3D Reconstruction and Motion Estimation Using Forward Looking Sonar

Hassan Assalih

Thesis Submitted
for the
Degree of Doctor of Philosophy

Heriot-Watt University

School of Engineering and Physical Sciences

July 2013

Abstract

Autonomous Underwater Vehicles (AUVs) are increasingly used in different domains including archaeology, oil and gas industry, coral reef monitoring, harbour's security, and mine countermeasure missions. As electromagnetic signals do not penetrate underwater environment, GPS signals cannot be used for AUV navigation, and optical cameras have very short range underwater which limits their use in most underwater environments.

Motion estimation for AUVs is a critical requirement for successful vehicle recovery and meaningful data collection. Classical inertial sensors, usually used for AUV motion estimation, suffer from large drift error. On the other hand, accurate inertial sensors are very expensive which limits their deployment to costly AUVs. Furthermore, acoustic positioning systems (APS) used for AUV navigation require costly installation and calibration. Moreover, they have poor performance in terms of the inferred resolution.

Underwater 3D imaging is another challenge in AUV industry as 3D information is increasingly demanded to accomplish different AUV missions. Different systems have been proposed for underwater 3D imaging, such as planar-array sonar and T-configured 3D sonar. While the former features good resolution in general, it is very expensive and requires huge computational power, the later is cheaper implementation but requires long time for full 3D scan even in short ranges.

In this thesis, we aim to tackle AUV motion estimation and underwater 3D imaging by proposing relatively affordable methodologies and study different parameters affecting their performance. We introduce a new motion estimation framework for AUVs which relies on the successive acoustic images to infer AUV ego-motion. Also, we propose an Acoustic Stereo Imaging (ASI) system for underwater 3D reconstruction based on forward looking sonars; the proposed system features cheaper implementation than planar array sonars and solves the delay problem in T configured 3D sonars.

Dedication

To the soul of my father M. Sameer Assalih, who encouraged me and inspired me from the first step I took in this life. To my mother Aida Labania who gave me the confidence and continuous support. To my brothers and sisters: Lubna, Hazem, Batoul and Imam. To my small family and big love Nadin and Majd.

To the land of freedom, the land of purity... To Syria ...

Acknowledgment

I would like to express my sincere gratitude to my supervisor Prof Yvan Petillot for his continuous support and cordial encouragement, I really cannot find words to acknowledge the support he offered when most needed.

I wish also to thank Prof Judith Bell who inspired me and introduced me to this interesting field.

Also, I would like to thank my friend Jamil for great discussions over the course of my study, thanks also to Monkiz, Ali, Ghazi, Manhal and Amin for being great friends.

Finally, I would like to express my deepest appreciation to Prof Shahriar Negahdaripour for his great collaboration and friendly hosting in my research visit to Miami-FL.

ACADEMIC REGISTRY

Research Thesis Submission

Name*:*			
School/PGI:			
Version: *(i.e. First, Resubmission, Final)*		Degree Sought (Award **and** Subject area)	

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted*.
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
E-thesis Submitted (***mandatory for final theses)***			
Signature:		Date:	

Contents

List of Figures

Acronyms

ADCP	Acoustic Doppler Current Profiler
APS	Acoustic Positioning System
ASI	Acoustic Stereo Imaging
AUV	Autonomous Underwater Vehicle
BSL	Baseline (Acoustic Positioning System)
DIDSON	Dual-Frequency Identification Sonar
DOA	Direction Of Arrival
DOF	Degrees Of Freedom
DOG	Difference-Of-Gaussian
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
FAU	Florida Atlantic University
FOG	Fibre Optic Gyroscopes
FOV	Field Of View
FOVA	Field Of View Area (the common area between two beams in the ASI system)
FOVV	Field Of View Volume (the common volume between two sonars in the ASI system)
FS	Forward Scan [Sonar]
GPS	Global Positioning System
INS	Inertial Navigation System
KF	Kalman Filter
LBL	Long Baseline
MBES	MultiBeam EchoSounder
MDUD	Modified Discrete Uniform Distribution
MEMS	micro-electro-mechanical
MI	Mutual Information
ML	Maximum Likelihood
MC	Monte Carlo simulation
MRF	Markov Random Fields

MRLG	Monolithic Ring Laser Gyro
POI	Point Of Interest
RANSAC	Random Sample and Consensus
RLG	Ring Laser Gyroscopes
ROV	Remotely Operated Vehicles
SA	Simulated Annealing
SCS	Sorting the Correspondence Space
SIFT	Scale Invariant Feature Transform
SIR	Sampling Importance Resampling
SNR	Signal-to-Noise Ratio
SBL	Short Baseline
SSBL	Super Short Baseline
SVD	Singular Value Decomposition
SURF	Speeded Up Robust Features
TBN	Terrain Based Navigation
TOF	Time Of Flight
TVG	Time Varying Gain
UAVs	Unmanned Air Vehicles
USBL	Ultra-Short Baseline (Acoustic Positioning System)

Chapter 1
Introduction

1.1 Motivation and challenges

Sea turtles usually migrate long distances across entire oceanic basins for mating, nesting, and food; when displaced from their area, they often home to the site of capture. In fact, sea turtles set an example for true autonomy while performing their activities underwater; they show astonishing navigational capabilities by travelling long journeys (~8000km) depending on their own “navigational senses” [1]. Many research centres and commercial companies are striving to achieve similar autonomy in Autonomous Underwater Vehicles (AUVs) that are designed to perform different underwater missions such as coral reef monitoring, ship hull inspection, harbour patrolling, mine counter-measures, pipeline survey, dam inspection, and seabed mapping.

Advances in battery capacity and the advent of hydrogen fuel cells enabled AUVs to accomplish long missions over an extended period of time [2][3][4]. These developments demanded accurate navigational sensors and algorithms to serve long AUV missions. An AUV navigational system is an indispensable component for successful vehicle recovery. In addition, it is a critical factor for precise data collection over the path of the mission. As GPS signals are not available underwater, motion estimation for AUVs is an active research topic, many methods and sensors have been proposed to solve the problem such as inertial sensors and acoustic positioning systems (APS).

Classical inertial sensors, usually used for AUV motion estimation, estimate the vehicle ego-motion by using accelerometers and gyroscopes to measure linear acceleration and angular velocity of the vehicle respectively; these measured parameters are used in dead reckoning to estimate the current vehicle position; however, they usually suffer from large drift error over extended AUV missions. On the other hand, accurate inertial sensors are expensive which limit their deployment to costly AUVs.

Acoustic positioning systems (APS), such as super short base line (SSBL) systems, are used to infer the AUV position by using acoustic signals between the vehicle and a supporting platform (mother ship, buoy .. etc). For example, in the SSBL system, the mother ship is equipped with an acoustic transmitter and a receiver array, and the AUV is equipped with an acoustic transponder. The mother ship calculates the position of the AUV by measuring the time of flight (TOF) and direction of arrival (DOA) of the acoustic signal returned from the AUV. As the AUV's position is measured on the ship, the calculated position of the AUV incurs a considerable delay in the measurement. In short, acoustic positioning systems (APS) require costly installation and calibration. Moreover, they have poor performance in terms of the inferred resolution. Above all, using APS methodologies raises the question about the autonomy of the supported AUV.

Underwater 3D imaging is another challenge in the AUV industry as 3D information is increasingly demanded to accomplish many AUV missions including dam and bridge inspection, bathymetric mapping, search and recovery, subsea metrology, and iceberg monitoring. Underwater 3D imaging can be defined as the whole process of visualizing the surrounding environment in 3D using optical and/or acoustic devices. In this sense, underwater 3D reconstruction is the first stage of underwater 3D imaging which includes subsequent visualization and modelling stages. Different systems have been proposed for underwater 3D imaging, such as planar-array sonars and T-configured 3D sonars. While the former features fast 3D scan, it is expensive and requires huge computational power, the later is a cheaper implementation but requires a long time for full 3D scan even for short ranges.

In this thesis, we focus on the aforementioned two challenges: 1) Underwater 3D reconstruction and 2) motion estimation for AUV. In both domains, we depend on two sonar views of the surrounding environment to infer the required information; in 3D reconstruction, we use two sonar views from two sonars with known geometric configuration between them to infer 3D coordinates of the insonified objects. In a similar manner, for motion estimation, we use consecutive views from one sonar to infer the geometric relation between these views. In motion estimation for AUVs, we only have one sonar insonifying the environment in two different positions representing a virtual stereo system. Using sonar views for motion estimation enhances the vehicle autonomy as it uses its "own sensors" with no support from other platforms; this procedure reduces the cost of the AUV operation compared with other systems requiring a supporting platform (ship, buoys … etc).

Both domains i.e. 3D reconstruction and motion estimation in AUVs answer the same question "*where am I*" in two different ways (see Figure 1.1); 3D reconstruction gives 3D coordinates of all landmarks sensed by the AUV while motion estimation gives 3D motion vector relative to another position in the AUV path.

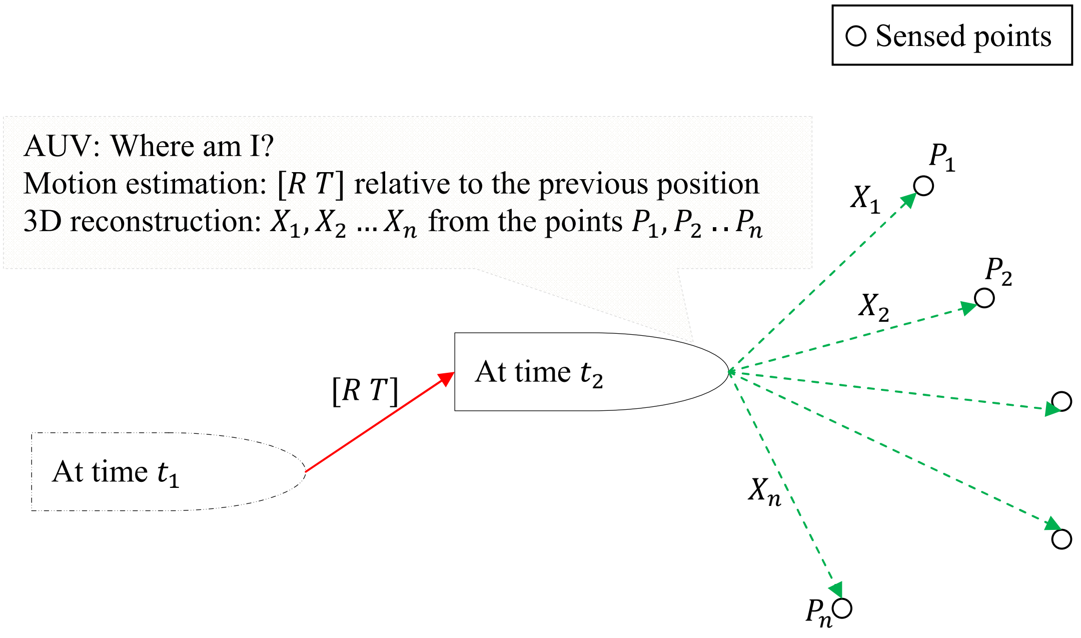

○ Sensed points
AUV: Where am I?
Motion estimation: $[R\ T]$ relative to the previous position
3D reconstruction: $X_1, X_2 \ldots X_n$ from the points $P_1, P_2 \ldots P_n$
P_1
X_1
P_2
X_2
$[R\ T]$
At time t_2
At time t_1
X_n
P_n

Figure 1.1: Motion estimation and 3D reconstruction answers for "*where am I*" question

The remaining part of this Chapter will develop the points introduced in this section. First, section 1.2 describes forward looking sonars which we use for 3D reconstruction and motion estimation. Section 1.3 introduces the proposed acoustic stereo system for underwater 3D reconstruction. Section 1.4 presents our methodology for motion estimation using acoustic images. Section 1.5 discusses image registration as a prerequisite to solve motion estimation and 3D reconstruction. Section 1.6 outlines our main contributions. Finally, section 1.7 explains the structure of the thesis and demonstrates its plan.

1.2 Forward looking sonar

AUVs are commonly equipped with forward looking sonars as a tool for obstacle avoidance, object identification and inspection. Forward Looking Sonar acts as a gap filler in front of the AUV which is usually equipped with side scan sonar. The Dual-Frequency Identification Sonar (DIDSON) is a good example of forward looking sonars; it can easily replace optical systems in turbid waters as it features high resolution and rapid refresh rate. The acoustic lenses in DIDSON, which are made of polymethylpentene plastic, form a set of divergent beams for transmitting/receiving the acoustic signals. As the middle lens moves between the other two lenses, the focus changes from 0.42m to its maximum range. DIDSON, as the name suggests, operates at two frequencies, 1.8 MHz or 1.1 MHz, and forms 96 beams spaced 0.3° apart, or 48 beams spaced 0.6° apart respectively [5][6]. It is relatively compact which makes it suitable for AUVs. Also, it consumes 25 Watts which is important to AUVs with a power budget. Figure 1.2 illustrates DIDSON 300 and its rectangular lenses; Appendix A lists DIDSON's technical specifications.

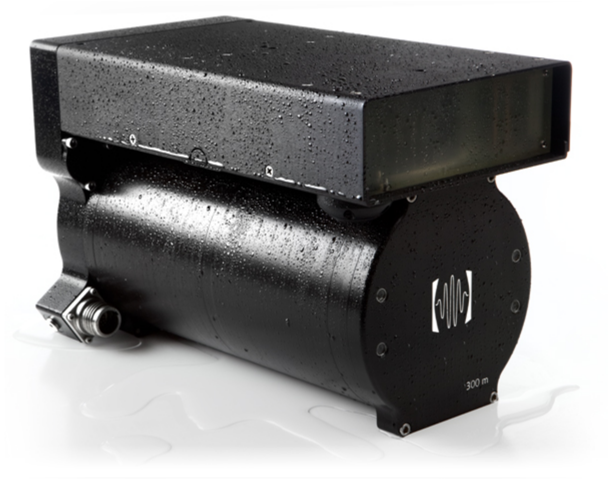


Figure 1.2: DIDSON 300 sonar

In addition to DIDSON, we have used BlueView P900 for motion estimation and 3D reconstruction. BlueView P900-130 is a multibeam sonar with wide field of view (130°) as it emits 768 beams with 0.18° beam spacing, it operates at 900 KHz frequency and features low power consumption and lightweight head which makes it perfect for AUVs [7]. Figure 1.3 illustrates BlueView P900-130 sonar head. Appendix B lists its technical specifications.

Figure 1.3: BlueView 900-130 sonar head

1.3 3D Reconstruction using Acoustic Stereo Imaging (ASI)

In this thesis, the design and assessment of a real time underwater 3D reconstruction system depending on multiple sonar views are presented. The new system which is dubbed Acoustic Stereo Imaging (ASI) consists of two relatively inexpensive acoustic sensors situated in specific geometric configurations. The acoustic sensors are multi-beam, high resolution sonars. The ASI vertical configuration suggests two sonars situated above one another and assumes that all beams of the first sonar are aligned with the corresponding beams in the second sonar. We have derived the ASI geometry which alleviates the well known correspondence problem in all stereo systems.

ASI geometry turns searching for correspondences between two sonar images into very small regions, thus each point in the first image can be located in a well defined region in the second image. Furthermore, the ASI geometry increases the chance to use image registration techniques in real time, since they will be applied in small regions. We have simulated the ASI system to study different parameters affecting the system performance; these parameters represent fundamental configurations like the distance between the two sonars and the angle between them.

1.4 Motion estimation using acoustic imaging

2-D Imaging systems, e.g., optical or acoustic, encode rich visual cues about the geometry of the world that is imaged and the position of the sensor relative to the world. In motion vision applications dealing with robotics platforms, one of the objectives is to determine the trajectory of a mobile system from the variations in the 2-D scene imagery. While this problem has been addressed extensively for numerous terrestrial applications [8][9][10] and to some extend in underwater by optical imaging [11] [12], less than a handful of earlier studies have explored application to 2-D sonar imaging systems [13][14].

In this thesis, a new framework for the analysis of 2-D sonar video image is presented, comprising the principles of Modified Discrete Uniform Distribution (MDUD), sonar projection function (SPF) and sonar arc sampling. These concepts are applied to address the problem of motion estimation using successive acoustic images. This will facilitate various routine tasks of ROVs/AUVs, including the inspection of various subsea structures, autonomous navigation, and target localization and classification.

Furthermore, we examine an earlier motion estimation framework [14] which employs an iterative method based on a maximum likelihood (ML) paradigm to compute 3-D motion from a pair of 2-D sonar views. We identify certain factors that affect the accurate estimation of 3-D motion. These primary factors include the inherent low signal-to-noise ratio (SNR) of 2-D sonar imagery, limited number of features that can be detected and matched in a motion sequence, and the horizontal field of view (FOV) of the acoustic cameras. We study the significance of each factor based on the variances of the estimated motion parameters. We analyze the results from a large number of numerical simulations with random data, and the typical parameters of existing 2-D sonar imaging systems. Our results offer quantitative measures, collectively in agreement with earlier observations and conjecture on the role of each parameter. In addition, we also present various results from an experiment with a real data set, where we apply different measures to establish ground truth for the quantitative assessment of the motion estimation accuracy. The results highlight some advantages offered in the estimation of 3-D motion by sonar imaging.

1.5 Acoustic image registration as a prerequisite

Image registration is an indispensable component in computer vision; it represents the core problem in many tasks like motion estimation, 3D reconstruction, medical image analysis, mosaicing and stereo-matching, to name a few. The problem can be easily described as the task of assigning correspondences for a set of points in the first image to their counterparts in the second image. In other words, the problem is to find the transformation which maps the first set of points into the other. The problem may look trivial for humans but it is not for computers. This fact justifies the opulent literature in the topic [15] which introduced many methods to solve the problem for optical images. Acoustic image registration is a requirement for 3D reconstruction and motion estimation. Unfortunately, common image registration techniques used for optical images normally fail in acoustic image registration [16]. In this thesis we review some important work in image registration and point matching algorithms. Also, we develop a new algorithm, named Sorting the Correspondence Space, to tackle acoustic image registration.

1.6 Main contributions

The main contributions of the thesis can be outlined in the following points:

- Novel approach for underwater 3D reconstruction; the proposed ASI system features cheaper implementation than array-based sonars and solves the time delay problem in the T–configured 3D sonars.
- New motion estimation framework for AUVs using forward looking sonar images; the introduced framework transforms onboard acoustic camera to a motion sensor.
- Detailed performance and accuracy study on an earlier motion estimation algorithm for AUVs [14].
- Novel, fast, and robust algorithm for affine point matching. The algorithm, named Sorting the Correspondence Space (SCS), was specially designed to tackle acoustic image registration as it is the core problem in motion estimation and 3D reconstruction alike.

1.7 Thesis structure

The thesis is organized as follows: Chapter 2 is a literature review on three major topics: underwater 3D imaging, motion estimation for AUVs, and image registration techniques. Chapter 3 explains the proposed Acoustic Stereo System in detail and study different parameters affecting its performance. Chapter 4 presents new motion estimation framework based on sonar arc sampling and Modified Discrete Uniform

Distribution (MDUD). Chapter 5 studies different aspects affecting the performance of an earlier motion estimation algorithm [14] [17]. Chapter 6 introduces a new algorithm for point matching; the algorithm is called sorting the correspondence space which can be used for acoustic image registration. Conclusions and further work is presented in Chapter 7. Figure 1.4 illustrates the hierarchical structure of the thesis.

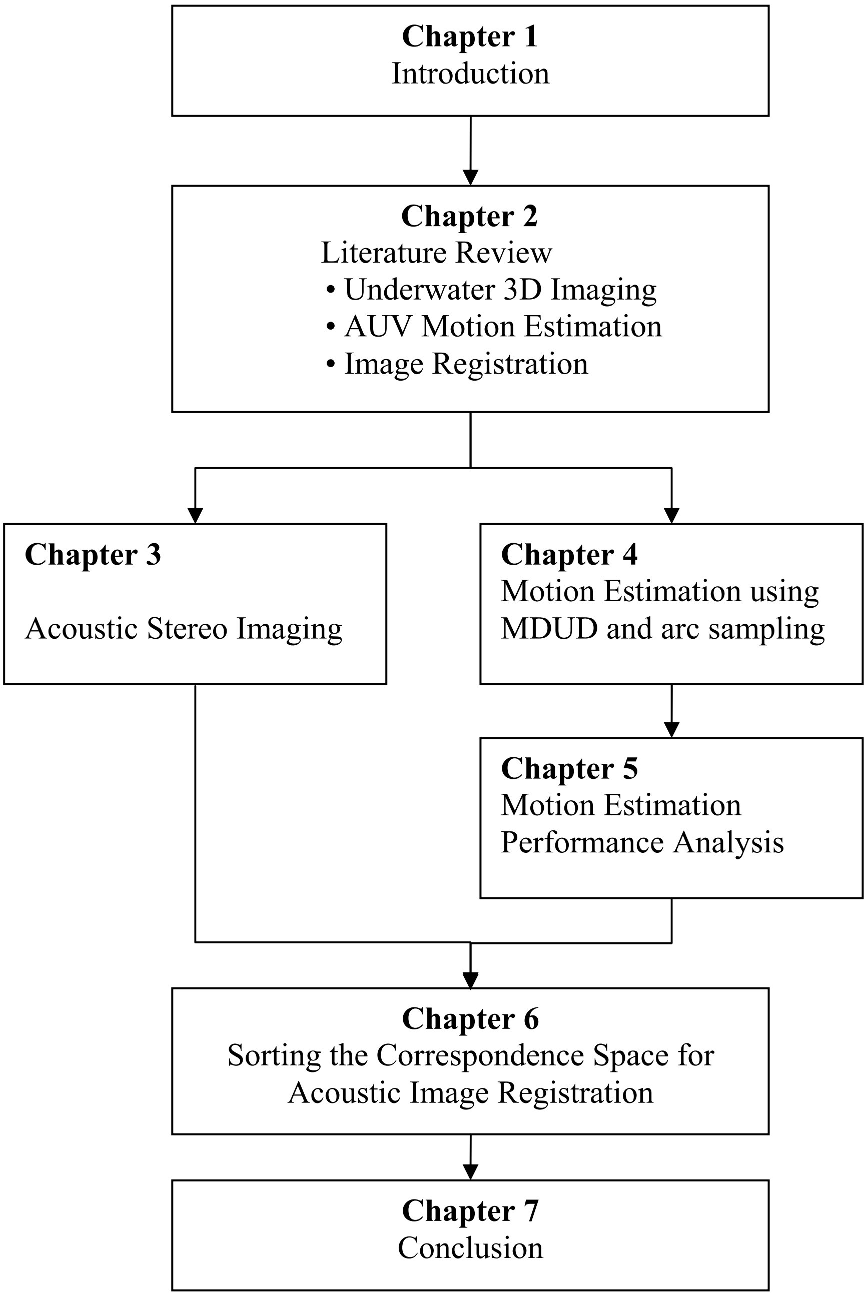


Figure 1.4: Hierarchical Structure of the thesis

Chapter 2
Underwater 3D Reconstruction and Motion Estimation

2.1 Introduction

In this chapter, we review different methodologies for underwater 3D reconstruction and motion estimation for AUVs, also, we discuss the correspondence problem between two acoustic images as it is the core problem in 3D reconstruction and motion estimation alike.

Underwater 3D reconstruction is reviewed from two main perspectives: 1) 3D sonar systems and 2) Algorithms for 3D reconstruction using 2D optical and/or acoustic images. Moreover, we discuss motion estimation for AUVs which is usually implemented in two main stages: 1) using motion estimation sensors, including the inertial navigation systems, Doppler Velocity Log (DVL), and acoustic beacon technologies; and 2) Motion estimation algorithms such as Kalman filters and particle filters. Image registration is a prerequisite to accomplish both 3D reconstruction and motion estimation using successive sonar views, thus the important methodologies for image registration is reviewed, mainly, SIFT (Scale Invariant Feature Transform), RANSAC (RANdom Sample And Consensus) and their descendants.

2.2 3D sonar systems

In most 3D acoustic systems, the scene is insonified with an acoustic pulse and the backscattered signals are collected using 2D sensors array. Many references emphasize that “a 2D sensors array is mandatory to obtain a 3D image” [18][19][20]. In this section, we review acoustic systems that use 2D array of sensors. They are categorized in three major kinds depending on the signal processing methodology, including: beamforming systems, holographic systems and acoustic lens-based systems. Moreover, we discuss the problems shared between all of these methodologies and point out the attempts to resolve them. Furthermore, T-configured 3D sonars that use two crossed linear arrays are reviewed as well as the lens concentrator systems that use mechanical methods to steer the acoustic beam. Also, interferometric sonar is discussed briefly. Finally, real and commercial 3D sonars are presented.

2.2.1 Beamforming systems

A beamforming system receives the backscattered signals using a 2D array of sensors and processes these signals so the signal which is coming from the steering direction is amplified and all other signals are largely attenuated. As a result, 3D data can be inferred in the steering direction and a complete 3D image can be constructed by repeating this process with adjacent steering directions similar to the raster scan method. The beam pattern in the beam forming systems is illustrated in Figure 2.1, it can be written in a closed form if the sensors are equispaced with the distance d and centred in the coordinate origin as follows [20]:

$$BP_{BMF}(\omega,\beta,\theta) = \frac{\sin[\omega N d(\sin\beta - \sin\theta)/2c]}{\sin[\omega d(\sin\beta - \sin\theta)/2c]} \qquad (2.1)$$

Where N is the number of sensors; θ is the steering direction; d is the distance between two adjacent sensors; $\omega = 2\pi f$ angular frequency; f frequency; c sound velocity in the medium, $\beta \in [-\phi_{max}, \phi_{max}]$, and $2\phi_{max}$ is the angular vertical coverage of the sonar.

The beam pattern is an important tool to evaluate the system performance; it acquaints us with two important key points: main lobe shape and sidelobes' level. The width of the main lobe represents the angular resolution which is mentioned as the lateral resolution in some references [21]. The side lobe level affects the noise level in the final output, reducing the side lobe level decreases the artifacts in the generated image, this is achieved by applying Adaptive Beamforming algorithms and Optimized Simulated Annealing Algorithms to solve this problem [22] [19], basically, these algorithms use sparse arrays rather than fully sampled arrays to reduce hardware cost and curtail computational burden, these requirements are achieved by: 1) optimize the sensors' positions with fixed weights to get the required beam pattern, or 2) fix the sensors' positions and optimize their weights, or 3) optimize the positions and weights concurrently. However, reducing the side lobe level increases the width of the main lobe which means reducing the angular resolution, trade-off solutions is proposed in [23] [24] [25].

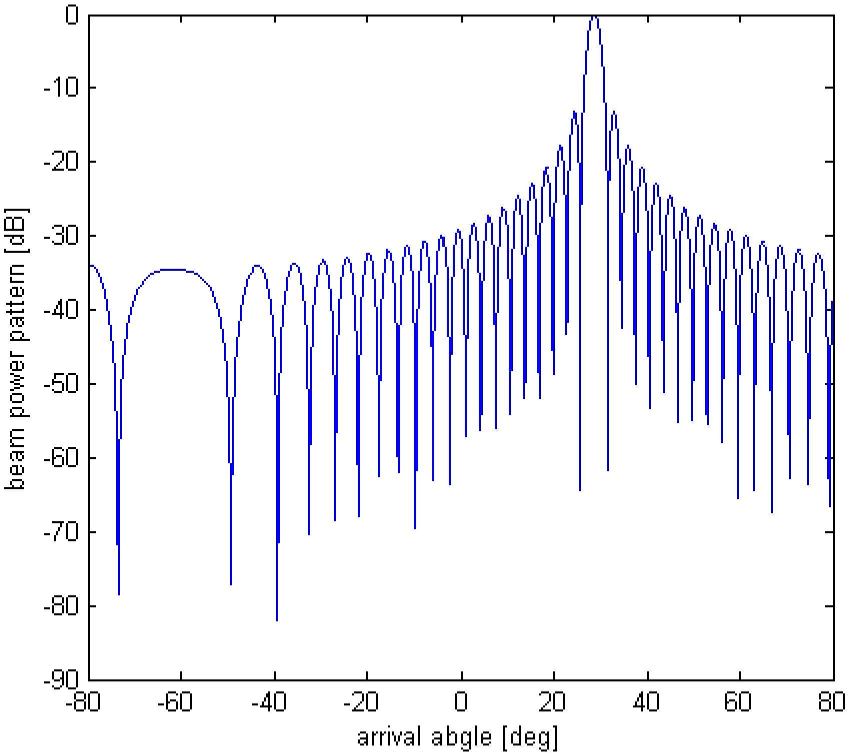


Figure 2.1: beam pattern of 64-element array with 1.5 mm spacing, steering θ = 30° frequency = 300 KHz

The beam pattern equation in (2.1) deals with a linear array of sensors, this is to simplify the equation at first instance; however, a linear array is not sufficient to produce 3D image in terms of the beam forming methodology. To produce a 3D image we need to steer the beam in 3D space and extend our equations to the 3D space, we can write the beam pattern equation in 3D space if we have a square array with equispaced sensors as follows:

$$BP_{\circ B}(\omega, \beta_\alpha, \beta_e, \theta_\alpha, \theta_e) = BP_{BMF}(\omega, \beta_\alpha, \theta_\alpha) BP_{BMF}(\omega, \beta_e, \theta_e) \tag{2.2}$$

Where θ_α and θ_e are the azimuth and elevation steering angles respectively; Figure 2.2 illustrates the beam pattern in 3D space produced from an array of transducers with the steering angels: $\theta_e = 0°$, $\theta_a = 30°$

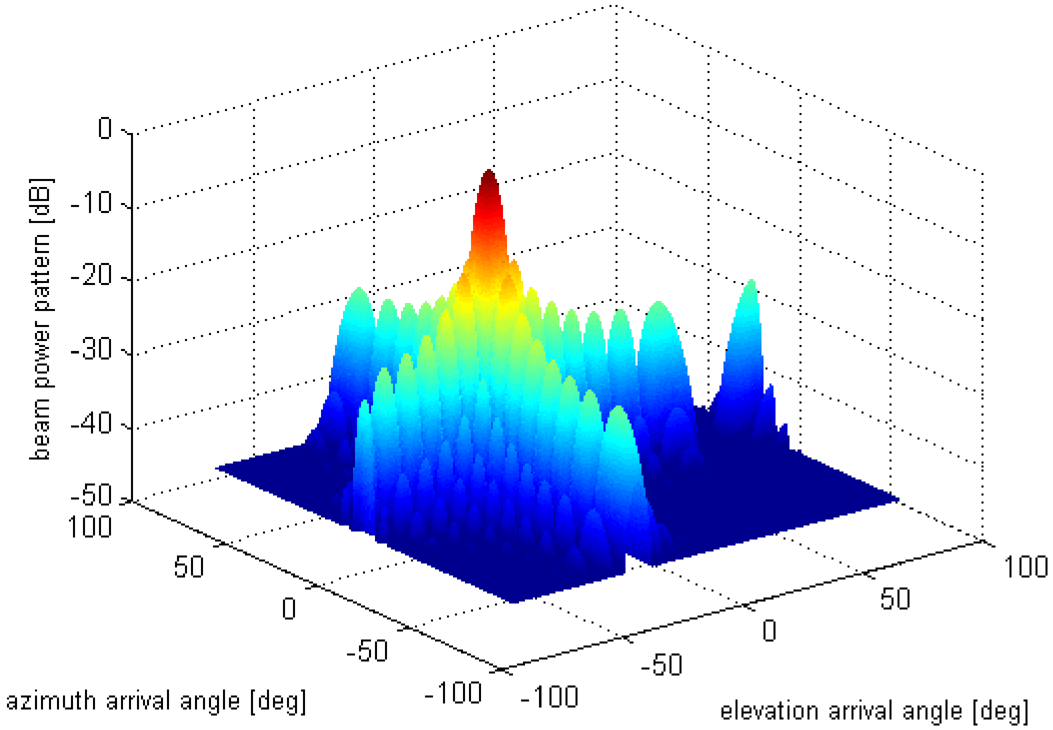


Figure 2.2: beam pattern of 2D array with 15x15 elements with $\lambda/2$ spacing, steering angles $\theta_e 0°$, $\theta_a = 30°$ frequency = 300 KHz

2.2.2 *Holographic systems*

There are two types of holographic algorithms, the first one use Fourier/Fresnel approximation in the narrow band systems; the second type of holographic algorithms use singular value decomposition to estimate the scene reflectivity in the wide band systems [20] as implemented in the Echoscope sonar [26].

