)

4.2.2 Matching

The point matching, in the sense of associating the image projections of the same 3–D point, is a challenging task. Contributing factors to the difficulty include acquisition noise and low image texture and contrast, which are frequent in underwater imaging applications.

In this work, a correlation-based matching procedure was implemented. It takes a list of features selected from the first image I_1, and tries to find the best match for each, over a second image I_2. The cost criterion, that drives the search on the second image, is the sum of squared differences (SSD) [1]. For a given feature $\mathbf{f}_i = (u_i, v_i)$, it is defined as

$$SSD(x, y) = \sum_{(u,v)\in W_i} \left[I_1(u, v) - I_2(u - x, v - y)\right]^2 \tag{4.1}$$

where W_i is an image patch around $\mathbf{f}_i$. A modified version of the above criterion was proposed by Santos–Victor [94, 95] in the context of underwater point matching of 3–D scenes, where a term is added to penalize deviations from the epipolar constraint.

The assumption of large overlap of image content between the two frames can be used to significantly reduce the computational burden of the matching. This is achieved by limiting the search area in I_2. In order to compute the appropriate limits, the two images are cross-correlated and a global displacement vector $\mathbf{d}_G$ is obtained. By applying a threshold to the cross-correlation image, we can estimate a bounding box around $\mathbf{d}_G$, that can be loosely interpreted as a confidence area for the global displacement. Then, for a given feature $\mathbf{f}_i$ the search area on I_2 is constrained to the rectangular area with the size of the bounding box and centered on $\mathbf{f}_i + \mathbf{d}_G$. Figure 4.3 illustrates the procedure.

A similar procedure is applied if we have prior information on the expected image motion. This information is representable in the form of an expectable homography matrix,

hereafter referred to as a *pre–homography*. When processing a sequence of images captured at a high rate, most often we can assume some degree of motion constancy between frames. Under such conditions, the motion of the features in the current images will be similar to that of the previous pair. Thus, the last estimated homography is used as the current pre–homography. The use of the pre–homography in the matching is twofold. First, it serves to define the location of the areas to search for the features in I_2. Second, is allows for prewarping the image areas around each feature, such that the appearance of the feature is closer to what expected in I_2. The feature prewarping is particularly important in the case of large rotation or perspective distortion between the images.

4.2.3 Robust Motion Estimation

In this section we will describe a procedure for the estimation of the motion parameters for a sequence of images.

The images are processed as shown on the diagram of Figure 4.4. For each image I_k, a set of features is extracted and matched directly on the following image I_{k+1}, as described above. The result of the matching process are two lists of coordinates of corresponding points. Due to the error prone nature of the matching process, it is likely that a number of point correspondences will not relate to the same 3-D point thus calling for the use of robust motion estimation methods.

Let $\mathbf{u}^i$ be a point on frame i, and $\mathbf{u}^{i+1}$ be its correspondence on frame $i+1$, where $\mathbf{u}^i$ and $\mathbf{u}^{i+1}$ are projections of the same world point U living in a 3–D plane Π. If $H_{i,i+1}$ is the homography matrix induced by Π which relates frames i and $i+1$, then the point coordinates are related by

$$\widetilde{\mathbf{u}}^i \doteq H_{i,i+1}\ \widetilde{\mathbf{u}}^{i+1}\ .$$

Let $\mathbf{u}_n^i$ be the location of the n^{th} feature extracted from image i, and matched with $\mathbf{u}_n^{i+1}$ on image $i+1$. The homography $H_{i,i+1}$ is robustly estimated by minimizing the median of the square distances,

$$H_{i,i+1} = \arg\min_H \operatorname*{med}_n \left[d^2\left(\widetilde{\mathbf{u}}_n^i, H \cdot \widetilde{\mathbf{u}}_n^{i+1}\right) + d^2\left(\widetilde{\mathbf{u}}_n^{i+1}, H^{-1} \cdot\ \widetilde{\mathbf{u}}_n^i\right)\right] \tag{4.2}$$

where $d(\cdot,\cdot)$ takes two point locations in homogeneous coordinates and returns their (affine) distance. The minimization is performed by random sampling, using the two-step variant of the least-median-of-squares (LMedS), described in Section 2.2.2.

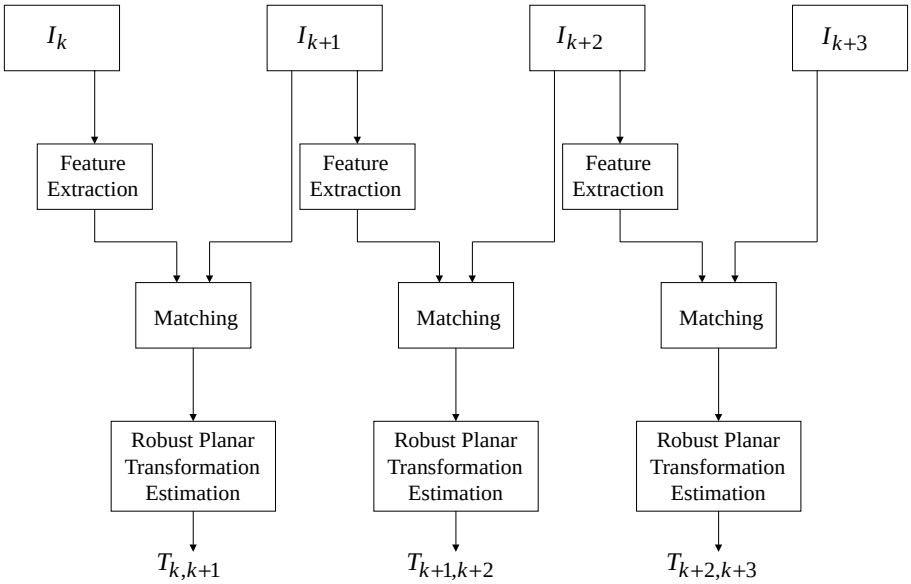


Figure 4.4: Block diagram of the sequence of operations on the images I_k for the motion parameter estimation. The output is the set of planar transformation matrices $T_{k,k+1}$.

Figure 4.5 shows the robust matching result for the image pair of Figure 4.2. Approximately 120 features were selected from the first image, where each feature is a square patch of 9×9 pixels. In order to avoid the extraction of closely separated features, a minimum distance of 15 pixels was imposed during the extraction process between any pair of features. The 4 d.o.f. similarity motion model was used for the homography estimation. The two steps of random sampling were run with a maximum of 100 and 50 iterations, respectively. After the random sampling, the set of 32 inliers were selected using a threshold of 3 pixels. All the inliers were used to estimate the final homography, under least–squares.

As mentioned above, the computed homography for the current pair of images is used to restrict the correlation search over the next pair. If, after the random sampling LMedS, the image matching is not successful then it is repeated with larger correlation areas. The sequence of correlation areas is $\frac{1}{4}$, $\frac{1}{2}$ and the full area of the image.

4.2.4 Frame Selection during Acquisition

In underwater vision applications it is very common for the image acquisition rate to be high when compared to the camera motion. This results in video sequences with high image redundancy between consecutive frames.

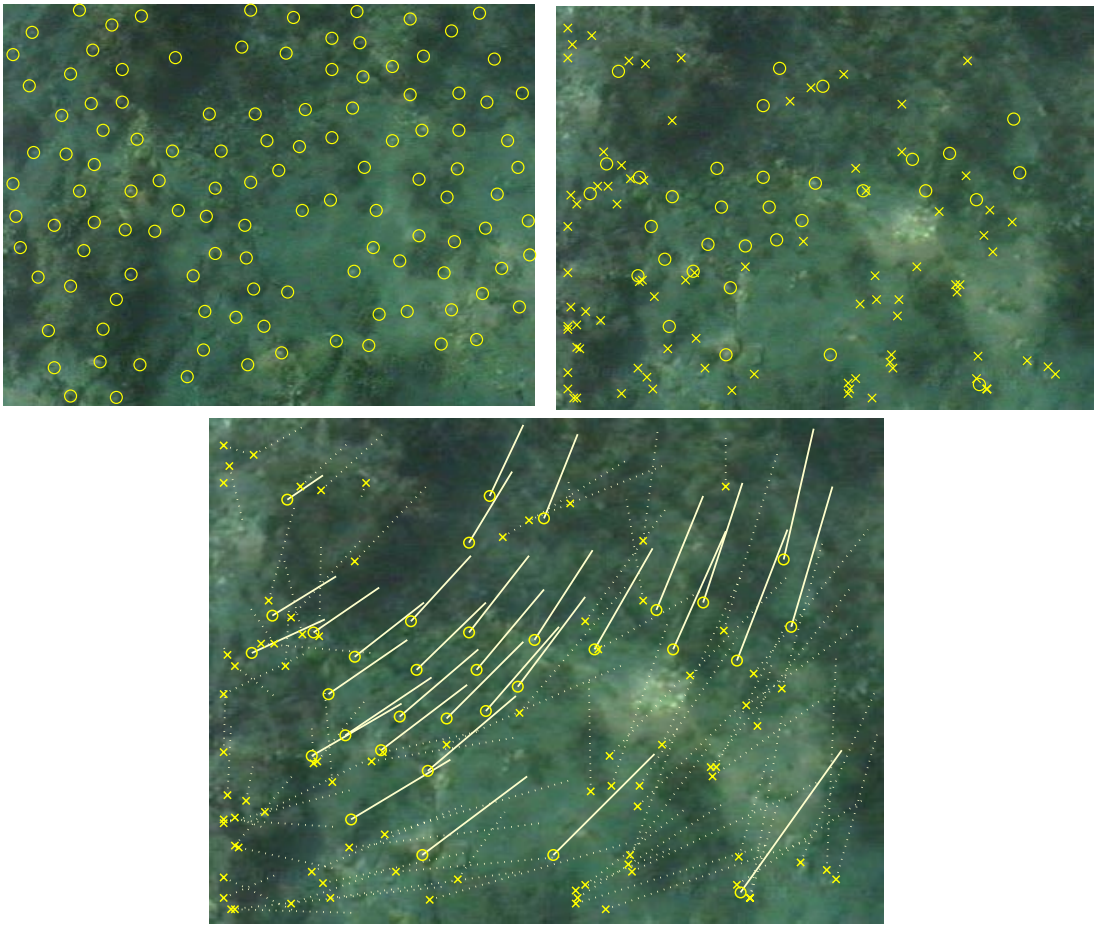


Figure 4.5: Robust feature matching example. A set of features is extracted from one image (upper left) and matched over the other image (upper right). From a set of 120 extracted and matched features, 32 were selected as inliers, using a similarity motion model and a distance threshold of 3 pixels. The inliers are marked as circles while the outliers are marked as crosses. The apparent feature motion is shown in the matching disparity (lower centre). The outliers are shown as dashed lines.

This condition motivated the use of a frame selection procedure, which allowed for the reduction of the memory and processing requirements in the following stages of the mosaic creation process. The frames are selected such that their superposition is the smallest above a given minimum acceptable overlap percentage. This threshold was chosen based on the results of preliminary matching trials. Under these trials, a set of representative underwater video sequences was used to infer the minimum superposition level which is still able to support adequate motion estimation[1]. The overlap threshold was found to be around 55%.

The implemented pair–wise robust image matching algorithm can process 7 images pairs per second, on a 800MHz Pentium PC. The frame selection step is performed on-line during the mosaic image acquisition. As a by product, it allows for the real-time creation of simpler strip mosaics, without global constraints. This proves to be very useful for the maneuvering of the vehicle during the acquisition, as it provides visual information on the approximate trajectory of the vehicle.

4.3 Iterative Topology Estimation

The previous section described the estimation of consecutive homographies $H_{i,i+1}$ from an image sequence. With it, one can find the transformation $H_{j,k}$ that relates any two images j and k with $k > j$, by appropriately cascading the homographies as

$$H_{j,k} = \prod_{i=j}^{k-1} H_{i,i+1} \ . \tag{4.3}$$

If the camera revisits a previously imaged region of the sea floor (for example by performing a loop), then there will be superposition between non time–consecutive images. Since small registration errors tend to accumulate during the sequential motion estimation, one should exploit such superpositions in order to reduce the accumulated error. This is done by matching non–consecutive images that overlap, and readjusting the homographies between the images.

We start by presenting the notion of *mosaic topology* (or simply *topology*) to refer to the undirected graph, comprising nodes and edges, that describes the connections between images that are matched. In this graph, nodes represent the location of the image centers,

[1]The criterion for considering a successful matching was the selection of at least 10 correct matches, confirmed by human inspection. The average total number of features was around 100 per image.

while the edges connect the image pairs that overlap and have been successfully matched. This concept was introduced by Shawney *et al.* in [97]. Similar concepts are used in [8] and [56] for global mosaic creation.

The topology is estimated by performing consecutive steps of *image matching* and *global adjustment*. The first step establishes correspondences between non-consecutive images, while the second updates the topology by taking into account the information from the new image matches. This cycle is repeated until no new image pairs can be matched. The two steps are now described.

Image matching — A matrix of the superposition level between every frame is constructed. For the image pairs whose predicted overlap is large, the image matching is attempted using the algorithm described in Section 4.2. Here, the pre–homography is computed by cascading the homographies over the loop.

Global adjustment — Upon finding new matches, the topology is adjusted by means of a global optimization procedure. The cost function to be minimized is the sum of distances between each correctly matched point and its corresponding point after being projected onto the same image frame,

$$F(X,\Theta) = \sum_{i,j} \sum_{n=1}^{N_{i,j}} \left[d^2\left(x_n^i, H(\Theta_i,\Theta_j)\cdot x_n^j\right) + d^2\left(x_n^j, H^{-1}(\Theta_i,\Theta_j)\cdot x_n^i\right)\right] \ , \quad (4.4)$$

where $N_{i,j}$ is the number of correct matches between frame i and j, and $H(\Theta_i,\Theta_j)$ is the homography constructed using the motion parameter vectors Θ_i and Θ_j. These vectors contain the parameters that relate frames i and j with a reference frame (the first frame). The minimization is carried out using a non-linear least squares algorithm [9], based on the interior trust–region method described in [10], which obtains a local minimizer

$$\widehat{\Theta} = \arg\min_{\Theta} F(X,\Theta) \ .$$

4.3.1 Efficient estimation

The above algorithm is computationally heavy. The total number of parameters p to be estimated depends upon the number of images n_{img} and the number of free parameters in the homography n_{par} as

$$p = (n_{img} - 1)\, n_{par} \ . \quad (4.5)$$

For a typical set of 150 images, using the most general homography model with 8 degrees of freedom requires the estimation of 1192 parameters. This precludes the use in applications requiring very fast mosaic creation. Nonetheless, the efficiency of the algorithm is significantly improved by taking into account algorithmic and parameterization issues, discussed below.

In underwater mosaicing applications, most often the imaging setup allows for the use of restricted motion models. An example of this is the covering of the region of interest at an approximately constant altitude [108], or having the camera image plane parallel to the sea–floor plane [76, 29, 90]. If the image motion constraints can be adequately modelled by a restricted homography (for example the ones described in Table 2.2), then such parameterization should be used. The main contributing reasons are :

- The reduction of the total number of parameters involved in minimizing Eq. (4.4).
- The reduction of the effects of fitting noisy data to an over–parameterized motion model.

The later reason is particularly important in the topology estimation process. The use of a over–parameterized motion model promotes the fitting of the error in the noisy data over the excessive degrees of freedom. When such homographies are cascaded, the accumulated error grows comparatively faster than with an adequately constrained model. Since the topology estimation algorithm depends upon the composition of homographies to predict superpositions, the accumulated errors may prevent such prediction.

Error accumulation due to excessive degrees of freedom in the motion model is illustrated in Figure 4.6. The same underwater sequence was used to estimate the set of homographies using four different parameterizations. This image sequence is a subset of a larger set (the *rock* sequence) and comprises a single closed loop with superposition. The images were captured in very shallow waters at an approximately constant altitude to the sea–bed, so that no large scale changes are observed. Therefore, an adequate motion model is the 2–D image translation plus rotation, with 3 degrees of freedom. The use of this model is shown in the first figure. The other models account for similarity, affine and full planar transformations, and are arranged in the order of increasing number of parameters.

The spatial arrangement of the frames which most accurately describes the actual image positions is given by the figure at the bottom. It contains the result of the full

mosaicing algorithm, after the topology estimation and the accurate registration step described in the next Section.

For every diagram in Figure 4.6, the corresponding superposition tables are shown in Figure 4.7. These tables were obtained by computing the superposition level for every pair of images using the sequential inter–frame homographies, and arranging the results as a matrix. The lighter colors correspond to the images where the superposition level is high. The homography model that better approximates the "ground–truth" table (lower right) is the 2D translation and rotation (upper right), closely followed by the similarity transform. For the results presented in this thesis, both models were implemented.

In addition to the use of restricted models, the size of the estimation problem is also reduced by using a sub-mosaic aggregation scheme. At the start of each iteration, the complete sequence is initially divided into sets of consecutive images that form small rigid sub-mosaics. Inside each sub-mosaic the homographies are considered static and only the inter-mosaic homographies are taken into account in the optimization algorithm. For selecting the number of images for of each sub–mosaic, a simple and effective rule–of–thumb is used. Under this, the sub-mosaic comprises 2 frames if those frames have overlap with any other non time–consecutive frame. Otherwise, longer sub-mosaic of 5 images are used. This scheme provides more degrees of freedom to the regions where there is more superposition. The use of sub–mosaics significantly improves the speed of the cost function evaluation, and does not affect the capability for inferring the appropriate topology.

On an algorithmic level, it should be noted that the minimization in Eq. (4.4) requires the computation of the residuals of point projections that depend only on a small number of parameters. The cost function can be written in an alternative form, as the squared norm of a vector $\mathbf{v}$,

$$F(X, \Theta) = \mathbf{v}^T \mathbf{v} \tag{4.6}$$

where $\mathbf{v}$ is the vector of distance residuals,

$$\mathbf{v} = \left[\begin{array}{c} d\left(x_n^i, H(\Theta_i, \Theta_j) \cdot x_n^j\right) \\ d\left(x_n^j, H^{-1}(\Theta_i, \Theta_j) \cdot x_n^i\right) \end{array} \right]_{i,j,n} . \tag{4.7}$$

The notation $[\cdot]_{i,j,n}$ represents the (vertical) stacking of the elements inside the bracket, obtained by iterating in i, j, and n under the appropriate limits. The dependency of each

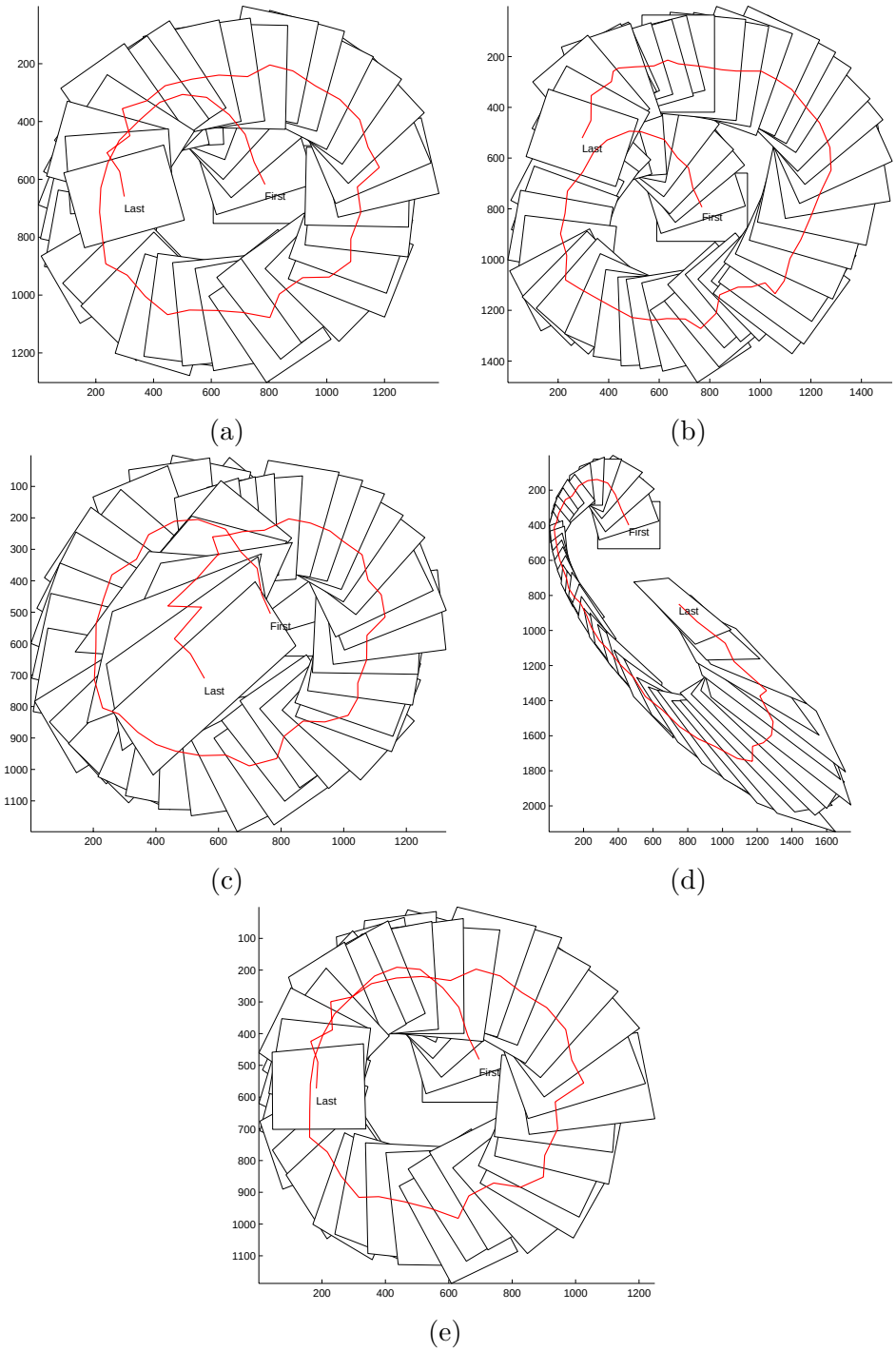


Figure 4.6: Sequential motion estimation using different motion models. The figures illustrate the spatial distribution of the images frames for motion estimation using a 2–D Translation and Rotation (3 dof) (a), similarity (4 dof) (b), affine (6 dof) (c) and full planar (8 dof) (d) motion model. Figure (e) presents the result obtained after the complete mosaicing algorithm, and serves as "ground–truth".

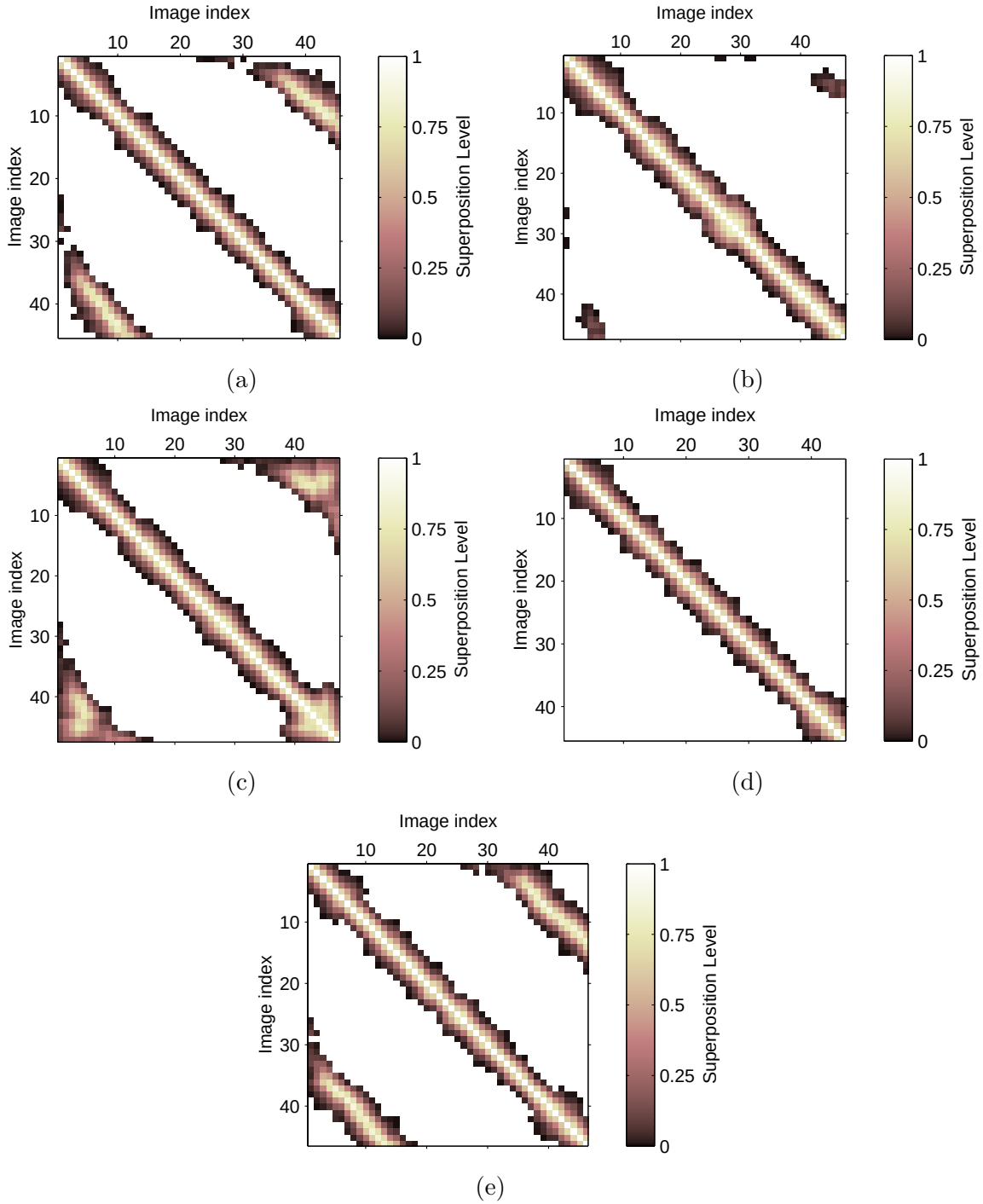


Figure 4.7: Sequential motion estimation using different motion models. This figure shows the matrices containing the superposition level for all images, corresponding to the image motion estimation cases of Figure 4.6. The matrices are arranjed in the same order.