In the holographic narrow band systems the algorithm phase-shift the signals collected from the 2D sensors and apply discrete Fourier transform to move from the index of the sensors $l = 1 \ldots N$ to the index of the spatial frequencies k [27]. The derived beam pattern from this algorithm can be written as follows [20]:

$$BP_{HOL}(\omega, \beta, k) = \frac{\sin\left[\frac{\omega N d}{2c}\left(\sin\beta - \frac{2\pi k c}{\omega N d}\right)\right]}{\sin\left[\frac{\omega d}{2c}\left(\sin\beta - \frac{2\pi k c}{\omega N d}\right)\right]} \tag{2.3}$$

Where N is the number of sensors; k is the index of the spatial frequencies; d is the distance between two adjacent sensors; $\omega = 2\pi f$ angular frequency; f frequency; c sound velocity in the medium, $\beta \in [-\phi_{max}, \phi_{max}]$, and $2\phi_{max}$ is the angular vertical coverage of the sonar.

It is clear that equation (2.3) is identical to the beam pattern equation (2.1) in the beam forming systems, so the relation between the k index and the steering direction θ can be written as follows [28]:

$$\sin\theta = \frac{2\pi kc}{\omega Nd} \tag{2.4}$$

Thus, all drawbacks and limitations of the Beamforming systems are applicable on the holographic systems using narrow band frequencies. These include the side lobe problem, the aliasing problem when $d > \lambda/2$ (d is the distance between two adjacent sensors; λ is the wave length), and the imperfect focusing. Again, lots of algorithms are applied to optimize sensors deployment in the planer arrays such as Monte Carlo simulation (MC), Simulated Annealing (SA), Intra-Block Monte Carlo method (IBMC), and combined SA-IBMC algorithm [29]. The beam pattern in this case is identical to the one illustrated in the Figure 2.1.

In the second type of the holographic systems, the problem is re-formulated into matrix equations. The pressure field is denoted by $s(\omega, p)$ as $N \times 1$ vector where N is the number of sensors placed in p_l and $l = 1 \dots N$; the $M' \times 1$ vector $c(\omega, r)$ contains the reflectivity of each resolution cell in the imaged space (see Figure 2.3), the relation between these matrixes is depicted as follows:

$$s(\omega, p) = U(\omega, p, r)c(\omega, r) \tag{2.5}$$

Where $U(\omega, p, r)$ is an $N \times M'$ transfer matrix, in (2.5) the whole imaging problem is turned into estimating the vector $c(\omega, r)$ given the transfer matrix $U(\omega, p, r)$ and the measured vector $s(\omega, p)$. Singular value decomposition (SVD) [30] is utilized here to solve for the pseudo-inverse matrix of $U(\omega, p, r)$ so the estimated $\tilde{c}(\omega, r)$ vector is defined as follows:

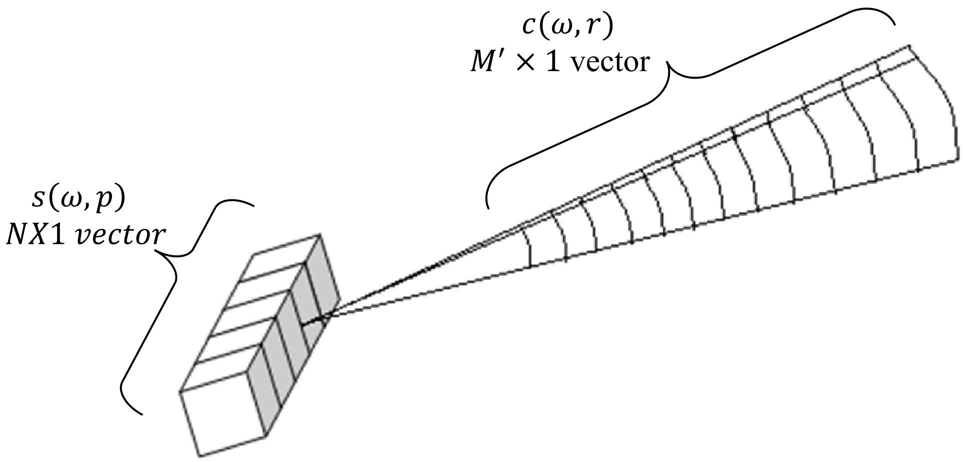


Figure 2.3: representation of the matrixes in the holographic methodology for one beam, the figure represents linear array only, in 3D sonars the sensors are situated in 2D array.

$$\tilde{c}(\omega, r) = U^H (U.U^H)^+ s \qquad (2.6)$$

Where ^{H} is the complex conjugate and transpose; $^+$ is the pseudo-inverse of the matrix. It worth mentioning that (2.6) is usually written for each layer in the imaged space which is divided to many concentric spherical layers to minimize the size of the computed matrixes.

2.2.3 Acoustic lens systems

Acoustic lens systems work in a similar way to the optical ones, basically, an acoustic-lens system consists of an acoustic lens to focus the backscattered signals on a 2D retina of sensors [31]. Each sensor receives a signal that represents the echo from a well-defined direction. Measuring the time of flight for these signals gives us the opportunity to build 3D image rather than 2D image as in the optical systems.

The phase shift which is implemented computationally in the holographic systems happens naturally in the acoustic lens when the signals pass through them. In the acoustic-lens systems the sensed field is limited to the aperture of the lens, so it can be calculated using the following equation:

$$S'(\omega, x) = \begin{cases} S(\omega, x)e^{j\omega x^2/2CF} & -A/2 \leq x \leq A/2 \\ 0, & otherwise \end{cases} \tag{2.7}$$

Where F is the focal length of the lens; A is the aperture of the lens; the beam pattern of the acoustic lens is defined as follows [20]:

$$BP_{lens}(\omega, \beta, \theta) = 2\frac{\sin[\omega A(\sin\beta + \sin\theta)/2c]}{\omega A(\sin\beta + \sin\theta)/2c} \tag{2.8}$$

The beam pattern in the acoustic lens systems and the beam forming systems is very similar; however, acoustic lens systems suffer some drawbacks according to the physical nature of the lens i.e. reducing sidelobes is not achievable through ad hoc weight coefficients as in the beamforming systems [32] [33] . Moreover, regulating the focusing distance to make it close to the scatterer distance requires complex mechanical arrangement. Above all, the phase shift performed by the lens is applicable only in Fresnel approximation zone. These problems are tackled in [34] [35] [27] [36]. Next we discuss the problems of the aforementioned planar array methodologies.

2.2.4 Problems of acoustic 3D imaging using planar arrays

Developing 3D acoustic imaging system by using planar array is hindered by two main factors: first of all, the cost of building the planar array which is composed of huge number of acoustic sensors [37]; secondly, the computational load to process the signals obtained from the acoustic sensors. These two linked problems are shared between all three systems explained earlier i.e. Beamforming system, Holographic system, and Acoustic-lens system, in the later one the computational load is abated at the cost of adding the acoustic lens, moreover, it requires dense retina as each sensor correspond to one direction only in one-to-one relation, obviously, this is not the case in the Beamforming and Holographic systems where the number of sensors is a little bit smaller than the number of beams; however, Beamforming and Holographic systems require more computational power and large storage capabilities compared to lens based systems.

Designing the planar array have to be accomplished by considering the aliasing problem, this means that the space between the acoustic sensors should not exceed a given value expressed in terms of the wave length $(\lambda/2)$. On the other hand, obtaining a fine resolution requires an array with wide spatial extension. As a result, the planar array will be composed of thousands of acoustic sensors to meet these requirements. To solve this problem, 3D imaging by sparse array has been presented [19] [38]. The problem of designing the planar array is turned into an optimization problem to reduce the number of acoustic sensors in the planar array and to find optimum positions and weights for them; the objectives of the optimization problem here are to lower the sidelobe level, narrow the main lobe width, and to avoid grating lobe in the beam pattern. This procedure reduces the cost of the planar array and truncates the computational burden required to produce 3D map.

3D underwater imaging inherits most problems of the 2D underwater imaging, namely, the speckle noise, specular reflections and spatial and temporal coherences of the insonified medium. Speckle noise is caused by the interference of the backscattered signals from rough surfaces, it can be reduced by using adaptive filtering [39] or by using the statistical components of the noisy signal [40]. On the other hand, specular reflection tends to exist while imaging man-mad objects, these objects tend to have smooth surfaces where acoustic signals bounce back to produce images with poor details and high contrast. Finally, the spatial and temporal properties of the medium distort the images in long-range scans; they can be safely neglected herein as we are focusing on short range systems only.

2.2.5 T - Configuration

Many sonar systems accomplish 3D-scan of the scene by using successive horizontal beamforming transmission and vertical beamforming reception. In other words, a vertical transmitting array insonifies a thin slice in elevation, then a horizontal receiving array collects the signals from different azimuth angles across each emitted slice. This arrangement is called the “T Configuration” or “Mill’s Cross Configuration”; its main advantage is the small number of transducers and transceiver channels which means cheaper and simpler systems, on the other hand, it has a critical disadvantage, the time to scan the 3D scene is very long compared to the 2D array systems, for example to get 128x128 points in 100m range by using the T configuration, the system needs to emit 128 successive pulses with 0.13s minimum delay between each pulse, therefore the system needs 17.06 s for complete 3D scan of the scene. Consequently, these systems are useful only for very short ranges.

2.2.6 Concentrator lens

Concentrator-lens systems can be considered as the mechanical version of the Acoustic-lens systems mentioned in a previous section, in the concentrator-lens systems, underwater 3D data can be deduced using one 2D Sonar equipped with concentrator Lens to form pencil-like beams (like DIDSON), by rotating this sonar 45^0 degrees we can perform full 3D scan in the field of view [41]. The lens' concentrator reduces the vertical beamwidth from the default 14^0 degrees to either 3^0 degrees or 1^0 degree depending on the concentrator type [6]. This procedure was implemented in Yamanaka lake-Japan to get full 3D image of the aquatic plants, the methodology requires GPS and motion sensors with heading, rolling, and pitching angles to compensate for slight DIDSON movements; as it uses mechanical rotation system, this methodology requires more time to scan the field of view compared to all other systems, one of the first articles which explains this methodology can be seen in [42].

2.2.7 Bathymetric systems

Multibeam echosounder (MBES) is typically used to generate bathymetric maps of the sea floor. The sonar's swath is produced from hundreds of narrow adjacent beams in fan-like form, see Figure 2.4. 3D bathymetric map can be generated as a result of the movement of the mounting platform. To generate 3D bathymetric map, the mounting platform may move in a linear path or simply rotate the sonar head to acquire 3D map of the dome under the sonar.

There are two ways to process the acquired data from MBES to generate 3D seafloor maps. The first approach filters data points to remove outliers and smooth the generated surfaces by gridding the surviving data points. In other words, interpolation is performed on the filtered data points to produce digital terrain model of the insonified area. Different algorithms have been proposed within the circle of this approach including utilizing band pass filter for outliers removal, and applying wavelets and quad-tree data structures to build surfaces [43]. The second approach models the seafloor terrain using Triangulated Irregular Network (TIN) [44]. This approach filters the acquired data as the previous approach but uses the filtered data directly with no gridding. Triangulated Irregular Network algorithm presents better details in the modelled terrain than the gridding approach at the cost of complex implementation [45].

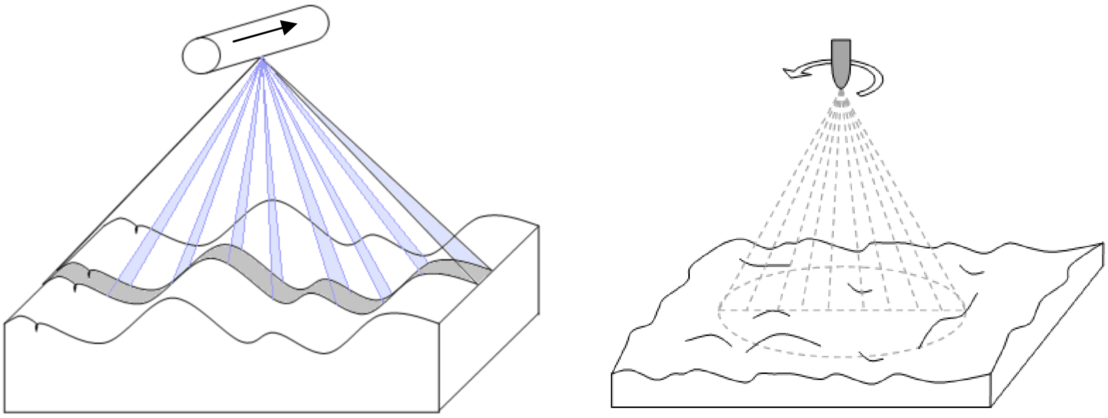


Figure 2.4: 3D bathymetric map can be generated using multibeam echosounders mounted on a moving AUV or a rotating platform.

2.2.8 Interferometric Systems

Interferometric sonar utilizes phase measurement at each receiving element to estimate the angle of the scatterer which reflected the acoustic pulse. The receiving elements, placed at known distances, allow determining the angle θ of the scatter by calculating the phase offsets of the acoustic return. Interferometric sonar doesn't depend on beamforming technology, but rather on the accurate measurement of the phase at each receiving element, it can produce bathymetric maps by combining the range measurement and the calculated θ angle. Interferometric sonar has been used for decades, in its early days, ambient and internal noise were the main problems facing its development. In addition, the problem of resolving multiple angles of arrival (simultaneous returns from different angles) and the shifting footprint effect (where acoustic returns from close parts of the seabed being received at different receiving elements simultaneously) were considered as significant problems affecting phase fluctuations in interferometric sonar data [46]. However, interferometers have been significantly improved with the advent of new technologies in electronics and transducer manufacturing, in addition to different algorithms serving the phase-measurement of the acoustic returns.

2.2.9 Real acoustic 3D imaging systems

Echoscope 3D Sonar is one of the remarkable 3D sonars available off-the-shelf, the sonar was designed and developed by a Bergen-based company called OmniTech which was acquired by CodaOctopus in 2002. Currently, the 3D camera consist of 128x128 beams (16,384 total) with 50°x50° Angular Coverage and 0.39° beam spacing, they use 375kHz frequency which allows for up to 150m maximum range [26] [47]. The first version of the Echoscope started with 40x40 beams only, it follows the holographic approach presented in the section (2.2.2) which avoids Fresnel Approximation, Figure 2.5 shows one image from the Echoscope.

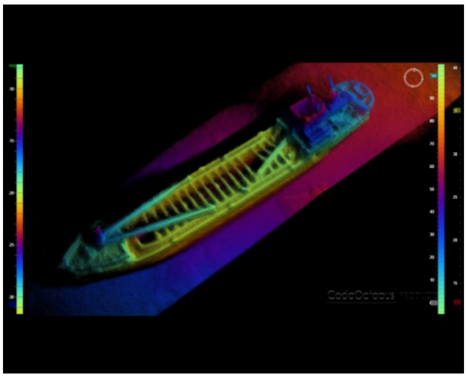


Figure 2.5: Real-time 3D image of the MV GB Church [48]

Cheaper 3D sonars follow the T configuration at the cost of longer scan times, two commercial sonars from this category are Eclipse from Tritech International Limited, and Pin-Point 3D intruder detection sonar from Marine Electronics Ltd.

Eclipse is 3D imaging sonar with 3-dimensional measurement capability. A single housing contains an orthogonal pair of dual transmit and receive multi-beam arrays, see Figure 2.6; it has 256 beams operating at 240kHz with a range of up to 120 meters [49]. Eclipse scans a 3D volume with 120 by 45 degrees. By electronically steering the horizontal or vertical transmit beam, a volume is illuminated in front of the sonar. This is converted into full 3-dimensional image. 3D Volume Visualization provides a realistic graphical representation of scanned objects.

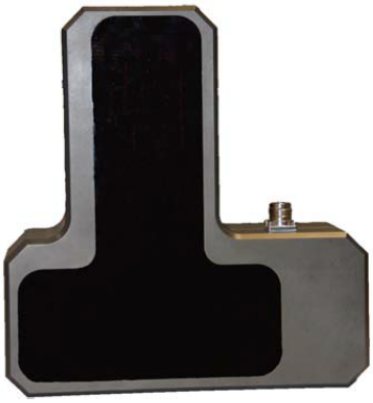


Figure 2.6: Eclipse sonar head demonstrating the T configuration

The pin-point intruder detection sonar is another forward-looking active sonar that follows the T configuration to provide a 3D display of an area, see Figure 2.7. The Pin-Point scans both horizontally and vertically to produce a three dimensional representation of the area in front of the sonar with a minimum range equal to 20 meters and a maximum range of 500m. The “T configuration” transducer array can be either fixed or vessel mounted and connected by cable to the electronics processing unit which is controlled from the operating PC via an ethernet connection [50]. The sonar scans a horizontal sector of 90° to a 1.5° resolution and a vertical sector of 20° to a 1° resolution for every “ping” of the transmitter. It has 32 transmitter beams and 64 receiver beams.

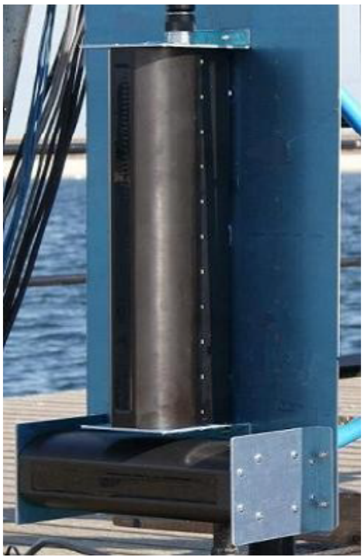


Figure 2.7: Pin-Point Intruder detection sonar as an example of the T Configuration

Another type of the 3D sonar systems is the moving multibeam sonar, these are sonars which generate multibeam swath and depend on the rotating head to generate full 3D scan, other variations of this methodology rely on the movement of the mounting platform to generate 3D bathymetric map. 3D Profiling Sonar 2001 from Marine Electronics Ltd stand as an example of the 3D sonars with rotating multibeam head, its acoustic transducer scans a horizontal swath and is then rotated by a small angle and another swath captured until a complete circular area underneath the sonar dome is covered [51], it operates with 1MHz frequency with 1.8° transducer beamwidth and varying swath angles of 30°, 60°, 90°, 120°, 150° and180° , see Figure 2.8. BlueView's BV5000-1350 and BV5000-2250 are other examples of the 3D mechanical scanning sonars which depend on the same technology with slight differences, they feature 256 beams with 1°x1° beamwidth and 0.18 beams spacing, their field of view is 45° [52].

BlueView's MB1350-45 and MB2250-45 multibeam sonars deliver bathymetry data depending on the movement of the mounting platform (AUV, ROV…), they have 256 beams with 1°x1° beamwidth and 0.18° beam spacing, their swath angle is 45°. These two sonars feature low power consumption as they operate with 15 watts only.

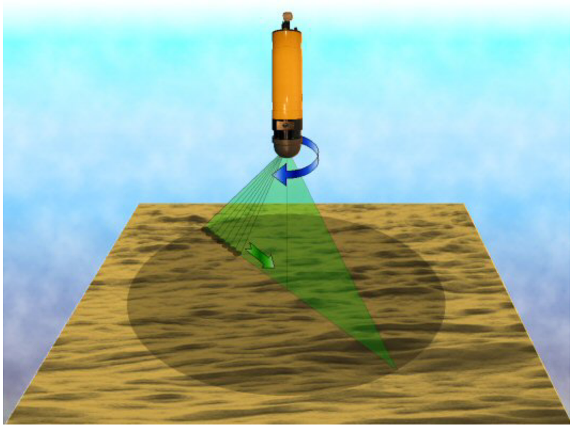


Figure 2.8: 3D Profiling Sonar 2001 from Marine Electronics Ltd

2.3 Algorithms for underwater 3D reconstruction

In this section we discuss three categories to perform underwater 3D reconstruction using 2D images obtained from optical or acoustic cameras.

2.3.1 3D reconstruction using shadows in acoustic images

Deriving a shadow model for objects imaged by forward-looking sonar is of considerable importance. The conventional method for object reconstruction using shadows in sonar images is to segment the sonar image into three regions: background, shadow, and objects [53]. Thereafter, the shadow and the object segments are used to estimate the elevation of each object individually. Merging these estimates give the desired 3D map. In a similar approach [54], the Markov Random Fields (MRF) method has been used to segment sonar images to three kinds of regions, namely, echo, shadow and background. The proposed scheme uses shadow regions to infer the elevation map. Also, it uses echo regions to infer a reflection map. These maps are merged in a final stage to build the 3D reconstructed scene, the main drawback of this methodology that it uses GPS information to estimate the motion between acquisition view points.

Another approach was implemented for 3D object recognition and reconstruction by exploiting the shadow data in DIDSON Images [55]. This method depends on a supposed DIDSON model and on the shadow data of the objects. Mainly, it uses correlation to measure the similarity between the actual shadow and the predicted shadow (predefined one). However, this method can be used with simple shapes only (cube, cone, and cylinder) so it cannot be utilized in real environments.

2.3.2 3D reconstruction using underwater optical images

Computer vision techniques commonly applied for 3D reconstruction in land have found the way to underwater applications [56][57][58][59]. Recently, significant attempts have been accomplished to perform 3D reconstruction from optical images acquired from moving AUV/ROV [60][61][62]. In [63] the authors proposed a framework to build large 3D maps based on AUV optical surveys. The first stage in the framework uses modified Harris corner detector [64] and Tuytelaars method [65] for feature extraction. These features are used in RANSAC-based algorithm to search for the essential matrix which best describes the transform between two successive images. Next the transform is used to infer 3D features that can be interpolated to obtain 3D sup-map in the reference frame. Finally, the sub-maps are matched using RANSAC and Horn's algorithm [66] to build global 3D map of the surveyed terrain. In addition, the framework exploits navigation data from inertial sensors to limit the search space and expedite the execution time in RANSAC phase.

There are fundamental problems in implementing 3D reconstruction based on underwater optical images, particularly the significant light attenuation and backscatter [67]. In fact, finding correspondences between the stereo views is required to accomplish 3D reconstruction. This problem is already challenging in air and it is even more challenging underwater as the image quality degrades significantly with severe light attenuation and backscatter from suspended particles. Moreover, the distances between the object and the stereo cameras are not the same which makes the surface radiance different in each view. These factors make the task of finding correspondences between two underwater optical views more challenging. Thus, applying the methods designed for open-air applications in underwater environment is limited to good visibility condition [68]. In practice, these methods often fail in low visibility conditions [69]. Furthermore, in lowlight condition, artificial light is needed which adds different shadow patterns to the views. Therefore, it increases the difficulty to find correspondences and adds additional burden in terms of the power budget in the AUV.

2.3.3 3D reconstruction using optic-acoustic images

There are some papers tackling underwater 3D reconstruction using optic-acoustic cameras, S. Negahdaripour has derived the epipolar geometry for optic acoustic-camera configuration in [70]. The author ended up with three solutions called the range solution, the azimuth solution and the third one is derived by expressing points' coordinates in the sonar reference frame using the rigid transformation between the sonar reference frame and the camera reference frame. The aim of these solutions is to calculate the missing Z-coordinate of a point in the real world. Obviously, these solutions give the same results in the virtual world (while modelling the sonar and camera). However, they will give different results in the real world because of the different distortions caused by the optical and acoustic sensors and all sorts of noise which affect the images. Another cause of these differences is the models adopted while deriving the epipolar geometry of the optic-acoustic cameras. These models (The pine-hole model for the optical camera and the polar- coordinate scan for the acoustic sensor) are approximated models of the real world.

The supposed epipolar geometry turns the search for the pair points in the sonar and optical images into one dimension search. Thus, if we have a point in the sonar image $p = (\theta, R)$ we can search for its corresponding point in the optical image by searching on the corresponding epipolar curve in the optical image. According to [70], these curves are ellipsis or hyperbolas curves. Whatever the case, this method does not give acceptable results due to the causes mentioned earlier. In other words, the complex correspondence problem remains to be the main impediment that hinders this 3D reconstruction approach to evolve.

Recently, S. Negahdaripour et al. have proposed a method to get around the correspondence problem to integrate the motion cues in optical and sonar videos for 3D positioning [71]. This method can be considered as the first step to solve the correspondence problem; simply it deals with each sensor alone without requiring matches between the sonar and the optical camera. Instead of searching for matches in the two images, it processes the sequences of images from each sensor individually, and deduces the 3D points from these sequences. The last step in this method is to optimize the 3D parameters of the deduced structure (or feature points). Clearly, this method does not apply the epipolar geometry but utilizes some of its results. As we have just mentioned, this method can be the initial step to solve the correspondence problem by reprojecting the optical pairs to the sonar image to obtain their matches (and repeating the same thing with the sonar pairs), these matches can be considered as the starting points (seeds) to propagate from and calculate other matches.

It has been noted that the epipolar geometry of the optic-acoustic stereo imaging has a theoretical nature [71]. In addition to that, we know that this system will only work in the perfect or very good conditions in terms of visibility and illumination. The optic-acoustic configuration depends on the optical sensor as well as the acoustic one. Thus, it cannot work in dark and/or turbid waters. This is preventing it from being applied in most applications ranged from surveillance to monitoring fish behaviour near the rivers' mouths where the turbidity is very high. In other applications, applying a synthetic light may affect the experiment/application itself.

Finally, we know that the optical sensors work within a metre or a fraction of a metre in turbid waters [5], this means that there is a little intersection between the FOV of the acoustic camera and the FOV of the optical one. For example, if we assume that the visibility is within 3 meters (which is considerably a good condition and it is not the case in most harbours) and by recalling that the range of the DIDSON Sonar is from 1m to 12m while working in the high resolution mode (1.8 MH) [72], then the intersection between these two FOVs is merely from 1m to 3m range. This intersection represents 18.18 % of the DIDSON's FOV. In the low resolution mode the measured range is 1m up to 40m, so the intersection is merely 5.12 % of the DIDSON's FOV. Obviously, these percentages speak for themselves which drives us far away from adopting the optic-acoustic configuration.

2.4 Motion estimation systems

In this section, we cover dominant methodologies and different sensors used to implement motion estimation systems for AUVs. Motion estimation for AUVs is a wide topic and can be viewed from different perspectives; our aim here is to present the bigger picture of the field rather than focusing on our specific methodology. In fact, in our motion estimation research, we concentrate on utilizing forward looking sonar as a motion sensor for the AUV; in other words, our aim is to feed the adopted filtering methodology on the AUV platform with motion measurements derived from the subsequent acoustic images obtained from the sonar. Filtering methods are not in the core of the conducted research herein; however, we review them briefly for the completeness of the work and to give an overall picture for navigational systems for AUVs.

2.4.1 Motion estimation using inertial sensors

Inertial sensors are mainly gyroscopes to detect the acceleration of the AUV, they can be the classic low-cost micro-electro-mechanical (MEMS) gyroscopes [73], the advanced and expensive Fibre Optic Gyroscopes (FOG) [74], or Ring Laser Gyroscopes (RLG) [75], in AUVs, these sensors are usually accompanied with Doppler Velocity Log (DVL) which can measure the AUV velocity near the sea bottom. Also, Acoustic Doppler Current Profiler (ADCP) is usually combined with the inertial navigation systems (INS) to correct for the drift in the estimated position especially if the AUV follow linear path or move slowly [76].

Figure 2.9 shows the KN-6050 INS/DVL/GPS combined sensor which is based on Kearfott’s Monolithic Ring Laser Gyro (MRLG), the Teledyne RDI Doppler Velocity Log (DVL), and internal GPS receiver [77]. An extensive study in [78] presented the performance of these sensors in the Arctic at latitudes exceeding 80°. Figure 2.10 illustrates HG1700 three Ring Laser Gyroscopes from Honeywell, it is widely used in UAVs (Unmanned Air Vehicles), and AUVs.

Figure 2.9: Kearfott integrated INS/GPS/DVL

All inertial navigation systems suffer from integration drift of different magnitudes depending on technology and methodology, as the error of the estimated position increase rapidly over time, inertial navigation systems require a fix from external reference systems such as GPS or APS (Acoustic Positioning System). Numerous studies presented algorithms to combine INS and GPS data in different robotics and/or control applications such as those in [75][79][80][81], a notable study, conducted in Japan Agency for Marine-earth Science and Technology (JAMSTEC), to fix the INS data in AUVs has been introduced in [82], the study presented two algorithms based on Kalman Filter (KF) and Extended Kalman Filter (EKF) to integrate super short baseline (SSBL) and INS methods, a similar approach is adopted in [83].

Improving the INS performance has a great impact on the robotic industries in air, land and sea as the INS systems feature complete autonomy compared with other navigational methodologies, a simple and effective way to enhance INS performance has been presented in [84] based on hardware redundancy followed by wavelet analysis to detect drift errors in INS sets; however, it has been widely noted that INS systems require regular fixes from an external reference system [82][83].

Figure 2.10: HG1700 three Ring Laser Gyroscopes from Honeywell

2.4.2 Motion estimation using acoustic beacons

Different navigational methodologies are included under the acoustic beacon umbrella as they all share the same simple principle: “the vehicle transmits/receives acoustic pings to determine its location” these systems include floating acoustic buoys, Long Base-Line (LBL) system, Short Base-Line (SBL) system, and Ultra Short Base-Line (USBL) system, the later is sometimes referred to as Super Short Base-Line (SSBL) system.

LBL system requires the installation of at least two transponders (“beacons”) on the sea bed or on the surface in the inverted LBL case. Transponders are usually positioned in well-defined places apart from each other; they immediately return an acoustic signal sent to them by the AUV, thus, the AUV can infer its position by using simple triangulation fed with the transponders’ positions, the depth measurement, and the time of flight of the signal to each transponder. LBL systems’ ranges may reach up to 10Km in deep water [85] but that decreases significantly in shallow water where the multipath problem has a major effect on the system. In theory, LBL beacon network can be extended as much as needed [86]; however, that will be accompanied with high installation and maintenance costs, another drawback factor is the requirement to calibrate and align the beacons as well as the AUV.

SSBL systems use single transponder attached to the AUV to act as an acoustic beacon, in this case, the vehicle can be tracked from the mother ship which is equipped with a transmitter and a receiver array, using these equipments, the mother ship can detect

Time of Flight (TOF) and Direction of Arrival (DOA) of the signal to determine the vehicle's position. In other words, the position of the AUV will be estimated on the mother ship and transmitted to the AUV using SSBL acoustic signal so the AUV will determine its self-position and fix any INS drift error.

In the inverted SSBL (ISSBL), the vehicle and the mother ship exchange roles, i.e. the transponder will be attached to the mother ship and the AUV will be equipped with a transmitter and a receiver array. In this scenario, the AUV will be able to estimate its absolute self-position based on the GPS data transmitted from the mother ship to the vehicle using the ISSBL acoustic signal [87]. The inverted SSBL is preferred over normal SSBL when the area of interest contains multiple AUVs at the same time. the main disadvantage in SSBL and ISSBL is the requirement for the mother ship which has to be in close proximity to the vehicles, this increases operation's cost and even raises the question about the autonomy of the vehicle, in fact, vehicles with SSBL navigation systems are described as untethered rather than autonomous [88].

Floating acoustic buoys systems have been introduced and studied by several authors [89][90][91][92][93]; the buoys are equipped with acoustic receivers/emitters, GPS receivers, and radio interconnection between them. Each buoy in the system determines the distance to the vehicle by listening to the acoustic ping sent from the synchronized vehicle, the master buoy then collects the data from the buoys using the radio connection and infers the vehicle position using simple triangulation, then the inferred position is sent to the vehicle using acoustic communication channel. This system, dubbed GPS Intelligent Buoy (GIB) system [94], has a major advantage of being able to serve multiple vehicles at once by differentiating their pings based on time, frequency, or pseudo-random code. However, its main disadvantage is the delay incurred in receiving the acoustic pings from the vehicle and sending the inferred position back using the acoustic channel.

In another approach, the buoys act as GPS satellites by broadcasting their GPS positions in regular times through acoustic pings. Using time-of-flight measurements of the received pings, the synchronized vehicle determines its absolute ego-position in a similar manner to those in GPS receivers. The system has been described in an early study in [95].

2.4.3 Motion estimation using geophysical features

Geophysical navigation systems use geophysical features in the terrain to estimate the AUV's position. Geophysical features may refer to magnetic, bathymetric, and gravitational features.

These systems can be grouped in two sub-categories; the first one is the Terrain Based Navigation (TBN) systems where the estimation is performed by providing the AUV with an existing map of the area and searching for the best match between the observed features and the provided map, a good survey on this approach can be seen in [96]. The second category in the geophysical navigation systems perform the estimation with no prior knowledge of the map, basically, they depend on feature registration in the subsequent observed acoustic/optical images. The latter category introduces more autonomy to the vehicle and enables it to explore new areas where the maps are not available.

The performance of the geophysical navigation system is largely affected by the number of features in the scanned scene and by the ability to extract them from the background, in fact, the challenging task of automatic feature extraction and feature registration using optic and/ or acoustic images is the main problem of the geophysical navigation systems. This is due to the low resolution of the sonar images, the high attenuation of light underwater, and the scarcity of features in the sea floor [85].

AUV navigation based on magnetic or gravitational variations has been proposed in earlier studies [97][98]. Passive gravity navigation utilizes gravity gradient measurements and compares it with gravity abnormality database, the system is proposed to be combined with other technologies using Kalman filter scheme. The performance of these systems has not been published. Other geophysical features include the tidal inlets and hydrothermal vents, however they are rarely encountered in real AUV missions [85].

2.5 Motion estimation algorithms

From methodologies point of view, we may consider two different techniques to estimate AUV positions based on measurements from the aforementioned sensors:

2.5.1 Kalman Filter (KF) and Extended Kalman Filter (EKF)

The Kalman filter estimates the AUV state position by using predict-update cycle; first of all, the filter estimates the state position of the vehicle depending on a proposed model and the control data, then it updates the predicted state based on the obtained feedback in the form of noisy measurements. In other words, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations [99].

If the system is Markovian and linear, and the noise can be modelled as Gaussian, then Kalman filter is an optimal Bayesian estimator [100]; however, the AUV physical model is highly nonlinear, which makes the Extended Kalman Filter (EKF) more suitable for estimating the position of the vehicle [101]. In the EKF, the first-order Taylor series is used to approximate the nonlinear model of the vehicle, in fact, the Jacobian matrix is used to linearize the model while the predict-update cycle survive in the EKF. Calculating the Jacobian matrices requires nontrivial operations and increases the complexity of the system. Furthermore, the EKF often underestimate the covariance of the state in the highly nonlinear problems which incurs large inaccuracies. Different studies concluded that the EKF achieve good results for AUV navigation with an INS and DVL if the vehicle obtains regular updates from the GPS [85]; however, the performance of this system will deteriorate significantly if the GPS signal is not available [73]. EKF evolved rapidly to serve AUV navigation and other applications, one can notice different flavours of the EKF, notably, the two-stage extended Kalman filter (TEKF) [102], adaptive two-stage extended Kalman filter (ATEKF) [79], unscented Kalman filter (UKF) [103], sigma-point unscented Kalman filter (SP-UKF) [104], and fuzzy adaptive federated Kalman filter (FAFKF) [105].

2.5.2 Particle Filters (PF)

The particle filter approach infers the current AUV position (referred as 'state' in the literature) by using a large number of weighted particles (or samples in the state space) to estimate the probability distribution of the state variable. The algorithm utilizes the observed history over the path of the AUV along with its control history; also, it assumes full or partial knowledge of the operation environment [106][107].

The particle filter approach is capable of estimating nonlinear systems, also, it doesn't assume Gaussian model for the noise as KF and EKF [108]. Moreover, it doesn't require initial position or orientation for the vehicle [109]. However, these advantages come at the cost of high demand for computational power resulting from the huge number of particles required to cover the state space. Another issue to mention here is the sample impoverishment problem induced by concentric particles in a small region of the state space, this problem represents a kidnapped vehicle in a wrong estimated position similar to the local minima problem in the optimization literature. In an analogous way to the simulated annealing algorithm, it has been suggested to dedicate a number of the particles to be randomly distributed in the state space regardless of the update process using the Sampling Importance Resampling (SIR) algorithm [106][107]; this arrangement avoids the sample impoverishment problem and guarantee to recover from a wrongly estimated position.

It is important to mention that there is no standard methodology to implement motion estimation for AUVs, researchers have always blended different methodologies and used full range of sensors to develop AUV navigational modules; this is mainly due to the complex nature of the problem and to the diversity of the employed applications in the AUV industry. For example, the authors in [106] suggested using particle filter to get an initial estimation and used extended Kalman filter after that, in an emergency situation, particle filter will take control again to re-estimate the position and give control back to Kalman filter. Also, federated Kalman filter based on the back-propagation neural network (BPNN) was used to implement the AUV navigation system [100]. Simultaneous Localization And Mapping (SLAM) is closely related to the field, dominant SLAM techniques utilize extended Kalman filter and/or particle filter [103]. SLAM received much attention in the last decade, nevertheless, SLAM is beyond the scope of the thesis, interested reader may refer to outstanding studies in [110] [111][112].

Using acoustic images to infer motion measurements has the same prerequisite as 3D reconstruction starting from the acoustic images, namely, image registration for the acquired images; next we review important algorithms prominent in optical and medical image registration domains.

2.6 Image registration techniques

Image registration is the problem of finding the transform function between two images to map points from the first image to same points in the second one. In other words, it can be described by raising a simple question: given two images for the same objects taken from different viewpoints, for each point in the first image where are their correspondences in the second image? This question can be easily answered by humans; unfortunately, it is not the case for computers! Figure 2.11 illustrates this problem.

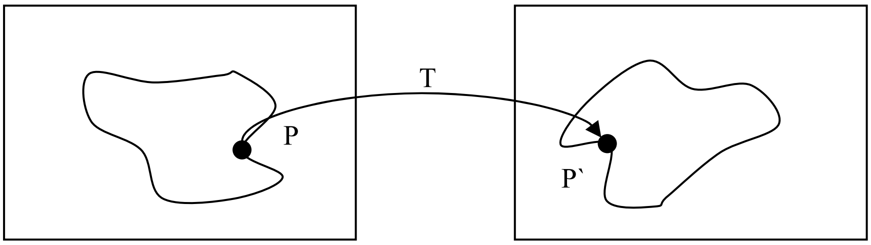


Figure 2.11: Image registration is the problem of finding T

This problem is well-known as the correspondence problem in the computer vision literature; it is faced in many applications where we have two or more sensors scanning the same objects or just one sensor moving while scanning objects of interest. 3D reconstruction, motion estimation, camera calibration, image mosaicing and object recognition are all hindered by this problem; to name but a few.

Image registration is a heavily studied problem, yet there is no complete solution which satisfies all applications where we face it. However, we can roughly set a general platform which all methods can fall into it. Figure 2.12 shows the general structure of the image registration methods. Next, we discuss local based techniques, feature based techniques, and point matching algorithms.

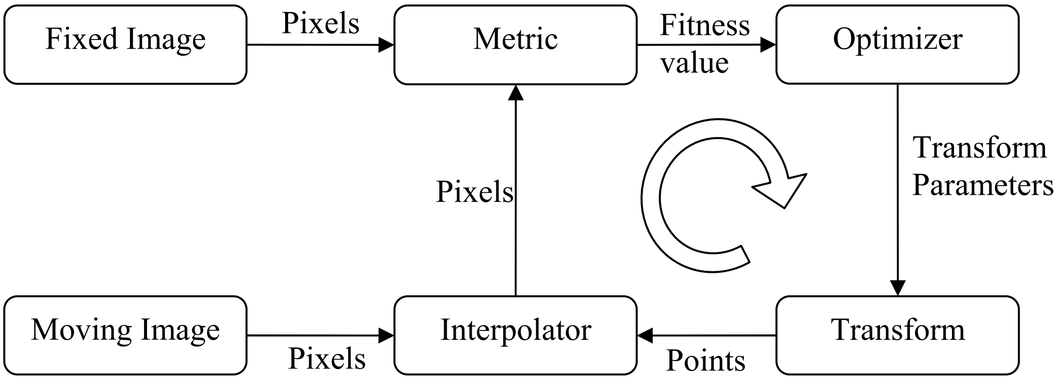


Figure 2.12: Structure of the image registration methods [113]

2.6.1 Local based techniques

2.6.1.1 Mutual Information (MI)

Mutual Information (MI) was originally introduced by Shannon [114] in 1948. There have been lots of applications for the MI, including basic statistics, communications theory, and complexity analysis [115]. Viola and Collignon separately used the MI to measure pixels similarities in 1995. They used the MI as a similarity metric to register multi-modal medical images such as computed tomography (CT) and magnetic resonance (MR) images [116].

Mutual Information depends on the entropy and the joint entropy of two random variables. In the context of the correspondences, the image pixels will form the random variables. Pixel values X can be considered as discrete random values with discrete density P so that the entropy H can be defined as follows:

$$\mathrm{H(X)} = -\mathrm{Ex}[\log(\mathrm{P(X)})] \tag{2.9}$$

The joint entropy is defined in a similar manner for two random variables X and Y by using the joint probability function P(X, Y). Mutual Information can be defined as follows:

$$\mathrm{MI}(\mathrm{X},\mathrm{Y}) = \mathrm{E}_{\mathrm{X},\mathrm{Y}}\left[\log\left(\frac{\mathrm{P}(\mathrm{X},\mathrm{Y})}{\mathrm{P}(\mathrm{X})\mathrm{P}(\mathrm{Y})}\right)\right] = \mathrm{H}(\mathrm{X}) + \mathrm{H}(\mathrm{Y}) - \mathrm{H}(\mathrm{X},\mathrm{Y}) \tag{2.10}$$

The minimum value of the Mutual information is 0 which means that X and Y are completely independent, where the maximum value of the MI is equal to $\min\left(\mathrm{MI}(\mathrm{X},\mathrm{X}), \mathrm{MI}(\mathrm{Y},\mathrm{Y})\right)$ which occur when both variables are identical or there is a one to one mapping T between both of them, this is because $\mathrm{MI}(\mathrm{X},\mathrm{X}) = \mathrm{MI}\big(\mathrm{X},\mathrm{T}(\mathrm{X})\big)$ and this justifies why the MI is a good similarity metric.

Mutual Information works well with different lightning conditions, even while using two sensors with different spectral responses. However, it cannot measure similarities between two regions if there is a considerable amount of Gaussian noise in one of the samples. Moreover Mutual Information is computationally expensive if compared with the typical convolution since it includes a step to estimate the probability density function of the variables.

2.6.2 Feature based techniques

2.6.2.1 SIFT

Scale Invariant Feature Transform (SIFT), introduced by David Lowe [117], is a very robust method to match features in different images with high confidence. SIFT can perform very well against affine distorted images with added noise and some differences in illumination. SIFT features exist in spatial and frequency domains which give them strong immunity from clutter, noise and occlusion. Large numbers of SIFT features can be extracted from typical image, they are highly distinctive which makes them suitable for many applications ranging from 3D reconstruction, stereo correspondence, motion tracking, image indexing and mosaicing to list but a few.

Generating SIFT features can be divided to four filtering-stages where the expensive operations are applied on locations after they pass an initial test. Lowe describes the main steps to extract image features in [118] with the following steps:

a. Scale-space extrema detection

In the first stage, keypoints are detected in all locations that are invariant to scale. This is achieved by searching all possible scales using the scale space function. Lindeberg shows in [119] that this function cannot be other than the Gaussian function. Therefore, detecting keypoints can be implemented efficiently by searching for the scale-space extrema in the DoG (difference-of-Gaussian). In other words, the image is convolved with the Gaussian kernel repeatedly with different σ (see equations (2.11), (2.12)). The first time the convolution is done with σ and the second time with kσ, and the third with $k^2\sigma$ and so on until we have doubled the σ (this will be called one octave). Then the produced images will be clustered in pairs and subtracted from each other to generate the DoGs, each DoG represents one scale in the scale space. Before going to the next octave, down sampling is performed on the last produced image (in the convolution process) with a factor of 2, then the same procedure is repeated again.

If we define the convolved image with Gaussian kernel with the equation:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.11}$$

Where $I(x, y)$ is the input image and:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.12}$$

Then we can easily get the images in the scale space with the equation:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.13}$$

Searching for the extrema in the scale space is performed by comparing each pixel in the DoG to its 26 neighbours (that is 8 pixels in the same scale, 9 pixels in the up-scale and 9 pixels in the down-scale). The pixel will be considered as a candidate only if it is smaller than all of them or larger than all of them.

There are two important parameters to be determined in this stage, the frequency of sampling in the scale domain and the frequency of sampling in the spatial domain. In the scale domain, Lowe shows in [118] that the best results can be obtained by sampling 3 scales per octave. While his study in the spatial domain suggests that the larger the σ the better the performance, but this comes at the cost of increased computational operations. $\sigma = 1.6$ can be considered as a trade-off between performance and efficiency.

b. Keypoint localization

There are three goals for this stage; the first one is to determine accurately the sub-pixel locus of the extrema in the scale space, the second one is to reject all keypoints with low contrast, and the third is to reject points along edges.

The first implementation of SIFT suggested that keypoints are positioned in the centre of the sample point, Brown in [120] developed a method to calculate the interpolated locus of the extrema using Taylor expansion of the scale-space function $D(x, y, \sigma)$.

$$D(x, y, \sigma) = D + \frac{\partial D^T}{\partial x} + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \tag{2.14}$$

The location of the extrema is determined by taking the derivative of $D(x, y, \sigma)$and setting it to zero, the equation yields:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial x^2} \frac{\partial D}{\partial x} \tag{2.15}$$

The derivates are approximated using the differences of the neighbour points, and the $\hat{x}$ value can be calculated. If $\hat{x}$ is larger than 0.5 in any direction then the extrema is positioned in another pixel.

All keypoints with low contrast are sensitive to noise; to reject these keypoints, the function value $\mathrm{D}(\hat{\mathrm{x}})$ is estimated. If this value is less than a fixed threshold then the keypoint will be rejected. To reject keypoints along the edges, we calculate the Hessian matrix for D and get the ratio of its eigenvalues (which is proportional to the squared trace of H divided by its determinant). This ratio should be less than a threshold; otherwise the keypoint will be rejected.

c. Orientation assignment

For each detected keypoint in the scale $\sigma 1$ the magnitude and orientation is calculated for each sample point in the smoothed image $\mathrm{L}(\mathrm{x}, \mathrm{y})$ in the scale $\sigma 1$. Again, the magnitude and the orientation are approximated using the pixel differences around the sample point, they are calculated depending on the following equations:

$$\mathrm{m}(\mathrm{x}, \mathrm{y}) = \sqrt{\big(\mathrm{L}(\mathrm{x}+1, \mathrm{y}) - \mathrm{L}(\mathrm{x}-1, \mathrm{y})\big)^2 + \big(\mathrm{L}(\mathrm{x}, \mathrm{y}+1) - \mathrm{L}(\mathrm{x}, \mathrm{y}-1)\big)^2} \tag{2.16}$$

$$\theta(\mathrm{x}, \mathrm{y}) = \tan^{-1}\left(\frac{\big(\mathrm{L}(\mathrm{x}, \mathrm{y}+1) - \mathrm{L}(\mathrm{x}, \mathrm{y}-1)\big)}{\big(\mathrm{L}(\mathrm{x}+1, \mathrm{y}) - \mathrm{L}(\mathrm{x}-1, \mathrm{y})\big)}\right) \tag{2.17}$$

Each calculated orientation around the keypoint is weighted with its gradient magnitude and by a Gaussian weighted circular with $\sigma 2 = 1.5 * \sigma 1$ the result is inserted to 36-bins histogram covering the 360 degrees. All orientations within 80% of the highest peak in the histogram are used to create a keypoint corresponding to that orientation.

d. Keypoint descriptor

The keypoint descriptor is formed from 16x16 sample points around the keypoint, the orientations and gradients for the entire sample points are calculated and inserted to 4x4 regions descriptor. In each region an 8-bin histogram is built to accommodate all orientations in that region. Each orientation (or bin) in the histogram is weighted by the sum of the gradient magnitudes within that orientation. The SIFT descriptor will be 4x4x8 = 128 elements. Figure 2.13 shows the keypoint descriptor as illustrated in [118].

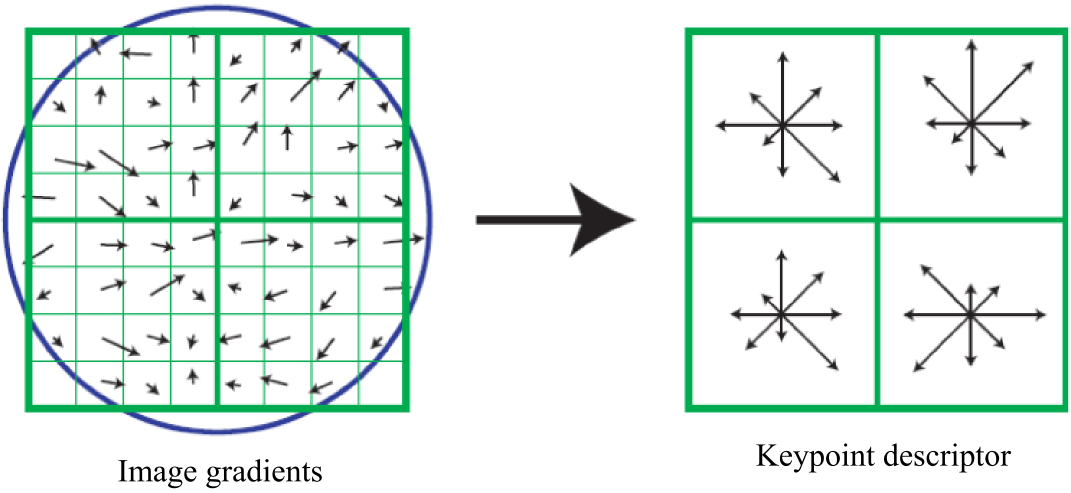


Figure 2.13: Keypoint descriptor as depicted in [118].

Figure 2.14 shows the results of matching two images using SIFT-128 (based on Lowe implementation), it is clear that SIFT doesn't suit acoustic images here as the number of correct matches is very small, we will discuss these results in details in Chapter 6.

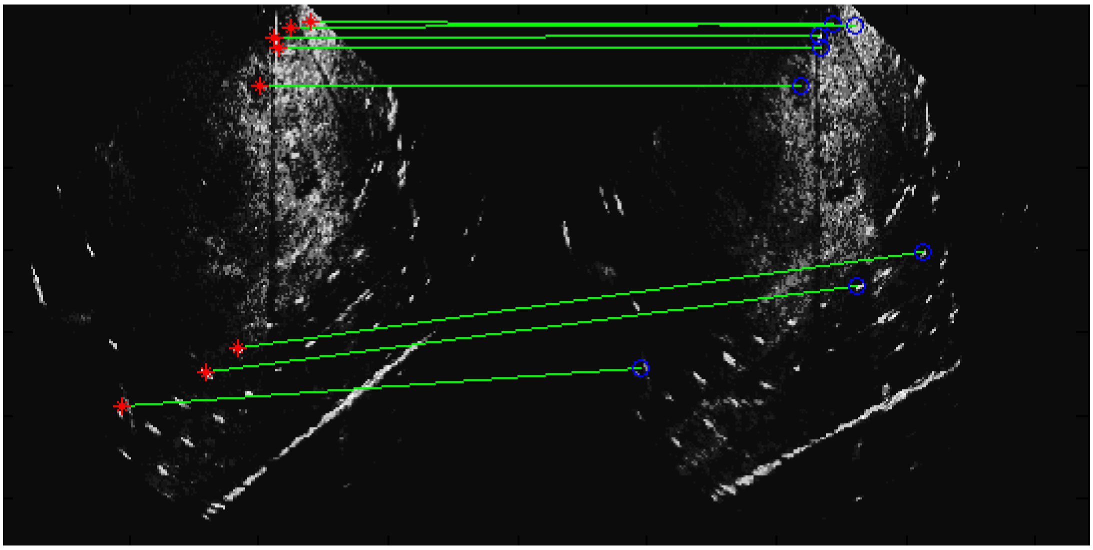


Figure 2.14: Matching two BlueView images using SIFT-128

2.6.2.2 Speeded Up Robust Features (SURF)

SURF was introduced by Herbert Bay and his colleagues in [121] as a new detector-descriptor scheme. The SURF detector is defined based on the Hessian matrix which is approximated by using box filters and integral images. Due to this approximation, the scale space can be produced by applying box filters with different sizes, rather than applying Gaussian smoothing repeatedly. Interest points are detected by applying non-maximum suppression in the produced scale space.

SURF descriptor is generated in two main steps. First of all, the interest point orientation is estimated, this is done by calculating the wavelet responses in the neighbourhood of the interest point and weighting this responses with a Gaussian mask centred at the interest point, these responses are represented with vectors in the x and y directions and they are used to estimate the dominate orientation in a sliding window covering $\frac{\pi}{3}$. The interest point will be associated with the most dominate orientation among all windows. The second step is to form the SURF descriptor; this is implemented by taking a square region in the centre of the interest point and aligning it with its orientation, then the square is divided to 4x4 sub-regions. The wavelet responses in the x and y directions are recalled within each sub-region, then they are weighted with a Gaussian mask. For each sub-region a vector descriptor is formed by summing the responses in the x and y directions along with their absolute values. This generates four elements for each sub-region so the total elements in the SURF descriptor will be 4x4x4 = 64 elements which is called SURF-64. The length of the SURF descriptor can be varied between 36 elements resulting from 3x3 sub-regions, or it can be 128 elements by summing negative and positive responses separately. However, SURF-64 gives the best results in terms of performance and efficiency.

Figure 2.15 shows two BlueView sonar images before matching, Figure 2.16 shows the results of matching both of them using SURF (based on the Open SURF Library), the algorithm found 27 matches from which there are 12 mismatches; we will discuss these results in details in Chapter 6.

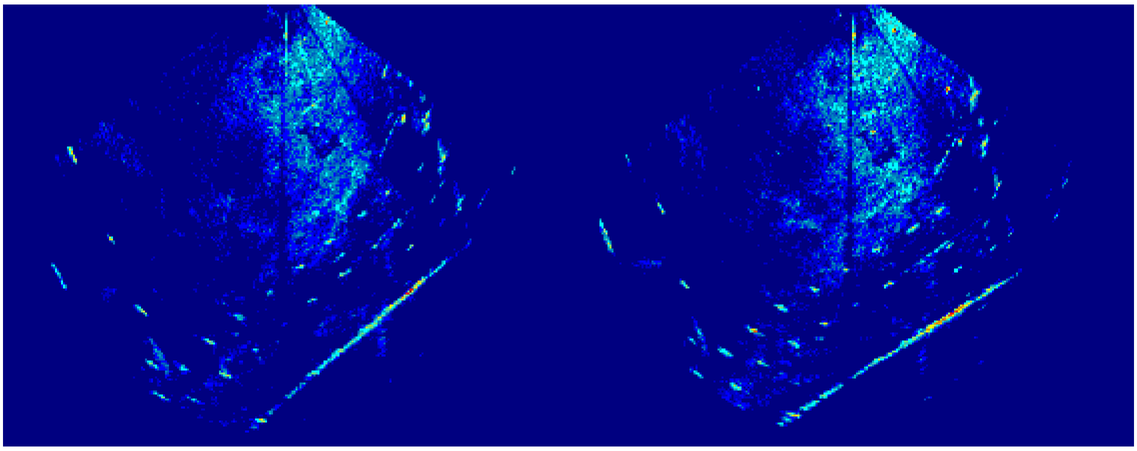


Figure 2.15: Two BlueView images before matching

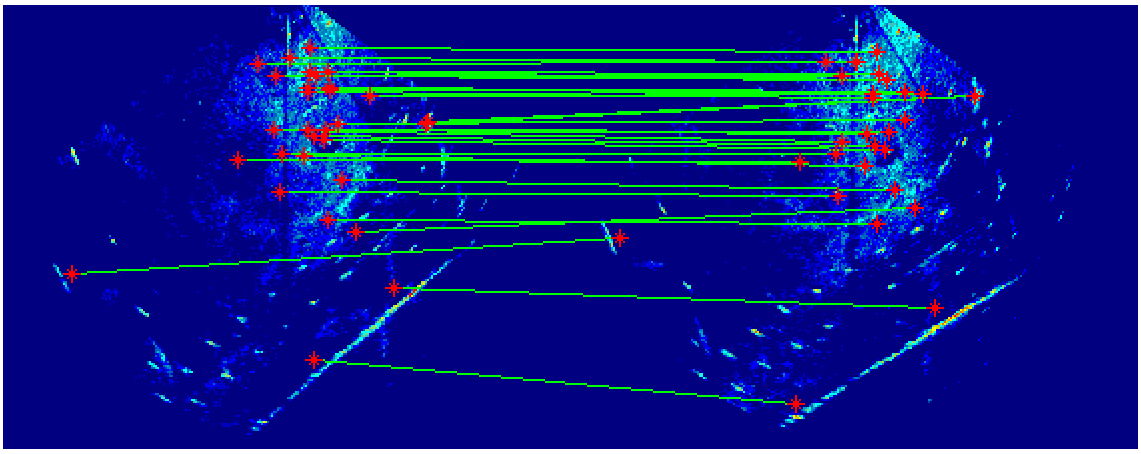


Figure 2.16: Matching two BlueView images using SURF, the algorithm found 27 matches from which there are 12 mismatches

2.6.3 Point matching algorithms

Point matching algorithms can be used to estimate the transform matrix between two sets of points, to accomplish image registration using point matching algorithms, the points are extracted from the images as a candidate features using fast and simple procedures, then point matching algorithms search for the best transform matrix to fit the first set into the other. We may categorize point matching algorithms in three groups; spectral, Iterative, and probabilistic methods.

Spectral methods are non iterative methods which depend on the eigenvector analysis of the proximity matrix, Scott and Longuet in [122] introduced one of the most cited algorithms in this category, their algorithm works well with translation, scaling and shearing; however, it fails rapidly with rotation and affine transformations. In [123], Li and Hartley introduced another spectral method based on Newton-Schulz factorization of the proximity matrix, Xiabi et al presented an algorithm in [124] based on Scott and Longuet framework but used Hausdorff distance between shape context patterns to build the proximity matrix. Practically, spectral methods suffer from large computational burden especially with large sets of points; moreover they require a rich proximity matrix where useful features can be extracted to solve for correspondences.

Iterative methods can be all grouped under the ICP (Iterative Closest Point) umbrella, the ICP algorithm presented by Besl and Mckay [125] is a standard algorithm in this category, it iteratively pairs correspondences based on the closest distance criterion and computes a rigid/affine transformation between the point sets in least-squares terms, when it reaches a local minima, it stops to provide the calculated transformation. Copious variations have been presented to enhance the ICP performance as in [126] and [127]. In general, ICP methods do not guarantee convergence to the right solution even with noiseless data, furthermore, they require the initial position of the point sets to be sufficiently close [128].

Probabilistic methods extend the binary correspondence assignments in ICP to the soft-assignment of correspondences based on some probability, Robust Point Matching (RPM) algorithm presented by Gold et al. [129] is an important algorithm in this category, it uses Levenberg-Marquardt optimizer to minimize the registration error, the algorithm has many descendants as in [130], [131], other methods use the EM (Expectation maximization) algorithm to optimize the likelihood function as in [132], [133],[134] and [135]. Generally speaking, probabilistic methods outperform conventional ICP especially with noisy inputs. Next we review in details RANSAC algorithm which is one of the most used point matching algorithms in computer vision literature.

2.6.3.1 Random Sample and Consensus (RANSAC)

RANSAC can be used to find the transformation between two related images. Usually, features' points are extracted from both images to form a reference point set from the reference image and sensed point set from the sensed (second) image [136]. RANSAC randomly selects correspondences to determine a hypothesized transformation T. The correctness of this transformation can be inferred depending on how well it transforms the sensed point set to the reference point set. If it doesn't transform enough points from the sensed point set to the reference point set, other random correspondences are selected to test another transformation. A point P in the sensed point set is considered to be transformed to the reference point set if the point $P' = TP$ falls within a small distance ε from another point in the reference point set.

Depending on the hypothesized transformation type, the required number of correspondences varies. If the transformation is translation only, one pair of correspondence points is enough to infer the transformation. If the transformation is rotation and translation, two pairs of correspondences is required. The affine transformation contains six parameters thus it requires three pairs of points to be inferred from six equations obtained from their 2D coordinates.

RANSAC performance is affected by three parameters: the maximum number of iterations N to select random correspondences, the distance tolerance ε to consider a point in the sensed point set is transformed to the reference point set, and the threshold r to consider that the transformation has transformed enough points from the sensed to the reference point sets. When N is small, the likelihood to find the correct transformation is small. If N is large, the algorithm requires more time to finish, but the likelihood to find the correct transformation is higher. The distance tolerance ε should reflect the noise in the points coordinates, so ε must be high if the noise is high and vice-versa. Similarly, the threshold r should reflect the inliers ratio in the point sets. Next, we summarise the algorithm steps after defining the used notation:

Notation:

- m: the number of points in the reference point set P'.
- n: the number of points in the sensed point set P.
- T: is the transformation which maps the sensed point set to the reference point set, p is the number of parameters in this transformation.
- ε is the distance tolerance to consider two points corresponding to each other, i.e. the distance between the transformed sensed point and its corresponding

point should be less than ε. ε should be less than half of the smallest distance between two points in the reference point set.

- r : is the minimum ratio of number of correspondences found and the number of points in the sensed set to consider the transformation correct.
- N is the maximum number of iterations.
- n_i the iteration number.
- n_c the largest number of correspondences obtained up to the current iteration n_i.

RANSAC algorithm can be outlined in the following steps:

1) Initialize $n_i = 0$ and $n_c = 0$
2) Select a random subset from each point set, the subset size is $q = p/2$.
3) For each possible correspondence arrangement between these subsets, do the following steps:
    a) Determine the transformation T_i from the q correspondences selected in step 2.
    b) Transform all sensed points P to the reference point set, if $\|P'_l - T_i P_k\| \leq \varepsilon$ then P'_l is the correspondence point for P_k where $(l = 1 \dots m\,; k = 1 \dots n)$
    c) If the number of true correspondences is t and $t > n_c$ then assign $n_c = t$ and save the obtained transformation $T_b = T_i$ as the best transformation so far. If $n_c/n \geq r$ then go to set 5.
4) Increase n_i by one and loop to step 2 if $n_i < N$
5) Use all pair points that satisfy the condition: $\|P_l - T_b P_k\| \leq \varepsilon$ to infer new transformation T_s by the least-squares method like SVD.
6) Return T_s as a solution and determine correspondences by using it to transform the sensed point set to the reference point set.