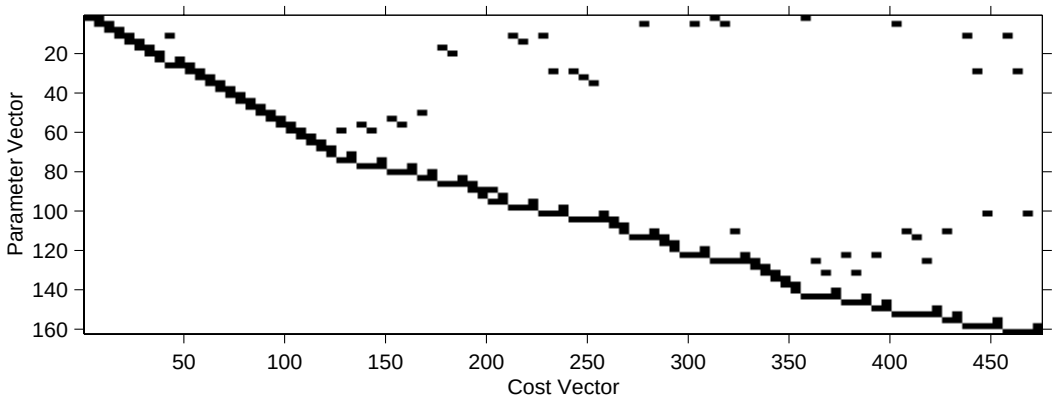


Figure 4.8: Example of the sparse structure of the Jacobian matrix used in the topology estimation process. This matrix relates the vector of homography parameters with the cost vector formed by the distances between matched points. The non–zero entries are represented in black.

element of $\mathbf{v}$ in a small number of parameters means that, for a given set of matched points and parameters (X, Θ), the Jacobians of $\mathbf{v}$ and Θ with respect to X are sparse with a predictable sparsity structure. This structure allows for very efficient implementations of the commonly used algorithms for non–linear least-squares. Examples of this are described by Capel [8] in mosaicing applications, or Hartley [40] in rotating camera calibration. The sparse structure of the Jacobian of $\mathbf{v}$ is illustrated in Figure 4.4 and is used in our algorithm.

A complete example of the topology estimation for the whole *rock* sequence is given in Figure 4.9, containing several loops. The graphs on the upper row illustrate the spatial arrangement of the images before (a) and after (b) the topology estimation step. The lower row shows a close–up of the corresponding mosaics. In order to emphasize the registration errors, the mosaics were rendered by stacking the images on the order of acquisition, so that the most recent are placed on top. The effects of the accumulated error from the sequential motion estimation are visible on the repeated pattern of algae and rocks in Figure 4.9(c). The same area is shown in Figure 4.9(d), where the matching of time–distant images allowed for the reduction of the registration error.

4.4 Accurate Global Registration

The main objective of the final stage of the algorithm is attaining a highly accurate registration. A more general parameterization for the homographies is therefore required,

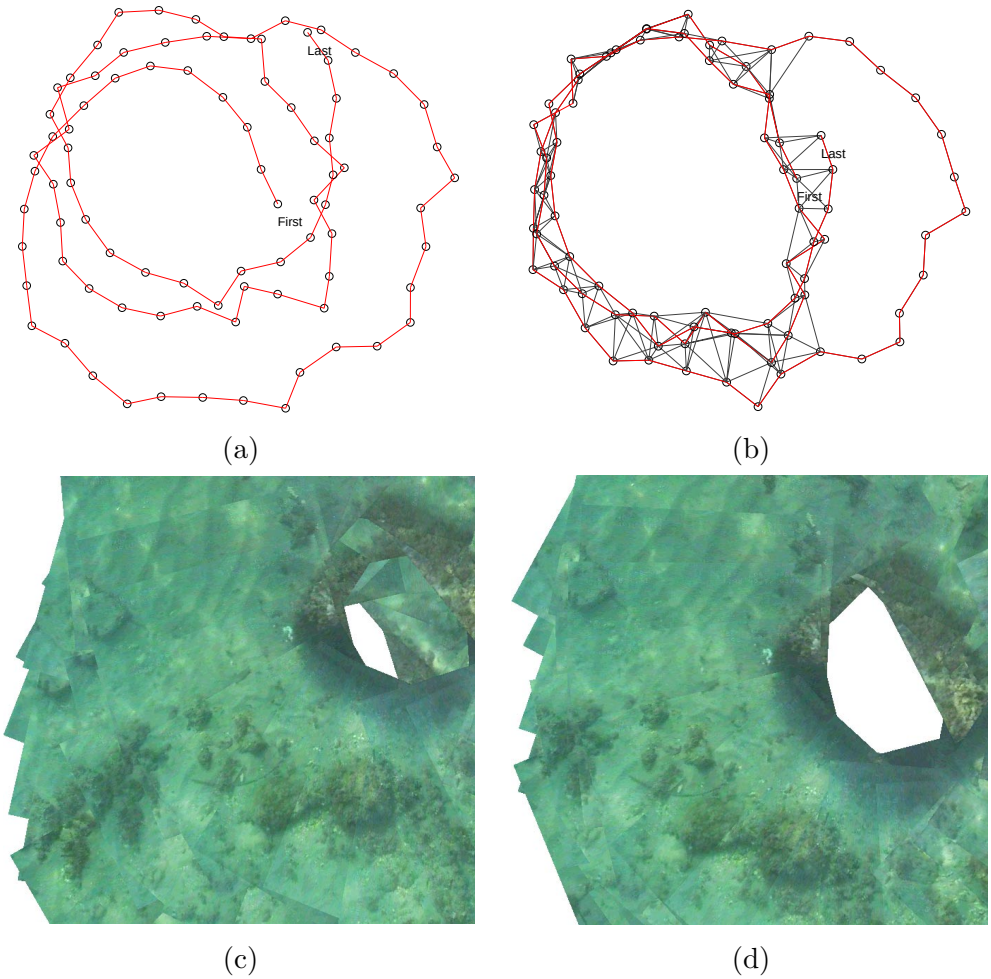


Figure 4.9: Topology estimation example for the complete *rock* sequence containing several loops – The graphs illustrate the spatial arrangement of the mosaic images before (a) and after (b) the topology estimation step. The graph nodes represent the image centers (marked as dots). The edges link the images that were successfully matched.
The lower row displays a close–up of the corresponding mosaics. The effects of the accumulated error (c) have been reduced by registering non-consecutive images (d).

capable of modelling the warping effects caused by wave-induced general camera rotation and changes on the distances to the sea floor. A parameterization was thus chosen in which all the 6 degrees of freedom of the camera pose are explicitly taken into account.

It should be noted that the estimation of the homographies for the 6 degrees of freedom model does not impose, *per se*, the condition of a single world plane from which the homographies are induced. This condition is therefore imposed by augmenting the overall estimation problem with additional parameters that describe the position and orientation of the world plane. The common world plane description is included in the parameterization of the homographies.

An important advantage of the following parameterization is that it allows for the full 3–D camera trajectory and world plane to be recovered during the process.

4.4.1 General parameterization

The overall parameterization scheme is the following. One of the camera frames (usually the first) is chosen as the origin for the 3–D coordinate frame, where the optical axis is coincident with the Z–axis. The world plane is parameterized with respect to this frame by 2 angular values that define its normal. As the trajectory and plane reconstruction can only be attained up to an overall scale factor, this ambiguity is removed by setting the plane distance to 1 metric unit[2], measured along the Z-axis.

Let Θ_i and Θ_j be the pose 6–vectors containing 3 rotation angles and 3 translations with respect to the reference 3–D frame of the first camera. Let $n(\Theta_p)$ be a 3-vector containing the normal to the world–plane (also in the 3–D reference frame), which is parameterized by the 2-vector Θ_p of angles. The homography relating frames i and j with the reference image frame is given by Eq. (2.2):

$$H_{i,1} = K \cdot \left[R(\Theta_i) + t(\Theta_i) \cdot n^T(\Theta_p) \right] \cdot K^{-1}$$

$$H_{j,1} = K \cdot \left[R(\Theta_j) + t(\Theta_j) \cdot n^T(\Theta_p) \right] \cdot K^{-1}$$

where $R(\Theta_i)$ and $R(\Theta_j)$ are rotation matrices, $t^T(\Theta_i)$ and $t^T(\Theta_j)$ are the translation components, as defined in Section 2.1.3. The homography relating frames i and j is given

[2]If additional information is available on the real distance to the sea floor (for example, from an altimeter), then it can be straightforwardly used here.

by

$$H_{i,j} = H_{i,1} \cdot H_{j,1}^{-1} = K \cdot \left[R\left(\Theta_i\right) + t\left(\Theta_i\right) \cdot n^T\left(\Theta_p\right)\right] \cdot \left[R\left(\Theta_j\right) + t\left(\Theta_j\right) \cdot n^T\left(\Theta_p\right)\right]^{-1} \cdot K^{-1} \tag{4.8}$$

4.4.2 Cost Function

The global registration is performed by bundle adjustment with a cost function similar to the one previously used in the topology adjustment in page 50, but using the parameters for the most general motion model. The distances between matched points are measured in their respective image frames, and summed over all pairs of correctly matched images, *i.e.*,

$$F(X, \Theta) = \sum_{i,j} \sum_{n=1}^{N_{i,j}} \left[d^2\left(x_n^i, H_{i,j} \cdot x_n^j\right) + d^2\left(x_n^j, H_{i,j}^{-1} \cdot x_n^i\right)\right] \tag{4.9}$$

For a set of M images, the total number of parameters to be estimated is $(M-1) \times 6 + 2$, comprising 6 parameters per camera (excluding the reference camera) plus the 2 angles of the normal to the world plane. As before, the cost function is minimized using non-linear least squares, to obtain

$$\widehat{\Theta} = \arg\min_{\Theta} F(X, \Theta)$$

The initial values for the parameter set are computed from the homographies obtained at the end of the topology estimation step (Section 4.3) which relate each image to the first camera. To do this, we use the general homography decomposition of Eq. 2.2, and the method described by Triggs in [115]. However, for each homography there is, in general, two distinct and valid solutions for the rotation, translation and plane orientation. This ambiguity can be solved by combining the information from two or more views [62]. Under the assumption of a single plane in the scene (as we do in this work), the ambiguity can be removed by choosing the solutions which correspond to the same plane orientation [24].

Our approach is to find the plane orientation which has the largest consensus from the group of all orientations. Given N homographies relating each images frame to the first image frame, let S_n be the set of all $2N$ plane normals obtained from decomposing each homography. Let n_i and n_j be vectors from S_n. We seek the most representative plane orientation $\widehat{n}$, such that

$$\widehat{n} = \arg\min_i \operatorname*{med}_j \left[\left(n_i - n_j\right)^T \left(n_i - n_j\right)\right] .$$

Given $\widehat{n}$, the criterion for removing the ambiguity is to select the set of rotation plus translation whose associated plane normal is closer to $\widehat{n}$.

4.5 Map Rendering

The final operation consists of blending the images, i.e., choosing the representative pixels to compose the mosaic image, that are taken from the spatially registered images.

A common method for image blending is to use the last contributing image. However, considering the intended application for navigation, an alternative method was used. Under this, the mosaic is created by choosing the contributing points which were located the closest to the center of their frames. In underwater applications it compares favorably with other commonly used rendering methods, such as the average or the median. This is due to the fact that it better preserves the textures and minimizes the effects of unmodeled lens distortion, which tends to be larger near the image borders.

The orientation of the world plane has been explicitly taken into account and estimated. Therefore, it is easy to compute a planar projective transformation that yields a fronto-parallel view of the mosaic. As we are interested in creating a navigation map, the fronto-parallel projection is the most appropriate in the sense it minimizes the perspective image distortions in the image-to-mosaic matching for vehicle configurations where the camera is pointing downwards.

A homography $H_{1,fp}$, relating the image frame of the reference camera with a virtual fronto–parallel camera, can be found just by applying an appropriate 3–D camera rotation, as illustrated in Figure 4.10. Let n be the 3-vector containing the normal to the world plane expressed in the 3–D camera reference frame. Then $H_{1,fp}$ is given by

$$H_{1,fp} = K \cdot R_{1,fp} \cdot K^{-1}$$

where $R_{1,fp}$ is any rotation matrix whose last column is $-n$. This family of matrices corresponds to 3–D rotations that align the optical axis of the reference frame with the normal of the plane, and is defined up to a rotation around the vertical axis.

For the navigation, we are interested in establishing a Euclidean 3-D world reference associated with the mosaic. As its location is purely arbitrary, it was chosen to have the origin in the intersection of the optical axis of the first image with the plane of the mosaic. The orientation is such that the mosaic plane has null z coordinate, and the x axis is

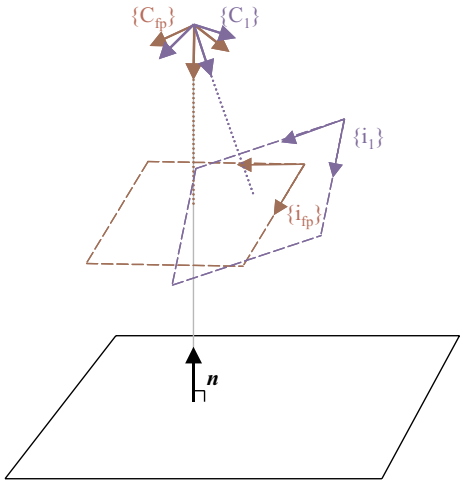


Figure 4.10: Rotation of the reference camera frame to yield a fronto–parallel view of the floor.

parallel to the first camera frame x axis. If the information about overall scale is available from a sensor such as an altimeter, it can be used here.

4.6 Results

The results on globally consistent mosaics were obtained from image sequences captured by a custom modified ROV [34]. Details on the platform setup are given in Appendix A.

An illustrative example of the mosaic creation process is given in Figure 4.11. The image sequence was acquired in shallow waters of about 2 meters depth, while the vehicle was manually driven around a squared shaped rock. During the acquisition, the inter-frame motion estimation was performed on-line, which allowed for the selection of 98 images based on a 60% superposition criteria. This resulted in the upper–left mosaic, where the effects of error accumulation are visible near the image top in the form of a repeated white stone. After 4 steps of topology estimation, 285 distinct pairs of non-consecutive images (combined from the selected image set) where successfully matched. The final topology of the upper-right mosaic was obtained. Next, the global optimization was carried out using full 3-D pose parameters for all cameras and world plane description. The outcome of this step allows for the creation of a fronto-parallel view of the mosaic, which can be used as the navigation map.

Another image sequence was obtained over a flat sandy area, fully surrounded by algae.

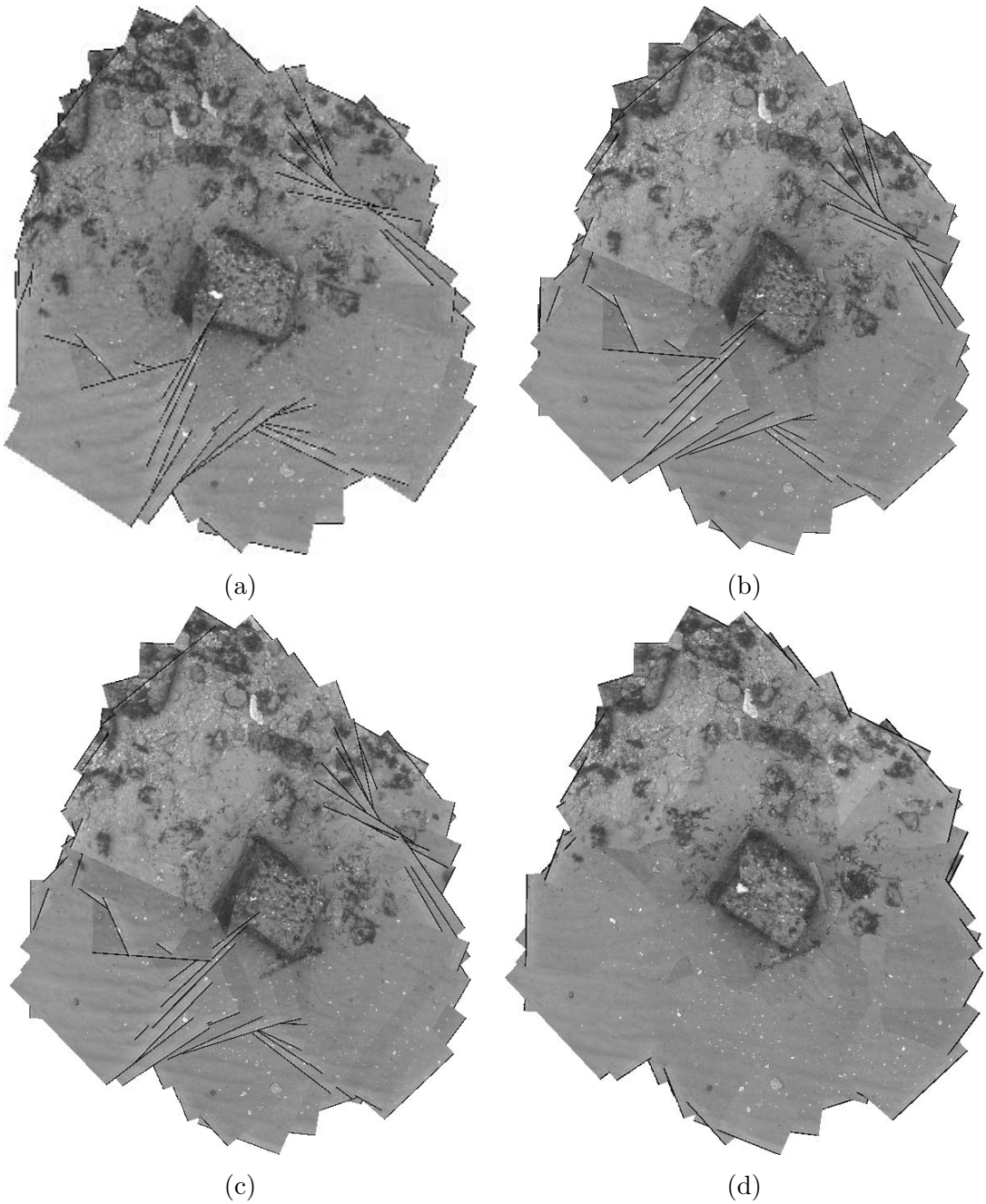


Figure 4.11: Mosaic creation example with intermediate step outcome – Consecutive image motion estimation (a), topology estimation and and non-consecutive image matching (b), high accuracy global registration (c) and final fronto-parallel view of the mosaic after global optimization (d). The first three mosaics were rendered using the pixels from the last contributing image, while the last was created with the contribution from the image whose pixels were closer to the frame center.

During the acquisition, the vehicle was manually driven to follow a zig-zag trajectory that covered most of the area. The sequence comprises 1000 images, corresponding to 6 minutes and 40 seconds of video. After the initial matching, a set of 129 images was selected using the criterion of minimal overlap above 50%, which resulted in an average overlap of 54.4%. The topology estimation was completed in 13 iterations, resulting in 322 pairs of images that were successfully matched. The evolution of the topology estimation is given in the four graphs of Figure 4.12. The graph (a) shows the spatial arrangement of the image centers as obtained by the first step of the mosaicing process, described in Section 4.2 (sequential motion estimation). The following graphs (b) and (c) correspond to the intermediate outcome and final result (d) of the topology estimation of Section 4.3. In each graph, line segments are used to link the centers of images that have been matched. Another graphical representation for the final result of the topology estimation is given in Figure 4.13. The image on the left is obtained from the superposition matrix, which contains the superposition level of each image with respect to all images. The image on the right illustrates the number of matches found for all pairs of images.

The mosaic obtained from the last stage of the algorithm, is shown in Figure 4.14. It was created by choosing the contributing points which were located the closest to the center of their frames. The chosen rendering operator uses the image intensity contributions of just one image, as stated above. The resulting mosaic exhibits a Voronoi–type space division for the borders of contributing images. The high quality of the final mosaic is illustrated by the fact that algae leafs, lying on the predominant ground plane are not disrupted along the visible boundaries of the contributing images.

A small section of the rendered mosaic is displayed in Figure 4.15 along with one of the original frames for the same area. The quality of the registration can be assessed from the fact that the visual features (such as small algae leafs) are not disrupted along the visible boundaries of the contributing images. The recovered 3D camera paths are illustrated in Figure 4.16.

The upper mosaic of Figure 4.17 was created from a set of 70 selected images. The image sequence was acquired approximately over the same sandy area as the previous sequence, but after a period of several months. For this experiment, the distance of the vehicle to the sea floor was measured by the on-board altimeter, during the acquisition of the first sequence image. After taking into account the displacement between the altimeter and the camera 3–D reference frames, the distance of 5.0 meters was obtained,

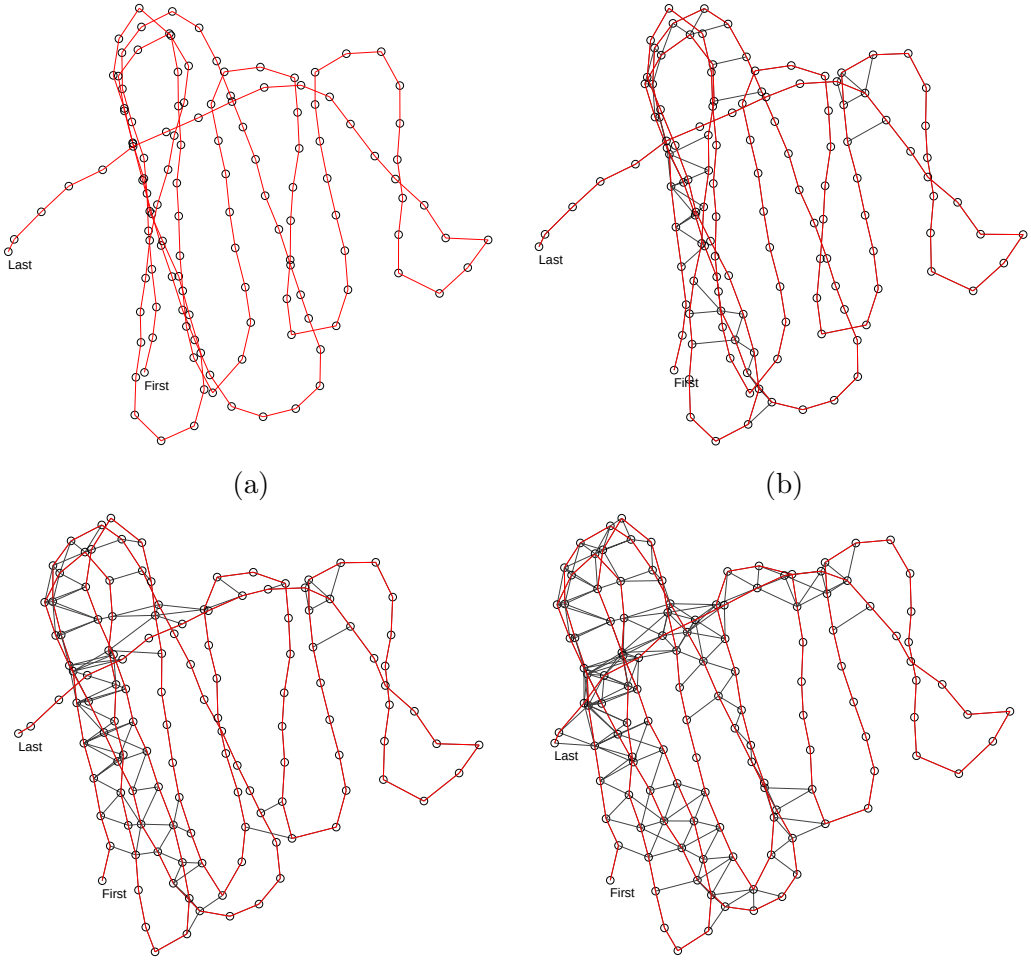
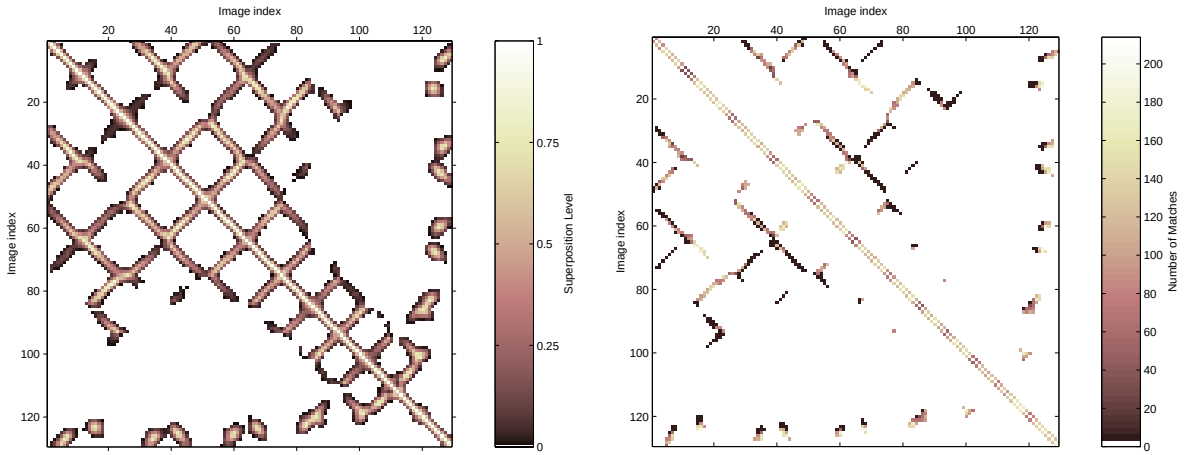


Figure 4.12: Topology estimation for the *bottle* sequence – The graphs illustrate the spatial arrangement of the images used for creating the mosaic of Figure 4.14. The first graph (a) refers to the initial motion estimation, in which the images are matched sequentially by the order of the frame aquisition. The following graphs were obtained after the 4^{th} (b), 9^{th}, and 13^{th} (final) topology iterations (d).