If the transformation between the sensed set and the reference set is the similarity transformation, then the number of selected pairs is two. Using these pairs, four equations can be inferred to determine four parameters in the supposed similarity transformation. In this case, the number of average iterations required to select two pairs is on the order of n^2. It becomes on the order of n^3 if the transformation is affine as three pairs are required to solve for six parameters in the affine transformation. Clearly, RANSAC cannot handle large point sets as the required number of iterations becomes very large.

There are lots of approaches to enhance RANSAC performance, for example, randomized RANSAC (R-RANSAC) has been suggested to expedite the performance of the error function [137]. Basically, the supposed transformation is not validated against the whole point set directly, but first it is validated against small random subset of the point set, if the transformation is valid for that subset then it will be validated against the whole subset. The subset size can be changed dynamically while the algorithm is running [138]. This methodology is very useful in rejecting unlikely transformation in a fast approach. It is particularly useful when the point sets are considerably large.

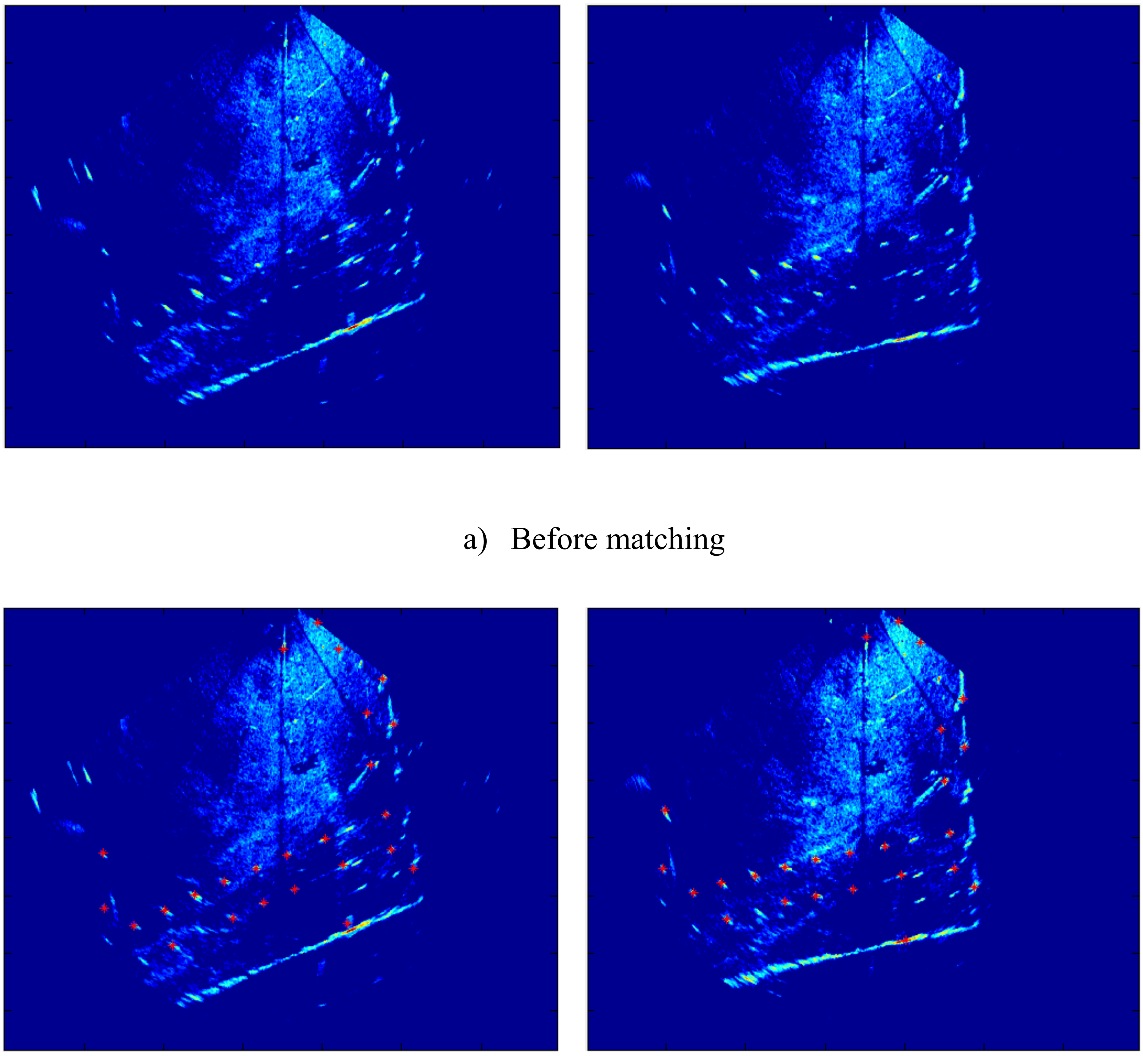
In a similar approach to the R-RANSAC, an early bail-out test has been introduced to enhance the RANSAC performance [139]. The algorithm uses a random subset of the point set to test the supposed transformation, if the percentage of inliers is considerably smaller than the best percentage obtained so far then the transformation is rejected. Again, the used percentage and the size of the subset depend on the size of the original point set and different RANSAC parameters.

Another approach to speed up RANSAC is based on the hierarchical search [140], in its essence, this approach searches for local transformation between two points and their k-nearest neighbours. If the local transformation is good locally then it will be promoted to the next stage of validating it against the whole point set. Optimal k depends on the size of the point sets and the error tolerance distance ε. A similar approach has been suggested based on cluster matching [141].

Preemptive RANSAC has been suggested as well to decrease the computational burden in searching for the optimum transformation [142]. The algorithm suggests predefined transformations and initializes a scoring system to select the best one. Each randomly selected pair of points increases the transformation score if the pair is considered inlier by that transformation. After selecting a large number of random pairs, the transformation with the highest score is selected as the best transformation. The main drawback of this algorithm is the difficulty of suggesting a small number of transformations at the first place.

In another trend to speed up RANSAC, additional features are used to increase the likelihood of selecting true correspondences. This procedure is adapted in the Maximum likelihood estimation sample consensus (MLESAC) [143] and in the guided sampling [144]. The basic principle of these algorithms is to select the pairs with similar features so they are more likely to be true correspondences [145]. This concept is further developed in the Progressive Sample Consensus (PROSAC) by selecting a small subsets with the most similar features and increasing the size of the subset progressively to cover the whole point set [146].

The aforementioned methods attempt to reduce the computational burden of RANSAC without solving the problem behind its overwhelming computational load. The main disadvantage of RANSAC is the fact of selecting samples randomly without any kind of sorting or ordering the space of possible solutions. Although this nature adds immunity against outliers and noise to RANSAC, it exponentially increases its load with increasing size of point sets and increasing number of parameters in the sought-after transformation. Furthermore, there is no guarantee in RANSAC for convergence to the right solution as the space of possible solutions cannot be tested completely. These problems are tackled in the new algorithm dubbed *Sorting the Correspondence Space* (SCS) which is presented in Chapter 6, Some of its results are illustrated in Figure 2.17.



a) Before matching

b) After matching

Figure 2.17 : Two acoustic images successfully registered using the SCS algorithm presented in Chapter 6, there is no mismatch as the algorithm search for the best transformation.

2.7 Conclusion

In this chapter, we presented various methods of underwater 3D imaging and discussed their weak and strong points; also, we have reviewed state of the art 3D sonars and discussed their technologies. Motion estimation for AUVs was reviewed from two main perspectives, firstly, motion estimation sensors serving the AUV industry, including the inertial navigational units, Doppler Velocity Log (DVL), and acoustic beacon technologies. Secondly, motion estimation algorithms such as Kalman filters and particle filters. Image registration is a prerequisite to accomplish both 3D reconstruction and motion estimation using successive sonar views, thus the important methodologies for image registration was examined, including RANSAC, SIFT and SURF.

Chapter 3
Acoustic Stereo Imaging (ASI) System

3.1 Introduction

3D Reconstruction can be achieved using two multibeam sonars using the ASI vertical configuration. In the ASI vertical configuration the sonars are situated above one another with known distance between them (D). The obvious advantage of this configuration is that pairs of beams from the two sonars are aligned in the same vertical plane, this makes the ASI geometry suitable to solve the correspondence problem incurred in classical stereo systems.

In this chapter, our goal is to derive the ASI geometry for the ASI vertical configuration and to study the effects of the system parameters on the 3D reconstruction of the scanned scene. In the vertical configuration the well known correspondence problem is alleviated significantly compared with the correspondence problem in the optic-acoustic configuration discussed in the section 2.3.3. Figure 3.1 illustrates how to calculate the elevation E for a suggested point P after being matched in both sonar images.

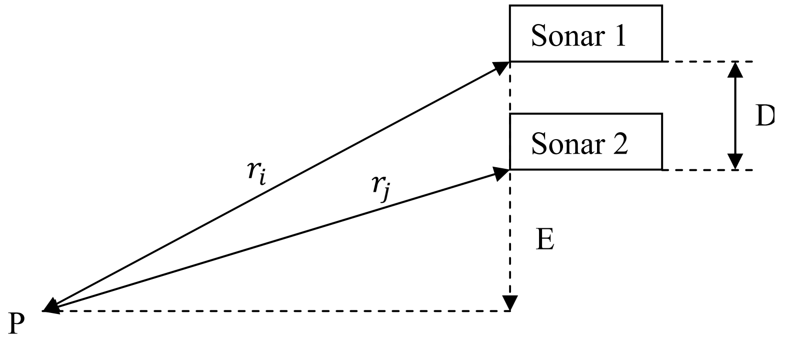


Figure 3.1: ASI sketch

Andrea Trucco in [147] has proposed a similar method using just one forward-scan sonar installed on a ship, the method uses the well known ship positions to calculate the elevation of a tracked feature using simple triangulation. Kalman filter was used for feature tracking in the sequence of sonar images. This method is too sensitive to the errors while determining the ship's position which is obtained from the GPS. As we can determine the position up to an acceptable resolution using the GPS, this method gives satisfactory results. Unfortunately, we cannot determine the AUV position with the same accuracy as the GPS signal doesn't penetrate underwater environments.

Synchronization problem has to be solved while using two sonars, because these two sonars cannot scan the same Field Of View (FOV) simultaneously without conflict and interference between the two signals transmitted by the first and the second sonar. This problem can be solved by using simple software and/or inexpensive compact hardware which can trigger both sonars in a successive manner. Alternatively, we may have two acoustic cameras scanning the same FOV simultaneously with different frequencies (in this case the shorter range should be considered). Figure 3.2 illustrates the ASI Vertical Configuration and the beams generated by the sonars.

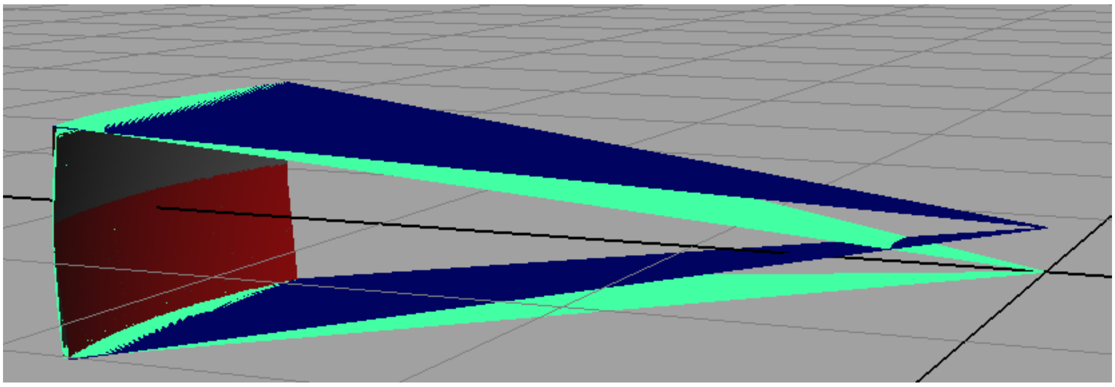


Figure 3.2: The ASI vertical configuration.

ASI vertical configuration resembles the T configuration discussed in the section 2.2.5. In fact, ASI vertical configuration has the same advantages of the T configuration, namely, cheaper implementation; furthermore, it solves the time delay disadvantage incurred in the T configuration as a result of using many transmit/receive cycles to insonify 3D volume. Recall that to get 128x128 points in 100m range by using the T configuration, the system needs to emit 128 successive pulses with 0.13s minimum delay between each pulse, therefore the system needs 17.06 s for complete 3D scan of the scene. In contrast, the ASI system requires 2 transmit/receive cycles only to scan the scene, which means 0.26 s to scan the same scene. (In some cases, it requires 4 transmit/receive cycles depending on the sonar technology and working frequency, which is the case in DIDSON while working in the short range).

3.2 ASI system architecture

The ASI system consists of three main modules: the sonar controller, pairs matcher, and 3D builder. All of these modules are configurable via an XML configuration file. The sonar controller is responsible for triggering the sonars at suitable times and feeding sonar images along with their time stamps to the pairs matcher. Moreover it determines the mode of the sonars (long range or short range).

Basically, the pairs matcher module is an implementation of the matching algorithm tuned to accommodate the ASI geometry. This module feeds the 3D builder module with candidate points to produce the final output of the system which is a 3D model of the scanned scene. The 3D builder utilizes RANSAC functionality or any other filter to fit points in smooth surfaces and reject outliers. Figure 3.3 shows the ASI structure.

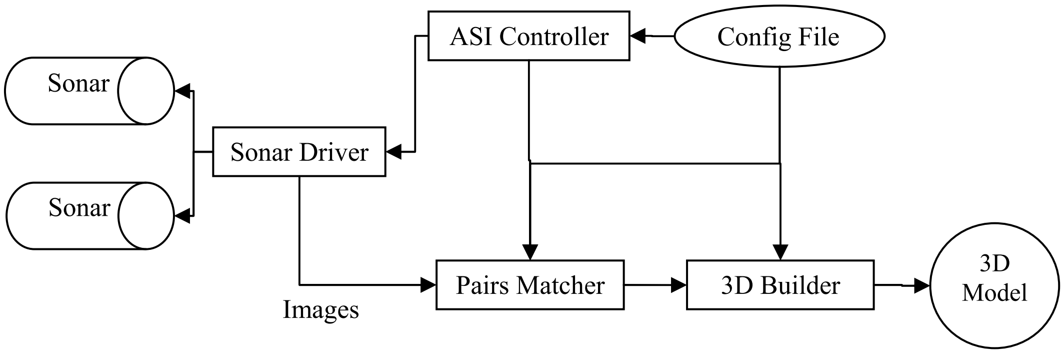


Figure 3.3: ASI structure

3.3 ASI vertical configuration

The vertical configuration suggests two sonars situated above one another and assumes that all beams of the first sonar are aligned to the corresponding beams in the second sonar. The distance between these two sonars is D. This is the simplest configuration which can be used to infer the actual 3D-coordinates of the POI (Point Of Interest) using two acoustic images. Other configurations could be useful as well, like the 90˚

ASI configuration and ASI horizontal configuration (both sonars on the X axis). Figure 3.4 shows the beam alignment in the vertical configuration and the XYZ directions used in this chapter.

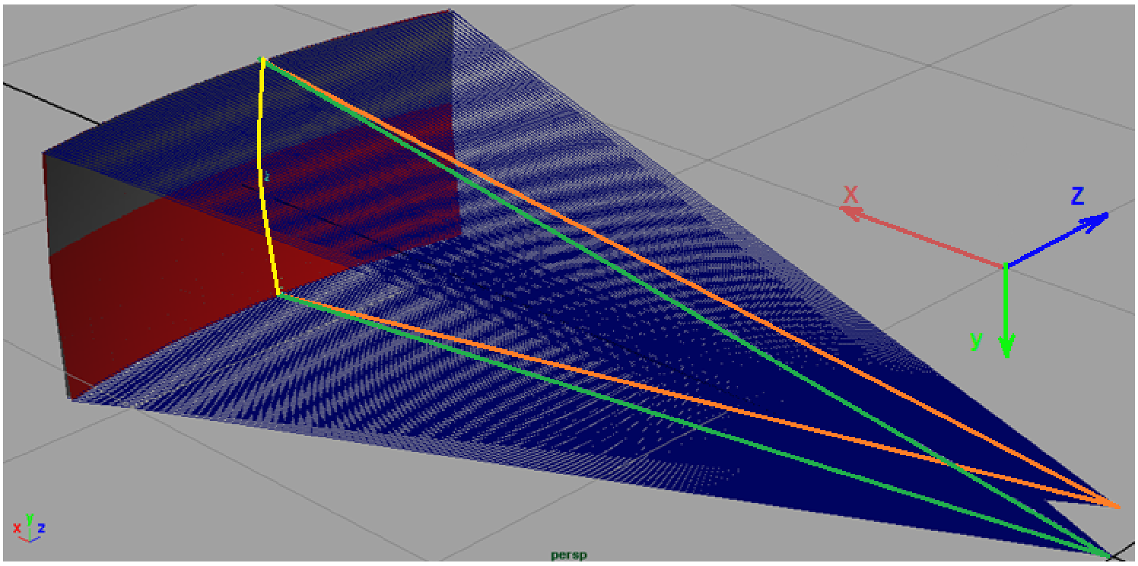


Figure 3.4: Beams alignment in the ASI vertical configuration.

In the ASI vertical configuration we associate local coordinates with each pair of beams, these are 2D-coordinates where the y axis matches the Y axis in the 3D- coordinates and the local x axis forms angle θ with the X axis in the 3D coordinates (the local x axis lies in the X-Z plane). If we know the 2D-coordinates of a POI in the local coordinates, then we can easily infer the 3D-coordinates of that point.

3.4 Notation

In the following sections, we use the following symbols in the local beam coordinate:

- R: The Sonar maximum range in the current mode (R≈14m in the 1.8 MHz mode of the DIDSON sonar [6])
- D: The distance between the two sonars, where the first is situated at the origin and the second is situated at the point (0, D).
- α_1: The angle between the x axis and the centre of the i^{th} beam from the first sonar, see Figure 3.5.

- α_2: The angle between the x axis and the centre of the j^{th} beam from the second sonar ($i = j$ as the beams are aligned).
- 2γ: The horizontal beam width (γ is 0.15˚ for the DIDSON).
- 2β: The vertical beam width (β=7˚ for the DIDSON).
- N: the number of beams in the current working mode (96 beams in 1.8MHz mode of the DIDSON)
- i: the index of the beams in the first sonar.
- j: the index of the beams in the second sonar.

Note that: α1 and α2 are equivalent to the depression/elevation angle for each sonar. In the global coordinates, we use the symbols θi and θj, these are the angles between the local x axis and the global X axis.

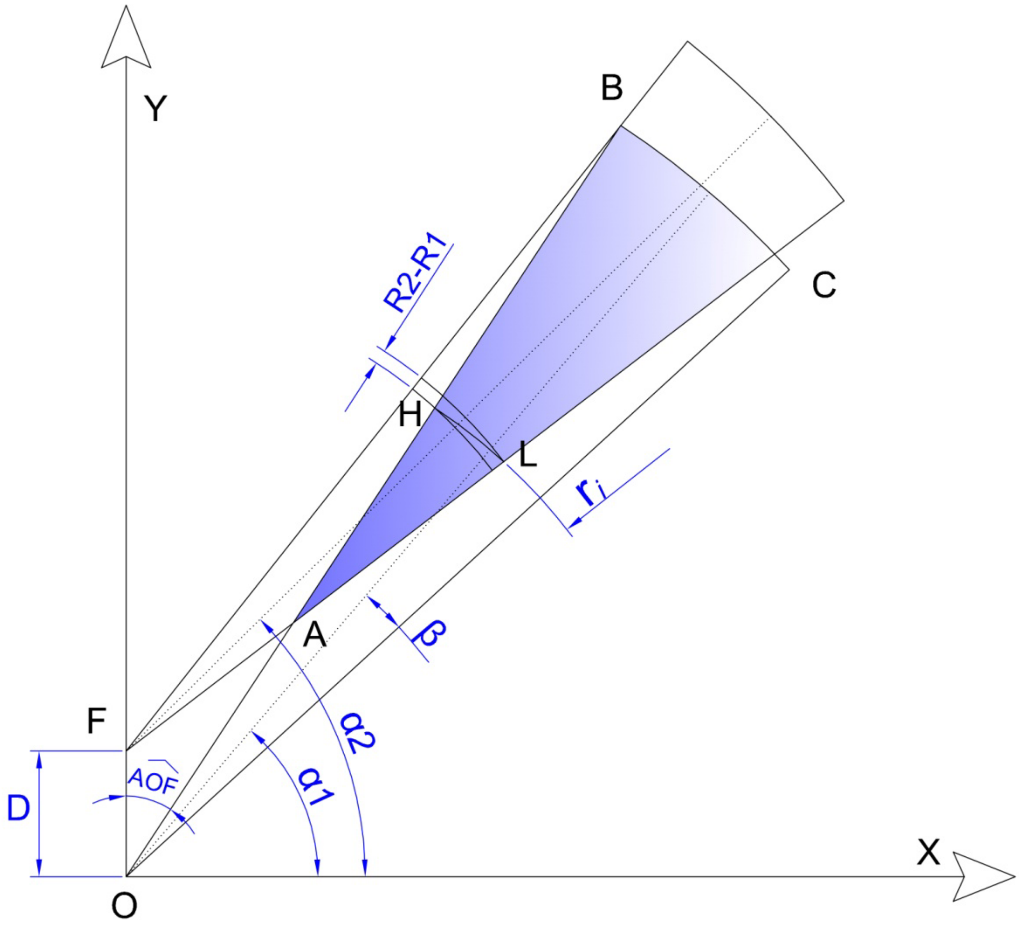


Figure 3.5: ASI system parameters and the FOV Area (FOVA) is shaded in the local beam-pair coordinate system

3.5 ASI field of view

In all stereo systems the FOV is one of the important factors which affects the system performance; the FOV in the stereo system has to be as wide as possible without reducing the resolution. In this section the FOV Area (FOVA) which is the common area between two beams is calculated. The ASI vertical configuration suggested previously is considered here.

Figure 3.5 illustrates the $FOVA(i,j)$ in the ASI system. The $FOVA(i,j)$ corresponds to the intersection of two beams (i,j) from both sonars. Using the aforementioned notation, the whole FOV volume (FOVV) is given in the equation (3.1):

$$\mathrm{FOVV} \approx \sum_{i=j=1}^{N} \mathrm{FOVA}(i,j) * \left(\frac{\mathrm{D}\cos(\alpha 2-\beta)}{\sin(\alpha 1-\alpha 2+2\beta)} + \mathrm{R}\right) * \tan\gamma \tag{3.1}$$

In the equation (3.1), we use the average beam width which can be calculated as follows:

$$\frac{2\mathrm{R}\tan\gamma + 2\,\overline{\mathrm{OA}}\tan\gamma}{2} = \left(\frac{\mathrm{D}\cos(\alpha 2-\beta)}{\sin(\alpha 1-\alpha 2+2\beta)} + \mathrm{R}\right) * \tan\gamma \tag{3.2}$$

Where $2\mathrm{R}\tan\gamma$ is the maximum beam width and $2\,\overline{\mathrm{OA}}\tan\gamma$ is the minimum beam width at range's window-start, see Figure 3.6 .

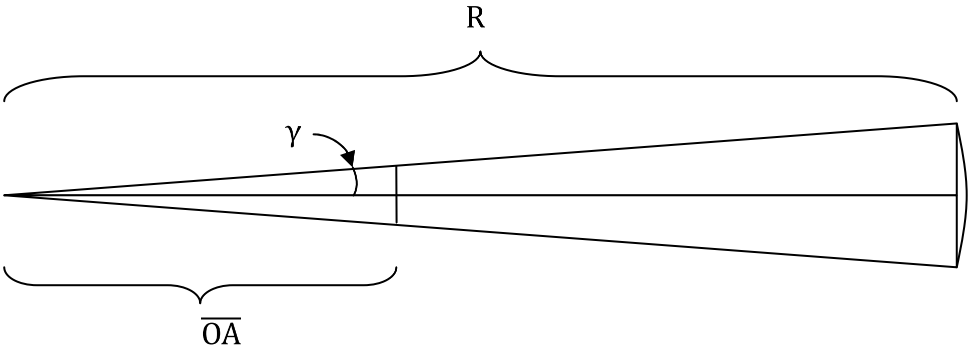


Figure 3.6: The horizontal beam width.

Next, we draw the statement for $\mathrm{FOVA(i,j)}$, from Figure 3.5 we can infer from $\overset{\Delta}{\mathrm{OFA}}$:

$$\overline{\mathrm{OF}} = \mathrm{D}\,;\;\; \widehat{\mathrm{AOF}} = \widehat{\mathrm{O}} = \frac{\pi}{2} - (\alpha 1 + \beta) = \frac{\pi}{2} - \alpha 1 - \beta \tag{3.3}$$

$$\widehat{\mathrm{OFA}} = \widehat{\mathrm{F}} = \frac{\pi}{2} + \alpha 2 - \beta \tag{3.4}$$

$$\widehat{\mathrm{FAO}} = \widehat{\mathrm{A}} = \pi - \widehat{\mathrm{O}} - \widehat{\mathrm{F}} = \alpha 1 - \alpha 2 + 2\,\beta \tag{3.5}$$

$$\overline{\mathrm{OA}} = \frac{\overline{\mathrm{OF}}\sin\widehat{\mathrm{F}}}{\sin\widehat{\mathrm{A}}} = \frac{\mathrm{D}\sin(\frac{\pi}{2} + \alpha 2 - \beta)}{\sin(\alpha 1 - \alpha 2 + 2\beta)} \tag{3.6}$$

$$\overline{\mathrm{AB}} = \mathrm{R} - \overline{\mathrm{OA}} = \mathrm{R} - \frac{\mathrm{D}.\cos(\alpha 2 - \beta)}{\sin(\alpha 1 - \alpha 2 + 2\beta)} \tag{3.7}$$

Also, from Figure 3.5 we can infer from $\overset{\Delta}{\mathrm{OAC}}$:

$$\overline{\mathrm{OC}} = \mathrm{R} \tag{3.8}$$

$$\overline{\mathrm{OA}} = \frac{\mathrm{D}\cos(\alpha 2 - \beta)}{\sin(\alpha 1 - \alpha 2 + 2\beta)} \tag{3.9}$$

$$\widehat{\mathrm{OAC}} = \hat{\hat{\mathrm{A}}} = \pi - \widehat{\mathrm{OAF}} = \pi - \alpha 1 + 2\alpha 2 - 2\beta \tag{3.10}$$

$$\Rightarrow \quad \sin\widehat{\mathrm{A}} = \sin\hat{\hat{\mathrm{A}}}$$

$$\mathrm{R}^2 = \overline{\mathrm{AC}}^2 + \overline{\mathrm{OA}}^2 - 2\overline{\mathrm{AC}}.\overline{\mathrm{OA}}.\cos\hat{\hat{\mathrm{A}}} \tag{3.11}$$

$$\overline{\mathrm{AC}} = \overline{\mathrm{OA}}.\cos(\hat{\hat{\mathrm{A}}}) \pm \sqrt{\mathrm{R}^2 - \overline{\mathrm{OA}}^2\sin^2(\widehat{\mathrm{A}})} \tag{3.12}$$

$$\overline{\mathrm{AC}} = -\mathrm{D}.\cos(\alpha 2 - \beta).\cot(\alpha 1 - \alpha 2 + 2\beta) + \sqrt{\mathrm{R}^2 - \mathrm{D}^2\cos^2(\alpha 2 - \beta)} \quad (3.13)$$

$$\mathrm{S} = \frac{1}{2}.\overline{\mathrm{AC}}.\overline{\mathrm{AB}}\sin(\widehat{\mathrm{A}}) \approx \mathrm{FOVA(i,j)} \quad (3.14)$$

From (3.7) and (3.13) we can approximate the FOV Area (FOVA) as follows:

$$\mathrm{S} = \frac{1}{2}.\left[\sqrt{\mathrm{R}^2 - \mathrm{D}^2\cos^2(\alpha 2 - \beta)} - \mathrm{D}.\cos(\alpha 2 - \beta).\cot(\alpha 1 - \alpha 2 + 2\beta)\right]. [\mathrm{R}.\sin(\alpha 1 - \alpha 2 + 2\beta) - \mathrm{D}.\cos(\alpha 2 - \beta)] \approx \mathrm{FOVA(i,j)} \quad (3.15)$$

Intuitively, the FOVA is reduced while we increase the distance (D) between the sonars; the shorter the distance between the sonars the more compact the ASI system is. However, reducing the distance affects the system performance as we will show later on. Figure 3.7 illustrate the relationship between the FOVA and the distance between the sonars (D).

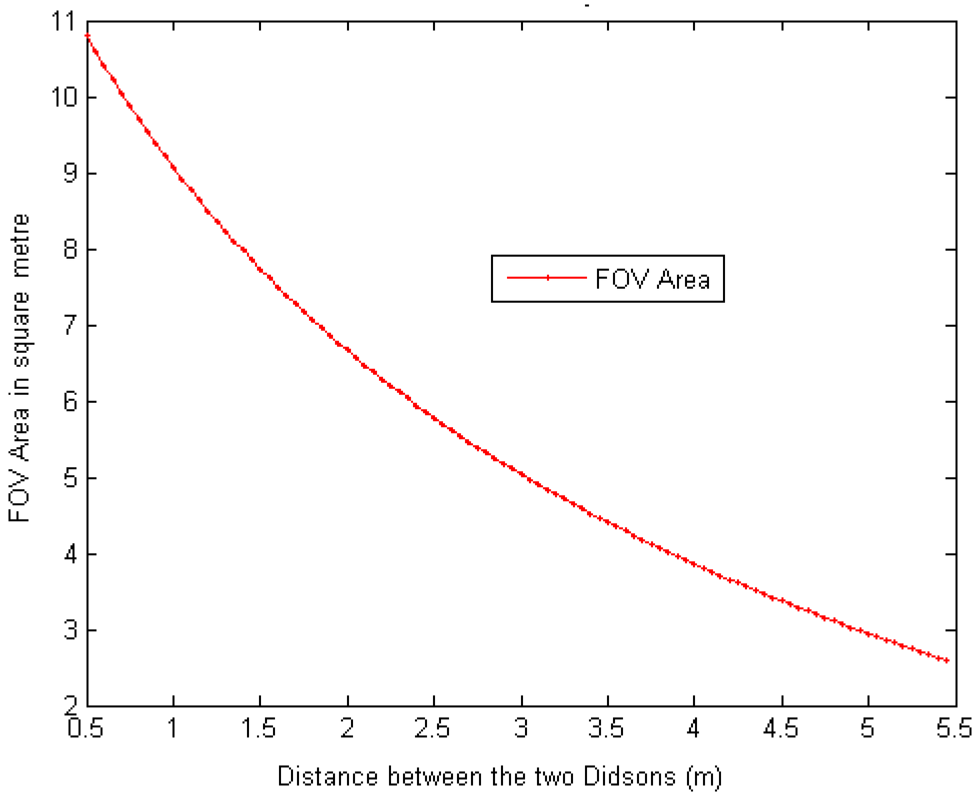


Figure 3.7: The relationship between the $FOVA(i,j)$ and D

3.6 ASI blind areas

All areas where there is no intersection between both beams are considered as blind areas in terms of the 3D imaging. The designer of the ASI system in the vertical configuration should maximize the FOVA and minimize blind areas. In other worlds, we have to determine the ASI system parameters carefully to maximize the FOVA and the system accuracy. The ASI system parameters which affect the FOVA are: D, α1, and α2. Figure 3.8 illustrates the blind areas.

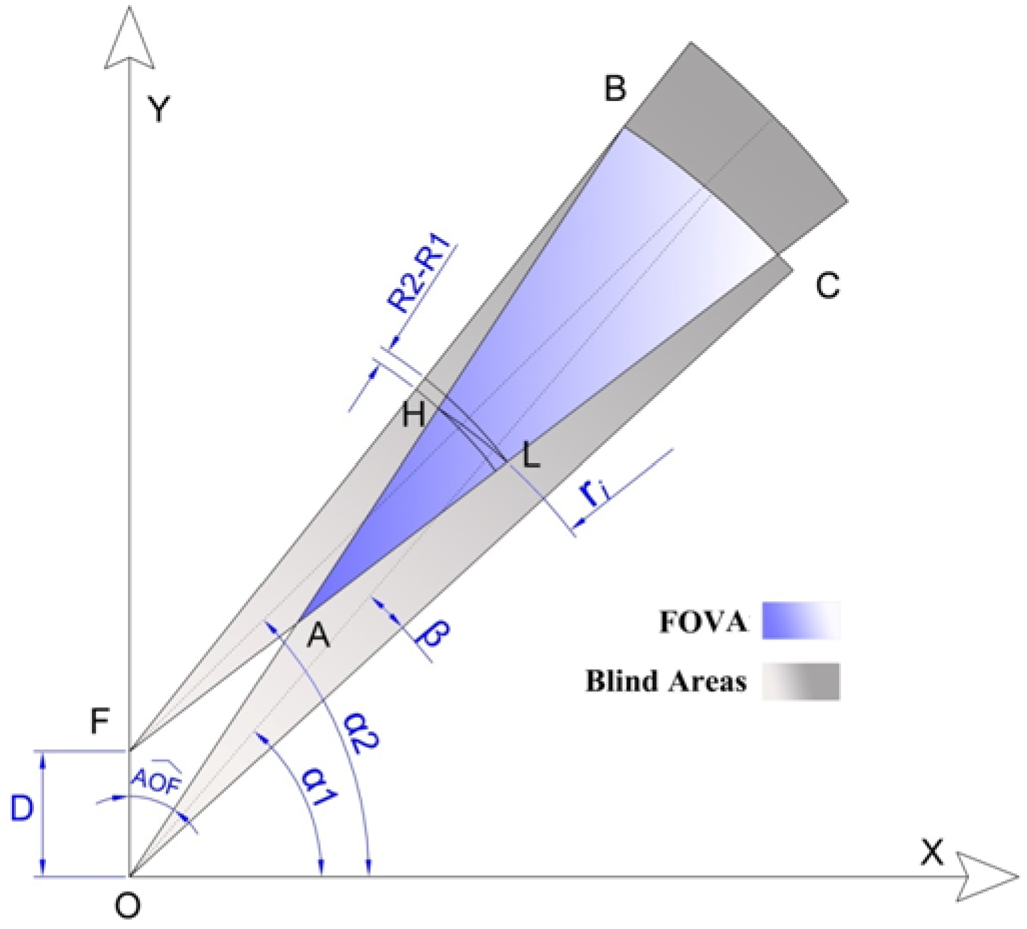


Figure 3.8: ASI blind areas

3.7 ASI geometry

ASI Geometry aims to reduce the search space for correspondences in both sonar images, so each point in the first image has a line segment in the second image where the corresponding point lies.

3.7.1 Sonar to sonar mapping

If we know r_i which is the distance between the first sonar and the point of interest then we want to know the possible positions of the same point in the second sonar image. Figure 3.9 illustrates the possible places of the POI in the second image. r_j is the actual distance between the POI and the second sonar, obviously $R_1 \leq r_j \leq R_2$

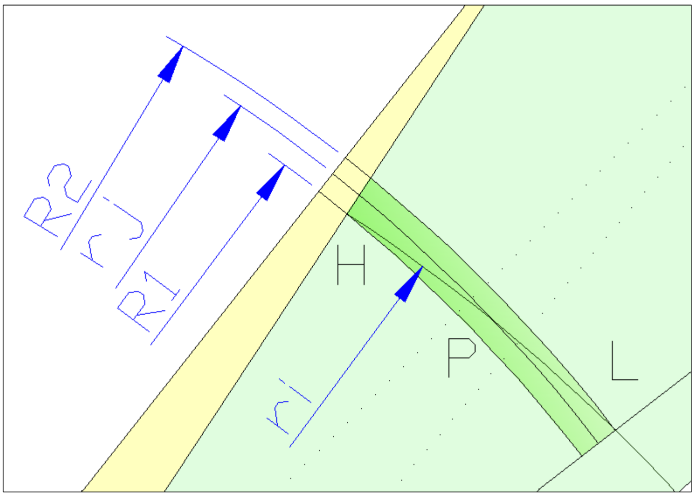


Figure 3.9: R1 and R2 Range

Our goal here is to calculate $\mathrm{R}_1$and R_2 for each POI in the first image. This procedure enables us to determine the possible positions in the second image for all points in the first image within the joint FOV. In this approach searching for correspondences between the two images is a really small search space. First of all, the search benefits from the ASI vertical configuration and assumes that all points in the beam θ_i in the first image correspond to all points in the beam θ_j in the second image. Secondly, for each point in the first image, its correspondence point in the second image lies between R_1 and R_2. We can calculate $\mathrm{R}_1$and R_2as follows:

3.7.1.1 Calculating R1

From Figure 3.5 we can calculate R1 and R2 by applying simple triangulations, from the imaginary triangle $\overset{\Delta}{\mathrm{OFH}}$ we can write

$$\overline{\mathrm{OH}} = \mathrm{r_i} \quad ; \quad \overline{\mathrm{OF}} = \mathrm{D} \quad ; \quad \overline{\mathrm{FH}} = \mathrm{R}_1 \tag{3.16}$$

$$\widehat{\mathrm{HOF}} = \frac{\pi}{2} - \alpha 1 - \beta \tag{3.17}$$

$$R_1^2 = \overline{FH}^2 = r_i^2 + D^2 - 2r_iD\cos(\frac{\pi}{2} - \alpha 1 - \beta) \qquad (3.18)$$

$$R_1 = \sqrt{r_i^2 + D^2 - 2r_iD\sin(\alpha 1 + \beta)} \qquad (3.19)$$

3.7.1.2 Calculating R2

In Figure 3.5 we can infer from the imaginary triangle $\overset{\Delta}{OFL}$:

$$\widehat{LFO} = \frac{\pi}{2} + \alpha 2 - \beta \ \ ; \ \ \overline{OL} = r_i \ \ ; \ \ \overline{OF} = D \ \ ; \ \ \overline{FL} = R_2 \qquad (3.20)$$

$$r_i^2 = R_2^2 + D^2 - 2R_2D\cos\left(\frac{\pi}{2} + \alpha 2 - \beta\right) \qquad (3.21)$$

$$R_2 = \sqrt{r_i^2 - D^2\cos^2(\alpha 2 - \beta)} - D\sin(\alpha 2 - \beta) \qquad (3.22)$$

The difference between R_2 and R_1 specifies the size of the search width which is dependent on the point position, Figure 3.10 illustrates the maximum difference between R_2 and R_1 with change in D. It is clear that the smaller the difference the better the performance of the matching algorithm, since it will search a smaller region.

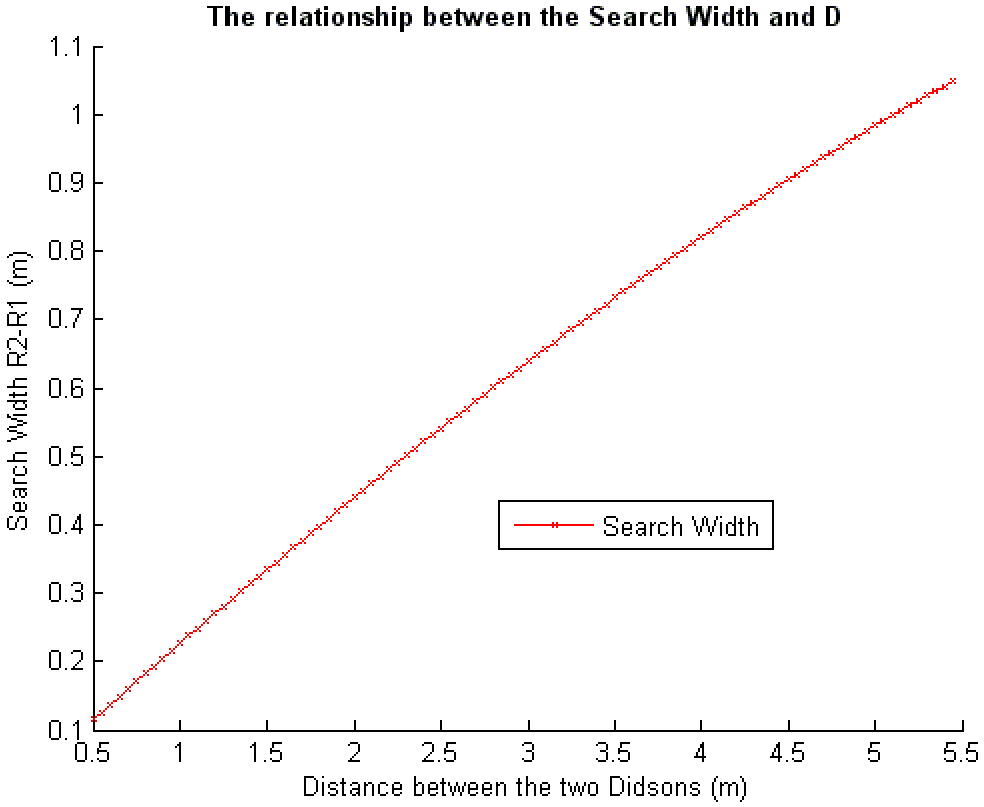


Figure 3.10: The relationship between the search width and the distance D

3.7.2 Determining the point coordinates depending on ri , rj

After searching for r_j in the range $[R_1, R_2]$ then we can calculate the point's coordinates as follows:

$$x_p = r_i \cos(\frac{\pi}{2} - \widehat{POF}) \quad \Rightarrow \quad x_p = r_i \sin(\widehat{POF}) \qquad (3.23)$$

$$y_p = r_i \sin\left(\frac{\pi}{2} - \widehat{POF}\right) \Rightarrow y_p = r_i \cos(\widehat{POF}) \qquad (3.24)$$

$$\cos(\widehat{POF}) = \frac{r_i^2 + D^2 - r_j^2}{2r_i D} \qquad (3.25)$$

$$y_p = \frac{r_i^2 - r_j^2}{2D} + \frac{D}{2} \tag{3.26}$$

$$x_p^2 = r_i^2 - y_p^2 \tag{3.27}$$

$$x_p = \sqrt{r_i^2 - (\frac{r_i^2 + D^2 - r_j^2}{2D})^2} \tag{3.28}$$

So far we have calculated the POI coordinates in the local coordinates associated with the beams in $\theta_i = \theta_j$. The 3D-coordinates of the POI are:

$$X_p = x_p \cos\theta_i \tag{3.29}$$

$$Y_p = y_p \tag{3.30}$$

$$Z_p = x_p \sin\theta_i \tag{3.31}$$

3.8 ASI simulation

An ASI Simulation has been implemented to study the errors generated while reconstructing points from two sonar images. Simulation along with real data shows a good reconstruction using the ASI geometry. The ASI system has to be tuned to satisfy the FOV requirement and the maximum error which can be accepted in each axis.

In the following simulation we have studied the error generated in 3 dimensions while reconstructing random points uniformly distributed over the FOV of the ASI system. In Figure 3.11 the ASI system parameters are D = 0.5m, $\alpha 1 = 19.75°$, $\alpha 2 = 17°$, this figure shows the relationship between the error produced in 3 axes and the range. The maximum Y error is 0.4m. It is clear that the error on the Y axis increases as the range of the points increases.

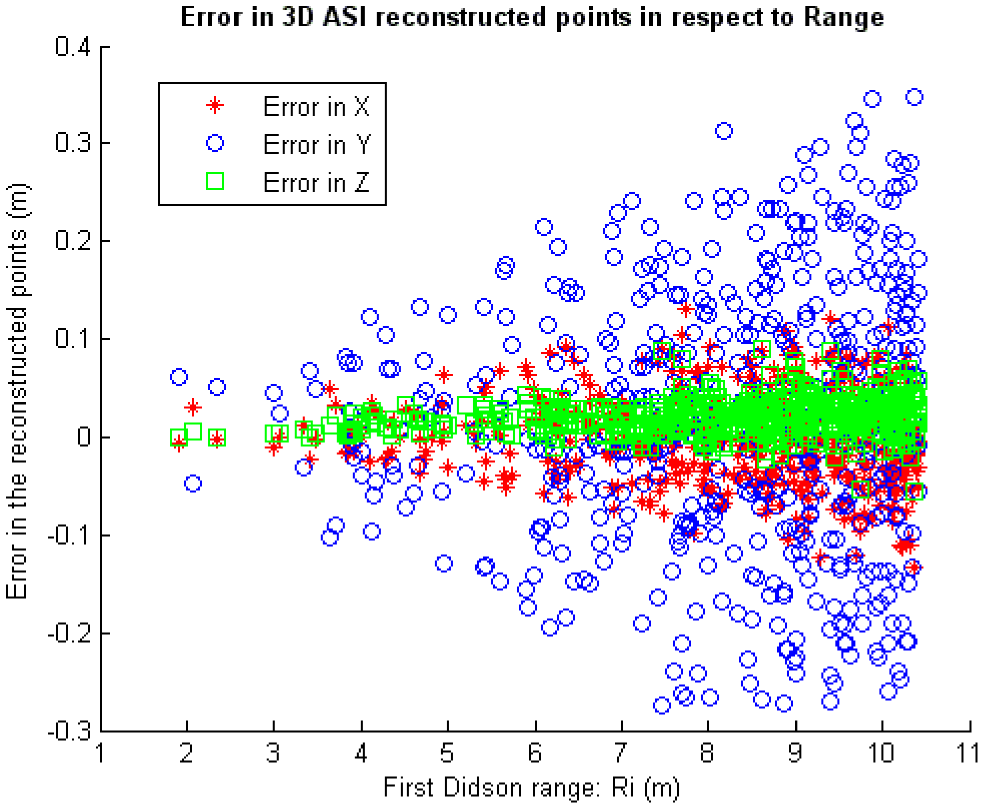


Figure 3.11: The relationship between the error and Range, the figure illustrates the error generated in 3 dimensions while reconstructing random points uniformly distributed over the FOV of the ASI system. Used parameters: D = 0.5m; α1 = 19.75°; and α2 = 17°

By changing the distance between the sonars from 0.5m to 1m the maximum error on the y axis is halved. Figure 3.12 shows the errors in an ASI system similar to the previous one but with distance D = 1m (max Y error is 0.2m). It is worth noticing that the FOVA starts at 3m in Figure 3.12 while it starts at 2m in Figure 3.11 depending on the used angles α1 = 19.75°, α2 = 17°.

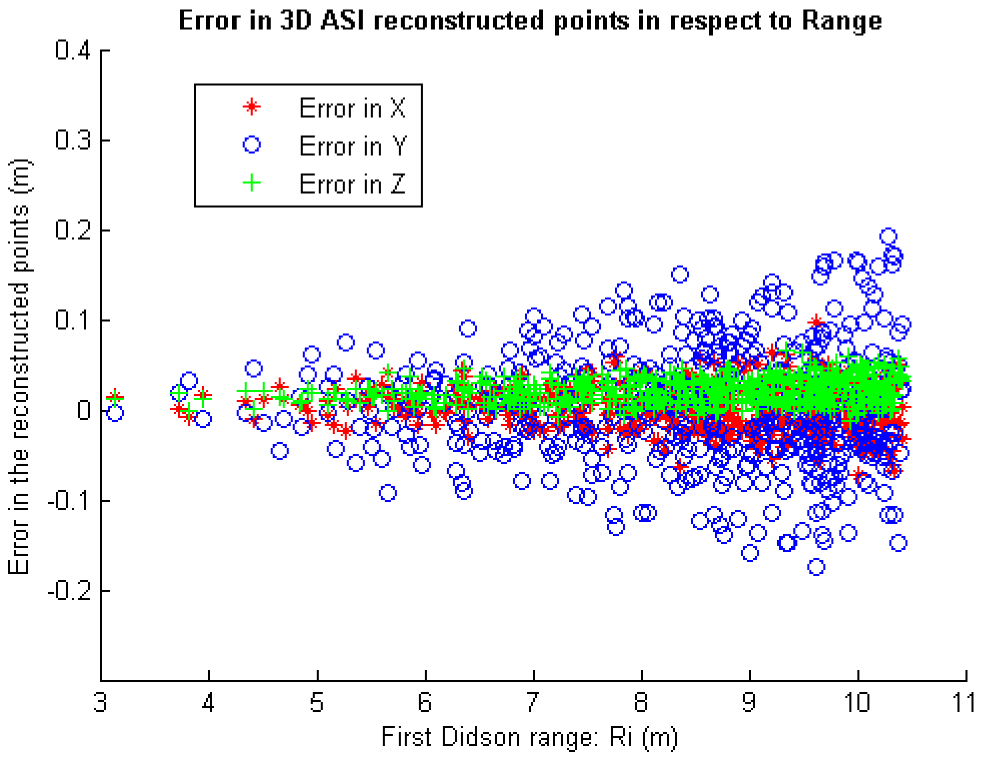


Figure 3.12: The relationship between the error and Range, D = 1m

3.8.1 The distance effect on the ASI system

Figure 3.11 and Figure 3.12 raise a question about the effects of the distance on the system performance. As we have seen earlier, the FOVA decreases while increasing the distance between the two sonars which can be considered as a disadvantage. Here, we would like to investigate the effects of the distance on the error generated while reconstructing points in the FOV. In this simulation we have increased the distance starting from 0.5m to 5.5m, $\alpha 2$ is fixed at 17° while $\alpha 1$ has been changed to maximize the FOVA (so $\alpha 1$ is calculated depending on the current distance). Each time we increase the distance by 0.05m, the simulator generates 1000 points uniformly distributed over the FOV. The reconstructed points are calculated and the errors are calculated on each axis separately. In this experiment we assume that the matching algorithm is 100% accurate. The mean error and the standard deviation of the errors are calculated for each new distance. Figure 3.13 shows the relationship between the distance and the mean error generated in three axes.

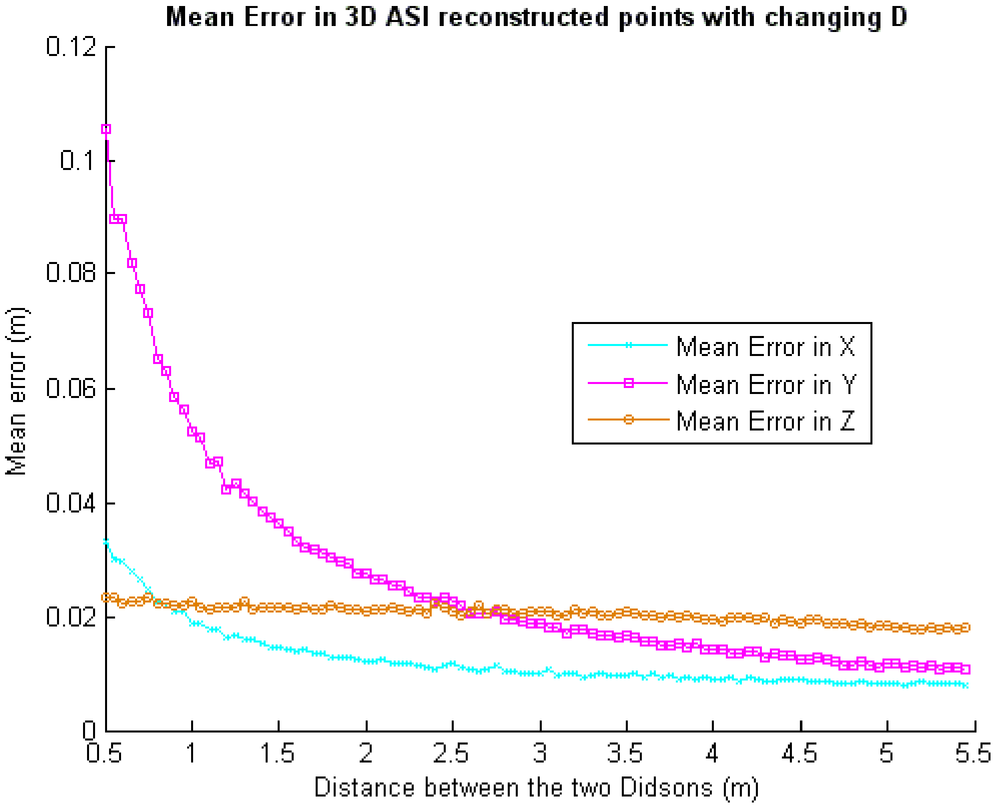


Figure 3.13: Mean Error in 3D reconstructed points with changing D starting from 0.5m to 5.5m. α2 is fixed at 17° while α1 has been changed to maximize the FOVA. The simulator generates 1000 points uniformly distributed over the FOV and the reconstructed points are calculated to find the errors on each axis separately.

As we can see from Figure 3.13, errors generated while calculating Z coordinates are flat and they are not affected by the distance. This is not surprising since Z coordinates can be calculated from just one image and they are only dependent on the horizontal beam width. Errors in X coordinates are acceptable if the distance is more than or equal to 1m. The issue here is the Y coordinates; the Y-mean error curve is very steep in the area between 0.5m and 1m. Increasing the distance will improve the performance in calculating the Y coordinates; however, increasing the distance will decrease the FOVA which is not desirable and it will increase the search width (the difference between R_2 and R_1) which has a negative effect on the matching algorithm. Building the ASI system requires finding the right balance between the FOV, the size of the search region, and the maximum error generated in the reconstructed points. Figure 3.14 shows the standard deviation of the errors in the previous experiment.

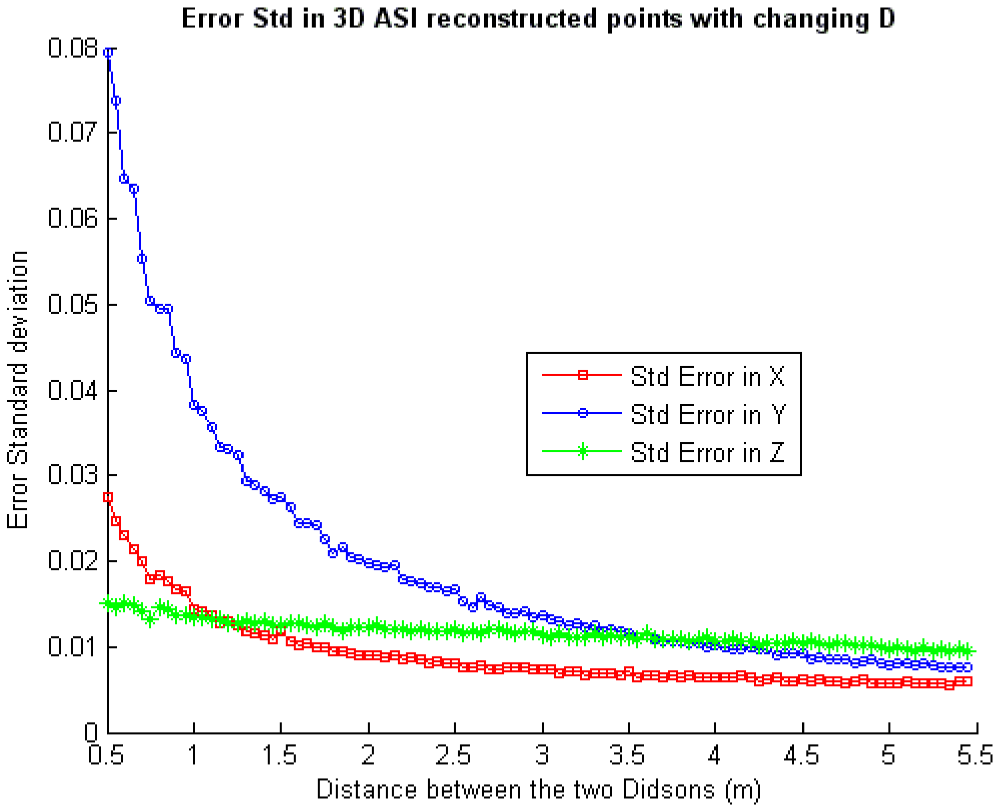


Figure 3.14: Standard deviation of the errors with changing D starting from 0.5m to 5.5m. α2 is fixed at 17° while α1 has been changed to maximize the FOVA. The simulator generates 1000 points uniformly distributed over the FOV and the reconstructed points are calculated to find the errors on each axis separately.

If the matching algorithm is 90% accurate so that 10% of the points are mismatched in the region betweenR_2 and R_1. Then errors in X and Y are significantly increased Figure 3.15 shows the mean error in 3 axes with 10% mismatch.

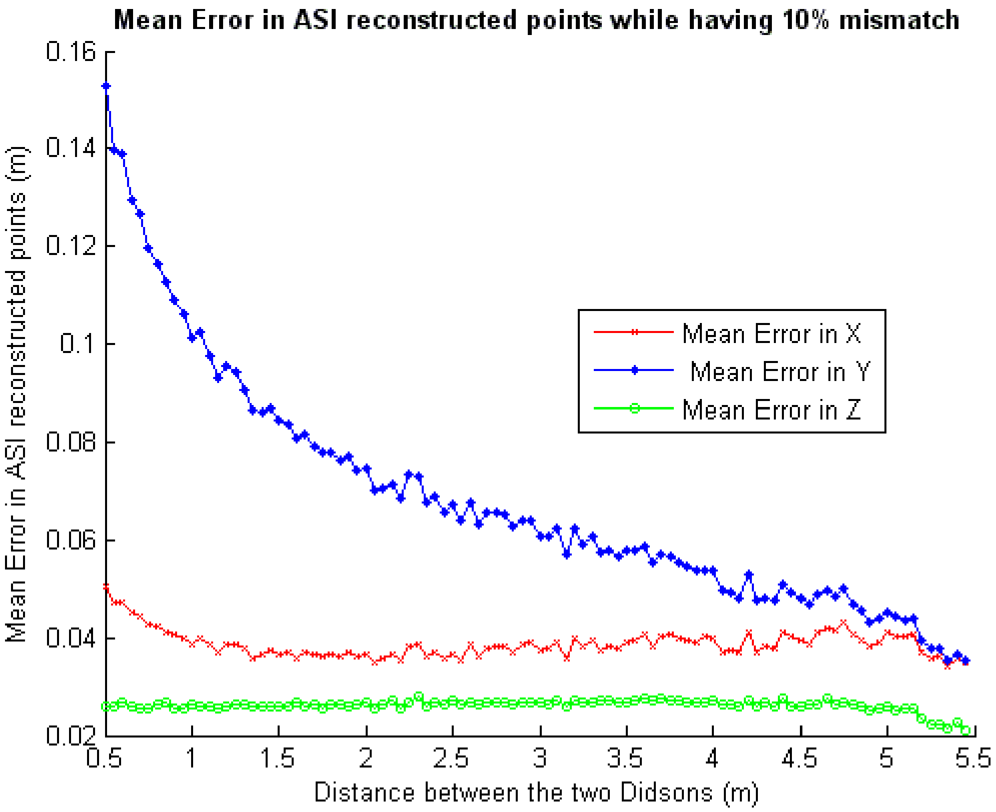


Figure 3.15: Mean Error in ASI reconstructed points while having 10% mismatch

3.8.2 The range bins effect on the ASI system

We have seen in Figure 3.12 the relation between errors in 3 axes and the range. In that simulation, the distance was 1m and the range was quantized with 512 bins as in the DIDSON sonar [6]. Increasing the number of bins will improve the accuracy of measuring r_i and r_j and this will improve the reconstructed Y coordinates significantly. Given the same parameters as the experiment in Figure 3.12 and adjusting the bins number to 1024, the maximum error will be halved compared with the same system but with 512 bins. This effect is similar to increasing the distance between the sonars to gain better performance. In other words, if we have a D = 0.5m ASI system we can obtain the performance of a D = 1m ASI system by doubling the range bins number, yet with bigger FOVA and smaller search width. Figure 3.16 shows the errors in 3 axes if we have the same parameters in Figure 3.12 but with doubled range bins number. If we have 100% accurate measurement of ri and rj then no errors will be produced in calculating Y coordinates and all errors in Z coordinates will be positive and all errors in X coordinates will be negative. Figure 3.17 shows this case.

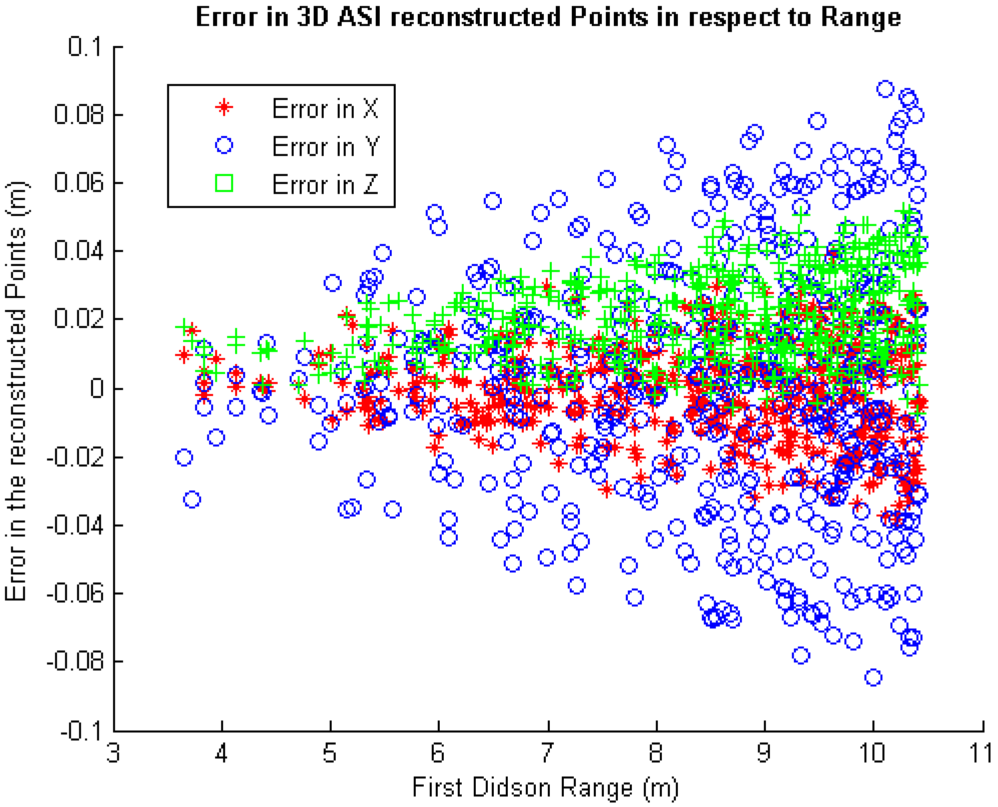


Figure 3.16: Errors will be halved if we double the Range Bins – see Figure 3.12

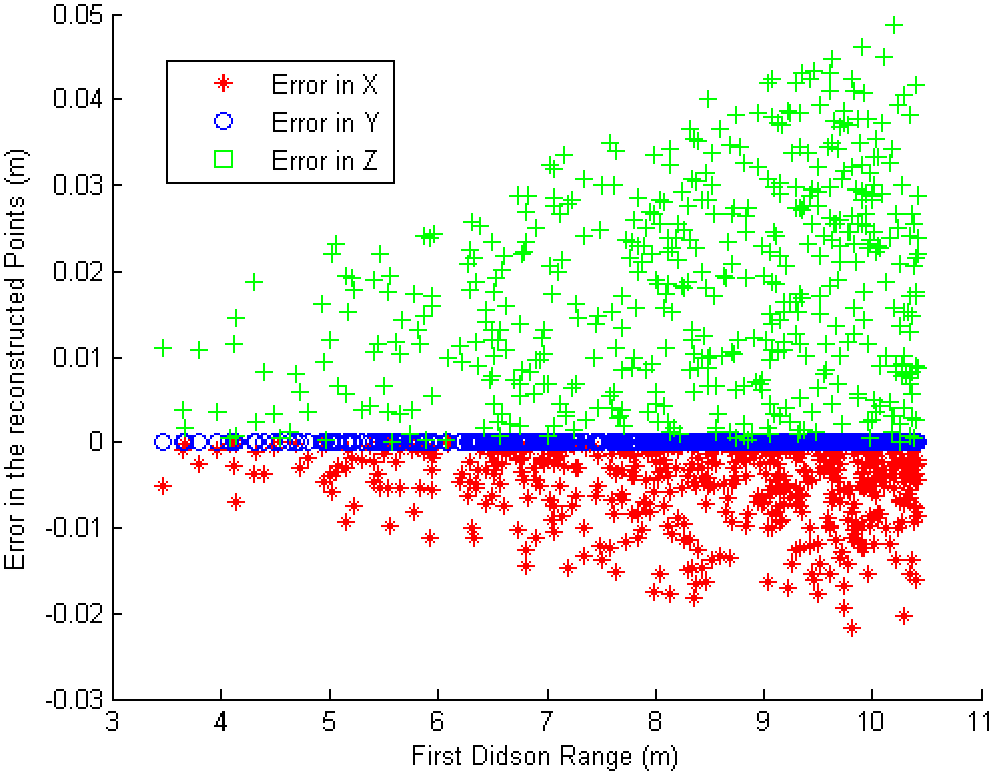


Figure 3.17: Y coordinates will be accurate if we have accurate measures for r_i and r_j

3.9 Results

We have conducted real experiments in an indoor tank in Heriot-Watt University, Edinburgh, UK, to prove the validity of the ASI Geometry and to show the feasibility of building a complete ASI system. To reconstruct 3D coordinates for some points we have used manual matching at this stage of the project. (Automatic matching is discussed in Chapter 6). To generate the images in Figure 3.18 and 3.19 we have used multi-beam sonar from Blue View, the ProViewerE P900E-20. We have one sonar only, so we moved the sonar from the first position to the second and assumed that both images are taken in a very short time. This assumption is valid since in the controlled tank environment when scanning stationary objects. The distance D between the two positions is 1m; $\alpha 1 = 30°$, $\alpha 2 = 10°$, and $\beta = 10°$. (β is a parameter of the sonar itself which is half of the vertical beam width in P900E-20 sonar).