(a) (b)

Figure 4.13: Superposition level (a) and number of matched points (b), for the all the images of the set, after the final iteration of the topology estimation process.

from the camera frame of the first image to the seabed. The overall mosaic map scale was set accordingly. The mosaic covers approximately 92 square meters, from which 40 correspond to sand. Each pixel on the mosaic corresponds to a sea floor area of about 2×2 centimeters. The rectangular region that contains the mosaic area measures 14.2×14.3 meters. The lower part of the figure presents a similar mosaic, created from a set of 46 images. As before, the scale was set using the altimeter to measure the distance of 4.29 meters from the first camera frame to the floor. The mosaic covers approximately 64 square meters and is inscribed in a 10.8×9.5 meter rectangle.

It should be noted that the mosaic process was able to successfully cope with image contents that clearly departs from the assumed planar and static conditions. This is visible in the large percentage of the mosaic area used by moving algae.

4.7 Discussion

This chapter presented a method for the creation of mosaics comprising four main algorithmic steps. These are the sequential estimation of camera motion, topology inference, high accuracy trajectory estimation and fronto-parallel mosaic rendering.

Illustrative results were obtained from a sequence of shallow water images taken by a ROV. The images present some of the common difficulties of underwater mosaicing, such as non planar sea-bottom, moving objects and severe illumination changes. These challenging

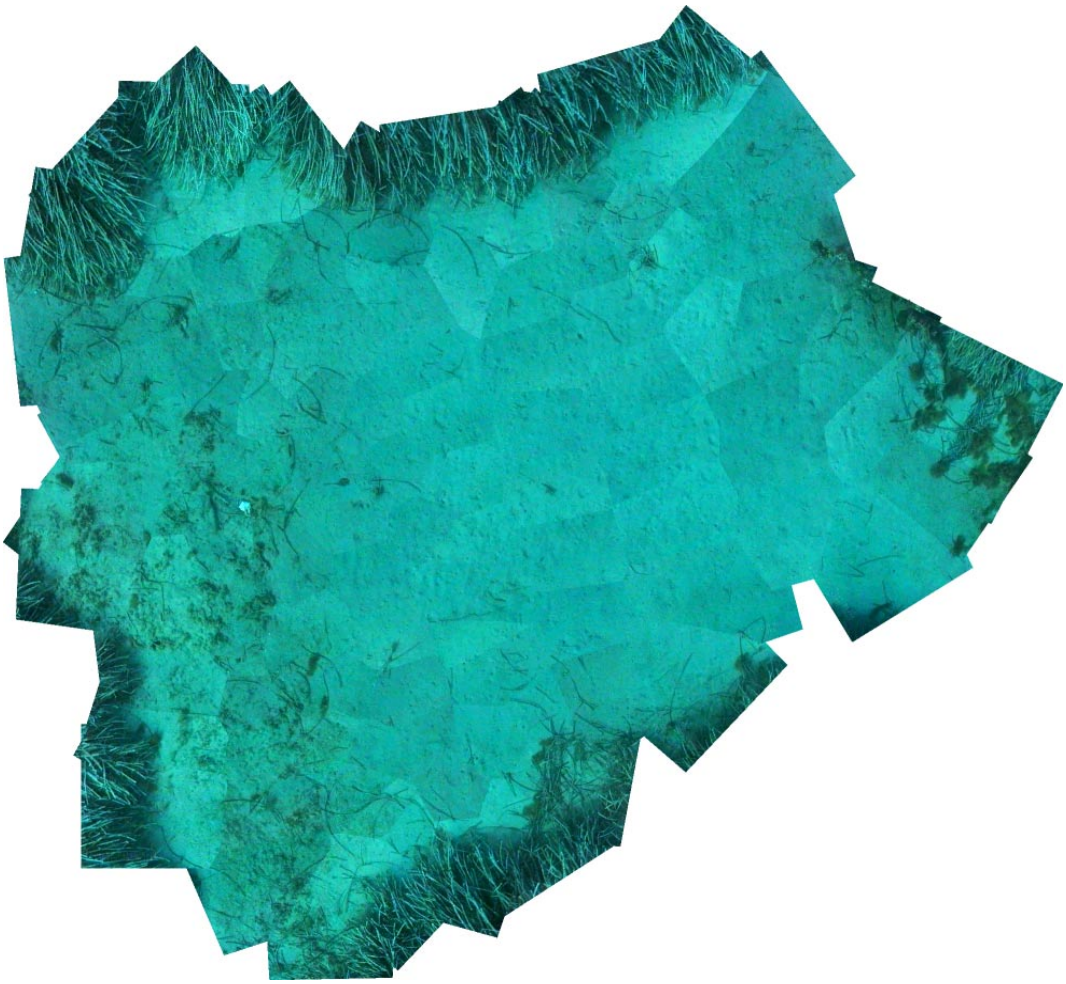


Figure 4.14: Final mosaic created using 129 images selected from the original set of 1000 and rendered with the *closest* operator. The seafloor area covered is approximately 42 m^2.

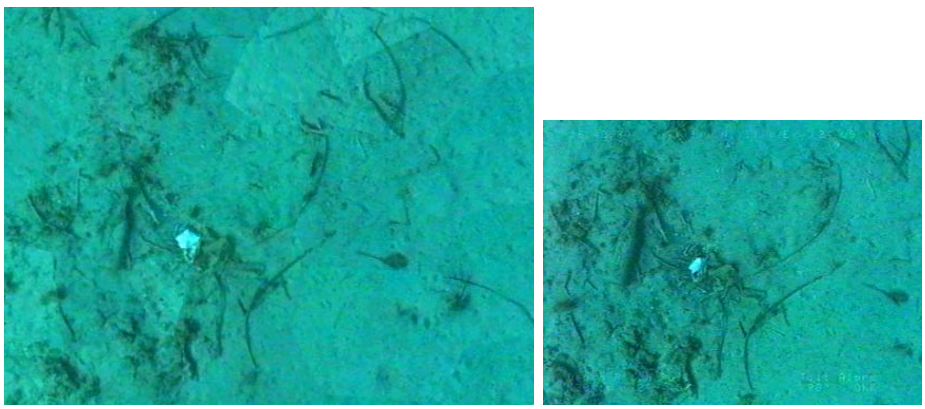


Figure 4.15: Area detail of the mosaic (left), and one of the original images (right).

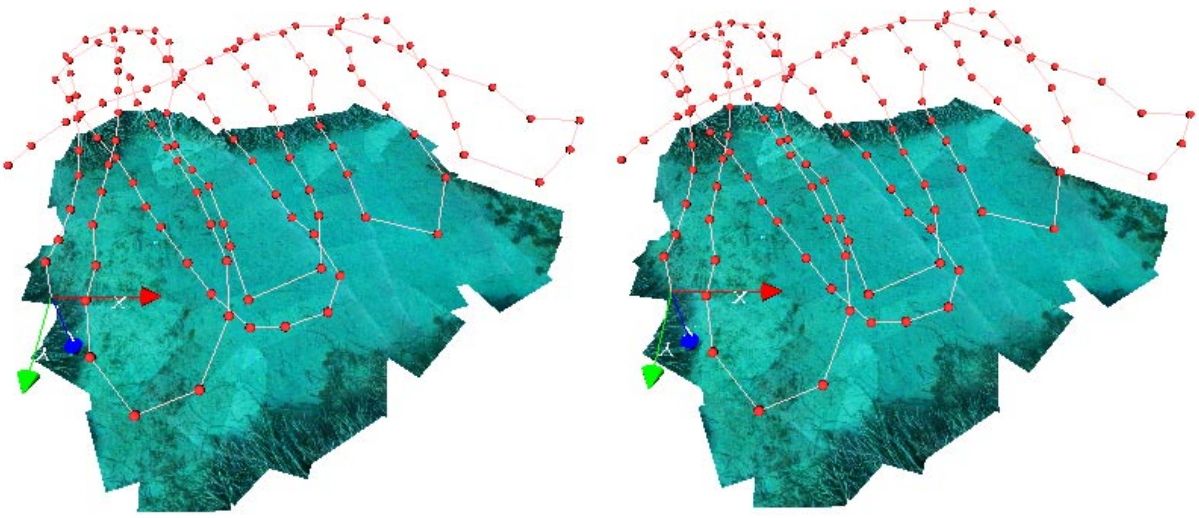


Figure 4.16: VRML stereogram of the camera path and mosaic. The world referential is illustrated by the system of axis, which is coincident with the first camera frame. The views are arranged for crossed eye fusion.

sequences are used to illustrate the robustness and good performance of the complete algorithm. The approach seamlessly integrates the problems of trajectory estimation and mosaic construction. Also, it provides the means of finding a geometric description of sea-bottom plane, and is able to reconstruct the camera path taking into account all the involved degrees of freedom.

The success of the global mosaicing process depends upon the accurate estimation of the motion, during the first step. Inaccurate results will hinder the ability to predict (and exploit) the non time–consecutive overlaps. As illustrated in Section 4.3, such overlaps are required to create spatially coherent mosaics.

When predicting the superposition, two types of errors can be considered:

- Type I – Detection of superposition when there is none. This is illustrated in Figure 4.6 (c) where the last frame erroneously overlaps with the first.
- Type II – No detection of superposition when there actually is. This is illustrated in Figure 4.6 (d) where the loop was not closed.

Under the our approach, only the first type of error can be detected. Such errors are signaled by the failure to match successfully the images involved.

However, the dependency on the accurate initial motion estimation can be alleviated

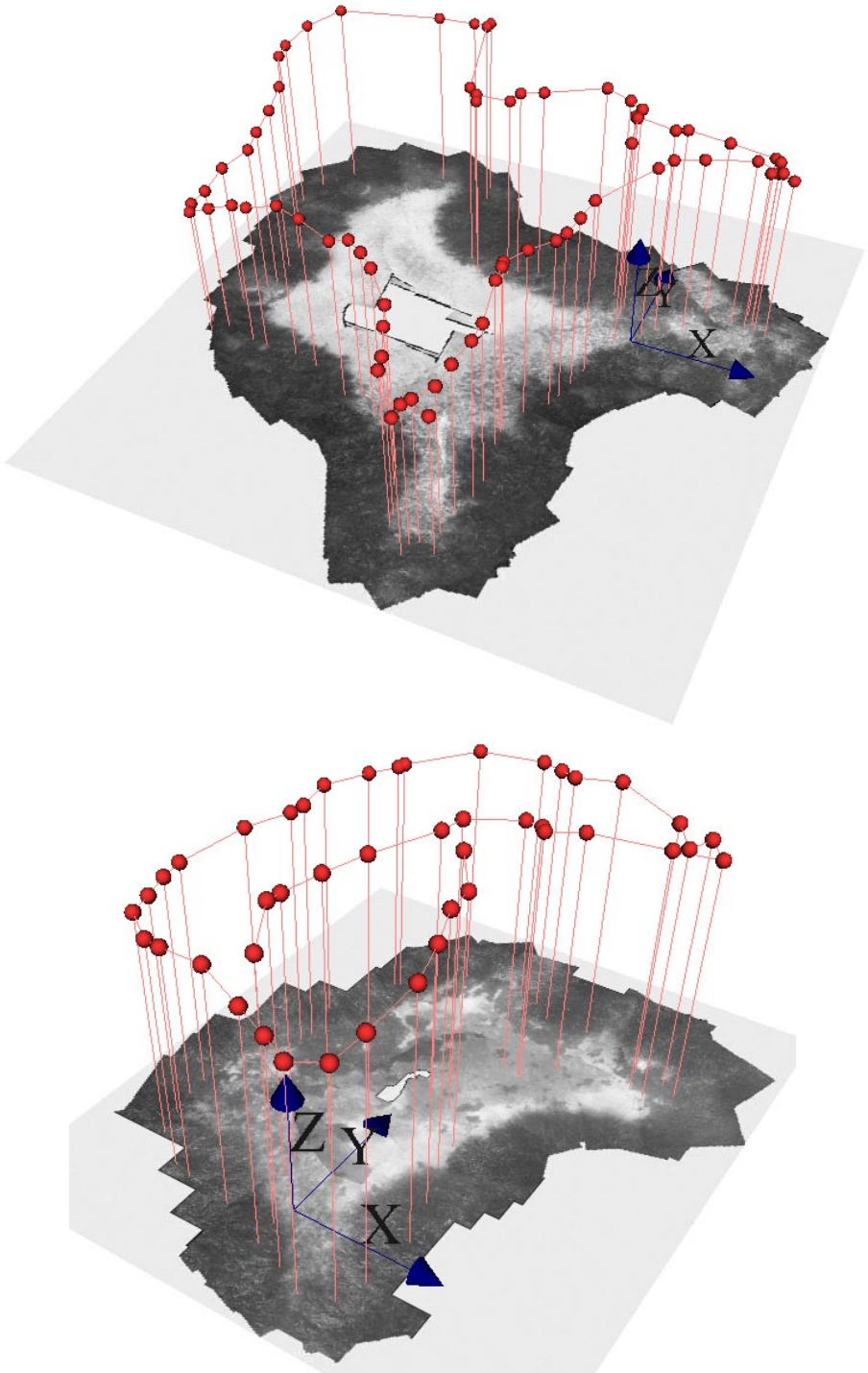


Figure 4.17: Perspective view of two of the mosaics used for the underwater navigation tests, with original camera path reconstruction. The small dots mark the 3–D position of the camera centres for the image set selected to create the mosaic. The world referencial is represented by the 3 perpendicular axes on the upper right of the image. Vertical lines were added to ease the perception of the 3–D trajectory. The upper mosaic comprises 70 images and the lower 46.

by insuring the presence of several overlapping regions in the video sequence. An example of such is the *bottle* sequence of Figure 4.12, where the large displacement between the last four frames and the rest of the mosaic was successfully corrected.

Chapter 5

Mosaic–based Pose Estimation

In this chapter we address the problem of using a previously created mosaic map for the 3–D localization of a vehicle. We assume the mosaic map has been built, and that a world coordinate frame has been associated with it. Having such map enables a camera-equipped autonomous vehicle to locate itself by finding point matches between the mosaic and the image frame.

The problem of pose estimation is addressed under the assumption of known camera intrinsic parameters. Two methods are presented which differ on the estimation accuracy and computational cost. The first is an well–known *algebraic method* to recover the pose parameters directly from the elements of the image–to–mosaic homography. Since it leads to a direct, non–iterative solution, it is suitable for real-time operation on setups of limited computing resources. The second method is a *maximum likelihood estimator* that recovers the pose by minimizing a cost function using the coordinates of matched points between the mosaic and the camera image. This method is optimal when the uncertainty in the coordinates of the matched points is modelled as additive Gaussian noise. However, it is computationally more demanding than the algebraic method, as it requires an iterative implementation to solve a non–linear least squares problem.

For both the algebraic and maximum likelihood methods, a first–order covariance propagation is performed. The validity and accuracy of covariance prediction is confirmed using statistical simulation. This chapter concludes with pose estimation results using an image sequence with known ground truth.

5.1 Pose Parameterization

In this chapter we will consider the most general parameterization for the pose, accounting for all the degrees of freedom in rotation and translation.

Let $\Theta = \begin{bmatrix} \alpha & \beta & \gamma & {}^{W}_{C}\mathbf{t}_x & {}^{W}_{C}\mathbf{t}_y & {}^{W}_{C}\mathbf{t}_z \end{bmatrix}^T$ be the 6-vector containing the camera pose in the form of 3 camera rotation angles and the location of the camera centre in world coordinates. The 3-D rigid transformation that relates points in the world and camera frames is given by

$$\begin{bmatrix} {}^{C}\mathbf{x} \\ {}^{C}\mathbf{y} \\ {}^{C}\mathbf{z} \end{bmatrix} = {}^{C}R_W \left(\begin{bmatrix} {}^{W}\mathbf{x} \\ {}^{W}\mathbf{y} \\ {}^{W}\mathbf{z} \end{bmatrix} - \begin{bmatrix} {}^{W}_{C}\mathbf{t}_x \\ {}^{W}_{C}\mathbf{t}_y \\ {}^{W}_{C}\mathbf{t}_z \end{bmatrix} \right) , \tag{5.1}$$

where ${}^{C}R_W$ is a rotation matrix. This matrix is parameterized by the X-Y-Z fixed angle convention [12],

$${}^{C}R_W = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} ,$$

where $s(.)$ and $c(.)$ represent the sine and cosine functions of the rotation angles.

Without loss of generality, we assume that a world coordinate frame was set such that all the world points belong to the plane defined by ${}^{W}\mathbf{z} = 0$. Therefore, a camera–to–world collineation can then be defined as

$$T_{i,W}(\Theta) \doteq K \cdot {}^{C}R_W \cdot \begin{bmatrix} 1 & 0 & -{}^{W}_{C}\mathbf{t}_x \\ 0 & 1 & -{}^{W}_{C}\mathbf{t}_y \\ 0 & 0 & -{}^{W}_{C}\mathbf{t}_z \end{bmatrix} , \tag{5.2}$$

where K is the 3×3 intrinsic parameter matrix[1]. The collineation $T_{i,W}(\Theta)$ relates the metric coordinates of the points in the world plane $\begin{bmatrix} {}^{W}\mathbf{x} & {}^{W}\mathbf{y} \end{bmatrix}^T$ with the pixel coordinates of their projections in the image $\begin{bmatrix} u & v \end{bmatrix}^T$, in the form

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \doteq T_{i,W}(\Theta) \cdot \begin{bmatrix} {}^{W}\mathbf{x} \\ {}^{W}\mathbf{y} \\ 1 \end{bmatrix} .$$

[1] Althought related, this equation cannot be directly obtained from Eq. (2.2), because we are now dealing with only two 3–D planes (the world plane and the image plane) whereas in Eq. (2.2) three plane are involved. Furthermore, we are now considering the 3-D reference frame to be on the world plane.

Since the mosaic is stored as a bitmap, establishing a world coordinate frame in the mosaic corresponds to defining a collineation that relates the mosaic image pixel coordinates with their metric counterpart. Let ${}^{W}N_{m}$ be the 3×3 matrix representing such collineation. Then the camera–to–mosaic homography can be written as

$$\Psi(\Theta) \doteq T_{i,W}(\Theta) \cdot {}^{W}N_{m} \ . \tag{5.3}$$

5.2 Algebraic Method

We will now review a fast algebraic solution to recover the camera pose from the homography relating the camera image with the calibrated mosaic (*i.e.* a mosaic bitmap with an associated coordinate frame). Using the knowledge on the intrinsic parameter matrix, a useful decomposition can be obtained for the collineation $T_{i,W}$ which relates planar world points with their camera projections. This solution is based on the method described by Sturm in [106]. Similar approaches are reported in [23, 21].

Let L be a (3×3) matrix constructed from the rotation matrix ${}^{C}_{W}R$, and the vector, ${}^{C}_{W}\mathbf{t}$,

$$L = \left[\begin{array}{cc} \overline{{}^{C}_{W}R} & {}^{C}_{W}\mathbf{t} \end{array} \right] \ ,$$

where, for a (3×3) matrix A, the notation $\overline{A}$ denotes the (3×2) submatrix comprising the first two columns. The homography $T_{i,W}$ can be written as

$$T_{i,W} = \lambda K L \ , \tag{5.4}$$

where λ is an unknown scale factor. In order to recover the pose information embedded in L, the unknown scale factor λ has to be determined. The absolute value of λ can be obtained by noting that the first two columns of L have unit norm. By denoting M as

$$M = \overline{T_{i,W}}^{T} \cdot K^{-T} \cdot K^{-1} \cdot \overline{T_{i,W}} = \left[\begin{array}{cc} \lambda^2 & 0 \\ 0 & \lambda^2 \end{array} \right]$$

we have $\lambda^2 = M(1,1) = M(2,2)$. The two possibilities for the sign of λ result in valid solutions for Eq.(5.4). However, only one corresponds to the camera height being above the sea floor.

The last column of ${}^{C}_{W}R$ can obtained by computing the cross product of the first two columns. Let ${}^{C}_{W}R_1$ and ${}^{C}_{W}R_2$ be the two candidates for ${}^{C}_{W}R$, corresponding respectively to

the scaling by $+|\lambda|$ and $-|\lambda|$. The matrices ${}^{C}_{W}R_1$ and ${}^{C}_{W}R_2$ are related by

$$ {}^{C}_{W}R_1 = {}^{C}_{W}R_2 \cdot \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} . $$

The corresponding optical centre locations are given by

$$ {}^{W}_{C}\mathbf{t}_1 = -\frac{1}{|\lambda|}\, {}^{C}_{W}R_1^T \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \text{ and } {}^{W}_{C}\mathbf{t}_2 = \frac{1}{|\lambda|}\, {}^{C}_{W}R_2^T \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} , $$

where $\begin{bmatrix} t_1 & t_2 & t_3 \end{bmatrix}^T$ is the last column of λL. The locations of the optical centres differ by the last coordinate which is symmetric. Both solutions for ${}^{C}_{W}R$ and ${}^{W}_{C}\mathbf{t}$ are in accordance with $T_{i,W}$, and are geometrically valid. In the application of this work, we are only interested in the positive $\overrightarrow{z}$ axis solution for ${}^{W}_{C}\mathbf{t}$, which corresponds to the camera being above the plane of the floor.

Due to the limited resolution of the matching process, the measured value $\widehat{T}_{i,W}$ will not exactly follow the structure of Eq. (5.4). In order to recover the pose, we are interested in finding the $T_{i,W}$ which best approximates the noisy measurement $\widehat{T}_{i,W}$, while keeping the noise-free structure.

Using the Frobenius norm to measure the distance between matrices, the problem can be formulated as

$$ \lambda, L = \arg\min_{\lambda,L} \left\| \lambda L - K^{-1}\widehat{T}_{i,W} \right\|^2_{frob} \text{ subject to } \overline{L}^T\overline{L} = I_2 . \tag{5.5} $$

Since the last column of L is not restricted, the above problem can be solved by dividing it into two independent subproblems. The first problem corresponds to the first two columns of L which are constrained, which is formulated as

$$ \lambda, \overline{L} = \arg\min_{\lambda,L} \left\| \lambda\overline{L} - \overline{K^{-1}\widehat{T}_{i,W}} \right\|^2_{frob} \text{ subject to } \overline{L}^T\overline{L} = I_2 . \tag{5.6} $$

As pointed out in [106], the solution of Eq. (5.6) can be solved using the Singular Value Decomposition, in the following manner. Let $U \cdot \Sigma \cdot V^T$ be the SVD of $\overline{K^{-1}\widehat{T}_{i,W}}$. Then $\overline{L}$ is given by

$$ \overline{L} = U \cdot V^T . $$

The scale factor can be found independently. By imposing the condition of null derivative at the minimum,

$$ \frac{d}{d\lambda} \left\| \lambda\overline{L} - \overline{K^{-1}\widehat{T}_{i,W}} \right\|^2_{frob} = 0 , $$

one gets

$$\lambda = \frac{\mathrm{tr}(\overline{L}^T \cdot \overline{K^{-1}\widehat{T}_{i,W}})}{\mathrm{tr}(\overline{L}^T \cdot \overline{L})}$$

which, taking into account the SVD of $\overline{L}$ and $\overline{K^{-1}\widehat{T}_{i,W}}$, is simply

$$\lambda = \frac{\mathrm{tr}(\Sigma)}{2} \ .$$

Once $\overline{L}$ is known, the second problem of finding the last column of L is solved directly by

$$\overset{C}{\underset{W}{\mathbf{t}}} = K^{-1} \cdot \widehat{T}_{i,W} \cdot \begin{bmatrix} 0 \\ 0 \\ \frac{1}{\lambda} \end{bmatrix}$$

and

$$L = \begin{bmatrix} \overline{L} & \overset{C}{\underset{W}{\mathbf{t}}} \end{bmatrix} \ .$$

The method presented in this section illustrates the use of the image–to–mosaic homography for estimating the pose. However, the pose information is embedded in the coordinates of the matched points. This has motivated an estimator for recovering the pose directly from the coordinates, that is presented in the following section.

5.3 Maximum Likelihood Estimation

We will now present a maximum likelihood estimator for the pose. The underlying assumptions are of independent additive Gaussian noise on the image the coordinates of the matched points.

After the matching process, we consider the following observation equation

$$\begin{bmatrix} \mathbf{x}_n^i \\ 1 \end{bmatrix} = \frac{\Psi(\Theta)}{\lambda_n} \begin{bmatrix} \mathbf{x}_n^m \\ 1 \end{bmatrix} + \begin{bmatrix} \varepsilon_n \\ 0 \end{bmatrix} \ , \tag{5.7}$$

where $\mathbf{x}_n^i$ and $\mathbf{x}_n^m$ are the coordinates of point correspondences in the camera frame and in the mosaic image respectively, and ε_n is a Gaussian random vector of zero mean and known covariance. The 3×3 matrix Ψ contains the camera–to–mosaic homography as described in Section 5.1. The scale factor λ_n is given by

$$\lambda_n = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \Psi(\Theta) \cdot \begin{bmatrix} \mathbf{x}_n^m \\ 1 \end{bmatrix} \ .$$

By stacking all the observations in the vector $\mathbf{x}^i = \begin{bmatrix} \mathbf{x}_1^i & \dots & \mathbf{x}_N^i \end{bmatrix}^T$ we can write the observation equation for all points as

$$\mathbf{x}^i = \mathbf{Q}(\mathbf{x}^m, \Theta) + \varepsilon \ .$$

The vector $\mathbf{Q}(\mathbf{x}^m, \Theta)$ contains the projections of mosaic points in a camera with pose Θ,

$$\mathbf{Q}\left(\mathbf{x}^m, \Theta\right) = \begin{bmatrix} \mathbf{Q}(\mathbf{x}_1^m, \Theta) \\ \vdots \\ \mathbf{Q}(\mathbf{x}_N^m, \Theta) \end{bmatrix}$$

where, for a generic point $\mathbf{x}_n^m$ in the mosaic, $\mathbf{Q}(\mathbf{x}_n^m, \Theta)$ represents the 2-vector of its projection in the image, *i.e.*,

$$\mathbf{Q}\left(\mathbf{x}_n^m, \Theta\right) = \frac{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot \Psi(\Theta) \begin{bmatrix} \mathbf{x}_n^m \\ 1 \end{bmatrix}}{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \Psi(\Theta) \cdot \begin{bmatrix} \mathbf{x}_n^m \\ 1 \end{bmatrix}} \ .$$

The error term ε is assumed to be a vector of independent, equally distributed, Gaussian random variables of zero mean and covariance given by

$$R = \mathrm{cov}(\varepsilon) = \sigma^2 I_{2N} \ . \tag{5.8}$$

This hypothesis will be tested empirically in Section 5.5.1.

Given the observation equation and the error distribution, the conditional probability of observing $\mathbf{x}^i$, given the pose parameters Θ, is

$$Like\left(\Theta\right) = P\left(\mathbf{x}^i \mid \Theta\right) = \frac{1}{\left(2\pi \cdot \sigma^2\right)^{\frac{N}{2}}} e^{-\frac{1}{2\sigma^2}\left(\mathbf{x}^i - \mathbf{Q}(\mathbf{x}^m, \Theta)\right)^T \left(\mathbf{x}^i - \mathbf{Q}(\mathbf{x}^m, \Theta)\right)} \ . \tag{5.9}$$

The expression above is referred to as the Likelihood function since it quantifies the "likelihood" of Θ being the sought parameters.

Given $\mathbf{x}^i$ and $\mathbf{x}^m$, we are interested in finding the pose Θ that best explains the data. This can be carried out by maximizing $P\left(\mathbf{x}^i \mid \Theta\right)$ with respect to Θ or, in an equivalent manner, minimizing the logarithm of the inverse of $P\left(\mathbf{x}^i \mid \Theta\right)$,

$$\begin{aligned} l\left(\Theta\right) = \ & \tfrac{N}{2}\log(2\pi) + N\log(\sigma) \\ & + \tfrac{1}{2\sigma^2}\left(\mathbf{x}^i - \mathbf{Q}\left(\mathbf{x}^m, \Theta\right)\right)^T \left(\mathbf{x}^i - \mathbf{Q}\left(\mathbf{x}^m, \Theta\right)\right) \end{aligned} \ . \tag{5.10}$$

The first two terms of $l(\Theta)$ do not depend on Θ, and can thus be removed in the minimization process. The maximum likelihood estimate of Θ is thus,

$$\Theta_{ML} = \arg\min_{\Theta} l(\Theta) = \arg\min_{\Theta} \left\| \mathbf{x}^i - \mathbf{Q}(\mathbf{x}^m, \Theta) \right\|^2 . \tag{5.11}$$

This minimization is carried out using a non–linear least squares algorithm [87]. The initial value for Θ is provided by the algebraic solution.

5.4 Uncertainty Propagation

The method used for estimating the uncertainty is based on the first order approximation of the Taylor series of a cost function. This cost function is minimized by estimated parameters and the data used in the estimation process. The method, overviewed in Appendix B, allows for the computation of the covariance of the noise on the estimates as a linear function of the covariance of the additive noise in the data.

5.4.1 Propagation for the Algebraic Method

For the algebraic method, the covariance in the pose estimate is computed in two steps. The first step propagates the covariance from the noise in the matched points to the elements of the image–to–mosaic homography Ψ. The second step propagates from these elements to the pose.

Matched points to homography

The computation of the image–to–mosaic homography is performed using the most general model, with 8 degrees of freedom. More specifically, the elements of the homography are computed by using the SVD to solve the constrained minimization

$$\Psi = \arg\min_{\Psi} \|H(X) \cdot \mathrm{vec}(\Psi)\|^2 \text{ subject to } \|\mathrm{vec}(\Psi)\| = 1 , \tag{5.12}$$

where $H(X)$ is a matrix comprising the matched point coordinates as defined in Eq.(2.4) in Chapter 2, and $\mathrm{vec}(\Psi)$ is a column vector containing the 9 elements of homography in a row-wise fashion. To simplify the notation, we will denote the elements of the homography arranged as a column by Ψ, for the remaining of this section.

An alternative method would be the computation of Ψ by fixing one of the elements to a non-zero scalar (typically $\Psi_{3,3} = 1$), and performing unconstrained minimization

over the remaining 8 elements. However this will lead to numerical instability if the fixed element turns out to be close to zero[2].

To help us obtain an expression for the covariance in the estimate of Ψ, we will introduce some additional notation. This notation is required for representing the entities in the estimation process which are affected by noise (both data and parameters). Let X_0 be the noise-free $4n \times 1$ vector containing the coordinates of the n matched points. Due to the limited resolution in the matching process, we can only observe a noisy version $\widehat{X}$ of X_0. We assume that the noise can be described as a vector of small amplitude zero–mean additive perturbations ΔX, such that $\widehat{X} = X_0 + \Delta X$, where the $4n \times 4n$ covariance matrix $\Sigma_{\Delta X}$ of ΔX is known.

By solving the problem in Eq.(5.12) using the observed data vector $\widehat{X}$, we obtain an estimate $\widehat{\Psi}$ for the homography elements. This estimate is also perturbed due to the unknown ΔX. We further assume that $\widehat{\Psi}$ can be expressed (by a first order Taylor expansion) as $\widehat{\Psi} = \Psi_0 + \Delta\Psi$, where Ψ_0 would be obtained using X_0 in Eq.(5.12).

Let F be a positive scalar function relating Ψ and X, such that F attains a minimum at $F(\widehat{X}, \widehat{\Psi})$

$$F(X, \Psi) = \|H(X) \cdot \Psi\|^2 = \Psi^T \left[H(X)\right]^T H(X) \Psi \ ,$$

and $s(\Psi)$ be a function such that $s(\Psi) = 0$ encompasses the constraint in Ψ,

$$s(\Psi) = \left(\Psi^T \cdot \Psi\right) - 1 = 0 \ .$$

Having introduced the perturbed versions of both data and parameters, the minimization problem in Eq.(5.12) can be written in the following equivalent form,

$$\widehat{\Psi} = \arg\min_{\Psi} F(\widehat{X}, \Psi) \text{ subject to } s(\Psi) = 0 \ . \tag{5.13}$$

Using the second order derivatives of F, a linear approximation can be found to relate $\Delta\Psi$ and ΔX, around $(\widehat{X}, \widehat{\Psi})$. This derivation is detailed in Appendix B. The predicted covariance $\Sigma_{\Delta\Psi}$ of the elements of Ψ is given by

$$\Sigma_{\Delta\Psi} = E \cdot A^{-1} \cdot B \cdot \Sigma_{\Delta X} \cdot B^T \cdot A^{-1} \cdot E^T \tag{5.14}$$

[2]Imposing $\Psi_{3,3} = 1$ fails to represent the case where the vector connecting the world referential to the optic center is perpendicular to the optic axis. Under such condition $\Psi_{3,3} = 0$.

where

$$E = \left[\begin{array}{cc} I_{9\times 9} & \mathbf{0}_{9\times 1} \end{array} \right]_{9\times 10} ,$$

$$A = \left[\begin{array}{cc} \frac{\partial^2 F}{\partial \Psi^2} & \left(\frac{\partial s}{\partial \Psi}\right)^T \\ \frac{\partial s}{\partial \Psi} & \mathbf{0} \end{array} \right]_{10\times 10} \quad \text{and}$$

$$B = \left[\begin{array}{c} -\left(\frac{\partial^2 F}{\partial \Psi \partial X}\right)^T \\ \mathbf{0} \end{array} \right]_{10\times 4n} .$$

Both the Hessian matrix of F and the Jacobian of s (inside matrix A) have a special structure, that is used to speed up the calculation of A, namely

$$\frac{\partial^2 F}{\partial \Psi^2} = H^T(X) \cdot H(X) \qquad \text{and}$$

$$\frac{\partial s}{\partial \Psi} = 2 \cdot \Psi^T .$$

The second derivative of F with respect to Ψ and X (in B) cannot be written as a compact expression, due to the structure of $H(X)$. For this reason, it was evaluated numerically using symmetric differences [87], for the expression

$$\frac{\partial^2 F}{\partial \Psi \partial X} = \frac{\partial}{\partial X}\left[2\left[H(X)\right]^T H(X)\,\Psi\right] .$$

It should be noted that the covariance propagation given above does not assume any particular distribution for the noise ΔX in the observations. The only assumptions are that ΔX should be sufficiently small to allow a first order approximation of $F(X, \Psi)$ and that $F(X, \Psi)$ has finite second order derivatives, so that Eq.(5.14) holds. This is detailed in Appendix B.

Homography to pose

The following step consists in propagating the covariance of the noise from the 9 elements of the homography to the pose vector.

As described above, given an image–to–mosaic homography, the pose can be recovered by solving the problem of Eq. (5.5). The solution is obtained by computing the singular value decomposition of a 3×3 matrix, which is a very fast process. Therefore it is computationally inexpensive to numerically estimate the Jacobian that relates small perturbations $\Delta\Psi$ in the elements of Ψ with the corresponding perturbations $\Delta\theta$ in the pose vector Θ. An estimator for the covariance $\Sigma_{\Delta\theta}$ in the pose is given by

$$\Sigma_{\Delta\theta} = J^T \cdot \Sigma_{\Delta\Psi} \cdot J ,$$

where $J = \frac{\partial \Theta}{\partial \Psi}(\Psi, \Theta)$ is the 9×6 Jacobian matrix. This matrix is computed by a numerical approximation.

In practical applications, it might be useful to also take into account the uncertainty on θ due to the uncertainty on the camera intrinsic parameters. If the uncertainty is modelled as small perturbations of zero mean around the nominal value, then the data vector (and associated covariance matrix) can be extended to include the elements of K that are affected by noise. In this case, care should be taken in properly scaling such values, to avoid the potential numerical problems related to having very disparate magnitudes in the entries of the (extended) data covariance matrix. An effective normalization is the division of intrinsic parameters by their largest value.

5.4.2 Propagation for the Maximum Likelihood Estimator

The pose estimates for the maximum likelihood estimator are obtained directly from the coordinates of the matched point in the mosaic and in the image. This is achieved by solving the unconstrained minimization problem of Eq. (5.11). The covariance $\Sigma_{\Delta\theta}$ of the pose estimates is predicted using the same approach as in Section 5.4.1. Since there are no constraints on Θ, the expression for $\Sigma_{\Delta\theta}$ is a simplified version of Eq. (5.14),

$$\Sigma_{\Delta\theta} = \left(\frac{\partial^2 F}{\partial \theta^2}\right)^{-1} \cdot \left(\frac{\partial^2 F}{\partial \theta \partial X}\right)^{T} \cdot \Sigma_{\Delta X} \cdot \frac{\partial^2 F}{\partial \theta \partial X} \cdot \left(\frac{\partial^2 F}{\partial \theta^2}\right)^{-1} ,$$

where the function F, which implicitly relates X and Θ, is

$$F(X, \Theta) = \|X - \mathbf{Q}(\mathbf{x}^m, \Theta)\|^2 = [X - \mathbf{Q}(\mathbf{x}^m, \Theta)]^T \cdot [X - \mathbf{Q}(\mathbf{x}^m, \Theta)] \ .$$

The covariance estimation involves the computation of second derivative matrices, that are evaluated numerically. Similarly to what was stated above, if we want to include the effects of the uncertainty from the intrinsic parameters, then the data vector and covariance matrix are extended to include the uncertain intrinsics.

5.4.3 Statistical simulation

A Monte Carlo [54] validation was carried out to test whether the error propagation could be satisfactorily estimated by the first order methods described above.

A reference pose was chosen, from which two lists of noise-free coordinates were obtained, comprising 34 point matches between the image and the mosaic. The points in the

image are contaminated with independent identically distributed additive Gaussian noise of 1 pixel standard deviation, for an image size of 320×240. This noise level is set as conservatively higher value than what was obtained by measuring the residuals resulting from feature matching between sets of underwater images with distinct visual content. To simulate the matching process as described in Section 4.2, we consider that only the point projections in the image are affected by noise, whereas the mosaic point coordinates are noise–free. It is assumed the noise is caused by the limited resolution of the matching procedure and from slight non-planarities in the scene.

For each noisy instance of the matched coordinates, the corresponding pose was estimated using both the algebraic method and the maximum likelihood estimator. The statistics of 1500 pose instances where then compared to the predicted values. The predicted covariance was computed around the mean value of the pose estimates.

For the algebraic method, the histograms for the elements of Ψ are illustrated in Figure 5.1. The diagonal entries of the predicted covariance $\Sigma_{\Delta\Psi_{vec}}$, corresponding to the variance of each element of Ψ, were used to draw a Gaussian distribution curve which was superimposed as a full line. This covariance matrix was used as the input for predicting the covariance $\Sigma_{\Delta\theta}$ associated with the pose parameters. The corresponding histograms for the elements of Θ are shown in Figure 5.2, where the predicted covariance on the parameters was used to superimpose a Gaussian curve. The two figures show that the prediction is accurate for the amount of noise involved.

For the maximum likelihood method, the histograms for the elements of Θ are illustrated in Figure 5.3. The covariance prediction is also accurate.

An additional test was conducted, with different levels of noise, to gain insight on the limits of the approximation validity. In order to compare the real and the predicted covariance matrices, a distortion measure was devised based on the normalization of the real covariance matrix. By using the singular value decomposition, one can find the linear transformation on the parameter space that maps the empirical covariance matrix onto the identity[3]. By applying the same transformation, both on the empirical and on the predicted covariance matrices, the measure returns the Frobenius norm of their difference.

The results for maximum likelihood estimator are presented in Figure 5.4. The noise levels ranged from 0.25 to 10 pixel standard deviation. For each noise level, 500 instances of the pose and predicted covariance were calculated. The orientation and translation

[3]provided the uncertainty spans all the parameter space.

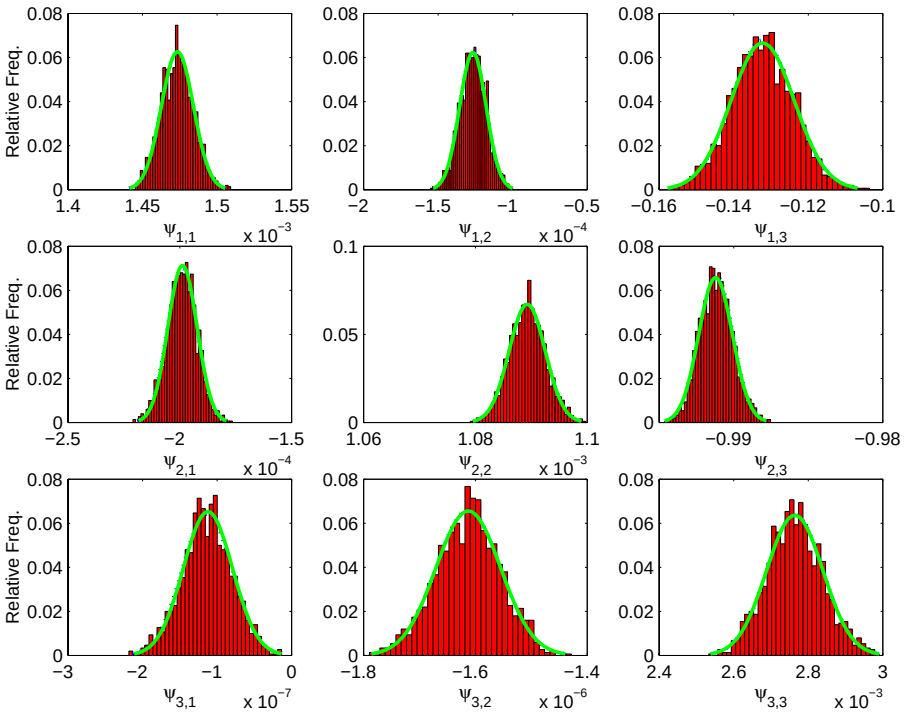


Figure 5.1: Results from Monte Carlo trials for testing the validity of the method for propagating the covariance from matched points to the elements of the image–to–mosaic homography Ψ. The histograms where created from 1500 instances of noise contamined coordinates, with a noise level of 1 pixel (standard deviation) and are arranged in accordance to the corresponding elements of Ψ.

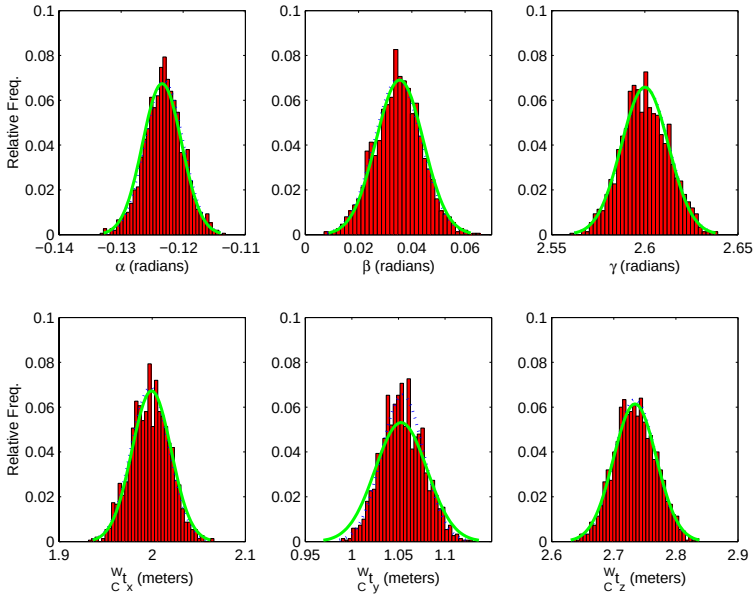


Figure 5.2: Histograms for the six pose parameters, obtained from the elements of the image–to–mosaic homography. The superimposed lines represent Gaussian distributions whose variances were obtained from the predicted covariance matrix.

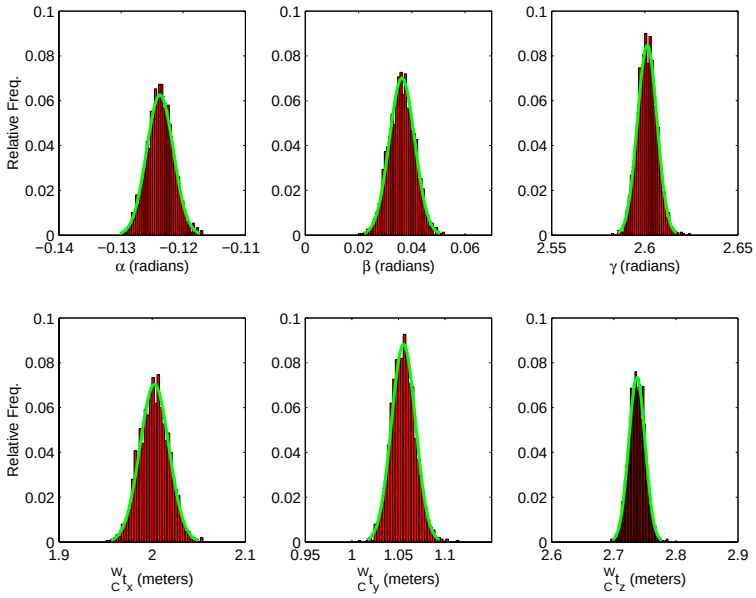


Figure 5.3: Histograms of the Monte Carlo simulations for the maximum likelihood estimator, where 1500 instances of the pose where obtained from the noisy coordinates of image points. The predicted covariance is represented as a full line. The graphs have the same scales as Figure 5.2.

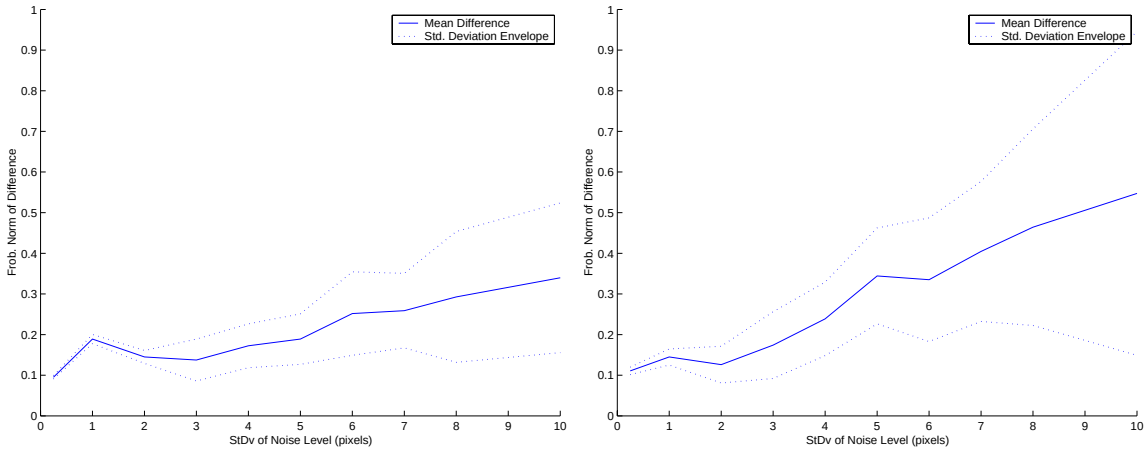


Figure 5.4: Differences in the predicted and empirical covariances with increasing noise levels for the orientation (left) and translation (right).

components of each predicted uncertainty were compared to the corresponding ones of the empirical covariance. The plot illustrates the average distortion with the standard deviation envelope. Although the average distortion grows in a smooth way, it can be inferred from the envelope that, for this pose, the prediction becomes unreliable for noise levels above 5 pixel standard deviation

5.5 Results

In order to evaluate the performance of the pose estimation algorithms, accurate ground-truth is required. For this reason we have used the mosaic of Figure 5.5 and synthesized new views according to a specified camera matrix and trajectory. These images are then used to retrieve the camera and position parameters. The mosaic was set to cover an area of 6 by 14.5 meters. The sequence comprises 40 images of 320×240 pixels taken by a camera on a moving vehicle combining 3D motion and rotation. The camera is pointing downwards and slightly forward, with a tilt angle of approximately 150 degree with respect to the horizontal. The used intrinsic parameters matrix K accounts for a skewless camera with

$$K = \begin{bmatrix} 480 & 0 & 160 \\ 0 & 480 & 120 \\ 0 & 0 & 1 \end{bmatrix} .$$

The uncertainty on the intrinsic parameters was also taken into account. Such uncertainty was modelled as a zero mean perturbation on the 4 elements of the skewless camera

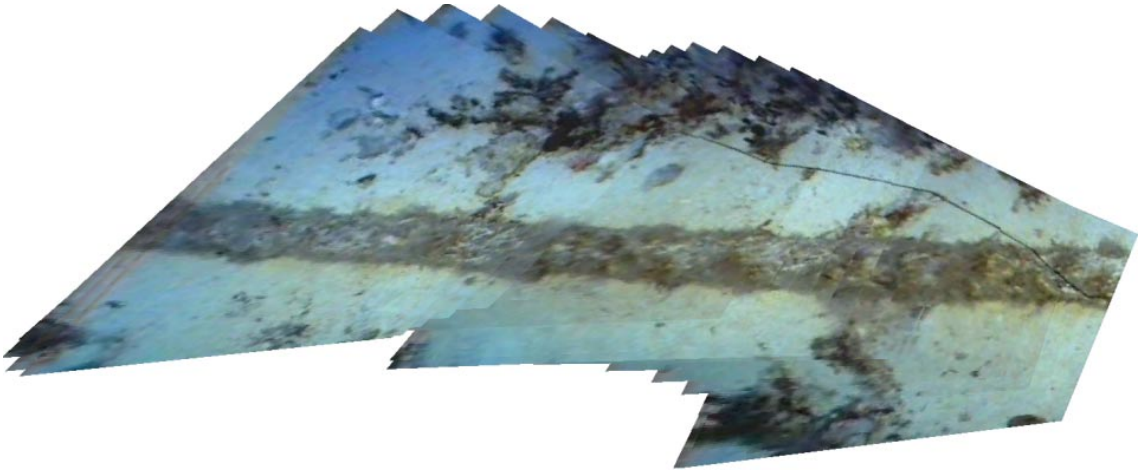


Figure 5.5: Underwater mosaic used for ground–truth, yielding a top view of the sea floor.

matrix, namely the scaling factors fk_u and fk_v, and the principal point location u_o and v_0, with the following variances

$$\sigma^2_{fk_u} = \sigma^2_{fk_v} = 25.0 \text{ and } \sigma^2_{u_o} = \sigma^2_{v_o} = 12.9 \ .$$

These variances were estimated from the calibration data of the ROV camera (Appendix A.2).