Figure 3.20 shows the first image obtained in the first position. The intersection between the red-dashed circle and the red-dashed line determine the point of interest which we would like to calculate its 3D coordinates (especially the Y coordinate). The real Y coordinate of that reflective point (which is a small ball in the middle of the tank) is 2.31m. Figure 3.21 shows the image obtained in the second position, again, the intersection between the red dashed circle and the red dashed line determine the same point of interest in the first image.

From Figure 3.20 and 3.21 we can get ri = 3.34m and rj = 2.90m. From these values we have calculated the Y coordinate for the point of interest which is 2.34m. In this case the error is within 3cm. we have tried many positions for the ball within the FOV as well as different D values, Table 3.1 shows these results. In most of these experiments results were consistent with the ASI simulation, some results are erroneous (number 2 and 9 in the table) due to the imperfection of the measuring tools and non-uniformity of the beams of the sonar.

The point of interest lies within the range between R_2 and R_1 which is shown in green dashed-dotted circles in Figure 3.21. Moreover, the POI lies on the red dashed line which has the same θ as the red dashed line in the first image (Figure 3.20), this comes directly from aligning beams in both positions and setting them in the same plane. Thus to search for the point of interest we have to search in the region underneath the red dashed line and between the green dashed-dotted circles. This is a key point of the ASI geometry developed in this thesis, it can be considered as a key contribution in this chapter.

Number	Distance D	Real Y	r_i	r_j	y_p	y_0	Constructed Y	Error
1	0.5	2.31	3.34	3.08	1.92	0.47	2.38	0.08
2	0.5	2.08	2.7	2.48	1.4	0.47	1.86	-0.22
3	0.5	1.75	2.26	1.98	1.44	0.47	1.9	0.15
4	0.6	2.31	3.34	3.07	1.74	0.47	2.21	-0.10
5	0.6	2.08	2.7	2.42	1.49	0.47	1.96	-0.12
6	0.6	1.75	2.26	2.02	1.15	0.47	1.62	-0.12
7	0.7	2.31	3.34	2.99	1.93	0.47	2.40	0.09
8	0.7	2.08	2.7	2.31	1.75	0.47	2.21	0.13
9	0.7	1.75	2.26	2.01	1.11	0.47	1.58	-0.17
10	0.8	2.31	3.34	2.99	1.78	0.47	2.25	-0.06
11	0.8	2.08	2.7	2.29	1.68	0.47	2.15	0.07
12	0.8	1.75	2.26	1.98	1.14	0.47	1.61	-0.14
13	0.9	2.31	3.34	2.96	1.78	0.47	2.25	-0.06
14	0.9	2.08	2.7	2.32	1.51	0.47	1.97	-0.11
15	0.9	1.75	2.26	1.94	1.20	0.47	1.67	-0.08
16	1	2.31	3.34	2.9	1.87	0.47	2.34	0.03
17	1	2.08	2.7	2.2	1.72	0.47	2.19	0.11
18	1	1.75	2.26	1.93	1.19	0.47	1.66	-0.9

Table 3.1: Different distances between the sonars to reconstruct the missing coordinate of the points in the tank, the larger the distance the better the performance at the cost of smaller FOVA. All numbers in meters.

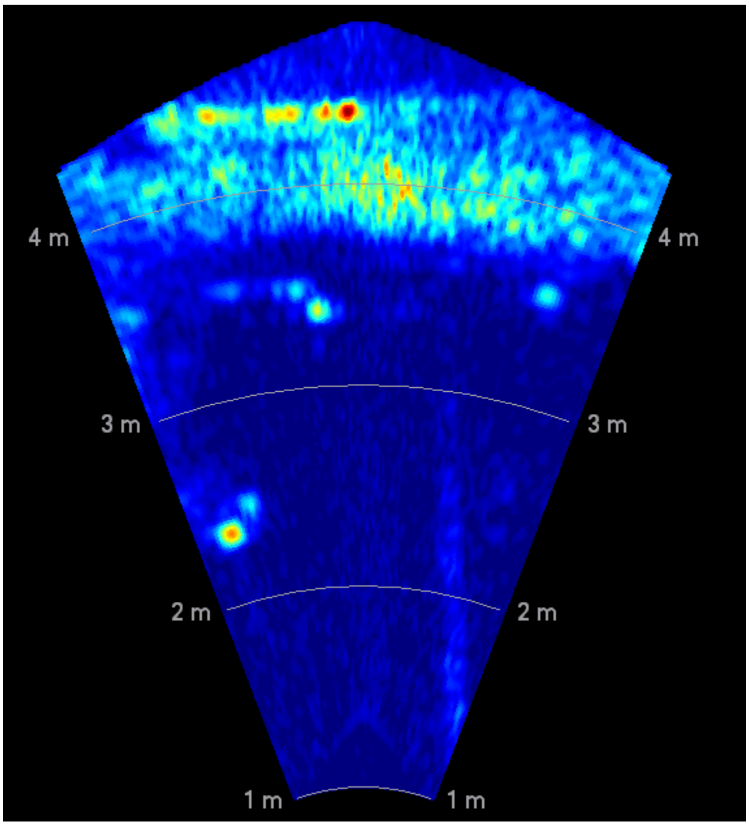


Figure 3.18: image obtained from the first position before matching, see text for details.

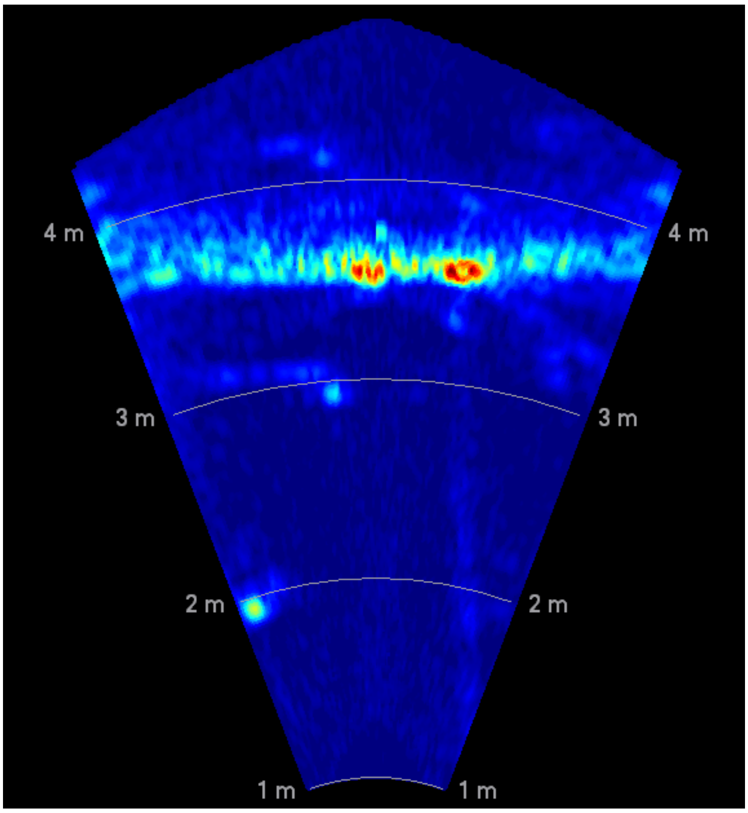


Figure 3.19: image obtained from the second position before matching, see text for details.

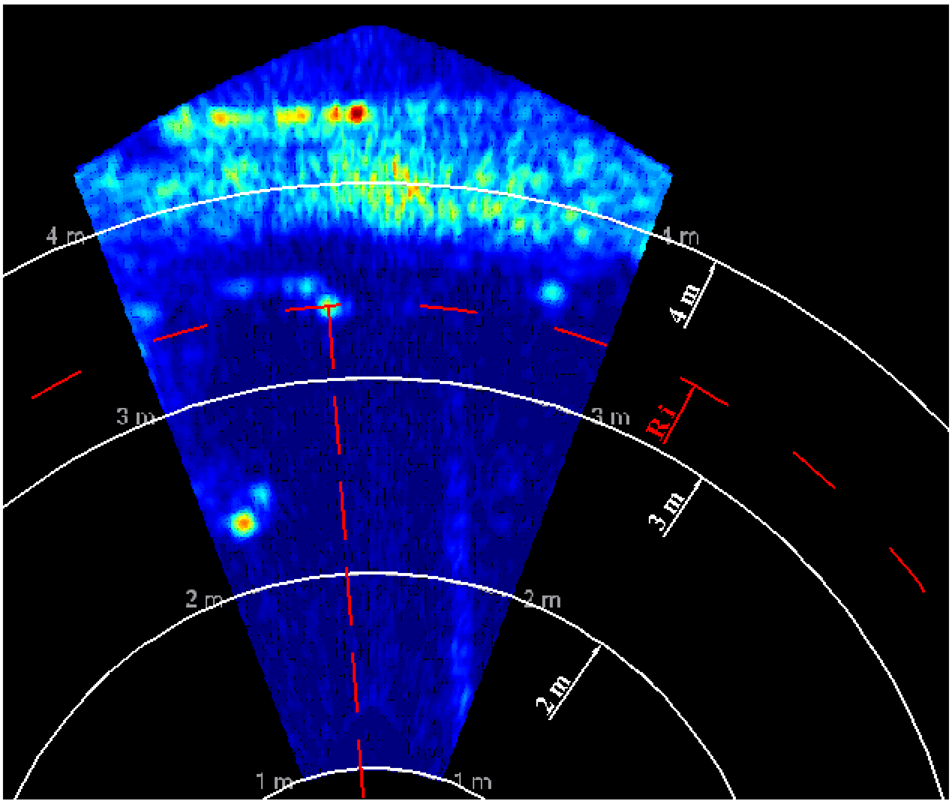


Figure 3.20: image obtained from the first position, see text for details.

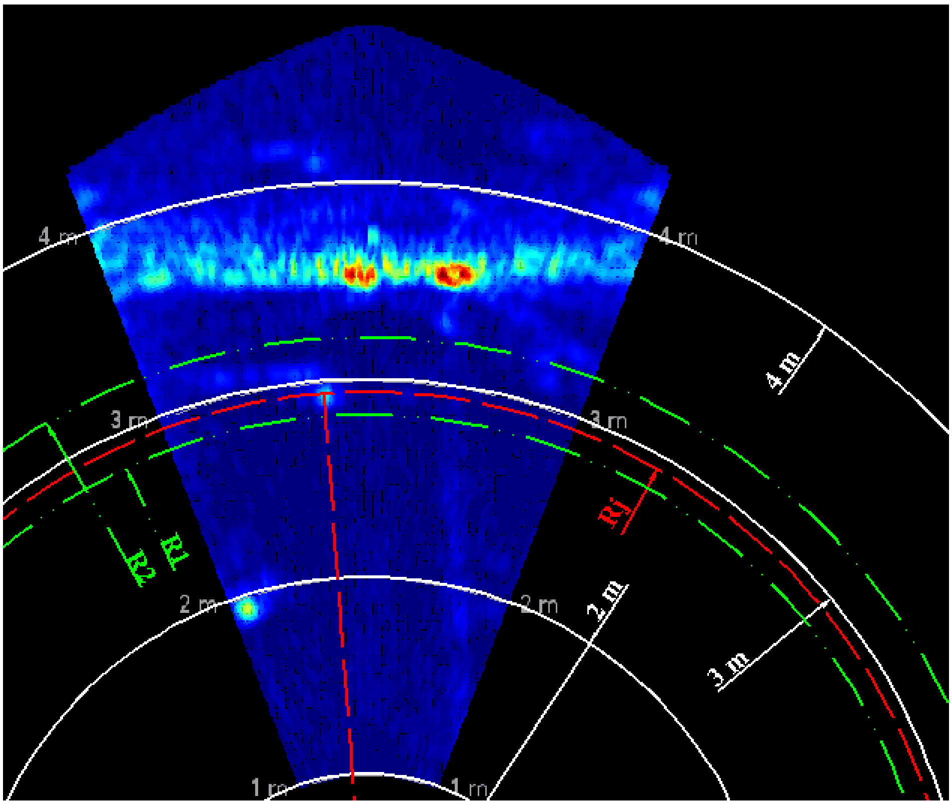


Figure 3.21: image obtained from the second position, see text for details.

3.10 Conclusion

In this chapter, we have discussed the feasibility of building a real time 3D reconstruction system starting from relatively inexpensive sensors. The ASI system which is introduced and studied in this chapter consists of two multi-beam, high resolution sonars situated one above the other. The ASI geometry has been introduced to facilitate matching correspondences between the two images obtained from the sonars. ASI simulation has shed the light on the ASI parameters and their effects on the system performance. Real data obtained from BlueView sonar have validated the ASI Geometry and 3D reconstruction for mid-water targets has been accomplished. The search for correspondences in a small linear range $[R_2, R_1]$ is the key finding in this chapter.

Chapter 4
Motion Estimation by Acoustic Imaging

4.1 Introduction

2-D Imaging systems, e.g., optical or acoustic, encode rich visual cues about the geometry of the work that is imaged and the position of the sensor relative to the world. In motion vision applications dealing with robotics platforms, one of the objectives is to determine the trajectory of a mobile system from the variations in the 2-D scene imagery. While this problem has been addressed extensively for numerous terrestrial applications[8] [9] [10] and to some extend in underwater by optical imaging [11][12], less than a handful of earlier studies have explored application to 2-D sonar imaging systems [13][14].

Recently, 2-D sonar imaging systems, e.g., DIDSON video cameras [72], have been installed on Remotely Operated Vehicles (ROVs) and AUVs for the inspection of ship hulls and other subsea structures in turbid coastal and harbour waters. The control systems of these platforms and (or) their designs are to eliminate the pitch and roll motions, namely the rotations around the X and Y axes of the platform [148] [149]. Therefore, they typically undergo movements with 4 degrees of freedom, which means no/negligible pitch and roll. Alternatively, these two rotation components (around the X and Y axes) can be measured by external sensor, e.g., gyros, while sensors for measuring rotations about the Z axis use magnets. Thus, while the former can be measured with relatively good accuracy, the performance of the latter may be affected by metal structures such as the ship hull, pipelines, etc. Also, magnets perform poorly in high altitudes closer to the Arctic Circle. Thus, it is important to devise a robust and accurate method for the estimation of rotations around the Z axis, namely heading motions.

In this chapter, a new framework for the analysis of 2-D sonar video image is explored, comprising the principles of Modified Discrete Uniform Distribution (MDUD), Sonar Projection Function (SPF) and sonar arc sampling. It can be applied to address the problems of motion estimation, correspondence problem and 3D reconstruction. This will facilitate various routine tasks of ROVs/AUVs, including the inspection of different subsea structures, autonomous navigation, and target localization and classification. The immediate application of interest is the estimation of sonar motion with 4 degrees of freedom. We present an algorithm that applies the proposed framework along with a weighted-Hough transform formulation to select a solution from among many potential candidates. Next we present essential background concepts including sampling DIDSON arc, Modified Discrete Uniform Distribution (MDUD), and probability adjustment based on shadow.

4.2 Background concepts

4.2.1 Sampling DIDSON arc

The sonar measurements (r, θ) comprise two components of the spherical coordinates $(\mathrm{r}, \theta, \varphi)$ of a 3-D point of interest (POI), see Figure 4.1. These measurements, i.e. (r, θ) define an arc in 3-D space as the locus of the POI, corresponding to an arbitrary elevation angle φ (doted-purple arc in Figure 4.1).

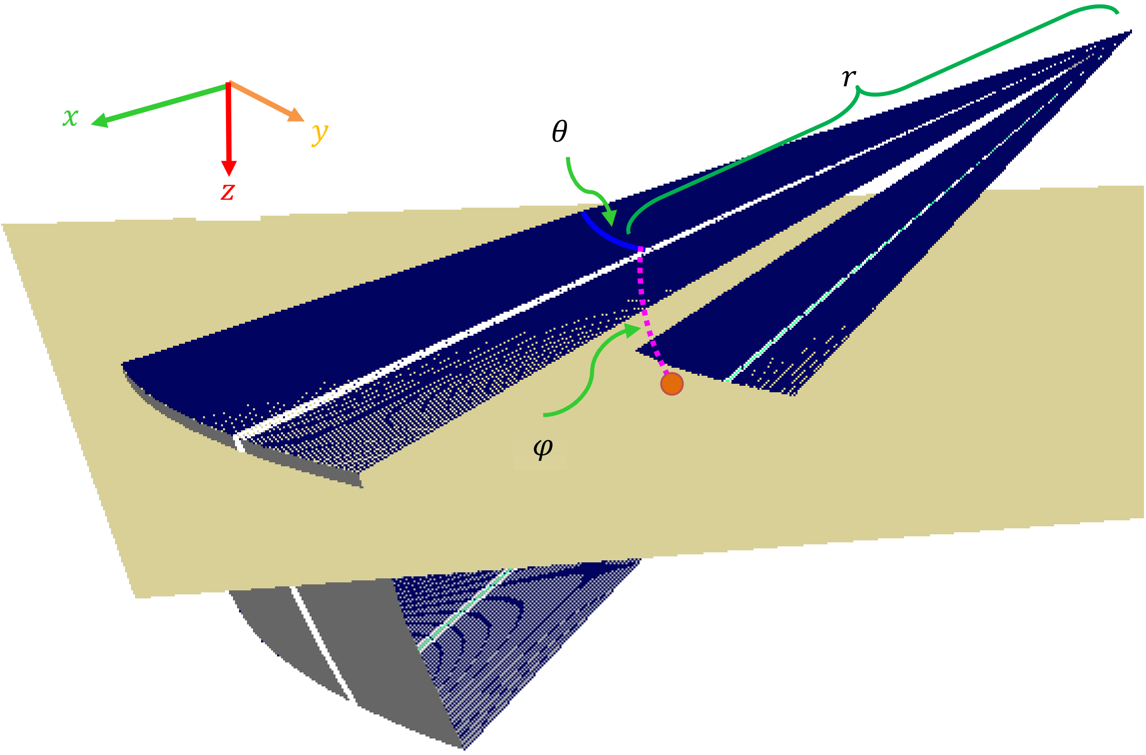


Figure 4.1: The spherical coordinates $(\mathrm{r}, \theta, \varphi)$ of a 3-D point of interest (POI). The locus of the POI is defined by the doted-purple arc in sonar measurements (r, θ).

In sonar measurements, the elevation angle φ of the POI cannot be retrieved so we cannot determine the 3D coordinates of the POI. Thus, any point on the arc described by (r, θ) (dotted-purple arc in Figure 4.1 and Figure 4.2) is a possible locus for the POI. To estimate the true position of the POI, we start by sampling this arc with N discrete points, each point being a candidate 3-D point for the POI as illustrated in Figure 4.2.

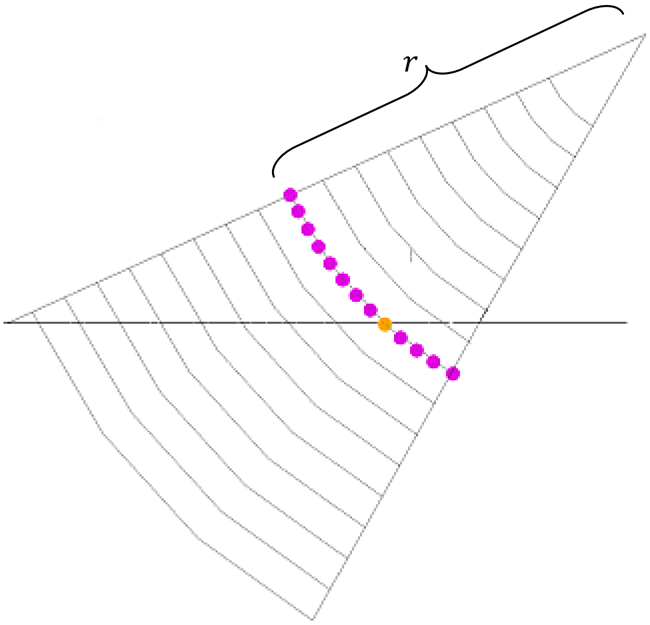


Figure 4.2: Each point in purple is a possible solution for the measured (r, θ) of the true POI (illustrated in orange). This figure depicts, in 2D, the beam which scans the POI illustrated in Figure 4.1.

By considering the axes as depicted in Figure 4.1, the 3D coordinates of the N sampled points can be inferred using the following equations in (4.1):

$$X_i = \begin{cases} x_i = r.\cos\theta\,.\cos\left(\frac{(i-1).\varphi_{max}}{N-1}\right) \\ y_i = r.\sin\theta\,.\cos\left(\frac{(i-1).\varphi_{max}}{N-1}\right) \\ z_i = r.\sin\left(\frac{(i-1).\varphi_{max}}{N-1}\right) \end{cases} \tag{4.1}$$

Where i = 1 ... N; φ_{Max} is the maximum vertical angle, we consider here that $\varphi_{\mathrm{Min}} = 0$

A naive look may consider that each point on the sampled arc has the same probability of being the true POI. In other words, each point can be assigned a pre-estimated probability from the uniform distribution, so the sampled 3D Points $\mathrm{X_i} = (\mathrm{x_i}, \mathrm{y_i}, \mathrm{z_i})$ are associated with the probability $\mathrm{P_i}$ of being the true POI as in equation (4.2). However, by considering the nature of sonar imaging, we introduce a different way of assigning probabilities to the sampled arc points as explained in the next section.

$$\sum_{i=1}^{i=N} P_i = 1 \qquad \Rightarrow \qquad P_i = \frac{1}{N} \tag{4.2}$$

4.2.2 Modified Discrete Uniform Distribution (MDUD)

The MDUD method for probability assignment is based on noticing the usual tilt angle of the sonar while in operation. The acoustic image of the insonified surface is formed properly when the sonar has a suitable tilt angle relative to the surface, otherwise, the acoustic image will be merely black image with few horizontal lines. The tilt angle of the sonar leads us to another observation; the region where the arc hits the flat surface depends on the range of the arc. In general, as the range increases, the elevation angle of the POI decreases. This is illustrated in Figure 4.3.

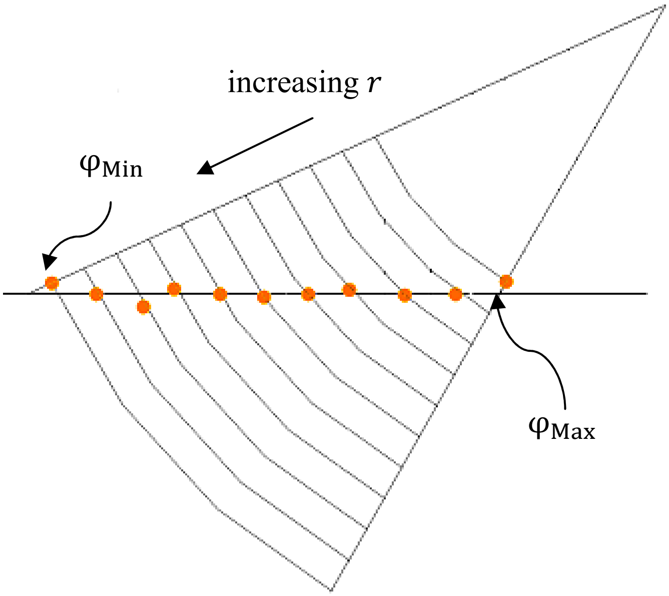


Figure 4.3: In general, as the range increases, the elevation angle of the POI decreases.

This observation leads us to define a set of points with high probability of being the true POI in the sampled arc. The position of this set depends merely on the range of the arc. This set is called points with high probability level; Figure 4.4 illustrates this set using red dots. The mathematical definition of this set will be introduced later on.

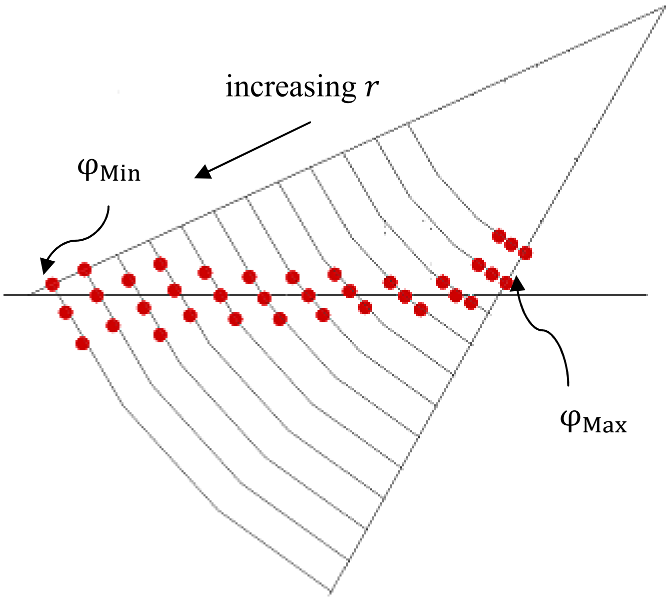


Figure 4.4: The high level probability points. Their elevation angle φ decreases as the range increases.

The definition of the points with high probability level brings the definition of two other sets; first, the set of points with low probability level where the elevation angle is larger than the elevation angle for the points in the high probability level, second, the set of points with medium probability level where the elevation angle is smaller than the elevation angle for the points in the high probability level. These two sets are illustrated in Figure 4.5.

Thus, each arc has N sampled points; these points are categorized in three sets with three probability levels reflecting the probability of being the true POI. The distribution of these sets on the sonar arc and their probability levels are governed by a new distribution dubbed the Modified Discrete Uniform Distribution (MDUD) which is explained next.

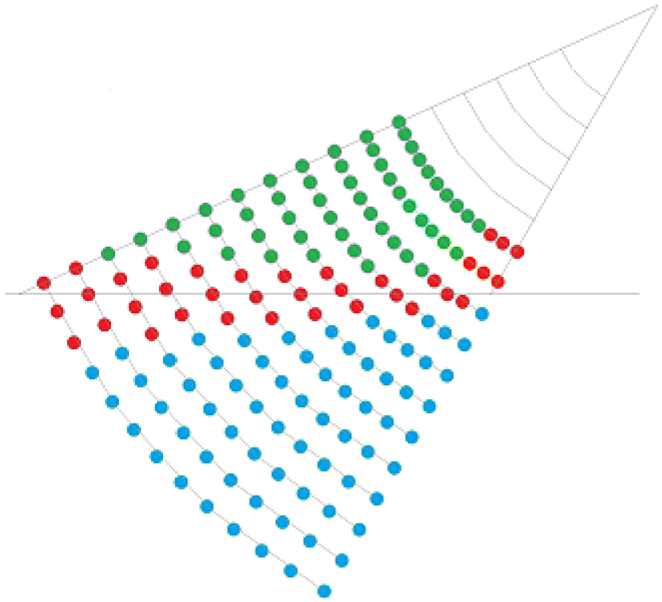


Figure 4.5: Sampled points in high level probability (red), medium level probability (green), and low level probability (blue). All arcs have the same number of sampled points, assuming a flat surface.

To assign probabilities for each point in the sampled arc (r, θ), we infer the values of $\mathrm{P}_1, \mathrm{P}_2 \ldots \mathrm{P}_\mathrm{N}$ based on a new version of the discrete uniform distribution. In this distribution, we introduce three levels of probability, in contrast to only one level in the uniform distribution. These levels are the low, high, and medium probabilities. Some sampled points fall in the medium level (ML), few in the high level (HL), and the rest in the low level (LL). This is illustrated in Figure 4.6 by blue, red, and green dots for low, high, and medium levels, respectively.

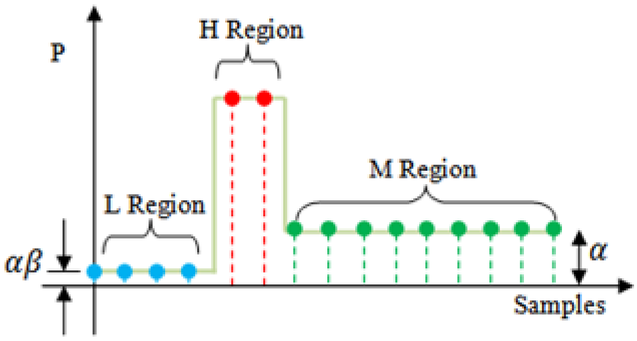


Figure 4.6: Modified Discrete Uniform Distribution

If N is the total number of samples in the arc, $\mathrm{N_m}$denotes the number of samples in the medium level, $\mathrm{N_h}$ is the number of samples in the high level, and $\mathrm{N_l}$ is the number of samples in the low level. It readily follows that:

$$\begin{aligned} &N = N_m + N_h + N_l \\ &0 < N_h \ \leq N \\ &0 \ \leq N_m \ < N \\ &0 \ \leq N_l \ < N \end{aligned} \tag{4.3}$$

Where $\mathrm{N}, \mathrm{N_m}, \mathrm{N_h}$ and $\mathrm{N_l}$ are all integers.