To simulate the vehicle drift induced by water currents a perturbation was added to the nominal forward motion of 0.23 meters/frame and to a nominal height above sea floor of 3 meters. The perturbations account for periodic drifts of around 0.4 meters in position and 15 degrees in orientation. For each frame, the combined movement of the camera is depicted in Figure 5.6, where the camera is represented with its optical axis.

5.5.1 Pose Estimation Results

The images from the created sequence were registered directly on the mosaic, using the robust point matching algorithm described in Chapter 4. For each frame, the number of selected inliers ranged from 16 to 39 pairs of points.

The maximum likelihood estimator was derived under the assumption of approximately Gaussian error on the observations (Eq. (5.8)). This assumption was tested by gathering the residues of all the point matches, and computing the empirical probability density function. The histogram for the set of 2208 residues is presented on the left hand side of Figure 5.7, with a superimposed normal density fit. The acceptably good fit indicates

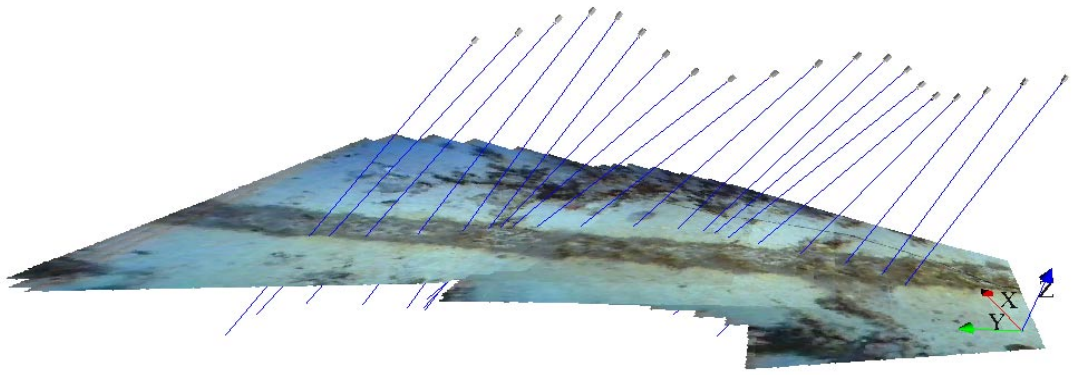


Figure 5.6: 3–D view of the camera positions and corresponding optical axes used for generating the sequence with available ground–truth. The origin of the 3–D world referential is represented by the system of three axes on the lower right, where each axis is drawn at 1 meter length.

Method	Position Errors (meters)		Angular Errors (degrees)	
	Avg. Norm	Avg. Unc. ($\times 10^{-3}$)	Avg. Norm	Avg. Unc. ($\times 10^{-3}$)
Algebraic	0.026	2.723	0.488	0.025
Max. Like.	0.016	0.874	0.254	0.008

Table 5.1: Trajectory recovery results for the algebraic and maximum likelihood methods. The average norm refers to the mean value of the error distance norm, while the average uncertainty refers to the mean value of the 50% uncertainty volume.

that the assumed probability model is justified. The isotropy on the projections errors is apparent on the right hand side of the same figure. Here, each dot represents the pair of residues associated with the (u, v) image coordinates of each image point.

For the covariance prediction, the uncertainty in the image projections was modelled as additive Gaussian noise, independent for each coordinate, with 1 pixel standard deviation.

Statistics on the reconstruction errors are presented in Table 5.1. The position errors were measured by taking the Euclidean distance between the ground-truth position and the estimated position. As for the orientation, the error was measured by computing the angle between the true and estimated camera frame orientations.

As expected, the lowest position and orientation errors are achieved with the maximum likelihood estimator. The difference in the methods performance can be explained by the fact that the two methods are not imposing the same constraints during the estimation process. The maximum likelihood estimator retrieves the pose directly from point

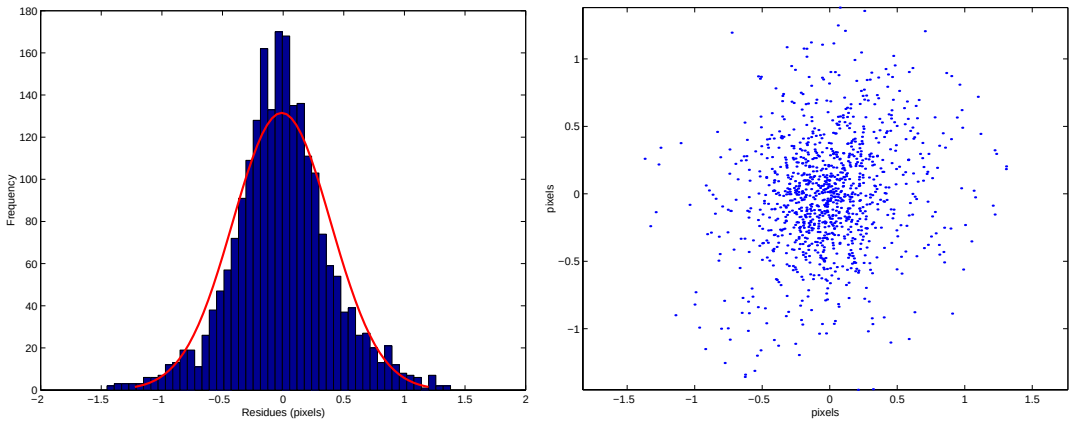


Figure 5.7: Distribution of the residues for the maximum likelihood pose estimator. The histogram on the left was created from the 2208 residues of all the selected point matches during the registration of the ground–truth sequence on the mosaic. A fitted Gaussian probability density function is superimposed. On the right, the spatial distribution of the pairs of residuals associated to each image point is plotted.

coordinates whereas the algebraic method uses the image–to–mosaic homography as an intermediate representation. The homography can model the mosaic view of any camera with arbitrary upper triangular intrinsic parameter matrix. As parameterized, it has 8 degrees of freedom, thus exceeding by 2 the number of pose parameters. Since the homography estimation (in the algebraic method) does not impose the particular structure due the projection camera intrinsics, the fit of the noisy image coordinates is worse than the maximum likelihood method.

The 3-D views of the recovered trajectories are depicted in Figure 5.8, where the larger uncertainty ellipsoids of the algebraic method are clearly visible.

The execution times were compared, to illustrate the differences in the numerical complexity of the methods. The results for a typical set of 34 matched points are presented in Table 5.2. Since no iterative minimization is required for estimating the pose in the algebraic method, the execution speed is more than an order of magnitude faster than the maximum likelihood estimator. For the case of the covariance prediction, the difference between the two methods is not as impressive. This is due to the derivatives of the cost function with respect to the coordinates of the matched point being computed by a numerical approximation, for both methods.

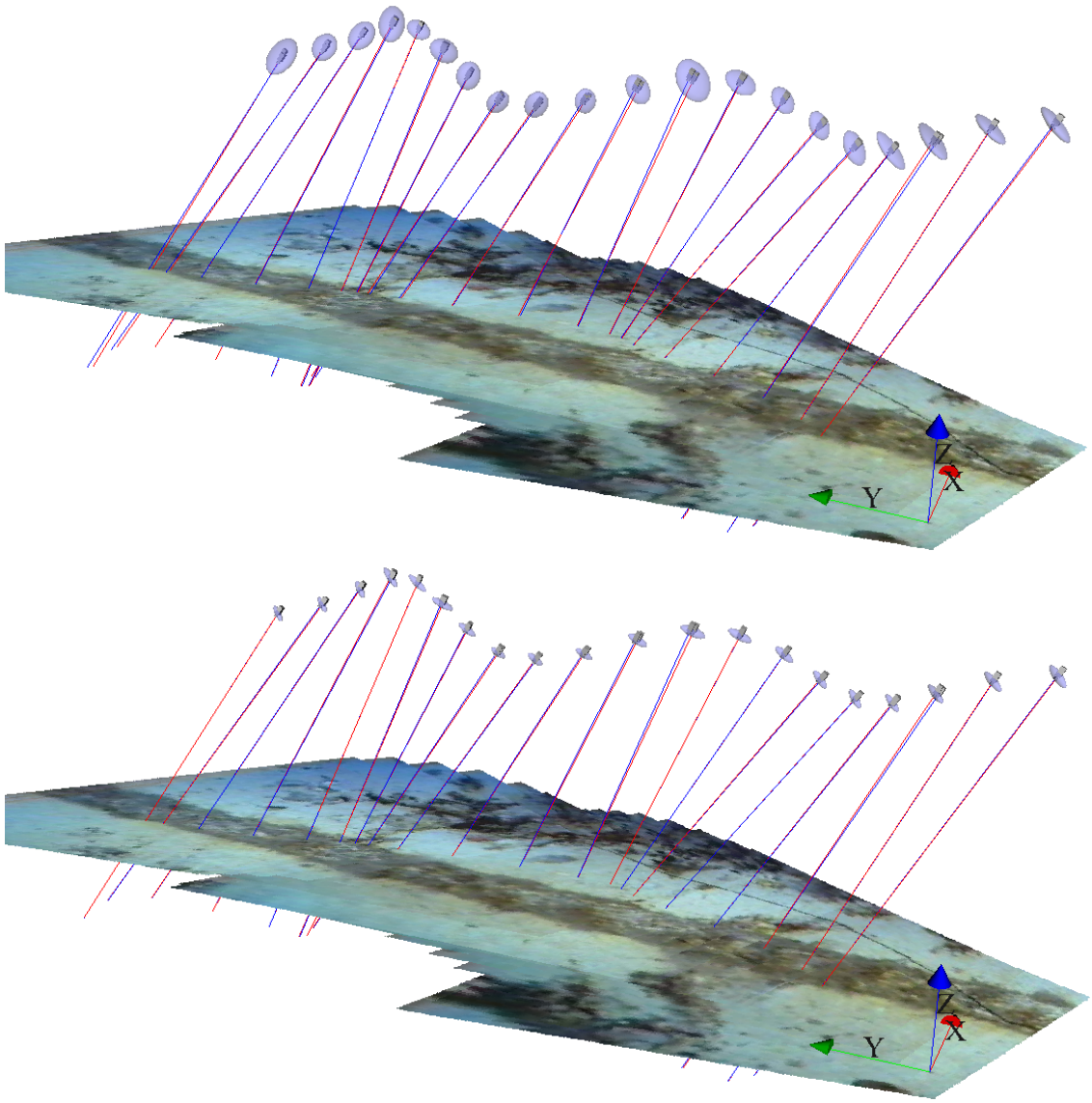


Figure 5.8: 3–D views of the estimated trajectory positions and uncertainty ellipsoids for the pose recovery. The upper image corresponds to the pose estimated from the image–to–mosaic homography Ψ, while the lower was obtained directly from the point match coordinates. The original camera axes are drawn in a darker colour (blue), while the recovered camera axes are drawn in a lighter colour (red). The size of the ellipsoids are set for a 50% probability, and only one out of two camera poses is plotted.

Method	Pose Estimation	Covariance Prediction
Algebraic	0.061 sec.	2.62 sec.
Max. Likelihood	0.767 sec.	4.44 sec.

Table 5.2: Execution times for the pose estimation and covariance prediction. The same set of 34 matched point coordinates was used for both methods. The methods were coded in Matlab (using built-in functions such as the SVD) and executed on a 800 MHz PC.

5.5.2 Pose from inter-image homographies

An additional experiment was conducted in order to compare the following image registration schemes:

- Image-to-mosaic homographies computed by direct mosaic registration

- Image-to-mosaic homographies computed by cascading inter-images homographies

The first scheme refers to the algebraic method using known intrinsic parameters. In the second, the true camera position and orientation is used for computing the first image-to-mosaic homography Ψ_1. The subsequent homographies are calculated by,

$$\Psi_i = \left(\prod_{k=2}^{i} T_{k,k-1} \right) \cdot \Psi_1 \qquad i > 1 \ ,$$

where $T_{k,k-1}$ are the inter-images homographies and the matrix product is computed by right-multiplying for each increment of the index k. The set of $T_{k,k-1}$ was estimated from the same sequence of images, and the number of used matched points varied from 10 to 76 pairs, with an average of 60.

Figure 5.9 and Figure 5.10 present, respectively, the plot of the positions errors for each frame, and a 3-D reconstruction of the two trajectories. It can be seen that the second scheme produces much less accurate results, due to the fact that small errors, inherent to the inter-image homography estimation, are accumulated. This phenomenon is in many ways comparable to the positioning errors arising from the use of dead-reckoning during navigation.

5.6 Discussion

This chapter presented two methods for solving the problem of recovering the 3D position and orientation of a camera, from a view of a previously created mosaic. An algebraic

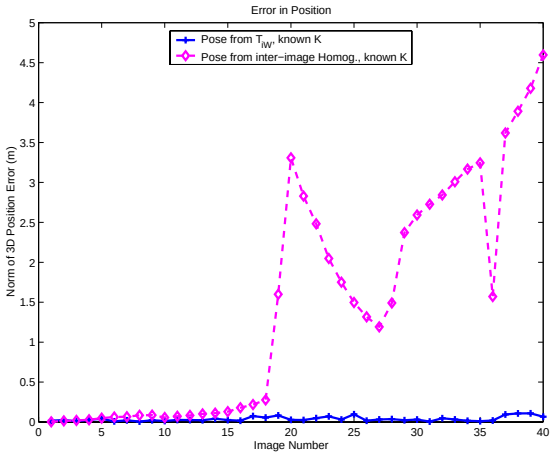


Figure 5.9: Position error for the pose recovery methods using direct mosaic registration, and inter–image homography cascading.

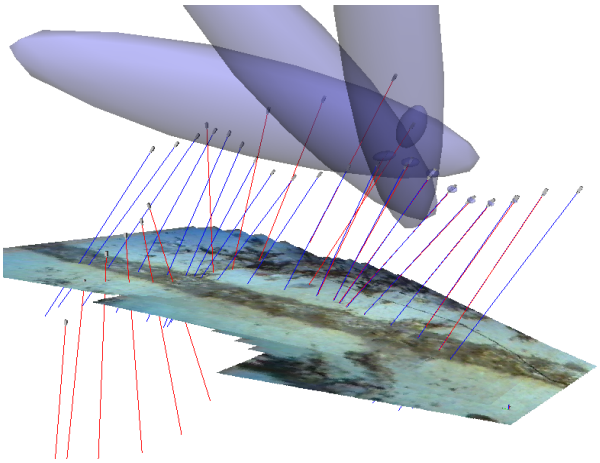


Figure 5.10: Estimated trajectory positions and uncertainty ellipsoids for pose recovery using inter–image homographies. Only one, out of every two recovered camera positions, is plotted. The ellipsoids are set for a 50% probability, but due to their rapid growth, only the first half are drawn.

solution was presented in which an intermediate image–to–mosaic homography was explicitly computed as a first step. In a second step, the pose was estimated directly from the 9 elements of the homography. This solution has the advantage of allowing for a very fast non–iterative implementation, since each step resorts to a single singular value decomposition. However, the special structure of the projective camera is not fully exploited during the homography estimation, since this would destroy the linear nature of the estimation problem (and imply an iterative optimization procedure).

For the cases where higher accuracy is required, an iterative estimator was devised, that uses directly the coordinates of point matches. By defining an observation equation and a model for the observation noise, a maximum likelihood solution was obtained.

For each method, the associated uncertainty in the pose parameters was implemented using a first order approximation. For the levels of noise involved, the approximation was validated by the good fit between the predicted and measured statistics. The pose estimation methods were evaluated using an image sequence with ground-truth. Their performance was compared both in terms of pose error and in terms of predicted estimate covariance.

The importance of the uncertainty estimation is twofold. Firstly, it provides quantitative measures for the comparison of different pose estimation algorithms. Secondly, for practical setups, it allows the on-line monitoring of the quality of trajectory reconstruction. This last aspect is of extreme importance in situations where there is a high cost associated with the risk of loosing a vehicle, due to poor positioning.

Chapter 6

Visual Navigation

This chapter presents an approach for vision–based autonomous navigation using mosaics. The main purpose is the validation of the mosaicing process presented in Chapter 4 as a way of creating navigation maps.

A navigation system is presented for an underwater robot, navigating close to the sea floor. As a design option, *only visual information* is used as sensor input for the generation of the motor commands. Such feature increases significantly the requirements on the reliability of the localization process. This contrasts with sensor fusion approaches, where a high degree of reliability is achieved by using information from several complementary sensors[1].

Using visual information alone also raises the issue of map adequacy in supporting localization. Different regions of the mosaic may have very different image content and will not support localization with the same level of accuracy. As an underwater example, in an area with slowly moving algae, the scene may be static enough for mosaicing. However, the appearance may change over time, making it useless for posterior localization. A similar problem occurs near the mosaic borders, including not only the outer perimeter of the mosaic, but also the vicinity of internal unmapped regions (*i.e.* mosaic "holes"). In such areas, the region of support for the localization may be reduced. A convenient way of overcoming this problem by forcing the vehicle to navigate along the areas where it can easily locate itself.

The proposed approach for mosaic–based navigation is schematically illustrated in

[1]It should be clear that, in terms of robustness, multi–sensor solutions are preferable provided adequate sensor characterization. Nonetheless, it is of scientific relevance to know how far can vision sensing go, when used by itself.

Figure 6.1. It comprises 3 distinct modules [36]:

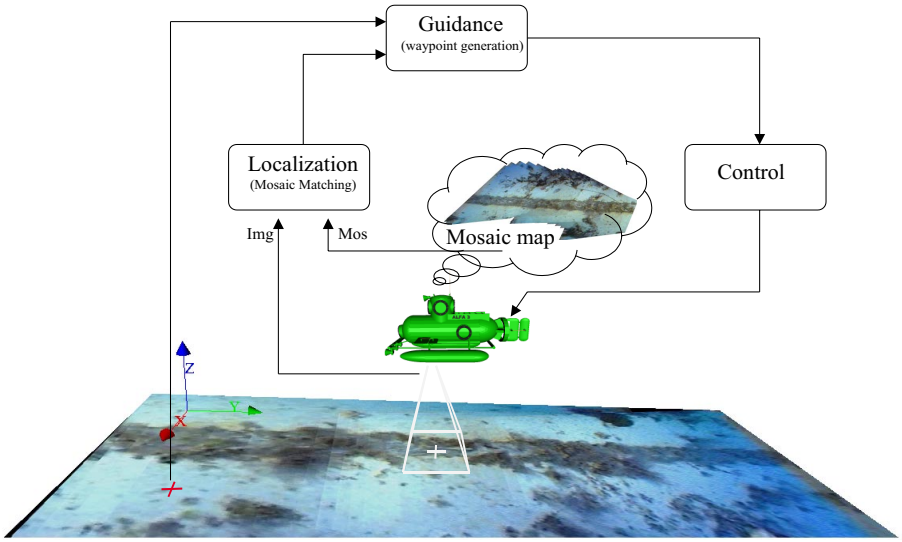


Figure 6.1: Overall visual servoing control scheme.

Localization – This module deals with real–time position sensing. It uses a previously created mosaic and the current image from the vehicle camera to provide all the information needed for navigation.

Guidance – A trajectory generation module defines a set of intermediate waypoints between the current vehicle position and the goal point. The resulting trajectory is optimized with respect to a criteria penalizing the total distance and favoring the regions that support accurate positioning.

Control signal generation – A control strategy, based on visual servoing is employed. A control law is devised using error measurements in the sensor space, *i.e.* image coordinates.

6.1 Localization

The first step of mosaic localization consists of finding the initial match between the current camera image and the corresponding area on the mosaic. Once the current image has been successfully registered, the on-line tracking of the vehicle position can be performed at a high update rate.

6.1.1 Initial Mosaic Matching

In order to avoid an exhaustive search over all the mosaic area, an estimate of the vehicle 3D position and orientation is desirable. This can be provided by some other modality of autonomous navigation in which a coarse global position estimate is maintained, such as beacon-based navigation or surface GPS reading. From this estimate, a corresponding homography H_{try} is computed and used for searching for point matches. In the experimental setup used in this thesis, no external positioning modality was available to provide an initial pose estimate. Therefore, this pose was computed from a very coarse matching of 3 points, that were manually provided.

Searching the vicinity

If the matching is not successful on the first attempt, then subsequent tries are performed around the vicinity of the first try. The matching attempts are performed using the same image, but over slightly different areas of the mosaic. Therefore, this process does not require moving the vehicle to acquire a new image at a different position.

A simple search strategy was implemented by considering different versions of the H_{try} matrix, corresponding to rotations around the image center or translations along one of the image axes. The rotations comprise -10, 0 and 10 degrees, while the translations are of a quarter the size of the image (along the respective axis). The matching with the rotated versions is attempted before translating. The translations follow a spiral-shaped search pattern around the original location. This pattern insures that the locations closer to the position of the original attempt are tried first. The sequence of matching tries is schematically illustrated in Figure 6.2. This process is repeated until a reliable match is found, or the search region is completely covered without success.

Limiting the search region

The uncertainty information on the external pose estimate can be used to limit the search region on the mosaic. This is achieved by propagating the uncertainty on the pose parameters Θ to the point υ where the optical axis intersects the world plane.

As defined in Section 5.1, the pose parameters $\Theta = \begin{bmatrix} \alpha & \beta & \gamma & {}^{W}_{C}\mathbf{t}_x & {}^{W}_{C}\mathbf{t}_y & {}^{W}_{C}\mathbf{t}_z \end{bmatrix}^T$ contain the 3 angles and 3 translation values that describe the camera frame in 3–D world coordinates. The point of intersection is given by the 3–D rigid transformation relating

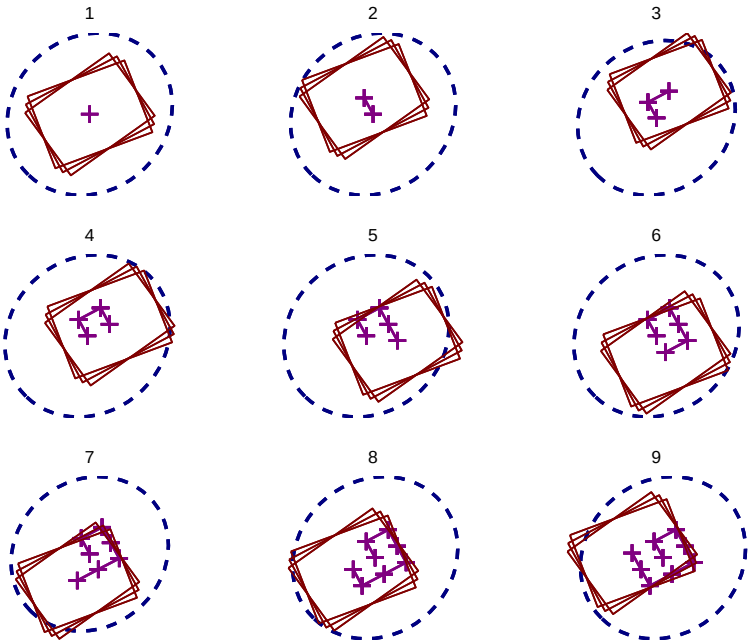


Figure 6.2: Sequence of attempts for the initial image–to–mosaic matching. Three rotated version are tried, before each translation. The translations follow a spiral pattern.

points in the world and camera frames

$$\begin{bmatrix} {}^{C}\mathbf{x} \\ {}^{C}\mathbf{y} \\ {}^{C}\mathbf{z} \end{bmatrix} = {}^{C}R_{W} \left(\begin{bmatrix} {}^{W}\mathbf{x} \\ {}^{W}\mathbf{y} \\ {}^{W}\mathbf{z} \end{bmatrix} - \begin{bmatrix} {}^{W}_{C}\mathbf{t}_{x} \\ {}^{W}_{C}\mathbf{t}_{y} \\ {}^{W}_{C}\mathbf{t}_{z} \end{bmatrix} \right) ,$$

with the additional constraints specifying the optical axis and the world surface

$$\begin{matrix} {}^{C}\mathbf{x} = {}^{C}\mathbf{y} = 0 \\ {}^{W}\mathbf{z} = 0 \end{matrix} .$$

This system is easily solvable for the intersection coordinates,

$$\upsilon = \begin{bmatrix} {}^{W}\mathbf{x} \\ {}^{W}\mathbf{y} \end{bmatrix} = {}^{W}_{C}\mathbf{t}_{z} \cdot \begin{bmatrix} \frac{\sin\beta}{\cos\beta\cos\gamma} \\ -\frac{\sin\gamma}{\cos\gamma} \end{bmatrix} + \begin{bmatrix} {}^{W}_{C}\mathbf{t}_{x} \\ {}^{W}_{C}\mathbf{t}_{y} \end{bmatrix} .$$

For small perturbations around Θ, υ can be approximated by its first-order Taylor expansion. Under the condition of small additive noise, the covariance matrix $\Sigma_{\Delta\upsilon}$ associated with υ, is given by

$$\Sigma_{\Delta\upsilon} = J(\Theta) \cdot \Sigma_{\Delta\Theta} \cdot J(\Theta)^{T} ,$$

where $J(\Theta)$ is the partial derivatives matrix of υ with respect to Θ,

$$J(\Theta) = \begin{bmatrix} 0 & \frac{{}^{W}_{C}\mathbf{t}_{z}}{\cos^{2}\beta\cos\gamma} & \frac{{}^{W}_{C}\mathbf{t}_{z}\,\sin\beta\sin\gamma}{\cos\beta\cos^{2}\gamma} & 1 & 0 & \frac{\sin\beta}{\cos\beta\cos\gamma} \\ 0 & 0 & -\frac{{}^{W}_{C}\mathbf{t}_{z}}{\cos^{2}\gamma} & 0 & 1 & -\frac{\sin\gamma}{\cos\gamma} \end{bmatrix} .$$

For a given probability P, v and $\Sigma_{\Delta v}$ define the ellipse over the mosaic whose inner region has a probability P of containing the actual intersection of the optical axis with the world plane.

A simulated example of using the pose uncertainty for setting the bound of the search area is given in Figure 6.3. In this example the pose estimate is such that the camera is fronto–parallel at the distance of 1 meter from the sea floor. The covariance matrices for the pose and intersection coordinates are

$$\Sigma_{\Delta\Theta} = \begin{bmatrix} 0.015 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.015 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.015 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.075 & 0.010 & 0 \\ 0 & 0 & 0 & 0.010 & 0.075 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.075 \end{bmatrix}, \qquad \Sigma_{\Delta v} = \begin{bmatrix} 0.089 & 0.010 \\ 0.010 & 0.089 \end{bmatrix}.$$

The ellipse is set for a 0.85 probability.

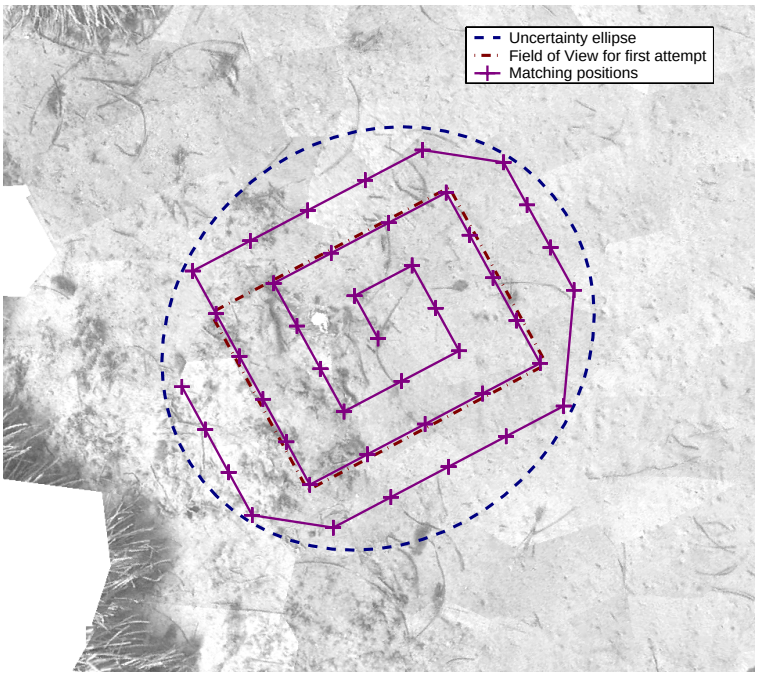


Figure 6.3: Example of the search area over the mosaic bounded by an error ellipse.

6.1.2 On–line tracking

This tracking comprises two complementary processes which run in parallel, at very distinct rates.

Absolute localization – The current image is matched directly over the mosaic, in order to have an absolute position estimate. This procedure is similar to the initial match, in the sense it uses the current position estimate to restrict the search area, and the spiral search pattern in case of the initial matching failure. This is a slow process.

Incremental tracking – This process measures the image motion by matching consecutive frames of the incoming video stream. This motion is integrated and used to update the vehicle position. The matching is performed using the same algorithm as described in Section 4.2 for the first part of the mosaic creation. The used motion model is the 4 d.o.f. similarity homography. The success of the image matching is assessed by the amount of point that are selected as correctly matched. In the case of unreliable measures, occurring when the number of selected matches is close to the minimum required for the homography computation, the resulting homography is discarded and replaced by the last reliable one[2]. This tracking is a fast process.

The complementary nature is illustrated by the fact that the two processes address different requirements of the position estimation needed for control and navigation: *real-time operation* and *bounded errors*. The absolute localization is a time-consuming task mainly due to the fact that a successful mosaic matching might not be achieved on the first attempt. However it provides an accurate position measurement.

Conversely, the incremental tracking is a much faster process but tends to accumulate small errors over time eventually rendering the estimate useless for our control purposes. It is also worth noting that this scheme is well fit for multiprocessor platforms, as the two processes can be run separately.

The contributions from the two processes are combined by simply cascading the image-to-image tracking homographies over the last successful image-to-mosaic matching. A typical position estimation update rate of 7 Hz is attained, on a dual–processor machine.

6.2 Trajectory Generation

As referred to before, the main purpose of generating trajectories is to guide the vehicle into avoiding the map areas in which the mosaic matching is more prone to failure. Examples of such are the areas of non-static algae, the mosaic borders or regions that were not

[2]This is valid under the assumption that the vehicle motion does not change abruptly between frames.

imaged during the mosaic acquisition phase.

For the results of this thesis, only the distance to the mosaic borders was considered, but the method can straightforwardly be used to avoid any region defined in the map. The trajectory generation is achieved by creating a cost map (offline) defining the regions to avoid, and by searching for a minimal cost path (online).

6.2.1 Cost Image

The first step consists of the creation of a cost map. The cost map is an array which contains the cost associated with navigating over every elementary region of the map image. The regions to be avoided will have higher cost than the others. The cost map is created using the Distance Transform [72] on a reduced size binary image of the non–valid region of the mosaic map.

The outcome of this operation is a cost image in which each pixel of the valid mosaic region contains a positive value that decreases with the distance to the border of the valid region.

6.2.2 Minimal Cost Path

Given the current position of the vehicle on the mosaic and the desired end position, we want to find the path that minimizes the accumulated cost over the cost map. This minimization problem can be formulated as a minimal path cost problem, where a path is defined as an ordered set of neighboring locations on the mosaic map. An efficient way to solve it is using Dijkstra's algorithm [82], which attains the optimal solution with a complexity of $O\left(m^2\right)$, where m is the number of pixels in the cost image. An example of the generation of trajectories using this method is presented in Figure 6.4.

The computation of the cost image is performed off-line, during the mosaic creation phase. Conversely, the generation of a new trajectory must be performed on-line during the mosaic servoing, whenever a new end-point is specified. For the purpose of avoiding the mosaic edges, a relatively small number of trajectory waypoints is required. Therefore the size of the cost image can be reduced, so that the computation of the trajectory does not compromise the on-line nature of the mosaic servoing.

Figure 6.4: Trajectory generation example – Valid mosaic region in white (left), cost image (center) and mosaic with superimposed trajectory (right). In the cost image, the darker regions have lower cost.

6.3 Vision Based Control

The geometry of the vehicle thrusters, with two horizontal and one vertical propeller, motivated the design of two decoupled controllers. The vehicle is controlled separately in the horizontal plane (over the sea floor to desired location), and in the vertical (maintaining a constant altitude). The heading is not controlled. The design is addressed within the framework of visual servoing strategies [50].

The implemented controllers were developed for visual station keeping and docking applications [122, 121], and are based on the approaches of Espiau *et al.* [18] and Malis *et al.* [64]. As the purpose of this chapter is the illustration of the navigation ability, only kinematic relations were used for the controllers. Also, no stability analysis is performed. Details on the modeling, identification and low-level control of the platform can be found in [27].

6.3.1 Servoing over the Mosaic

The objective of servoing over the mosaic is regulating to zero an error function relating the current position of the vehicle and the desired end–point (reference) in the mosaic, while rejecting external disturbances such as currents.

The coordinates of the end–point in the mosaic are initially defined in the mosaic bitmap image frame (for instance, by the human operator clicking over desired position). These coordinates can be translated into the vehicle's current image frame, since we assume the on–line tracking process to be operating, thus providing the current image–to–mosaic

homography.

In the following derivation we use normalized image coordinates, in order to simplify the equations. The normalized image coordinates relate to the standard image coordinates by a collineation. This collineation is defined by the intrinsic parameter matrix K, which is assumed to be known.

Let $\mathbf{s}_d = [x_d, y_d]^T$ be the desired reference point, in normalized image coordinates, and $s = [x_c, y_c]^T$ be the point in the image to be driven to the reference, as illustrated in Figure 6.5.

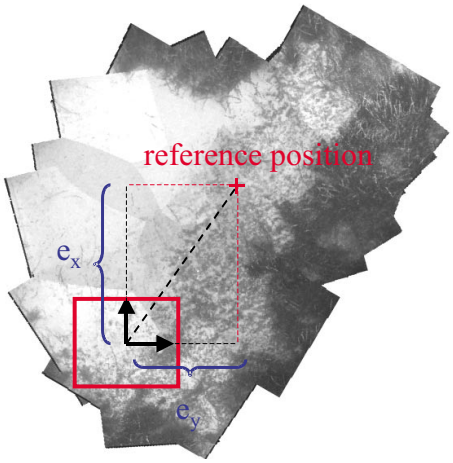


Figure 6.5: Definition of error measures on the mosaic. The current image frame is represented by the frame rectangle and the reference is marked by the cross.

The image error function is defined as

$$\mathbf{e} = \mathbf{s} - \mathbf{s}_d \ .$$

The instant velocity vector $\dot{\mathbf{s}}$ of a projected 3D point in the image, is related with the velocity of the camera. This kinematic relationship is represented by a matrix often referred to as the *image Jacobian* or the *interaction matrix* [50, 18] which satisfies

$$\dot{\mathbf{s}} = \mathrm{L}\mathbf{v}_{\mathbf{cam}} \tag{6.1}$$

where L is the image Jacobian and $\mathbf{v}_{\mathbf{cam}}$ is the 6×1 camera velocity screw [18],

$$\mathbf{v}_{\mathbf{cam}} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^{\mathbf{T}}$$

containing the linear and angular velocity components of the world frame with respect to the camera, expressed in the camera 3D coordinate frame.

For a generic $[x, y]^T$ image point, the image Jacobian defines a motion field [46] which is dependant on the depth Z to the originating 3D point, measured along the camera optical axis,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^2) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & (1+y^2) & -xy & -x \end{bmatrix} \mathbf{v_{cam}} \ . \tag{6.2}$$

As design option for the derivation of a control law, it is often desirable to impose an exponential decrease of the error function [64] along the time, such that

$$\dot{\mathbf{e}}(t) = -\lambda \cdot \mathbf{e}(t) \ ,$$

where λ is a positive constant that serves as a tuning parameter.

Using Eq.(6.2), we can then solve for the camera motion that guarantees this convergence,

$$\mathbf{v}^*_{\mathbf{cam}} = -\lambda \cdot \mathrm{L}(\mathbf{s}, \mathbf{Z})^\dagger \cdot (\mathbf{s} - \mathbf{s_d}) \ , \tag{6.3}$$

where $\mathbf{v}^*_{\mathbf{cam}}$ is the camera velocity that comprises the control objective and $L^\dagger$ is the pseudo-inverse of the image Jacobian.

It is now useful to relate the camera frame to the frame associated with the actuators (the vehicle frame). As we are dealing with velocity vectors, this relation can be expressed as a Jacobian J_{r2c} relating the two 3–D coordinate frames. If the camera is fixed with respect to the vehicle, the two frames do not change in time and $\mathrm{J}_{\mathrm{r2c}}$ can be computed beforehand. For the case of our testbed ROV, the vehicle frame is a pure translated version of the camera frame.

The relation between velocities is

$$\mathbf{v}_{cam} = \mathrm{J}_{r2c} \cdot \mathbf{v}_{robot} \ .$$

Given the underactuated nature of the platform, only a subset of the velocities in $\mathbf{v}_{robot}$ can be controlled by commanding the propellers. These refer to the surge (translation along the $\overrightarrow{x}$ axis), and yaw motion (rotation around the vertical $\overrightarrow{z}$ axis). The controllable degrees of freedom define a reduced velocity vector $\mathbf{v}^r_{robot}$

$$\mathbf{v}^r_{robot} = \begin{bmatrix} v_x & \omega_z \end{bmatrix}^T \ ,$$

and a corresponding reduced Jacobian J^r_{r2c} such that

$$\mathbf{v}_{cam} = \mathrm{J}^r_{r2c} \cdot \mathbf{v}^r_{robot} \ . \tag{6.4}$$

It is now possible to re-formulate the control objective in terms of desired vehicle velocity components, such that the image center is driven towards the reference point in the mosaic. Substituting (6.4) into (6.1), we obtain an expression that relates the image motion to the vehicle velocity:

$$\dot{\mathbf{s}} = \mathbf{L}\cdot \mathrm{J}_{r2c} \cdot \mathbf{v}^{r}_{robot} \ . \tag{6.5}$$

With this expression, we can solve for the vehicle velocity in the horizontal plane, necessary to guarantee the convergence of the image error function:

$$\mathbf{v}^{r*}_{robot} = -\lambda \cdot (\mathrm{L}(\mathbf{s},\mathbf{Z})\mathrm{J}_{\mathrm{r2c}})^{\dagger} \cdot (s - s_d) \ . \tag{6.6}$$

Figure 6.6 illustrates the structure of the complete visual servoing controller for the horizontal plane. The term B^{-1} was obtained from a dynamic model of the vehicle thrusters, and allows for the computation of the motor commands (PWM signals) that correspond to the required steady–state velocities [122].

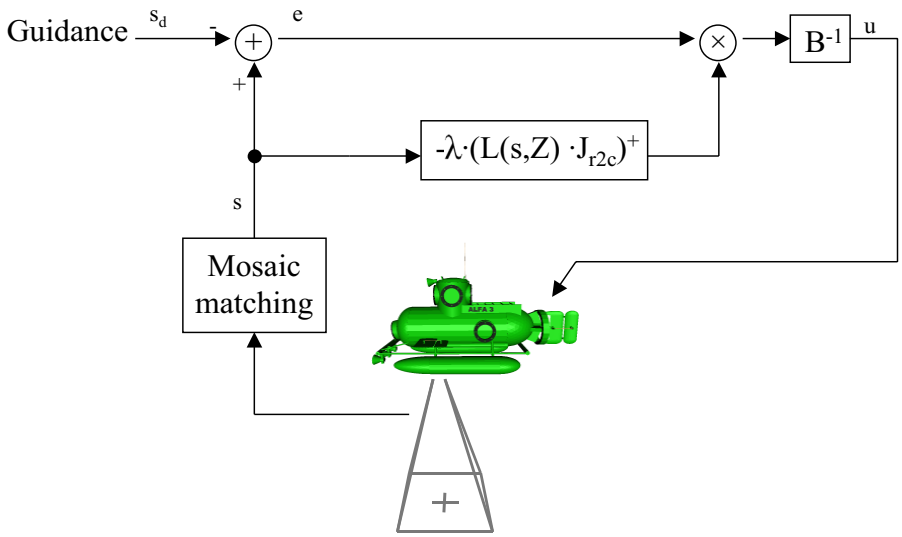


Figure 6.6: Control block diagram.

6.3.2 Altitude control

The controller for the vertical plane aims at maintaining the camera at a fixed altitude during navigation. This is achieved by regulating to zero the difference in scale between the current image–to–mosaic homography and a reference value.

The image scaling induced by applying a affine transform (defined in Table 2.2) can be recovered from the determinant of the upper left 2×2 submatrix [42]:

$$s = \sqrt{|\Psi_{2\times 2}|} \ . \tag{6.7}$$

This does not hold for general projective homographies (as the scale changes along the image), due to the projective distortion. However this is a suitable approximation if the camera is approximately fronto–parallel to the ground floor.

For the altitude controller, we consider the current image-to-mosaic homography and reconstruct its scaling factor. This is then compared to a reference scaling, taken from the initial image–to–mosaic homography, as to generate the control error

$$e = \ s - s_d \ .$$

A PID control action is used,

$$u_{robot} = -(K_p \cdot e + K_d \cdot \dot{e} + K_i \cdot \int e \ dt) \ , \tag{6.8}$$

in which the gains were manually tuned.

6.4 Results

An extensive set of mosaic servoing experiments were conducted at sea [35]. Some of the most illustrative are now presented.

As stated above, in Section 6.1.2, the on-line tracking comprises two complementary processes of position estimation, running simultaneously but at distinct rates. The mosaic matching was triggered in fixed intervals of 5 seconds, typically requiring 3 seconds to be complete if it was successful on the first attempt. The image–to–image tracking ran permanently over consecutive pairs of incoming images, and was used to update the current position estimate at approximately 7Hz. The image processing and servoing was run on a dual–processor 800MHz computer. The platform was not required to be fixed (*i.e.* motionless) during the mosaic matching since the image–to–image tracking was performed independently.

6.4.1 Visual Servoing

An illustrative underwater servoing experiment is presented in Figure 6.7 where a top-view of the ROV trajectory and references are plotted. The ROV completed several loop

trajectories and travelled for 165 meters, during a 11.8 minute run.

A more detailed view of part of the run is given in Figure 6.10, corresponding to 136 seconds and 3 end points. During this part, the platform took respectively 12, 22 and 20 seconds to navigate each of the 3 legs. The remaining 82 seconds were spent in keeping station at the end points[3].

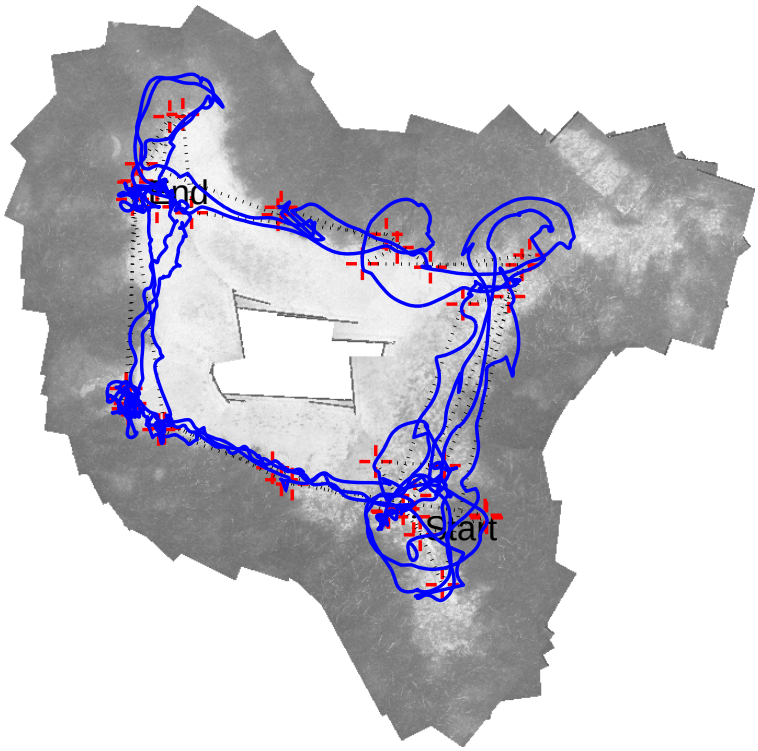


Figure 6.7: Underwater mosaic servoing experiment. This plot shows a top–view of the ROV trajectory for the complete run with the reference positions marked with crosses. The ROV trajectory was recovered for the on–line image–to–mosaic matching with updates from the image–to–image tracking, and is marked by the full line.

Another experiment is presented in Figure 6.9, where the ROV travelled for 159 meters, during a 7 minute run. Figure 6.10 details a 42 second part of the run. Here, the benefit of the path planning is clearly visible. During the second leg of the run (on the left), the vehicle avoids the undefined region in the center of the mosaic.

The end points were manually specified through a simple user interface, where the operator was required to click over the desired end position, directly over the mosaic map. A screen capture of the interface is shown in Chapter 1, Figure 1.2.

[3]Here, station keeping refers to sustaining the end point inside the camera field of view.

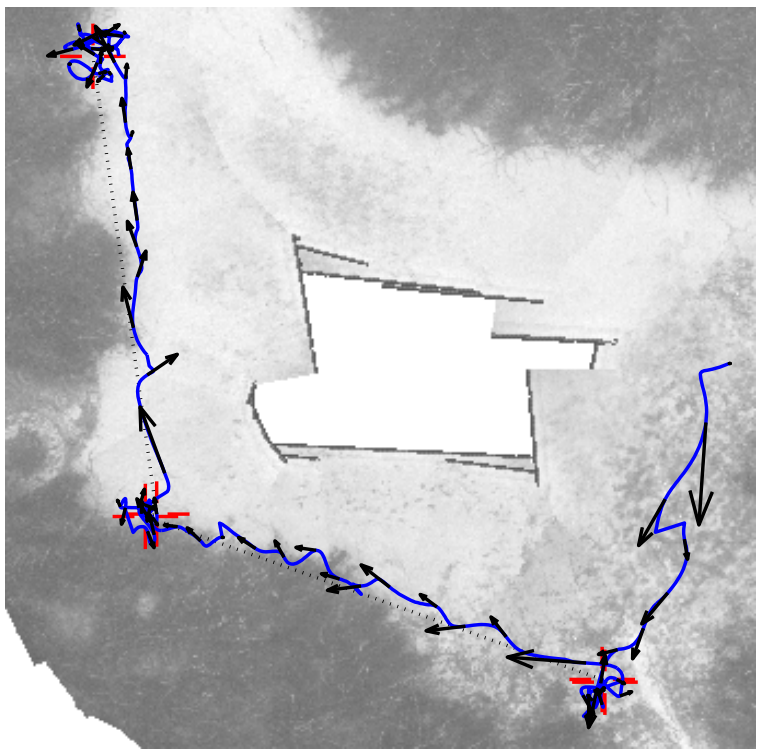


Figure 6.8: Trajectory detail comprising two endpoints – The generated path connecting the endpoints is marked by the dashed line. In order to allow the sense of speed, a set of arrows is superimposed. The arrows are drawn every 2 seconds and sized proportionally to the platform velocity. A sense of overall scale can be gained by noting the size of the car tyre on the left.

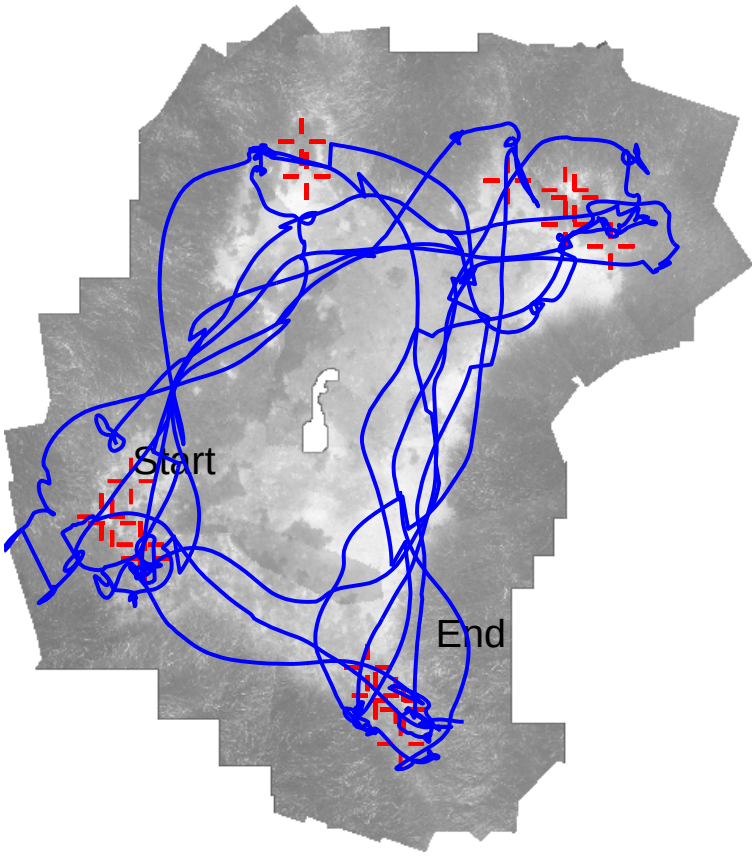


Figure 6.9: Underwater mosaic servoing experiment. This plot shows a top–view of the ROV trajectory for the complete run with the reference positions marked with crosses. The ROV trajectory was recovered for the on–line image–to–mosaic matching with updates from the image–to–image tracking, and is marked with the full line.