We define two parameters in determining the three-level probabilities. The first one α usually takes on values between 60% to 90% of that for the uniform distribution (i.e. $1/\mathrm{N}$), in fact, $(1-\alpha)$ defines the amount of probability taken from the points in the medium level and added to the points in the high level. The second parameter β usually takes on values between 15% to 35% of the medium level. Accordingly, the probability $\mathrm{P_i}$ is calculated as follows:

$$P_i = \begin{cases} \dfrac{\alpha.\beta}{N} & : 1 \leq i \ \leq N_l \\ \dfrac{1}{N}\left[1 + \dfrac{1}{N_h}[(1-\alpha)N_m + (1-\alpha\beta)N_l]\right] & : N_l < i \ \leq N_l + N_h \\ \dfrac{\alpha}{N} & : N_l + N_h < i \ \leq N \end{cases} \tag{4.4}$$

It worth mentioning that the definition for P_i in (4.4) satisfies the relation in (4.5), and the uniform distribution is a special case of the MDUD when $\alpha = \beta = 1$.

$$\sum_{i=1}^{i=N} P_i = 1 \tag{4.5}$$

The percentage of samples in the medium and low level regions compared to all available samples is denoted by γ, this usually takes a value between 70% to 95%. The relation between the sampled points on a DIDSON arc and this distribution is defined based on the range of the arc as a first guess. If the first strong return and the last strong return in the range direction are denoted $R_{min}$and R_{max}, respectively, we define $\mathrm{N_l}$, $\mathrm{N_h}$, and $\mathrm{N_m}$ as follows:

$$N_l = floor\left[\gamma N.\frac{r - R_{first}}{R_{last} - R_{first}}\right]$$

$$N_m = \ ceiling[\gamma . N - N_l] \tag{4.6}$$

$$N_h = N - N_m - N_l$$

Where r is the range of the POI. It is noted that, over all, we need to tune four parameters: $N, \alpha, \beta,$ and γ. Calculating $\mathrm{N_l}$ in (4.6) is simple approximation given that we have a decent sonar image, which means that the sonar is close to the recommended tilt angle. In theory, $\mathrm{N_l}$ depends on the tilt angle of the sonar relative to the seabed and on the range of the POI. The first and last strong returns are detected when the average value of the intensities exceeds pre-determined threshold.

4.2.3 Probability adjustment based on shadow

Determining $\mathrm{N_l}$, $\mathrm{N_h}$ and $\mathrm{N_m}$ for each point of interest is not dependent on range only, however the range contributes to the original distribution of the probabilities over the N samples. Calculations of $\mathrm{N_l}$, $\mathrm{N_h}$ and $\mathrm{N_m}$ in (4.6) are carried out by assuming a flat scene, but this is not the case everywhere. For example, suppose we are imaging the sea bottom floor. Usually, some structures may protrude from the sea floor; these structures would form shadows in the sonar image. In our method, we adjust the calculations of $\mathrm{N_l}$, $\mathrm{N_h}$ and $\mathrm{N_m}$ based on the length of the shadow cast behind the point of interest along the iso-azimuth contour, (see Figure 4.7.)

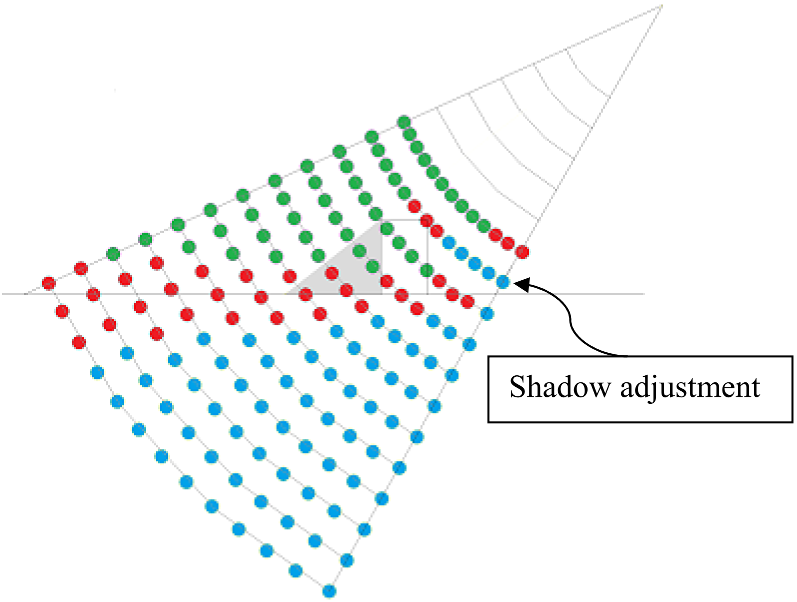


Figure 4.7: MDUD, and the shadow adjustment, blue dots are in the low region, red in the high and green in the medium

Consider the image in (r, θ) coordinates depicted in Figure 4.8. The yellow rectangle represents an object forming a shadow (black rectangle) in the image. In calculating N_l, N_h and N_m for each point in the yellow region, we do not use its own range (r) in (4.6), but rather we apply the adjusted range, which is calculated as follows:

$$r_{adj} = \frac{R_S}{R_A} r \tag{4.7}$$

The size of the shadow in Figure 4.8 is assumed to be the same for all of the sonar beams. In general, this is not the case, and so the shadow is measured along the same iso-azimuth scan (sonar beam) for the POI. Figure 4.9 shows a sample (r, θ) image from a DIDSON camera.

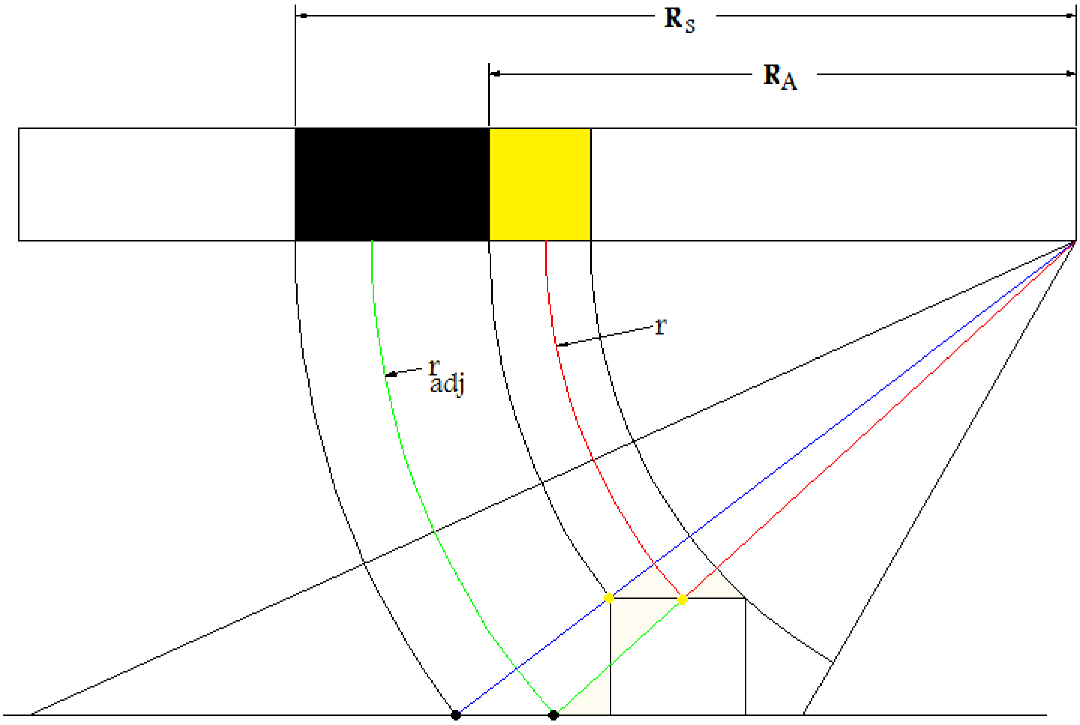


Figure 4.8: an Object and its shadow

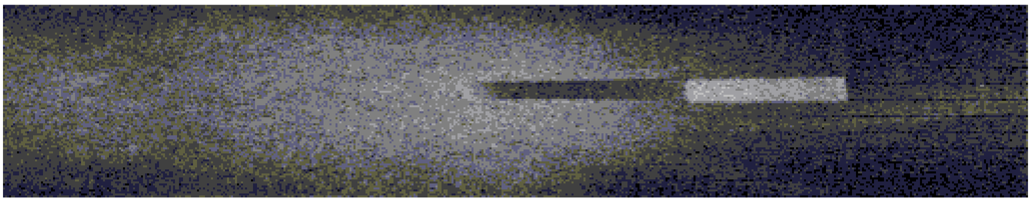


Figure 4.9: real data from DIDSON showing a block in R-Theta coordinates

4.2.4 Sonar projection function

For a 3-D point $\mathrm{X = (x, y, z)}$ we define the Sonar Projection Function (SPF) as follows:

$$SPF(X) = \begin{cases} (r, \theta) \; when \; X \; visible \\ (0, 0) \; when \; X \; outside \; field \; of \; view \end{cases}$$

(4.8)

$$r = \left[\sqrt{x^2 + y^2 + z^2}\right]_{Q\varepsilon}$$

$$\theta = \left[\tan^{-1}\frac{y}{x}\right]_{Q\lambda}$$

Where Qε means that r is quantized with ε resolution, Qλ means that θ is quantized with λ resolution as well. Note that the point of interest has to meet the following conditions:

$$\begin{gathered} \mathrm{R_{Min}} \leq \mathrm{r} \leq \mathrm{R_{Max}} \\ 0 \leq \theta \leq \theta_{\mathrm{Max}} \\ 0 \leq \tan^{-1}\frac{\mathrm{z}}{\mathrm{r_{xy}}} \leq \varphi_{\mathrm{Max}} \end{gathered} \tag{4.9}$$

Where $\mathrm{R_{Min}}, \mathrm{R_{Min}}, \theta_{\mathrm{max}}$ and φ_{max} are the specification of the sonar. We next demonstrate the application of MDUD and arc sampling to the estimation of sonar motion.

4.3 Application 1: sonar motion estimation

The sonar motion relative to the sea floor may comprise translation and rotation components. These can be described by the rigid transformation $[\mathrm{R}, \mathrm{T}]$. Then all space points move to a new position $\mathrm{X}_\mathrm{i}' = (\mathrm{x}_\mathrm{i}', \mathrm{y}_\mathrm{i}', \mathrm{z}_\mathrm{i}')$ relative to the sonar. Using the homogeneous coordinates, the new position is given by the equation in (4.10).

$$\mathrm{X}' = \begin{bmatrix} \mathrm{x}' \\ \mathrm{y}' \\ \mathrm{z}' \end{bmatrix} = [\mathrm{R}, \mathrm{T}] \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{z} \\ 1 \end{bmatrix} = [\mathrm{R}, \mathrm{T}]\, \mathrm{X} \tag{4.10}$$

$$\mathrm{SPF}(\mathrm{X}') = \mathrm{SPF}(\mathrm{x}', \mathrm{y}', \mathrm{z}') = (\mathrm{r}', \theta') \tag{4.11}$$

Where $\mathrm{X}_\mathrm{i}' = [\mathrm{x}_\mathrm{i}' \;\; \mathrm{y}_\mathrm{i}' \;\; \mathrm{z}_\mathrm{i}']^\mathrm{T}$ is the correspondence of $\mathrm{X} = [\mathrm{x} \;\; \mathrm{y} \;\; \mathrm{z}]^\mathrm{T}$. The motion estimation involves determining $[R\; T]$ given the corresponding measurement pairs (r, θ) and (r', θ') in two sonar views. We consider the simplified case where the rotation is solely about the Z-axis of the sonar camera. This is the case, when the sonar is installed on a vehicle that typically undergoes movements with 4 degrees of freedom; i.e. no/negligible pitch and roll.

Rotations around the X axis or the Y axis can be measured by an external sensor which depends on gravity, sensors used to measure rotations around the Z axis use magnets so their performance may be affected by fast movements and/or metal objects like the ship's hull. Moreover, most control systems in AUVs and ROVs try to eliminate rotations around the X axis and the Y axis, so these rotations tend to be very small. What is important here is the rotation about the Z axis to scan bigger area and this is what the algorithm tries to facilitate.

4.3.1 Calculating the sonar rotation (rotation around Z-axis)

If we have two points in the first image, and their correspondences in the second image we can calculate the rotation angle around the z axis. Figure 4.10 shows that the angle of rotation R_z is given by

$$R_z = A_1 - A_2 \tag{4.12}$$

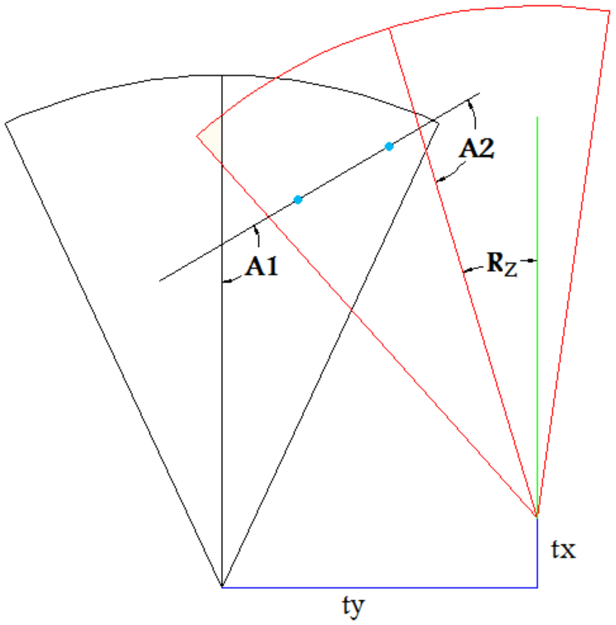


Figure 4.10: Calculating R_z using two POIs in two scans and drawing a line between them to infer A_1 and A_2

Where A_1 and A_2 each, is the angle between the line passing through the two points of interest in each of the two views (blue dots in the Figure 4.10) and the central beam of that view, in other words, $\mathrm{R_z}$ can be calculated as follows:

$$\mathrm{R_z} = \tan^{-1}\frac{\mathrm{X}_2 - \mathrm{X}_1}{\mathrm{Y}_2 - \mathrm{Y}_1} - \tan^{-1}\frac{\mathrm{X}'_2 - \mathrm{X}'_1}{\mathrm{Y}'_2 - \mathrm{Y}'_1} \tag{4.13}$$

Where $(\mathrm{X}_1, \mathrm{Y}_1), (\mathrm{X}_2, \mathrm{Y}_2)$ and $(\mathrm{X'}_1, \mathrm{Y'}_1), (\mathrm{X'}_2, \mathrm{Y'}_2)$ are the points of interest in the first and second view respectively.

If we have M pairs of correspondences, we can make $\mathrm{M}(\mathrm{M}-1)/2$ calculations for the rotation angle around the Z axis, but the estimates differ due to various sources of error, including uncertainty in the feature position and range-azimuth quantizations. One solution is to average these various estimates. In general some form of least-square error formulation can be applied. Here, we propose another solution.

One can readily note that the sources of error have minimal impact where the line between the two points of interest crosses more beams. Conversely, if the two points of interest lie on the same beam or on nearby beams, then each of A_1 and A_2 are more sensitive to the errors in locating the two image features and image quantization (range and azimuth resolutions).

In our proposed solution, we consider only those pairs of points where the line between them crosses more than a minimum number of beams (based on a pre-determined threshold). We then apply a voting scheme based on the Hough transform in the $[-\pi, \pi]$ space with a selected resolution (e.g., 0.5° resolution corresponding to 721 cells). The solution is the angle with the maximum votes in the linear Hough array.

To summarize, if we have $p(x_i, y_i)$ a point in the first view which corresponds to the point in the second view $p'(x'_i, y'_i)$ and $i = 1 \dots M$. Then to find out the rotation angle around the z axis we follow the algorithm as explained in the pseudo-code in the Table 4.1.

```
// P is an array of the points in the first view
// Pprim is an array of the points in the second view
// P[i] corresponds to Pprim[i], i = 0.. M-1
// HoughArray is the array where we collect votes

for (int i = 0; i < M ; i++)
{
    for (int j = 0; i < M ; j++)
    {
        if (i == j) continue;

        // the points should cross enough beams to be considered.
        if ((BeamDistance(P[i],P[j]) < threshold)||
            (BeamDistance(Pprim[i],Pprim[j]) < threshold)) continue;

        // atan is a function of the inverse tangent in degrees.
        A1 = atan((P[j].x-P[i].x)/(P[j].y-P[i].y));
        A2 = atan((Pprim[j].x-Pprim[i].x)/(Pprim[j].y-Pprim[i].y));

        // round Rz to the nearest 0.5 degree
        Rz = round((A1-A2)*2)/2;

        // The index of Rz in Hough array,
        index = 2*Rz+360;

        // vote in the Hough Array,  it has
        // 721 celles representing the range -180,180
        HoughArray[index]++;
    }
}
// return the index of the maximum cell.
Index = MaxIndex(HoughArray);
Rz = (Index-360)*2;
```

Table 4.1: Pseudo-code for calculating the rotation angle around the ***z*** axis.

4.3.2 Calculating the sonar translation

Once the rotation around the Z axis has been determined, we can calculate the translation matrix T from M correspondences, as we describe next. To do so, we first de-rotate the second image by the estimated angle R_z, so the transformation between the two views is solely due to translation.

Consider a selected correspondence. In the first view, the feature point gives N 3-D candidate points $X_i = (x_i, y_i, z_i)$ based on the sampling of the sonar arc; each assigned a probability according to the MDUD method. Similarly, the correspondence in the second view gives N 3-D candidate points $X_j' = (x_j', y_j', z_j')$ with their probabilities. While there are N^2 possible motions from a sample $X_i = (x_i, y_i, z_i)$ to $X_j' = (x_j', y_j', z_j')$, some are more likely based on the assigned probabilities of the corresponding 3-D points. To determine the most likely translational motion, we apply a weighted voting scheme based on Hough transform over all the corresponding pairs.

In our implementation, we sample the 3-D space of potential translations $\mathrm{T_k} \subset [-\mathrm{T_k^{max}}, \mathrm{T_k^{max}}]$ with a particular resolution $\Delta\mathrm{T_k}$where $(\mathrm{k} = \mathrm{x}, \mathrm{y}, \mathrm{z})$. Along each coordinate axis, there are $\mathrm{C_k} = 1 + 2\mathrm{T_k^{max}}/\Delta\mathrm{T_k}$ cells, for a total of $\mathrm{C} = \mathrm{C_x C_y C_z}$ cells. Each translation gives a weighted vote for its selected cell based on its assigned probability $\mathrm{P_i P'_j}$ .The solution is the cell with the largest vote.

4.4 Motion estimation algorithm

Herein we summarize the whole motion estimation algorithm, given two sonar images $\mathrm{I_1}$ and $\mathrm{I_2}$, with M pairs of correspondences called the "cor" set, execute the following steps:

1) Initialize the correspondences set as in (4.16):

$$\mathrm{cor} = \{(\mathrm{r_j}, \theta_\mathrm{j}) \in \mathrm{I_1} \;\; ; \; (\mathrm{r'_j}, \theta'_\mathrm{j}) \in \mathrm{I_2} \;\; \text{where } \mathrm{j} = 1 \ldots \mathrm{M}\} \qquad (4.14)$$

2) Calculate angles A1 and A2:

$$\mathrm{A_{Angles}} = \;\{\mathrm{A1_k}\,; \mathrm{A2_k}\,\} \text{ where } \mathrm{k} = 1, 2 \;\ldots \mathrm{M(M-1)/2} \qquad (4.15)$$

3) Discard those $\{\mathrm{A1_k}\,; \mathrm{A2_k}\,\}$ pairs that are computed from two points where the line between them crosses fewer beams than the pre-determined threshold. Denote Q the number of lines that survive the threshold test.

$$\mathrm{A_{Angles}} = \;\{\mathrm{A1_k}\,; \mathrm{A2_k} \text{ where } \mathrm{k} = 1 \ldots \mathrm{Q}\} \qquad (4.16)$$

4) Calculate the estimates $\mathrm{R_{Zk}}$ from all Q pairs

$$R_{Zk} = A1_k - A2_k \text{ where } k = 1 \dots Q \tag{4.17}$$

5) Initialize 1-D Hough array and collect votes for R_{Zk}.

6) Get the index of the maximum cell, which corresponds to the best estimate R_Z .

7) Each point in the "cor" set will produce N points by applying the sampling procedure in (4.2), Note that M is the number of correspondences, we have:

$$\begin{gathered} S = \{(x_{ij}, y_{ij}, z_{ij}) \,;\, (x'_{ij}, y'_{ij}, z'_{ij})\} \\ \text{where } i = 1 \dots N; j = 1 \dots M \end{gathered} \tag{4.18}$$

8) Associate the MDUD probabilities to the points in S, taking into account adjustments based on shadow size.

$$P = \{P_{ij} \,;\, P'_{ij} \text{ where } i = 1 \dots N; j = 1 \dots M\} \tag{4.19}$$

9) De-rotate the sampled points in the second image around the Z axis by $-R_Z$

$$\begin{gathered} S_{\text{derotated}} = \{(x_{ij}, y_{ij}, z_{ij}) \,;\, (x''_{ij}, y''_{ij}, z''_{ij})\} \\ \text{where } i = 1 \dots N; j = 1 \dots M \end{gathered} \tag{4.20}$$

10) Calculate $T_k = [tx_k \; ty_k \; tz_k]^T$:

$$\begin{gathered} tx_k = x_{ij} - x''_{hj} \\ ty_k = y_{ij} - y''_{hj} \\ tz_k = z_{ij} - z''_{hj} \\ P(T_k) = P_{ij} P'_{hj} \end{gathered} \tag{4.21}$$

$$\text{where } \; i = 1..N; h = 1..N; j = 1..M; k = i.j.h$$

11) Initialize 3-D Hough array, and collect votes $\mathrm{P}(\mathrm{T_k})$ for each of the C cell.

12) Get the index of the maximum votes, corresponding to the best estimate T.

13) The transformation between $\mathrm{I_1}$ and $\mathrm{I_2}$ is $[\mathrm{R_z}\ \mathrm{T}]$.

4.5 Application 2: Correspondence problem

Suppose that we have computed the motion $[\mathrm{R}, \mathrm{T}]$ for two sonar views, or it is determined by calibration (as is the case for a calibrated sonar stereo system). Given a particular measurement (r, θ) for some 3-D feature in one sonar view, can we estimate the measurement (r', θ') for the same feature in the second view?

For a given (r, θ) we have N candidate 3-D points by sampling the arc:

$$(\mathrm{r}, \theta) \Rightarrow [\mathrm{X_1}, \mathrm{X_2}, \mathrm{X_3}, \dots \dots \mathrm{X_N}] \tag{4.22}$$

These points are associated with MDUD probabilities $[\ \mathrm{P_1}, \mathrm{P_2}, \dots\ \mathrm{P_N}]$, each representing the likelihood that the associated sample $\mathrm{X_i}\ ; \mathrm{i} = 1{,}2 \dots \mathrm{N}$ is the true 3-D POI. To determine the match in the second image, we compute the 3-D position relative to the second sonar position, and the projection in the second view:

$$\mathrm{X'_i} = [\mathrm{R}, \mathrm{T}]\mathrm{X_i} \qquad \text{where i} = 1 \dots \mathrm{N} \tag{4.23}$$

$$\mathrm{SPF}(\mathrm{X'_i}) = (\mathrm{r'_i}, \theta'_\mathrm{i}) \quad \text{where i} = 1 \dots \mathrm{N} \tag{4.24}$$

The probability P_i is associated with each projected measurement $(\mathrm{r'_i}, \theta'_\mathrm{i})$. The correlation between (r, θ) and each $(\mathrm{r'_i}, \theta'_\mathrm{i})$ will give us a new measure $\mathrm{q_i}$. The estimates with low q_i and low P_i can be discarded, keeping the remaining ones as potential sought after match. This procedure can be easily adopted in the ASI system described in Chapter 3; it can be utilized in the vertical configuration and any other configuration.

4.6 Results

Several tests have been conducted to test the performance of the algorithm, results show that the z rotation angle can be inferred with ∓1° error for a single measurement. The first experiment was carried out in the University of Miami where we designed a special mounting platform to host the DIDSON so that it is possible to measure the Z rotation and the tilt of the sonar, see Figure 4.11. The experiment consists of four 5° rotations about the Z axis with no translations at all, this will give us an idea about the expected errors in the translation results, the sonar was tilted 26° to image the calibration board used in [70] [150], this board offers many distinct features in the sonar image as illustrated in Figure 4.12.

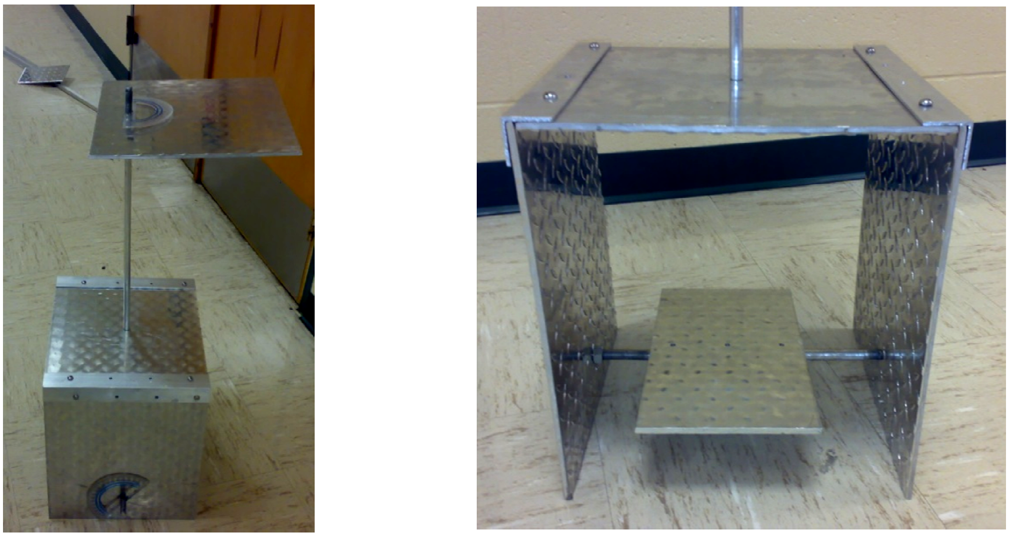


Figure 4.11: Mounting platform to host DIDSON sonar and measure the tilt and Z rotation.

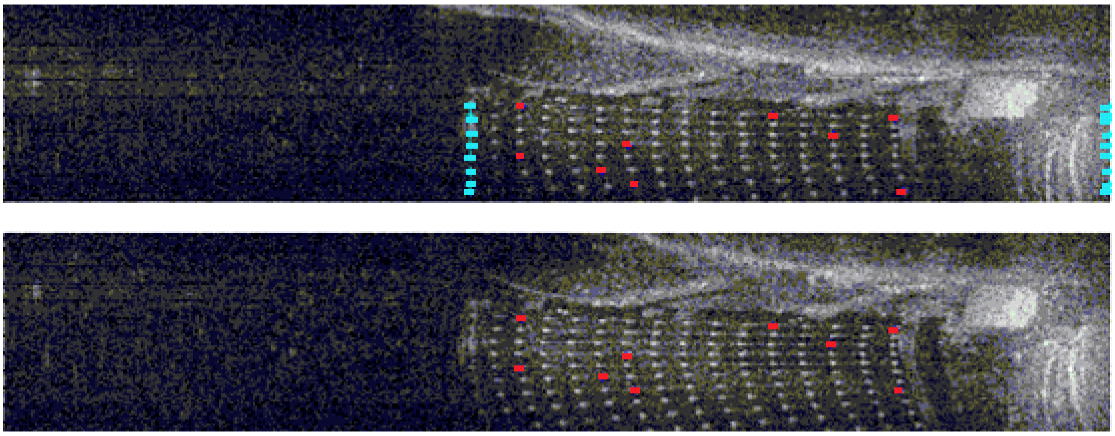


Figure 4.12: Two images for the calibration board in R-Theta coordinates, POIs appear in red and the first/last strong return appear in cyan in the first image

Table 4.2 shows the results obtained from the first experiment, it is clear that the Z rotation angle is estimated with small errors, however there are considerable errors in estimating the translation on the Y axis, after performing many tests, it turns out that errors on the Y axis increase as we rotate about the Z axis. And as we move forward on the X axis, errors increase on the Z axis. However, we can use these results as a first guess for other iterative methods such as in [14] if we are looking for more robust estimation. The strong point in this algorithm is its ability to give acceptable results in real time without any fear of de-convergence as in the iterative methods.

Real	R_z°	5	5	5	5
	T_x (m)	0	0	0	0
	T_y(m)	0	0	0	0
	T_z(m)	0	0	0	0
Estimated	R_z°	5	4	4	4.5
	T_x(m)	0.005	-0.005	-0.025	0
	T_y(m)	0.050	0.025	0.030	0
	T_z(m)	-0.015	0.005	-0.010	0

Table 4.2: Real and Estimated motions with Z-Rotation Only

To test the translation estimation solely, we moved the sonar in X, Y and Z directions at the same time, in each step we move the sonar 2.54 cm (1 inch) in each direction without any rotation. Table 4.3 shows the results of this experiment.

Real	R_z°	0	0	0	0
	T_x (m)	0.0254	0.0254	0.0254	0.0254
	T_y(m)	0.0254	0.0254	0.0254	0.0254
	T_z(m)	0.0254	0.0254	0.0254	0.0254
Estimated	R_z°	0	-0.5	-1	0.5
	T_x(m)	0.025	0.025	0.040	0.010
	T_y(m)	0.025	0.015	0.010	0.035
	T_z(m)	0.035	0.010	0.050	0.015

Table 4.3: Real and Estimated motions for 3-Directional translation.

It worth mentioning that we used Hough array with 0.005m cell's width; $\mathrm{N} = 14$; $\alpha = 0.60$; $\beta = 0.20$; and $\gamma = 0.90$ as parameters for the MDUD. Unfortunately, the setup doesn't allow for bigger steps in our controlled environment, so the results of these experiments do not reflect the performance of the algorithm in real movements, as the steps fall within the beams' resolution of the sonar, this is illustrated in Figure 4.13.