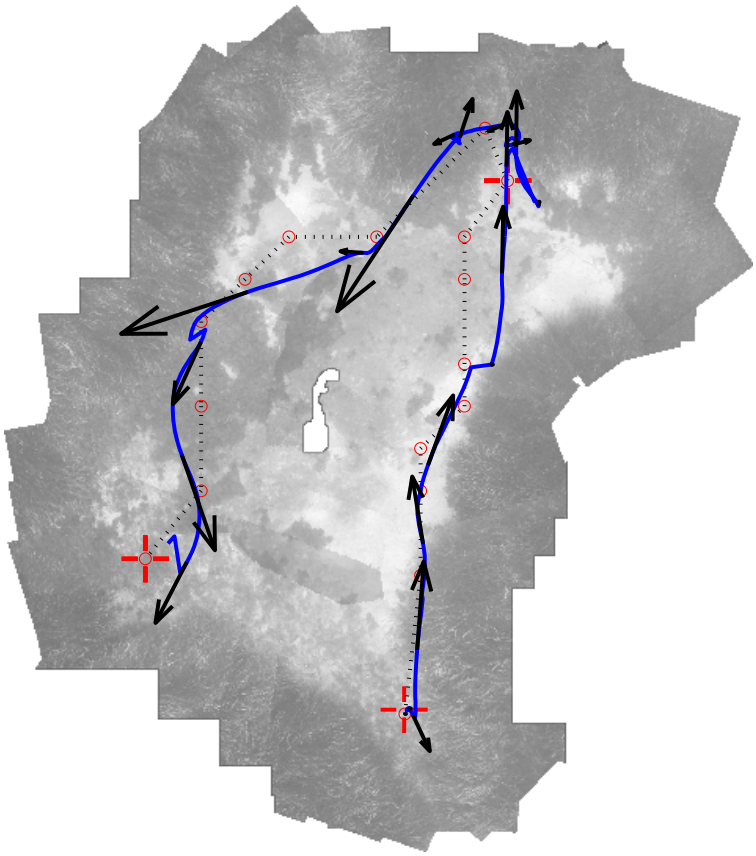


Figure 6.10: Trajectory detail comprising two endpoints – The generated trajectory, that connects the endpoints, is marked by the dashed line. The intermediate waypoints are circled. The arrows are drawn every 2 seconds and sized proportionally to the platform velocity.

During the run of Figure 6.9, 80 images were matched over the mosaic to obtain an absolute localization. More than 3000 images were used by the image–to–image tracking, to provide the fast position updates.

6.4.2 Uncertainty estimation

During the servoing experiments all the images that were matched over the mosaic, were also recorded on disk. As an off-line processing step, these images were re-matched over the mosaic, and the point correspondences were used to estimate both the full 6 d.o.f. pose and the associated uncertainty. For this computation, the following assumptions were considered:

- the only source of uncertainty was the limited accuracy on the point matching,
- point matches were affected with Gaussian noise, uncorrelated over the two coordinates,
- the standard deviation was 0.5 pixels for all coordinates. This value was experimentally measured from the residuals of the homography estimation[33].

The ellipsoidal uncertainty volumes associated with the translational part of the pose parameters, are represented in Figure 6.11. From the relatively flat, horizontally-levelled ellipsoids, it can be seen that the uncertainty on the camera position is larger along the $\overrightarrow{x}$ and $\overrightarrow{y}$ axes than along the $\overrightarrow{z}$.

6.4.3 Offline Matching

During the sea trials, the set of images used by the image–to–image tracker were recorded on disk for later processing. This allowed for the off-line matching of the whole sequence over the mosaic, using the same algorithms as the on-line mosaic matching.

It was therefore possible to recreate the trajectory using the 4 d.o.f. fronto-parallel parameterization for the pose. This trajectory was then used as *ground truth*, to evaluate the on-line estimates, which combined the incremental image–to–image tracking estimate with the last available mosaic matching.

Figure 6.12 plots the horizontal metric distance between the camera centres for the on-line and off-line estimates, during a selected period of 60 seconds. The duty cycle of the mosaic matching is represented as a square wave, where the rising edge corresponds

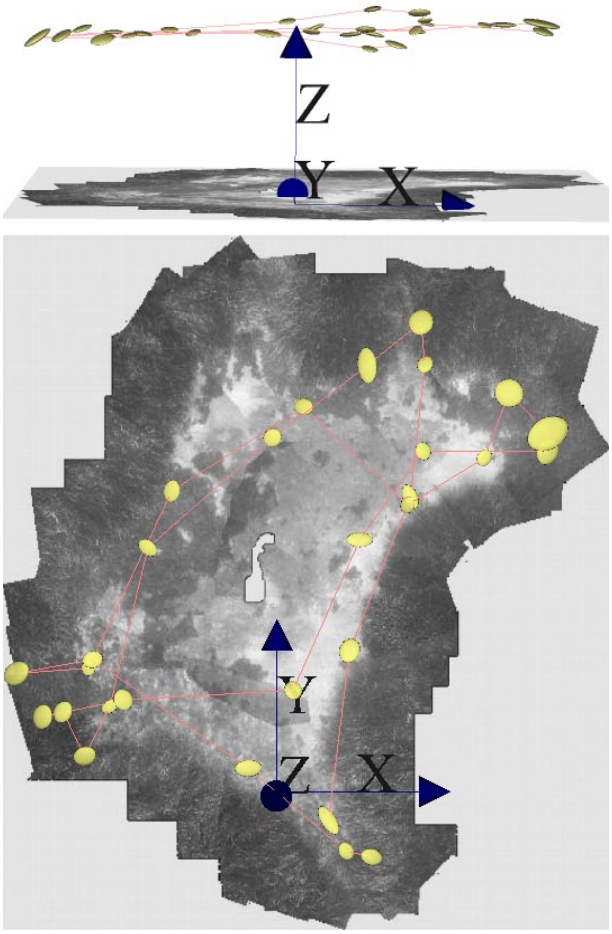


Figure 6.11: Mosaic servoing trajectory reconstruction – The two views show the camera positions associated with the images that were directly matched over the mosaic during the servoing run. The ellipsoids mark the estimated camera centers and convey the uncertainty assotiated with the translation part of the pose. The ellipsoid dimensions are set for a 50% probability. However, for clarity reasons, the ellipsoid axes sizes were enlarged by a factor of four, and only 143 seconds of the run are represented.

to the acquisition of a new image to be matched over the mosaic, and the falling edge corresponds to the instant when the mosaic matching information becomes available and the position is corrected. It can be noticed that the error does not fall to zero at the position correction instant. The reason behind this is that the mosaic matching position estimate is only available some time after the corresponding image was acquired. During such time interval (of around 3 seconds), the image–to–image tracker is integrating the incremental vehicle motion and therefore accumulating small errors.

This plot illustrates the need and importance of the periodic mosaic matching, which is apparent from the fast error build-up between mosaic matches, and in its fall once the matching is successful. This approach also presents the advantage of allowing the monitoring of the accumulated error during the on-line run, which can be directly measured immediately after a successful mosaic match. Although not taken into account in this set of tests, the magnitude of the accumulated error can be used to adjust the image–to–mosaic matching frequency, thus adapting to cases where the image–to–image tracking performance changes.

6.5 Discussion

The mosaic based servoing results show the feasibility of using vision as a single positioning modality for relatively large distances, and extended periods of time. The devised methods allow for the positioning for servoing, where the errors are bound by periodic mosaic matching, and for the uncertainty propagation, where the pose estimation quality can be assessed.

For the initial localization over the mosaic, an external pose estimate is used. As an alternative, one could resort to visual information alone to provide such estimate. A simple solution is performing area correlation of the image on the mosaic, over several orientations and scales. Such procedure was implemented in an early phase of this work, but proved too computationally heavy and inaccurate to be of practical interest for the sea trials.

During the navigation the pose estimate is maintained by combining mosaic matching with inter–frame tracking. The reason for not doing exclusively image–to–mosaic matching is the processing time involved. As stated above the image–to–mosaic matching for this sea trial conditions requires typically 3 seconds if the matching is successful in the

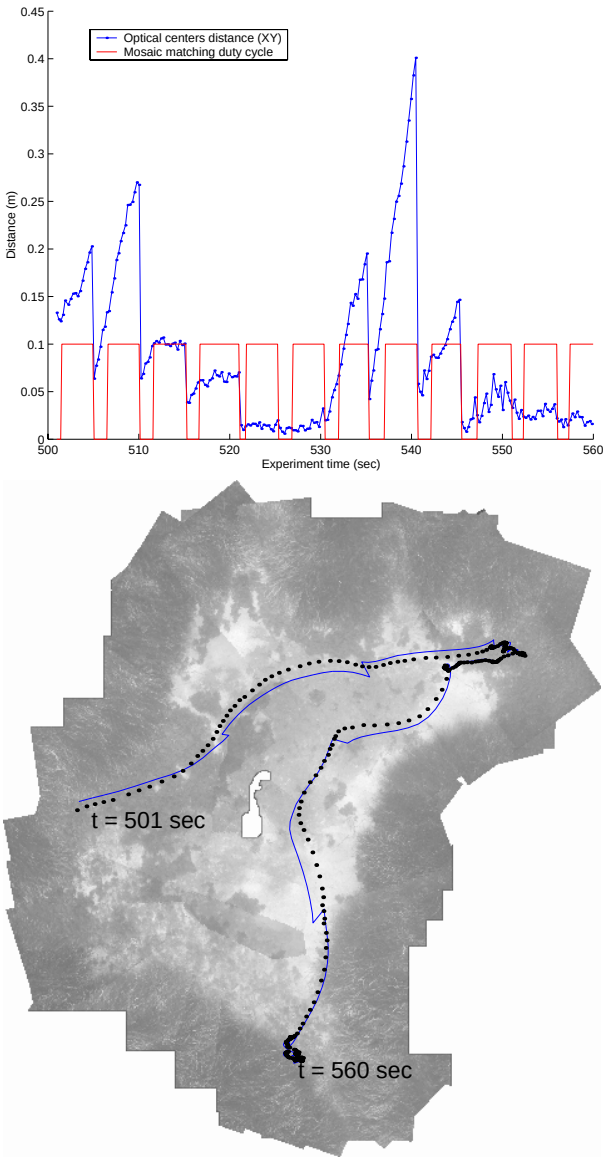


Figure 6.12: Difference between the online position estimate, using mosaic matching with inter–image tracking updates, and the offline estimate, obtained by maching all the images over the mosaic. The upper figure shows the horizontal (XY) distance error as a line with dots. On the lower figure the online trajectory is plotted as solid line, while the offline is marked by the dots. Each dot represents one image, acquired at 7 Hz.

first try. Otherwise it can take longer. Conversely the image–to–image can run at 7Hz. The difference in the processing times has to do with the dissimilarity between the on-line camera images and the corresponding areas of the mosaic. This is mainly due to illumination changes between the time the mosaic was acquired and the time it is used, and non-planarity and non-rigidity of the scene. To a lesser extent, the disparity of the processing times is also due to implementation issues, as we try to match a much larger number of correspondences and apply feature warping prior to the correlation. Even if the intervals between mosaic matches were reduced and a smaller number of correspondences were searched for, it would be difficult to achieve a position update frequency suitable for the visual servoing. However, being a computational issue, this trade-off between precision and availability is much dependant on the computing resources available.

Chapter 7

Conclusions

This chapter summarizes the work and contributions of this thesis. A set of interesting directions for some short term future developments is presented and discussed.

7.1 Summary and Achievements

This thesis addressed the problem of creating visual maps capable of supporting autonomous navigation.

Chapter 1 introduced the subject and overviewed the approach. The problem of underwater sensing for navigation was addressed. The most commonly used sensing modalities were presented and discussed, which allowed for pointing out the relative benefits and limitations of using vision. Chapter 2 reviewed some essential geometric and algorithmic aspects of the mosaic creation methods. Particularly important are the collineations in the 2–D projective space and robust model-based estimation which constitute the backbone of our mosaicing approach.

Chapter 3 discussed the most relevant techniques related with mosaic creation in robotics. Such techniques include image registration and the use of mosaics for robot navigation, both in land and underwater applications. Comparative information on the state–of–the–art was provided for a clearer understanding of the work in this thesis with respect to what has been accomplished before. A closing section discussed the most important features that are desirable on a mosaicing system intended for navigation.

In Chapter 4 a complete mosaicing approach was proposed with the purpose of creating accurate visual maps. The approach is able to deal with general 3–D camera motion and to exploit the time-distant superpositions due to loop trajectories. Such superpositions

are used to find new image matches. As a consequence, additional spatial constrains can be imposed thus promoting the accuracy and coherence of the resulting mosaic. The mosaicing algorithm is able to simultaneously create the mosaic, and to estimate the 3–D camera trajectory undertaken during the image acquisition, up to a scale factor. As a final step, a fronto–parallel view of the sea floor is obtained, since a geometric description of the world plane is also estimated. Illustrative results were obtained from image sequences acquired by a ROV in shallow waters. The images presented some of the common difficulties of underwater mosaicing, such as non planar sea-bottom, moving objects and illumination changes. The good performance and robustness of the complete algorithm was demonstrated.

The use of a mosaic as a map for localization is illustrated in Chapter 5. Two methods, addressing different requirements in precision and computation resources, were presented for recovering the complete 3–D pose from an image captured in a previously mosaiced area. An algebraic method provides a computationally inexpensive solution, by estimating the image–to–mosaic homography as an intermediate representation of the pose. This homography is then used to recover the 6 parameters that describe the position and orientation of the camera with respect to a world coordinate frame. For the cases where a pose estimate is required to have higher accuracy, a maximum likelihood solution was derived and illustrated.

An important feature of any sensor for navigation is the ability to provide not only accurate readings but also to provide an uncertainty measure associated with those readings. Taking this in mind, a first order propagation of the uncertainty in the pose (as a function of the uncertainty in the point matches) was illustrated. Such propagation was shown to be accurate to a conservatively high level of noise in the point matches. The uncertainty propagation is also useful in providing a way to compare the performance of the estimators.

Chapter 6 presented an approach for autonomous mosaic–based navigation. This approach served two main purposes. The first purpose was validating the mosaicing process of Chapter 4 as way to provide maps capable of sustaining navigation for periods of several minutes. The second purpose was assessing if visual sensing could be used alone to provide all the real–time information required for controlling the position of a underwater platform. The implementation and testing of the approach successfully demonstrated the feasibility and performance of both objectives.

The navigation system comprises 3 distinct modules. These modules account for the *localization* with respect to the map, the *guidance* through the creation of a suitable trajectory to the goal position, and the *generation of control signals* by performing visual servoing. Typical localization requirements for navigation and control are *limited errors* and *fast update rate* in the measurements. Taking this into account, a set of efficient visual routines were used for localizing the vehicle with respect to the mosaic. The routines combine inter–frame and image–to–mosaic matching, to successfully meet such requirements.

A real–time path planning method was applied to ensure that the vehicle avoids navigating near the borders of the valid regions of the mosaic, thus increasing the chances of correctly positioning itself. A visual control scheme, based on image measurements, was proposed to drive the vehicle. It attained good overall performance for the trajectory following, given the underactuated nature of the test bed, and the fact that no dynamic model of the vehicle motion was used.

The complete methodology was tested at sea, under realistic and adverse conditions. It showed that it is possible, and practical, to navigate autonomously over the previously acquired mosaics for large periods of time, without the use of any additional sensory information.

7.2 Discussion

This thesis dealt with a number of issues related to the appropriateness of mosaicing techniques for underwater robots. Some of the key subjects are formulated in the following questions.

- *Why use vision sensing underwater?*

  Reverting to Chapter 1 where this topic was addressed, the main advantages of using optical vision sensing are the purveying of position information in world fixed reference frame, with fast updates, using inexpensive and readily available hardware. In our work, these advantages were illustrated by the mosaic servoing capability. Such capability was demonstrated, in sustained operation, using a comparatively inexpensive underwater inspection platform and off–the–shelf processing hardware.

- *Are video mosaics an adequate representation of the underwater environment?*

  Mosaics are not only useful *per se* as *extended views*, easily interpreted by human

operators in common tasks such as initial site exploration using manually driven ROVs, but can also play an important role as *spatial representations* in automated operations such as habitat mapping and periodic monitoring, in the near future.

By registering and blending images captured near the seabed, video mosaicing is effective in overcoming the short range nature of underwater optics which prevents long range imaging. We have illustrated this using image sequences from approximately planar sea floors. In our mosaicing process, apart from the step where the single plane constraint is enforced to promote the accuracy of the of the global registration, there is no fundamental shortcoming in applying the same image registration and topology estimation techniques to non-planar but locally smooth areas of the sea floor. However care would have to be exercised in selecting the most adequate motion model for the pair wise image registration, which is strongly dependant on the pose of the camera with respect to the floor. This is a topic for future work.

From the navigation autonomy point of view, there is a growing belief that the development of truly autonomous underwater vehicles of moderate cost must avoid external position systems. These systems, such as acoustic long baseline transponders or underwater GPS, are currently expensive and may require complicated logistics. By contrast, autonomous vehicles should be able to observe and interact with the environment in order to extract all the relevant *natural* cues for navigation. Such information may consist of visual, acoustic or magnetic features (to name but a few), that need to be combined and organized into useful navigation representations.

The subsea medium can be highly unstructured, specially in the coastal regions. Most often, the presence of natural features, capable of being used for navigation, will vary strongly from place to place. An example of this is the succession of sand banks, algae covered areas and rocky slopes that were present in our test location. Clearly, different types of sensors will be better suited for the different environments.

Bearing this in mind, video mosaics can have an important role as a *local representation* for the regions where the video mosaicing is best suited, namely in relatively smooth and textured areas. By contrast, acoustic bathymetry is better fitted for mapping rocky slopes and may be used as a local representation for those regions. Such local maps are the building blocks of a larger *global representation* where the geometric arrangement (or topological structure) of the local maps is maintained.

- *Can vision alone provide enough reliable information for navigation?*

  Within the scope of this thesis, this point was considered a design option, and was successfully validated under the specific testing conditions. However, there is no doubt on the benefit in fusing data from other sensors. Even restricting our considerations to low cost sensors, a clear advantage is obtained from using a compass or a depth sensor, which provide drift-free orientation and scale information. As an example, such information can assist the initial pair–wise image motion estimation, thus making it less prone to the dead reckoning accumulated error.

7.3 Directions for Future Work

A number of directions can be set for future work.

As illustrated, robust feature matching allows for the adequate creation of mosaic maps even in the presence of small non-planarities and moving algae, on regions that are predominantly planar. However, alternative scene representations may be helpful in extending the sea–bed area to be mapped. One of such representations is obtained by relaxing the single plane assumption. Instead of considering just one plane, a more general representation would be obtained by approximating the sea-bed by a piece–wise planar surface. Such representation raises the challenging problem of the automatic surface segmentation, which may, for example, be treated under a *maximum a posteriori* framework, after imposing a convenient regularization prior on the number or extent of the planar faces. The coplanarity test for noisy data given by Kanatani in [55] can be used for detecting non-planar surfaces. Alternatively one can consider the approach of Peleg [85] to mosaic smooth surfaces as manifolds.

Another issue is the real–time construction of the maps. In this work, it was opted for the higher quality attainable by batch methods, which are inherently computationally heavy. Still, it is of obvious advantage to have the mosaicing scheme operating as fast as possible, ideally in real–time, without sacrificing the accuracy. During the mosaicing process, the phase which is more time–consuming is the topology estimation, due to the matching and optimization iterations. Upon loop closure, faster propagation of the effects of the newly found neighboring relations, may be attained using the smoothing technique described in [120].

The issue of adequate area covering was not addressed, since the original sequences for

the sea-bed mosaics were acquired while the vehicle was being hand–driven. An area covering strategy is important to insure that all the region of interest is covered, thus avoiding creating mosaics with gaps, and guaranteeing the existence of sufficient area overlap between swaths for the topology estimation. This is a commonly overlooked aspect in the underwater mosaicing literature, mainly because the mosaic creation has been traditionally regarded as a passive process. This contrasts with more active vision approaches, common in the land robotics literature, where perception and action are directly intertwined. In our work, consecutive image matching and motion estimation was performed in real–time, during acquisition. These measurements can be used under closed loop control, which is the first step towards trajectory following.

Another interesting development would be the real–time analysis of the scene structure in order to decide whether it is suitable for mosaicing. Under specific environments such as shallow waters, texture analysis can be successful in detecting algae boundaries [108], thus delimiting the areas adequate for mosaicing.

Appendix A

Underwater Experimental Setup

The video sequences acquisition and navigation experiments were conducted using a commercial grade ROV. The platform was provided by Deep Ocean Engineering [17] and was based on the Phantom 500 class of inspection robots. Custom modifications were performed during factory assembly, to allow for closed–loop control by a standard personal computer.

A.1 Vehicle Description

The ROV is illustrated in Figure A.1. The standard vehicle comprises a open frame structure which houses a single water–tight hull. The metal crash–protection frame is approximately 1 meter long by 0.65 meter wide and 0.65 meters high.

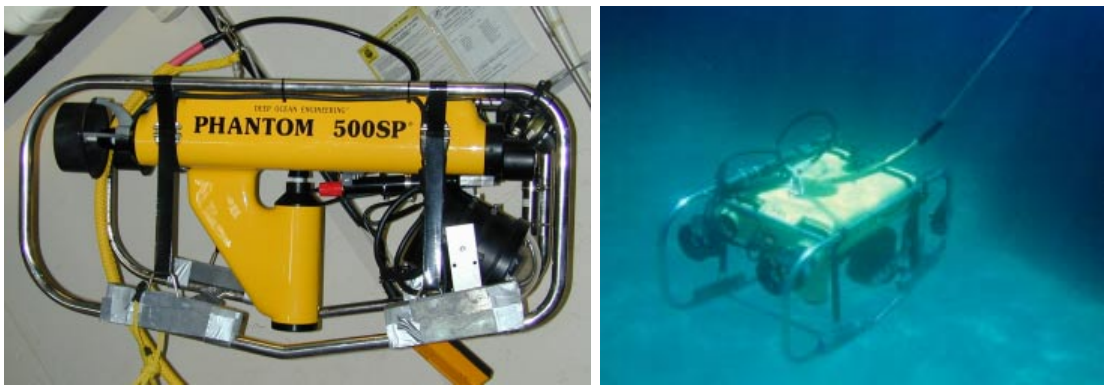


Figure A.1: Computer controlled Phantom ROV with the on–board camera. The camera housing is visible in the lower right, attached to the crash frame.

The vehicle is equipped with 2 horizontal propellers for surge and yaw motion. A

third propeller is mounted vertically near the center of mass, and provides heave motion. By construction, the platform is passively stable for roll and pitch motion, which are not actuated. Also, there is no lateral (sideways) actuation. The vehicle is therefore underactuated, which creates non-holonomic motion constraints.

The total weight of the vehicle (including fitted equipment and additional weights) is approximately 85 Kg in the air. In water, a set of weights was attached to the lower part of the frame to provide horizontal balance and slight positive buoyancy.

A set of standard sensors are fitted in the vehicle. These include an inertial navigation system with a rate gyro and accelerometer, flux–gate compass, a pressure depth sensor, a scanning sonar profiler (mounted on a controllable tilt head) and an acoustic altimeter. Custom sensors include a video camera and incremental encoders on the thrusters, for closed loop control of the propeller speed.

The ROV is linked to the surface by a flexible umbilical 120 meter cable, with neutral buoyancy. The cable provides electric power to the thrusters, an analog video link and a bi-directional serial communication link for the sensor triggering and data reception.

A large part of the experimental testing was conducted in the Mediterranean Sea, in France. The vehicle was deployed from a pier, and operated from shore (Figure A.2). For this range the water depth varied between 2 and 7 meters. The working area was fairly flat, with no abrupt depth changes. A large percentage of area was covered with non-static algae. This clearly departs from the general assumptions on static background. However, some sandy pits proved adequate for the map construction and navigation tests.

Figure A.2: View of the testing area in the Mediterranean sea.

A.2 Camera Calibration

The video camera is a Sony EVI equipped with a controllable pan–and–tilt head, zoom and focus. The image acquisition was performed in a Matrox Meteor II frame grabber.

Although not used in this thesis, the control of the pan and tilt head may be used to extend the visual field of the camera, thus compensating for the holonomic constraints of the platform when performing active visual tracking. The importance of this ability has been illustrated in visual station keeping [122] when tracking the same region of the sea floor during long intervals.

A special housing was constructed for the camera, comprising the spherical glass dome shown in Figure A.3. The camera is mounted so that the optical centre is approximately coincident with the dome centre. This configuration has the advantage of reducing the image distortion effect caused by refraction, since the optical rays are always cross the different medium interfaces perpendicularly to the interface surface.

Figure A.3: Close–up on the video camera housing with spherical dome.

The camera was calibrated underwater, in order to take into account the combined effects of lens distortions and dome refraction. The method of Heikkilä and Silvén [45] was used in the form of a publicly available MatLab toolbox. This method uses, as input, the coordinates of known scene 3D points and their point projections on several images. This method can be used with either planar or non-planar calibration grid. However, more accurate results are obtained with non-planar structure, given the same number of images. The method estimates the coefficients for both tangential and radial distortion, as well as the linear projective intrinsic parameters for a distortion–free equivalent camera. Two underwater images of a calibrations grid are shown in Figure A.4.

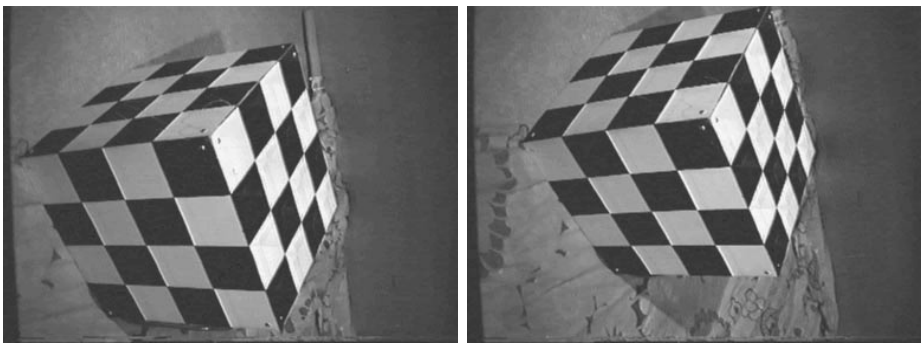


Figure A.4: Two underwater images of the grid used for camera calibration.

Appendix B

First Order Covariance Propagation

The method used for the uncertainty propagation follows that of Haralick in [38]. In that paper, a general method is presented for propagating the covariance matrix through any kind of linear or non-linear calculation. The method is devised for the case where the data and the parameters are *implicitly* related, through the minimization of a cost function. A similar derivation is given by Faugeras in [21].

Let $\widehat{X}$ be the $n \times 1$ *data vector* of noisy measurements, such that $\widehat{X} = X_0 + \Delta X$, where X_0 indicates the noise–free quantities and ΔX is random additive noise. Both X_0 and ΔX are unobservable vectors. Let $\widehat{\Theta}$ be a $k \times 1$ vector of *parameters* that are estimated from the calculation using the data $\widehat{X}$, such that $\widehat{\Theta} = \Theta_0 + \Delta\Theta$, where Θ_0 is the vector of ideal noise–free estimates and $\Delta\Theta$ is the associated random perturbation induced by ΔX. Similarly, both Θ_0 and $\Delta\Theta$ are unobservable.

For the case of constrained estimation of Θ, let $s(\Theta)$ be a $\mathbb{R}^k \rightarrow \mathbb{R}^m$ function describing the m constraints on Θ_0 such that $s(\Theta_0) = 0$, and Λ_0 be the vector of the Lagrange multipliers associated with the constraints.

Let $F(X, \Theta)$ be a scalar function, that implicitly relates the data and the parameters such that $F(X, \Theta) \geq 0$. For the unperturbed vectors, $F(X_0, \Theta_0) = 0$. Additionally the computed noisy estimates $\widehat{\Theta}$ are obtained by

$$\widehat{\Theta} = \arg\min_{\Theta} F\left(\widehat{X}, \Theta\right)$$

The method for covariance propagation assumes the following two conditions:

- The function $F(X, \Theta)$ has finite second partial derivatives.

- The random perturbations ΔX are small enough, so that $F(X_0, \Theta_0)$ and $F\left(\widehat{X}, \widehat{\Theta}\right)$ can be well related by a first order Taylor series expansion.

Let $g(X, \Theta)$ be the gradient of F with respect to Θ,

$$g(X, \Theta) = \frac{\partial F}{\partial \Theta}(X, \Theta) \quad \in I\!R^k$$

For the noise–free values of X_0 and Θ_0, the following relation holds

$$\frac{\partial}{\partial \Theta}\left[F(X_0, \Theta_0) + s(\Theta_0)^T \cdot \Lambda_0\right] = 0$$

Since $F(X_0, \Theta_0) = 0$, we have $g(X_0, \Theta_0) = 0$ and

$$\left[\frac{\partial s}{\partial \Theta}(\Theta_0)\right]^T \cdot \Lambda_0 = 0$$

which implies $\Lambda_0 = 0$, since $\frac{\partial s}{\partial \Theta}(\Theta_0)$ is expected to be full rank.

Let $S(X, \Theta, \Lambda)$ be defined as

$$S(X, \Theta, \Lambda) = \begin{bmatrix} g(X, \Theta) + \left[\frac{\partial s}{\partial \Theta}(\Theta)\right]^T \cdot \Lambda \\ s(\Theta) \end{bmatrix}$$

Writing the first–order Taylor series $S(X, \Theta, \Lambda)$ at $(X_0, \Theta_0, \Lambda_0)$, one gets

$$S(X_0 + \Delta X, \Theta_0 + \Delta\Theta, \Lambda_0 + \Delta\Lambda) - S(X_0, \Theta_0, \Lambda_0) \simeq \left[\frac{\partial S}{\partial X}\right]^T \Delta X + \left[\frac{\partial S}{\partial \Theta}\right]^T \Delta\Theta + \left[\frac{\partial S}{\partial \Lambda}\right]^T \Delta\Lambda \tag{B.1}$$

where the derivatives of S are evaluated at $(X_0, \Theta_0, \Lambda_0)$. Both terms on the left hand side of Eq.(B.1) are equal to zero, which leads to

$$-\left[\frac{\partial S}{\partial X}\right]^T \Delta X \simeq \left[\frac{\partial S}{\partial \Theta}\right]^T \Delta\Theta + \left[\frac{\partial S}{\partial \Lambda}\right]^T \Delta\Lambda$$

Writing the above equation in terms of $g(X, \Theta)$ and $s(\Theta)$, the following approximated equality is obtained

$$A \cdot \begin{bmatrix} \Delta\Theta \\ \Delta\Lambda \end{bmatrix} \simeq B \cdot \Delta X$$

where

$$A = \begin{bmatrix} \frac{\partial g}{\partial \Theta} & \left(\frac{\partial s}{\partial \Theta}\right)^T \\ \frac{\partial s}{\partial \Theta} & \mathbf{0} \end{bmatrix} \text{ and } B = \begin{bmatrix} -\left(\frac{\partial g}{\partial X}\right)^T \\ \mathbf{0} \end{bmatrix} \tag{B.2}$$

Given the covariance matrix $\Sigma_{\Delta X}$ of the additive noise in X, the first order approximation to the covariance of the perturbations in Θ is given by

$$\Sigma_{\Delta\theta,\Delta\Lambda} \simeq A^{-1} \cdot B \cdot \Sigma_{\Delta X} \cdot B^T \cdot A^{-1} \tag{B.3}$$

where A and B are evaluated at (X, Θ). An estimator of the covariance $\widehat{\Sigma}_{\Delta\theta,\Delta\Lambda}$ of the noisy parameters is obtained by evaluating Eq.(B.3) A and B at $\left(\widehat{X}, \widehat{\Theta}\right)$.

For the case of unconstrained estimation, the above derivation simplifies to

$$A = \frac{\partial g}{\partial \Theta} \text{ and } B = -\left(\frac{\partial g}{\partial X}\right)^T$$

An estimator for the covariance $\widehat{\Sigma}_{\Delta\Theta}$ of the noise in $\widehat{\Theta}$, is given by

$$\widehat{\Sigma}_{\Delta\Theta} = \left(\frac{\partial g}{\partial \Theta}\right)^{-1} \cdot \left(\frac{\partial g}{\partial X}\right)^T \cdot \Sigma_{\Delta X} \cdot \frac{\partial g}{\partial X} \cdot \left(\frac{\partial g}{\partial \Theta}\right)^{-1} \tag{B.4}$$

Bibliography

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, 1989.

[2] A. Arsénio and J. Marques. Performance analysis and characterization of matching algorithms. In *Proc. of the International Symposium on Intelligent Robotic Systems*, Stockholm, Sweden, July 1997.

[3] S. Ayer. *Sequential and Competitive Methods for Estimation of Multiple Motions*. PhD thesis, École Polytechnique Fédérale de Lausanne, 1995.

[4] R. Ballard, A. McCann, D. Yoerger, L. Whitcomb, D. Mindell, J. Oleson, H. Singh, B. Foley, J. Adams, and D. Piechota. The discovery of ancient history in the deep sea using advanced deep submergence technology. *Journal of Deep Sea Research I*, 41:1591–1620, 2000.

[5] J.K. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–78, 1994.

[6] M. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. of the 4th. International Conference on Computer Vision*, Berlin, Germany, May 1993.

[7] Lisa Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4):325–376, 1992.

[8] D. P. Capel. *Image Mosaicing and Super-resolution*. PhD thesis, University of Oxford, Oxford, UK, 2001.

[9] T. Coleman, M. Branch, and A. Grace. *Optimization toolbox for use with Matlab*. The MathWorks, Inc., Natick, MA, USA, 1999.

[10] T. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. Technical Report TR93–1342, Cornell University Library, April 1993.

[11] S. Coorg and S. Teller. Spherical mosaics with quaternions and dense correlation. *International Journal of Computer Vision*, 37(3):259–273, 2000.

[12] John Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley, 1989.

[13] J. Davis. Mosaics of scenes with moving objects. In *Proc. of the Conference on Computer Vision and Pattern Recognition*, Santa Barbara, CA, USA, June 1998.

[14] L. de Agapito, R. Hartley, and E. Hayman. Linear self–calibration of a rotating and zooming camera. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 15 –21, Fort Collins, Colorado, USA, June 1999.

[15] F. Dellaert, S. Thrun, and C. Thorpe. Mosaicing a large number of widely dispersed, noisy, and distorted images: A Bayesian approach. Technical Report CMU-RI-TR-99-34, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, March 1999.

[16] K. Duffin and W. Barrett. Globally optimal image mosaics. In *Graphics Interface*, pages 217–222, 1998.

[17] Deep Ocean Engineering. Phantom ROV test bed for NARVAL project. Press Release available at http://www.deepocean.com, August 1999.

[18] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.

[19] R. Eustice, H. Singh, and J. Howland. Image registration underwater for fluid flow measurements and photomosaicking. In *Proc. of the Oceans 2000 Conference*, Providence, Rhode Island, USA, September 2000.

[20] H. Everett. *Sensors for Mobile Robots: Theory and Application*. A K Peters, Ltd, Wellesley, MA, 1995.

[21] O. Faugeras. *Three Dimensional Computer Vision*. MIT Press, 1993.

[22] O. Faugeras, P. Fua, B. Hotz, R. Ma, L. Roberts, M. Thonnat, and Z. Zhang. Quantitative and qualitative comparision of some area and feature-based stereo algorithms. In W. Förstner and Ruwiedel, editors, *Robust Computer Vision*. Wichmann, Bonn, Germany, March 1992.

[23] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3):485–508, September 1988.

[24] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. Technical Report 856, INRIA, June 1988.

[25] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 6(24):381–395, 1981.

[26] Stephen Fleischer. *Bounded–Error Vision–Based Navigation of Autonomous Underwater Vehicles*. PhD thesis, Stanford University, California, USA, May 2000.

[27] J. P. Folcher and M. J. Rendas. Identification and control of the Phantom 500 body motion. In *Proc. of the Oceans 2001 Conference*, pages 529–535, Honolulu, Hawaii, U.S.A., November 2001.

[28] A. Fusiello, E. Trucco, T. Tommasini, and V. Roberto. Improving feature tracking with robust statistics. *Pattern Analysis & Applications*, 2:312–320, 1999.

[29] R. Garcia, J. Batlle, X. Cufí, and J. Amat. Positioning an underwater vehicle through image mosaicking. In *Proc. International Conference on Robotics and Automation (ICRA2001)*, pages 2779–2784, Seoul, Korea, May 2001.

[30] R. Garcia, X. Cufí, and J. Batlle. Detection of matchings in a sequence of underwater images through texture analysis. In *Proc. of the IEEE International Conference on Image Processing (ICIP)*, pages 361–364, Thessaloniki, Greece, 2001.

[31] R. Garcia, J. Puig, P. Ridao, and X. Cufi. Augmented state Kalman filtering for AUV navigation. In *Proc. International Conference on Robotics and Automation (ICRA2002)*, pages 4010–4015, Washington DC, USA, May 2002.

[32] N. Gracias and J. Santos-Victor. Automatic mosaic creation of the ocean floor. In *Proc. of the IEEE Oceans 98 Conference*, Nice, France, September 1998.

[33] N. Gracias and J. Santos-Victor. Trajectory reconstruction with uncertainty estimation using mosaic registration. *Robotics and Autonomous Systems*, 35:163–177, July 2001.

[34] N. Gracias and J. Santos-Victor. Underwater mosaicing and trajectory reconstruction using global alignment. In *Proc. of the Oceans 2001 Conference*, pages 2557–2563, Honolulu, Hawaii, U.S.A., November 2001.

[35] N. Gracias, S. Zwaan, A. Bernardino, and J. Santos-Victor. Results on Underwater Mosaic–based Navigation. In *Proc. of the Oceans 2002 Conference*, Biloxi, Mississippi, U.S.A., October 2002.

[36] N. Gracias, S. Zwaan, A. Bernardino, and J. Santos-Victor. Mosaic based Navigation for Autonomous Underwater Vehicles. *Accepted for publication in Journal of Oceanic Engineering*, 2003.

[37] Nuno Gracias. Application of robust estimation to computer vision: Video mosaics and 3–D reconstruction. Master's thesis, Instituto Superior Técnico, Lisbon, Portugal, April 1998.

[38] R. Haralick. Propagating covariance in computer vision. In *Proc. of the Workshop on Performance Characteristics of Vision Algorithms*, Cambridge, UK, April 1996.

[39] C. Harris. Determination of ego-motion from matched points. In *Proceedings Alvey Conference*, Cambridge, UK, 1987.

[40] R. Hartley. Self–calibration from multiple views with a rotating camera. In *Proc. of the 3rd. European Conference on Computer Vision*, volume I, pages 471–478, Stockholm, Sweden, May 1994. Springer–Verlag.

[41] R. Hartley. Self–calibration from stationary cameras. *International Journal of Computer Vision*, 22(1):5–23, February/March 1997.

[42] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision.* Cambridge University Press, 2000.

[43] R. Haywood. Acquisition of a micro scale photographic survey using an autonomous submersible. In *Proc. of the Oceans 86 Conference*, New York NY, USA, 1986.

[44] A. Healey, E. An, and D. Marco. On line compensation of heading sensor bias for low cost AUVs. In *Proc. of the 1998 Workshop on Autonomous Underwater Vehicles*, Cambridge, Massachusetts, USA, August 1998.

[45] J. Heikkilä and Olli Silvén. A four–step camera calibration procedure with implicit image correction. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Puerto Rico, June 1997.

[46] B. Horn. *Robot vision.* MIT Press, 1986.

[47] B. Horn and B. Shunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[48] R. Horn and C. Johnson. *Matrix Analysis.* Cambridge University Press, 1985.

[49] A. Huster, S. Fleischer, and S. Rock. Demonstration of a vision–based dead–reckoning system for navigation of an underwater vehicle. In *Proc. of the IEEE Oceans 98 Conference*, Nice, France, September 1998.

[50] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651–670, October 1996.

[51] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proc. of the 5th IEEE International Conference on Computer Vision*, Cambridge, Massachusetts, June 1995.

[52] A. Jain, editor. *Fundamentals Digital Image Processing.* Prentice Hall, 1989.

[53] A. Jepson and M. Black. Mixture models for optical flow computation. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 760–761, New York, May 1993.

[54] M. Kalos and P. Whitlock. *Monte Carlo Methods, I: Basics.* John Wiley & Sons, 1986.

[55] K. Kanatani, editor. *Statistical optimization for geometric computation: Theory and practice.* Elsevier Science, 1996.

[56] E. Kang, I. Cohen, and G. Medioni. A graph–based global registration for 2D mosaics. In *Proc. of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000.

[57] S. Kang and R. Szeliski. 3–D scene data recovery using omnidirecional mulibaseline stereo. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 364–370, 1996.

[58] A. Kelly. Mobile robot localization from large scale appearance mosaics. *International Journal of Robotics Research (IJRR)*, 19(11), 2000.

[59] S. Kruger and A. Calway. Image registration using multiresolution frequency domain correlation. In *British Machine Vision Conference*, pages 316–325, 1998.

[60] C. Kuglin and D. Hines. The phase correlation image alignment method. In *Proc. of the IEEE IEEE Conference on Cybernetics and Society*, pages 163–165, September 1975.

[61] M. Larsen. High performance Doppler–inertial navigation – Experimental results. In *Proc. of the Oceans 2000 Conference*, Providence, Rhode Island, USA, September 2000.

[62] H.C. Longuet-Higgins. The reconstruction of a plane surface from two perspective projections. *Proc. of the Royal Society, London*, 227:399–410, 1986.

[63] J. Maciel. *Global matching: Optimal solution to correspondence problems.* PhD thesis, Universidade Técnica de Lisboa, Lisbon, Portugal, August 2001.

[64] E. Malis and F. Chaumette. 2 1/2 D Visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, June 2000.

[65] S. Mann and R. W. Picard. Video orbits of the projective group: A new perspective on image mosaicing. Technical Report TR-339, MIT - Media Lab., 1995.

[66] S. Marapane and M. Trivedi. Multi–primitive hierarchical (MPH) stereo analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):227–240, March 1994.

[67] R. Marks, S. Rock, and M. Lee. Using visual sensing for control of an underwater robotic vehicle. In *Proc. of IARP Second Workshop on Mobile Robots for Subsea Environments*, Monterey, USA, May 1994.

[68] R. Marks, S. Rock, and M. Lee. Real-Time video mosaicking of the ocean floor. *IEEE Journal of Oceanic Engineering*, 20(3):229–241, July 1995.

[69] M. Massey and W. Bender. Salient stills: Process and practice. *IBM Systems Journal*, 35(3 and 4):557–573, 1996.

[70] P. McLauchlan and A. Jaenicke. Image mosaicing using sequential bundle adjustment. In *Proc. of the British Machine Vision Conference BMVC2000*, Bristol, U.K., September 2000.

[71] P. Meer, D. Mintz, A. Rosenfeld, and D. Kim. Robust regression methods for computer vision: a review. *Int. Journal of Computer Vision*, 6(1):59–70, 1991.

[72] A. Meijster, J. Roerdink, and W. Hesselink. A general algorithm for computing distance transforms in linear time. In *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 331–340. Kluwer, 2000.

[73] NARVAL – Navigation of Autonomous Robots via Active Environmental Perception, Esprit–LTR Project 30185, 1998–2002.

[74] S. Negahdaripour. Closed–form relationship between the two interpretations of a moving plane. *Journal of the Optical Society of America A*, 7(2), February 1990.

[75] S. Negahdaripour. Revised definition of optical flow: Integration of radiometric and geometric cues for dynamic scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(9):961–970, September 1998.

[76] S. Negahdaripour and P. Firoozfam. Positioning and image mosaicing of long image sequences; Comparison of selected methods. In *Proc. of the IEEE Oceans 2001 Conference*, Honolulu, Hawai, USA, November 2001.

[77] S. Negahdaripour and B. Horn. Direct passive navigation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(1), January 1987.

[78] S. Negahdaripour and X. Xu. Mosaic–based positioning and improved motion–estimation methods for automatic navigation of submersible vehicles. *IEEE Journal of Oceanic Engineering*, 27(1):79–99, January 2002.

[79] S. Negahdaripour, X. Xu, and A. Khamene. Applications of direct 3D motion estimation for underwater machine vision systems. In *Proc. of the IEEE Oceans 98 Conference*, Nice, France, September 1998.

[80] S. Negahdaripour and C. Yu. Passive optical sensing for near–bottom station–keeping. In *Proc. of the IEEE Oceans 90 Conference*, Washington DC, USA, September 1990.

[81] S. Negahdaripour and C. Yu. On shape and range recovery from image shading for underwater applications. In J. Yuh, editor, *Underwater Robotic Vehicles – Design and control*, pages 221–250. TSI Press, 1995.

[82] G. Nemhauser and L. Wolsey, editors. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[83] F. Odone, A. Fusiello, and E. Trucco. Layered representation of a video shot with mosaicing. *Pattern Analysis and Applications*, 5(3):296–305, August 2002.

[84] A. Ortiz, M. Simo, and G. Oliver. A vision system for an underwater cable tracker. *International Journal of Machine Vision and Applications*, 13(3):129–140, 2001.

[85] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1144–1154, October 2000.

[86] C. Plakas and E. Trucco. Developing a real-time robust video tracker. In *Proc. of the IEEE Oceans 2002 Conference*, Providence, Rhode Island, USA, September 2000.

[87] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1988.

[88] B. Reddy and B. Chatterji. An FFT–based technique for translation, rotation, and scale–invariant image registration. *IEEE Transactions on Image Processing*, 5(8):1266–1271, 1996.

[89] S. Rolfes, M.J. Rendas, and J. Side. Using autonomous underwater vehicles for seabed habitat mapping. In *Proc. of the International Council for the Exploration of the Sea*, September 2000.

[90] C. Roman and H. Singh. Estimation of error in large area underwater photomosaics using vehicle navigation data. In *Proc. of the Oceans 2001 Conference*, pages 1849–1853, Honolulu, Hawaii, U.S.A., November 2001.

[91] P. Rousseeuw and A. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, 1987.

[92] J. Ruanaidh and T. Pun. Rotation, scale and translation invariant digital image watermarking. In *Proc. IEEE International Conference on Image Processing*, volume 1, pages 536–539, Santa Barbara, CA, USA, October 1997.

[93] Y. Rzhanov, L. Linnett, and R. Forbes. Underwater video mosaicing for seabed mapping. In *Proc. IEEE International Conference on Image Processing*, Vancouver, Canada, September 2000.

[94] J. Santos-Victor. *Visual Perception for Mobile Robots: From Percepts to Behaviours*. PhD thesis, Universidade Técnica de Lisboa, Lisbon, Portugal, November 1994.

[95] J. Santos-Victor and J. Sentieiro. Image matching for underwater 3D vision. In *International Conference on Image Processing: Theory and Applications*, San Remo, Italy, June 1993.

[96] H. Sawhney, S. Ayer, and M. Gorkani. Model–based 2D & 3D dominant motion estimation for mosaicing and video representation. In *Proc. of the 5th IEEE International Conference on Computer Vision*, Cambridge, Massachusetts, June 1995. IEEE Computer Society Press.

[97] H. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local to global alignment. In *Proc. European Conference on Computer Vision*. Springer-Verlag, June 1998.

[98] H. Sawhney and R. Kumar. True multi–image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(3):235–243, March 1999.

[99] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two–frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1):7–42, May 2002.

[100] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2):151–172, 2000.

[101] J. Shi and C. Tomasi. Good features to track. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, Washington, USA, June 1994.

[102] H. Shum and R. Szeliski. Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *International Journal of Computer Vision*, 36(2):101–130, 2000.

[103] C. Silva and J. Santos-Victor. Direct egomotion estimation. In *Proc. of the 13th Int. Conference on Pattern Recognition*, Vienna, Austria, August 1996.

[104] D. Sinclair, A. Blake, and D. Murray. Robust estimation of egomotion from normal flow. *International Journal of Computer Vision*, 13(1):57–70, September 1994.

[105] H. Singh, J. Howland, L. Whitcomb, and D. Yoerger. Quantitative mosaicking of underwater imagery. In *Proc. of the IEEE Oceans 98 Conference*, Nice, France, September 1998.

[106] P. Sturm. Algorithms for plane–based pose estimation. In *Proc. of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 706–711, South Carolina, USA, June 2000.

[107] R. Szeliski. Image mosaicing for tele–reality applications. Technical Report CRL 94/2, Digital Equipment Corporation, May 1994.

[108] A. Tenas, M. J. Rendas, and J. P. Folcher. Image segmentation by unsupervised adaptive clustering in the distribution space for AUV guidance along sea-bed boundaries using vision. In *Proc. of the Oceans 2001 Conference*, pages 538–544, Honolulu, Hawaii, U.S.A., November 2001.

[109] L. Teodosio and W. Bender. Salient video stills: Content and context preserved. In *Proc. of the ACM Multimedia Conference*, Anaheim, August 1993.

[110] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. MINERVA: A second generation mobile tour–guide robot. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, Michigan, U.S.A., May 1999.

[111] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour–guide robot minerva. *International Journal of Robotics Research*, 19(11):972–999, 2000.

[112] S. Thrun, W. Burgard, and D. Fox. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1-3):29–53, 1998.

[113] P. Torr. Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. *International Journal of Computer Vision*, 50(1):35–62, October 2002.

[114] P. Torr and D. Murray. The development and comparision of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, September/October 1997.

[115] B. Triggs. Autocalibration from planar scenes. In *Proc. of the European Conference on Computer Vision*, pages 89–105, Freiburg, Germany, June 1998.

[116] E. Trucco and A. Verri. *Introductory Techniques for 3–D Computer Vision*. Prentice–Hall, 1998.

[117] R. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

[118] R. Tsai and T. Huang. Estimating three–dimensional motion parameters of a rigid planar patch, II: Singular value decomposition. *IEEE Transaction on Acoustics, Speech, and Signal Processing*, 30(4):525–534, August 1982.

[119] R. Unnikrishnan and A. Kelly. A constrained optimization approach to globally consistent mapping. In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, Lausanne, Switzerland, September 2002.

[120] R. Unnikrishnan and A. Kelly. Mosaicing large cyclic environments for visual navigation in autonomous vehicles. In *Proc. International Conference on Robotics and Automation (ICRA2002)*, pages 4299–4306, Washington DC, USA, May 2002.

[121] S. van der Zwaan. Vision based station keeping and docking for floating robots. Master's thesis, Instituto Superior Técnico, Lisbon, Portugal, May 2001.

[122] S. van der Zwaan, A. Bernardino, and J. Santos-Victor. Visual station keeping for floating robots in unstructured environments. *Robotics and Autonomous Systems*, 39(3–4):145–155, June 2002.

[123] J. Weng, T. Huang, and N. Ahuja. Motion and structure from line correspondences: Closed form solution, uniqueness and optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(3):318–336, March 1992.

[124] L. Whitcomb, D. Yoerger, H. Singh, and D. Mindell. Towards precision robotic maneuvering, survey, and manipulation in unstructured undersea environments. In *Robotics Research – The Eighth International Symposium*, pages 45–54. Springer–Verlag, 1998.

[125] X. Xu. *Vision–based ROV System.* PhD thesis, University of Miami, Coral Gables, Miami, May 2000.

[126] Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. Technical Report 2927, INRIA, Sophia–Antipolis, France, July 1996.

[127] J. Zheng and S. Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–76, October 1992.

[128] Q. Zheng and R. Chellappa. A computational vision approach to image registration. *IEEE Transactions on Imaging Processing*, 2(3):311–326, July 1993.