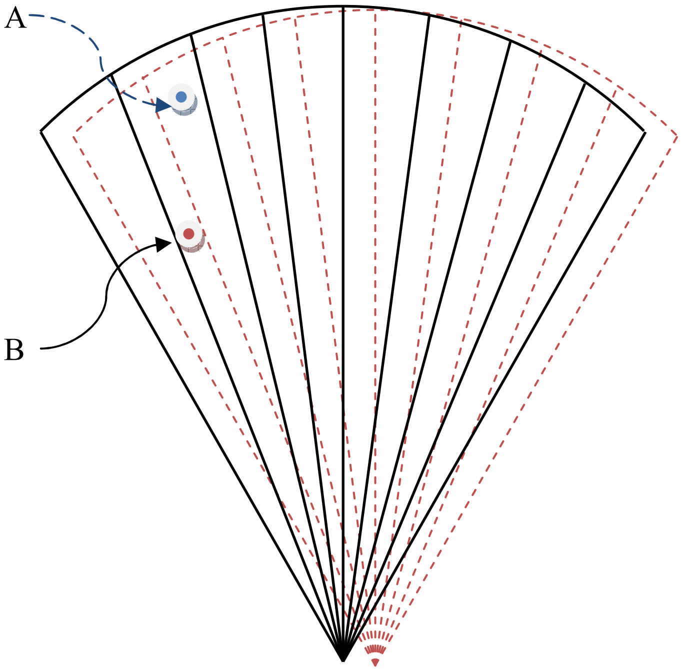
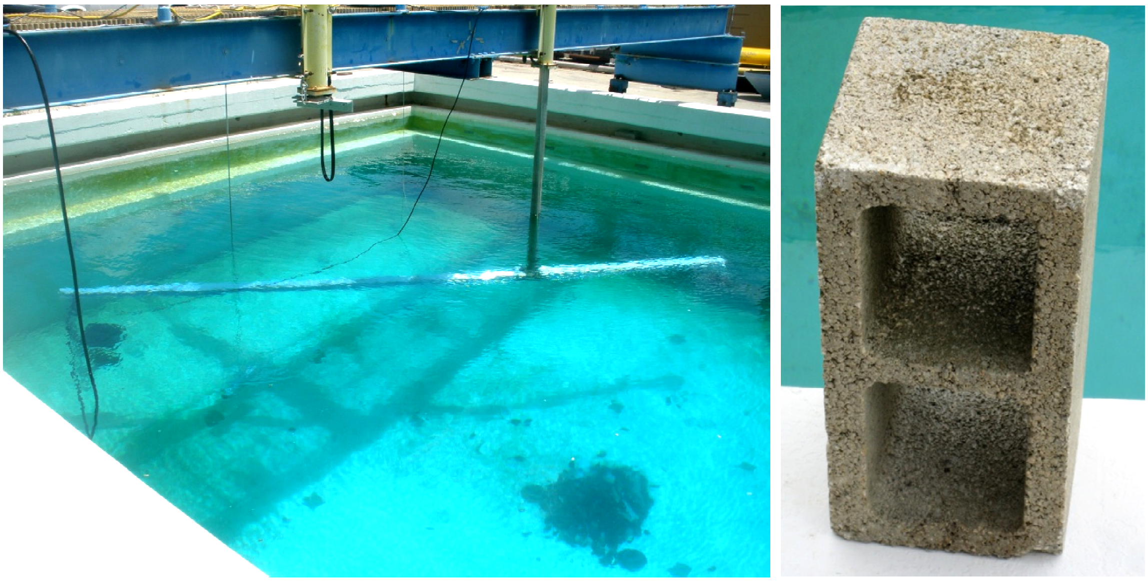


Figure 4.13: Small sonar movements cannot be detected as the change may or may not alter the position of the features in the image. For example, after the small movement of the sonar, the point A did not change its position in the acoustic image, while the point B changed its position from the second beam in the left to the first beam to the left. Please note that the figure is for demonstration purposes only and does not reflect true number of the beams.

To shed another light on the algorithm we have conducted another experiment with relatively longer track. In the second experiment we used DIDSON data obtained in Florida Atlantic University to measure the Z rotation angle, the experiment involved a DIDSON rotating full 360° counterclockwise around a cinder block.

First, the DIDSON was positioned at a 25° mounting angle from the horizontal. The mounting pole was then attached to the longer end of the cantilever beam in the Boca Raton Campus pool as illustrated in Figure 4.14. The cantilever beam was lowered into the pool so that the DIDSON was placed approximately 1.33m from the pool floor, and 2.76m away horizontally from the pivot point, giving a 3.07m slant distance from the DIDSON to the block as illustrated in Figure 4.15.



a) The sonar setup

b) The target

Figure 4.14: The DIDSON's position was set to zero on the gantry and was rotated around the target a full 360° counterclockwise

The viewing window parameters were changed to the values: Window Start: 1.88 m; Window Length: 2.25 m; Focus: 2.98 m. The recorded data contains 1400 frames, we used every 50th frame from them (see Figure 4.16 and Figure 4.17) to reconstruct the track of the sonar with its direction (blue arrows) as illustrated in Figure 4.18. The total length of the track is 17.34m which was reconstructed as 17.72m giving an error of +0.38m resembling 2.2% error. As we rotated the sonar manually with the help of a diver, the estimated positions are not equispaced. On the *Z* axis, the positions fluctuated between 0 and -0.35 m as illustrated in Figure 4.19.

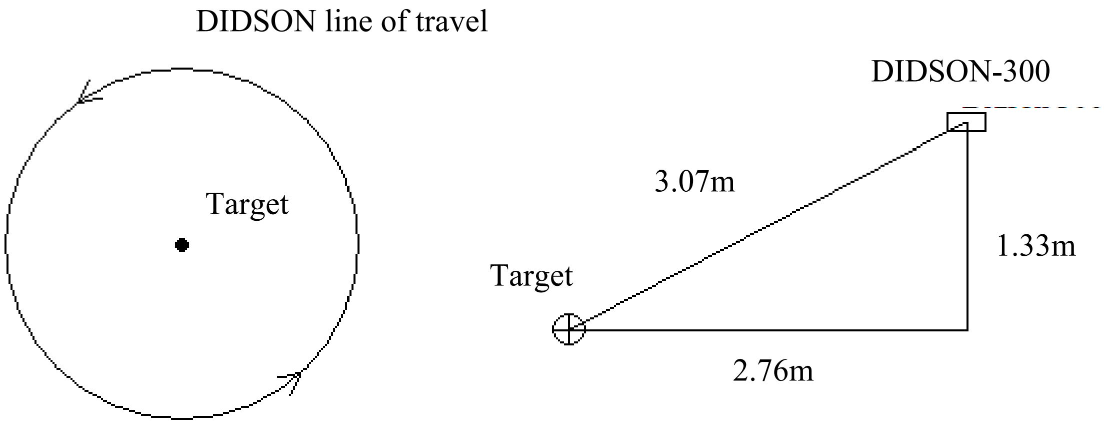


Figure 4.15: The track and geometry of the sonar/target

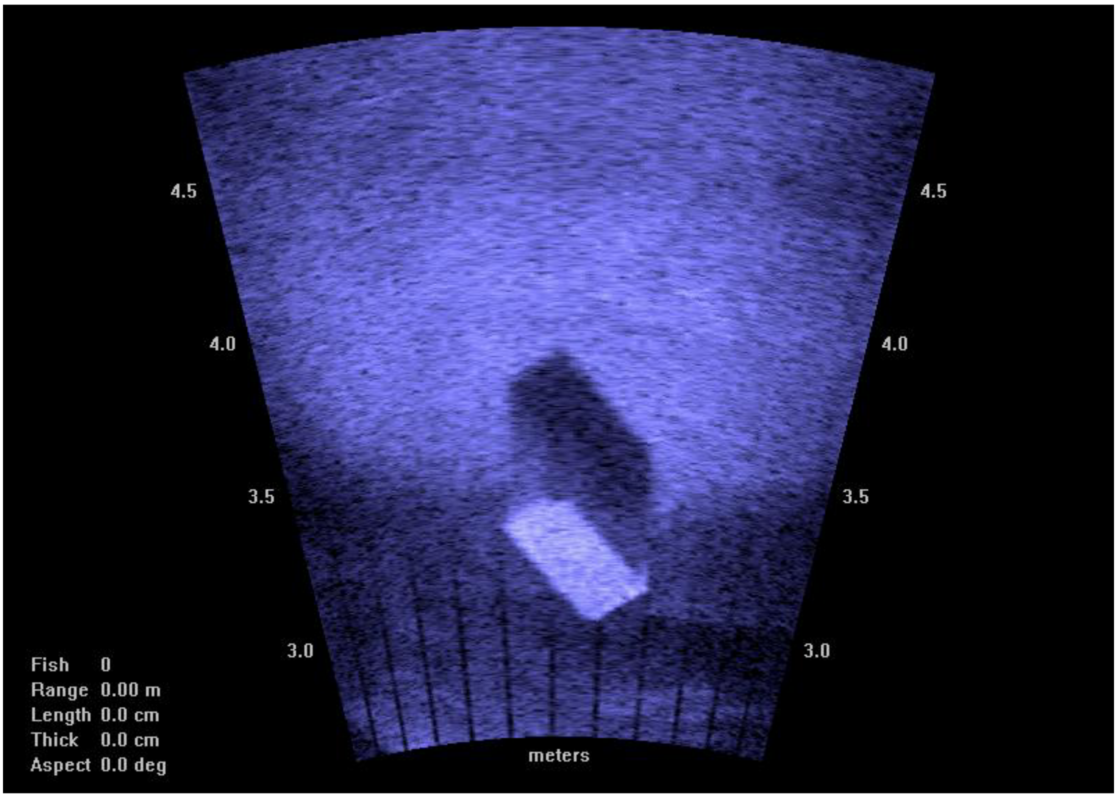


Figure 4.16: example frame of the imaged cinder block. (at 240°)

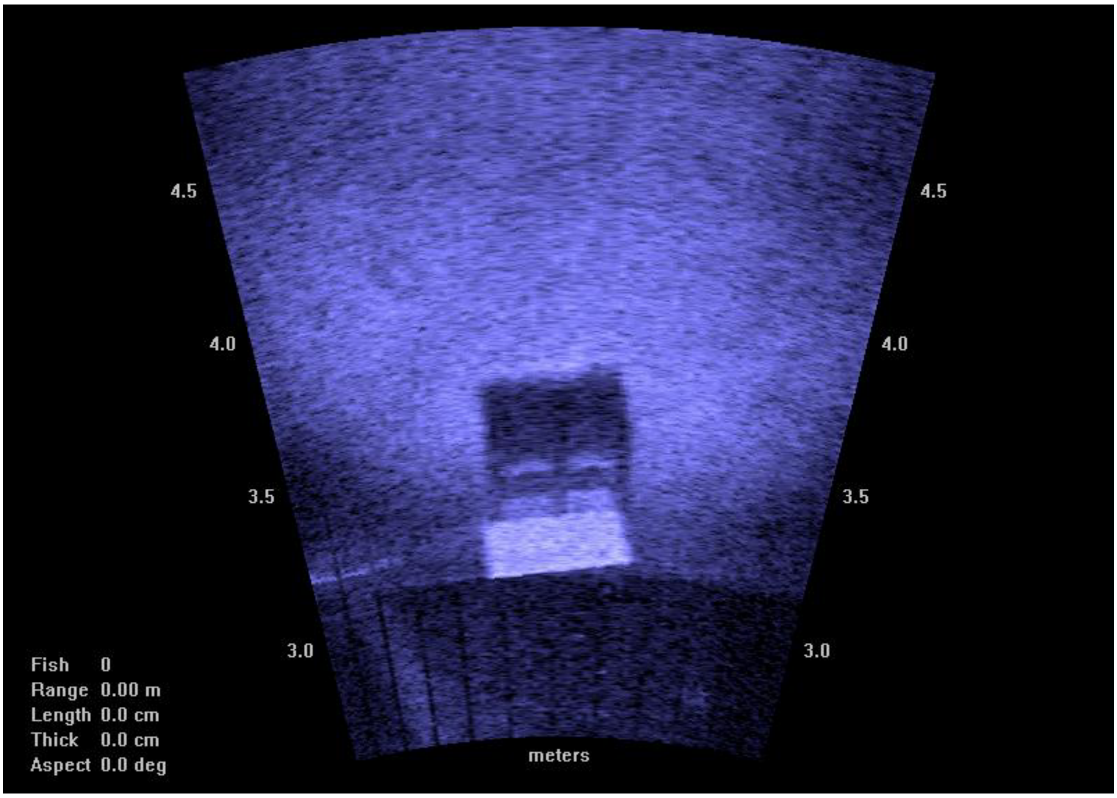


Figure 4.17: example frame of the imaged cinder block. (at 300°)

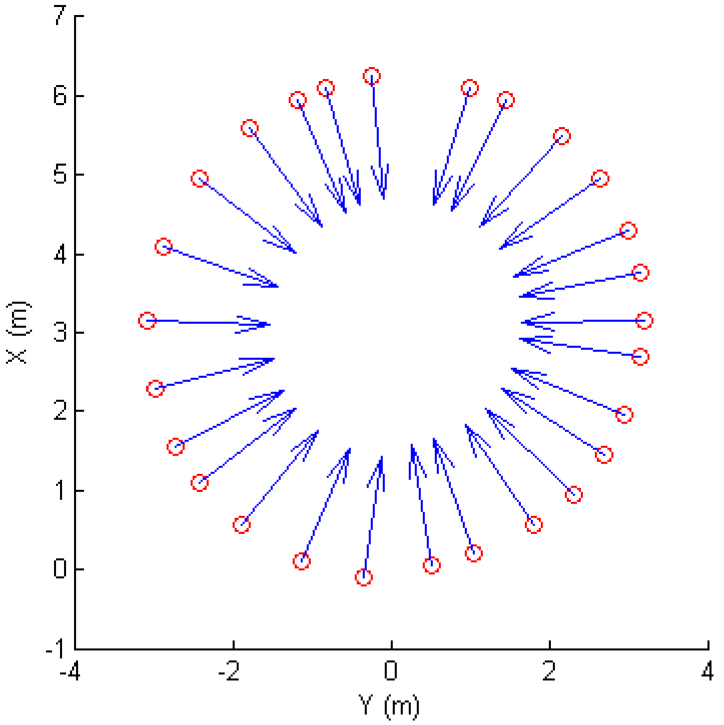


Figure 4.18: A reconstructed track for DIDSON sonar using the proposed method, The total length of the track is 17.34m which was reconstructed as 17.72m giving an error of +0.38m resembling 2.2% error.

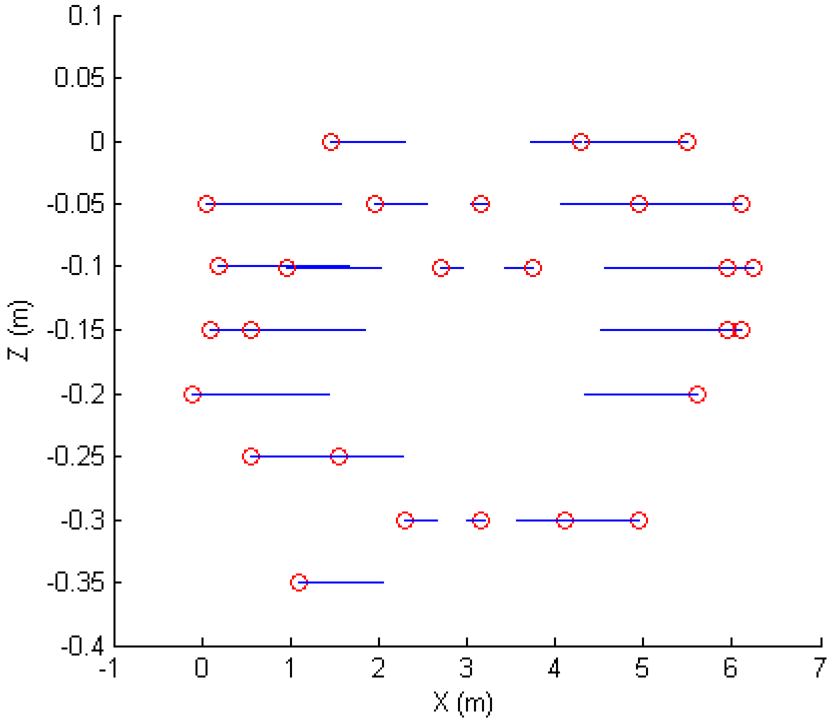


Figure 4.19: The reconstructed sonar positions in the XZ plane, the positions fluctuated between 0 and -35 cm.

4.7 Conclusion

In this chapter, we have presented a new framework to build motion estimation algorithm for underwater vehicles, the framework introduces new concepts including Modified Uniform Distribution and sonar arc sampling. The algorithm uses two Hough arrays to select the best answer for translation and rotation angle. Real experiments have been conducted to validate the methodology and acceptable results were obtained. The framework can be integrated with iterative motion estimation methods to provide good starting point for robust motion estimation.

Chapter 5
Motion Estimation Performance Analysis

5.1 Introduction

In addition to its numerous terrestrial applications, the capability to compute self (or a target's) 3-D motion is important for many underwater operations. While optical systems are the most common imaging modality above the sea, serious complexities arise when the mission is to be carried out beneath the sea surface, with turbidity being the most prohibiting factor.

Acoustic-based imaging and methods, because of the ability to penetrate silt and other sources of turbidity, offer a more effective solution. It is thus desired to devise robust motion vision techniques comparable to those realized for optical cameras. In this chapter, we explore the impact of selected key factors on the accuracy of motion estimation from 2-D sonar cameras. These factors comprise the field of view of the camera, the number of feature matches in a motion sequence, and the inaccuracy in the knowledge of their image positions. The significance of each factor is analyzed based on the variances of the sought after motion parameters. We also present various results from an experiment with real data, where we apply different measures that provide us with ground truth for the quantitative assessment of the motion estimation accuracy.

The estimation of 3-D motion from optical images has received extensive treatment in the motion vision literature for its numerous applications in the terrestrial domain. The domain-independent feature-based approaches have been applied effectively to realize many desired capabilities for underwater operation, e.g., optical station keeping, target-based positioning and photo-mosaicing, to name a few (see [151],[152] for a survey of relevant literature). Among various methods, the computation of 3-D motion relative to planar scenes arises in many underwater applications dealing with target detection and mapping over relatively flat surfaces, e.g., exploration of seabed habitats, ship hull inspection, as well as mine detection and localization. Unfortunately, turbidity or low visibility can impede the utility or prohibit the deployment of traditional optical cameras in certain environments, e.g., polluted harbours and certain marine sanctuaries. Here, acoustic-based imaging and methods offer a key advantage, because of the ability to penetrate silt and other sources of turbidity. In particular, a new generation of high-frequency 2-D forward-looking (FL) sonar systems offers a reliable technology for high-resolution imaging at near-video data rate [7][72]. In optical imaging, the epipolar geometry of projections rays from two or more views offer the framework for 3-D scene or motion interpretation. When moving rigidly in the world, a linear mapping describes the transformation of the coordinates of the planar surface points from one viewing position to the next. The underlying transformation is the homography between two perspective views of the scene plane [8], which can be fixed by the correspondences of minimum 4 points in the two views [153] [154].

In contrast, range and azimuth comprise the measurements of 3-D scene surface features in 2-D FL imaging. Imaging a plane with a 2-D FL sonar, the geometric transformation from one view to the next depends on 1) the linear mapping between the Cartesian coordinate systems of the two views, namely, the affine components from the homography of two perspective views; 2) the unknown elevations of the features being tracked [155]. Being dependent on the target feature point’s elevation, the transformation varies over the image. The nature of sonar imaging and available measurements can introduce some inherent complexities in the estimation of 3-D motion from the frame-to-frame feature displacements, when analyzing sonar video or motion sequences.

An iterative method based on a maximum likelihood estimation (ML) framework has been proposed for the computation of 3-D motion from a pair of 2-D sonar views [14]. Results from the reported experiments, although, limited to a small number of real data sets, identified certain complexities and bottlenecks in the accurate estimation of 3D motion. Among these, primary factors include the inherent low signal-to-noise ratio of 2-D sonar imagery, limited number of features that can be detected and matched in a motion sequence, and the very small field of view (FOV) of the earlier cameras (less than 30° horizontal FOV). The latter may no longer be a key factor with the advent of new models and technologies that offer much larger FOVs (as large as 130°). However, it is still desired to explore and quantify the impact of these factors on the motion estimation accuracy.

In this chapter, we study the significance of each factor based on the variances of the sought after motion parameters. We analyze the results from a large number of numerical simulations with random data, and the typical parameters of existing 2-D sonar imaging systems. Our results offer quantitative measures, collectively in agreement with earlier observations and conjecture on the role of the named influential factors. In addition, we also present various results from an experiment with a real data set, where we apply different measures to establish ground truth for the quantitative assessment of the motion estimation accuracy. The results highlight some advantages offered in the estimation of 3-D motion by sonar imaging.

5.2 Background concepts

5.2.1 Coordinate systems

A 3-D point $\mathrm{P_s}$ has coordinates $\mathrm{P_s} = (\mathrm{X_s}, \mathrm{Y_s}, \mathrm{Z_s})^{\mathrm{T}}$ in the world reference system. For the sonar view, we employ the spherical coordinates $[\mathcal{R}, \theta, \phi]^{\mathrm{T}}$ comprising the range, azimuth angle and elevation angle. The point $\mathrm{p_s} = (\mathrm{x_s}, \mathrm{y_s}) = \mathcal{R}(\sin\theta, \cos\theta)$ in the zero-elevation plane of sonar coordinate system may be viewed as the 2-D sonar projection of a 3-D point P. The relationship between the Cartesian and spherical coordinates is useful in analyzing the sonar data:

$$\mathrm{P_s} = \begin{bmatrix} \mathrm{X_s} \\ \mathrm{Y_s} \\ \mathrm{Z_s} \end{bmatrix} = \mathcal{R} \begin{bmatrix} \cos\phi\sin\theta \\ \cos\phi\cos\theta \\ \sin\phi \end{bmatrix} \tag{5.1}$$

$$\begin{bmatrix} \mathcal{R} \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{\mathrm{X_s^2} + \mathrm{Y_s^2} + \mathrm{Z_s^2}} \\ \tan^{-1}(\mathrm{X_s}/\mathrm{Y_s}) \\ \tan^{-1}\left(\mathrm{Z_s}/\sqrt{\mathrm{X_s^2} + \mathrm{Y_s^2}}\right) \end{bmatrix} \tag{5.2}$$

5.2.2 Sonar image model

A 2-D sonar image $\mathrm{I}(\mathcal{R}, \theta)$ represents acoustic reflections from 3-D points at ranges $\mathcal{R}$ (within a down-range window $[\mathcal{R}_{\mathrm{min}}, \mathcal{R}_{\mathrm{max}}]$) and azimuth direction θ (within cross-range field of view $[-\theta_{\mathrm{max}}: \theta_{\mathrm{max}}]$). The elevation angle ϕ, unknown in a 2-D imaging sonar, is constrained by the vertical width of each transmitted beam $[-\phi_{\max}, \phi_{\max}]$, which is 7° in DIDSON sonar [6] and 10° in Blue View P900-130 [7].

5.2.3 *Coordinate transformation*

The transformation between the coordinate systems at viewing positions can be defined in terms of 3 × 3 rotation matrix R and 3-D translation vector t:

$$\mathrm{P}' = \mathrm{RP} + \mathrm{t} \tag{5.3}$$

Generally, the transformation parameters $\{\mathrm{R}, \mathrm{t}\}$ describe the motion of the observer or imaging system – a sonar camera here – from one position to the next.

5.2.4 *Plane representation*

A point P on a plane Π with normal n satisfies the dot product equation: $\mathrm{P}.\mathrm{n} = -1$. The planar surface normal can be determined in the coordinate system of the second view from the equation in (5.9) which is derived as follows:

$$n'.P' = -1 \qquad \Rightarrow \qquad n'\,{P'}^{T} = -1 \tag{5.4}$$

$$n'(RP + t)^{T} = -1 \quad \Rightarrow \quad n'(P^{T}R^{T} + t^{T}) = -1 \tag{5.5}$$

$$\leftarrow\times\, Rn \qquad \Rightarrow \qquad n'(P^{T}R^{T}Rn + t^{T}Rn) = -Rn \tag{5.6}$$

$$R^{T}R = I \qquad \Rightarrow \qquad n'(P^{T}n + t^{T}Rn) = -Rn \tag{5.7}$$

$$P^{T}n = -1 \quad \Rightarrow \quad n'(-1 + t^{T}Rn) = -Rn \tag{5.8}$$

$$\mathrm{n}' = \left(\frac{1}{1 - t^{T}Rn}\right)Rn \tag{5.9}$$

A general analogy with classical motion vision of two optical views can be established by noting that the elevation angle ϕ can be expressed in terms of the surface normal, and two coordinate $\{\mathcal{R}, \theta\}$. More precisely, we can first express the surface equation in the form (5.10).

$$(\mathrm{n_x} \sin\theta + \mathrm{n_y} \cos\theta) \cos\phi + \mathrm{n_z} \sin\phi = -\frac{1}{\mathcal{R}} \tag{5.10}$$

Which enables us to solve for the elevation angle ϕ in terms of the surface normal:

$$\phi = -\gamma + \eta \tag{5.11}$$

Where

$$\gamma = \tan^{-1}\left(\frac{\mathrm{n_x} \sin\theta + \ \mathrm{n_y} \cos\theta}{\mathrm{n_z}}\right) \tag{5.12}$$

$$\eta = \sin^{-1}\left(\frac{-1}{\mathcal{R}\sqrt{\left(\mathrm{n_x} \sin\theta + \mathrm{n_y} \sin\theta\right)^2 + \mathrm{n_z^2}}}\right) \tag{5.13}$$

5.2.5 Image-to-image transformation of planar points

Using the plane equation $\mathrm{P.n} = -1$, we can write (5.3) in the form

$$\mathrm{P_s'} = \left(\mathrm{R} - \mathrm{tn^T}\right)\mathrm{P_s} = \mathrm{QP_s} \tag{5.14}$$

Where $\mathrm{Q} = \mathrm{R} - \mathrm{tn^T}$describes the underlying linear transformation between the coordinates of a 3-D points in reference frames of two viewing points. We can derive the frame-to-frame transformation of 2-D sonar image points $p_s(x_s, y_s)$, $\mathrm{p_s'}$ (x'_s, y'_s) described in (5.15) and (5.16) [17] . The transformation is given by (5.17).

$$(x_s, y_s) = \mathcal{R}(\sin\theta, \cos\theta) \tag{5.15}$$

$$(x'_s, y'_s) = \mathcal{R}'(\sin\theta', \cos\theta') \tag{5.16}$$

$$\begin{bmatrix} x'_s \\ y'_s \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha q_{11} & \alpha q_{12} & \beta q_{13} \\ \alpha q_{21} & \alpha q_{22} & \beta q_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = H \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} \tag{5.17}$$

Where $\alpha = \cos\phi / \cos\phi'$, $\beta = \mathcal{R}\sin\phi / \cos\phi'$ and q_{ij} is the $\{i, j\}$ element of 3×3 linear transformation matrix $Q = (R - tn^T)$.

It is important to note that the sonar rotation involves only 3 unknowns, expressed by the 3×1 vector representation $\omega = (\omega_x, \omega_y, \omega_z)^T$. Here, $R = R_x R_y R_z$, where $R_u (u = x, y, z)$ denotes 3×3 matrix representing rotation by ω_u about u axis.

It trivially follows that the homography in (5.17), while a complex nonlinear constraint, can be expressed in terms of the 9 motion and surface parameters $\{t, \omega, n\}$. A distinct difference with traditional two-frame motion problem is the fact that no scale factor ambiguity exists with two sonar views. In other words, projections into two views of a small number of features allow us to determine the unknown motion and plane parameters. Simple count reveals that a minimum of 5 points is necessary, providing us with 10 constraints in terms of 9 unknowns. If the motion is modelled by pure translation, a minimum of 3 points is necessary for a solution. While such issues as ambiguous configurations, number of possible solutions, etc., are intriguing theoretical problems that also provide insight into solution degeneracies, we are interested with analyzing factors affecting the motion estimation method adopted in [17], these factors include localization accuracy of feature positions, field of view, and the number of points.

5.3 Analyzing the motion parameters

Equation (5.17) comprises two constraints relating the correspondence of points in two sonar views to the sonar motion and parameters of the planar scene i.e.:

$$x_s' = \alpha(q_{11}x_s + q_{12}y_s) + \beta\, q_{13} \tag{5.18}$$

$$y_s' = \alpha(q_{21}x_s + q_{22}y_s) + \beta\, q_{23} \tag{5.19}$$

From this, variations in $(dx_s', dy_s', dx_s, dy_s)$ can be expressed in terms of variations in the motion and scene parameters, namely, $dt_x, dt_y, dt_z, d\vartheta_x, d\vartheta_y, d\vartheta_z, dn_x. dn_y, dn_z$. By reversing this relationship, we can analyze the variations in the motion and surface parameters due to uncertainty in the correspondences and their image positions. More precisely, given that the sonar data is rather noisy and image positions of sonar features cannot be established perfectly, it is important to know the impact on the motion estimation accuracy. Some tedious algebra in derivation of the underlying equations can be avoided by making use of computer symbolic processing, a single equation can reach up to three pages length. Still, the expressions are complicated due to highly nonlinear nature of the frame-to-frame transformation model in (5.17) and a large number of intermediate parameters. For simplicity, we express the nonlinear system of sensitivity equations in terms of some of the intermediate parameters, and then give the final form.

Differentiating the equations (5.18) (5.19) yields to the relation between the derivatives of $(dx_s', dy_s', dx_s, dy_s)$ and all other motion derivatives and the derivatives of the surface vector i.e. $dt_x, dt_y, dt_z, d\vartheta_x, d\vartheta_y, d\vartheta_z, dn_x. dn_y, dn_z$, finding these relations is extremely tedious due to the nonlinear nature of the equations involved, to overcome this we used Matlab symbolic processor to find these derivatives by feeding it with the essential equations discussed above.

$$\begin{aligned} dx_s' = {} & (q_{11}x_s + q_{12}y_s)d\alpha + (q_{13})d\beta + (\alpha q_{11})dx_s + \\ & (\alpha\, q_{12})dy_s + (\alpha x_s)dq_{11} + (\alpha y_s)dq_{12} + (\beta)dq_{13} \end{aligned} \tag{5.20}$$

$$\begin{aligned} dy'_s = (q_{21}x_s + q_{22}y_s)d\alpha + (q_{23})d\beta + (\alpha q_{21})dx_s + \\ (\alpha\, q_{22})dy_s + (\alpha x_s)dq_{21} + (\alpha y_s)dq_{22} + (\beta)dq_{23} \end{aligned} \tag{5.21}$$

Equations (5.20) (5.21) illustrate the first step to find these derivatives; these two equations can be re-written in the form

$$\begin{aligned} (q_{11}x_s + q_{12}y_s)d\alpha + (q_{13})d\beta + (\alpha x_s)dq_{11} + (\alpha y_s)dq_{12} + \ (\beta)dq_{13} \\ = dx'_s - \ \alpha(q_{11}dx_s + q_{12}dy_s) \end{aligned} \tag{5.22}$$

$$\begin{aligned} (q_{21}x_s + q_{22}y_s)d\alpha + (q_{23})d\beta + (\alpha x_s)dq_{21} + (\alpha y_s)dq_{22} + (\beta)dq_{23} \\ = dy'_s - \ \alpha(q_{21}dx_s + q_{22}dy_s) \end{aligned} \tag{5.23}$$

The left hand side of the equations (5.22) and (5.23) was written in terms of nine derivatives, six derivatives for the motion and three for the surface normal i.e. these two equations can be rearranged in the form

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{19} \\ a_{21} & a_{22} & \cdots & a_{29} \end{bmatrix} M = \begin{bmatrix} dx'_s - \alpha(q_{11}dx_s + q_{12}dy_s) \\ dy'_s - \alpha(q_{21}dx_s + q_{22}dy_s) \end{bmatrix} \tag{5.24}$$

Where

$$M = [dt_x \ \ dt_y \ \ dt_z \ \ dn_x \ \ dn_y \ \ dn_z \ \ d\vartheta_x \ \ d\vartheta_y \ \ d\vartheta_z]^T \tag{5.25}$$

$t = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T$ is the translation vector, $n = \begin{bmatrix} n_x \, n_y \, n_z \end{bmatrix}^T$ is the normal vector, $\vartheta = \begin{bmatrix} \vartheta_x \, \vartheta_y \, \vartheta_z \end{bmatrix}^T$ is the rotation vector for the rotation angles around the x, y and z axes. The rotation matrix can be obtained from (5.29).

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\vartheta_x & -\sin\vartheta_x \\ 0 & \sin\vartheta_x & \cos\vartheta_x \end{bmatrix} \tag{5.26}$$

$$\mathrm{R_y} = \begin{bmatrix} \cos\vartheta_{\mathrm{y}} & 0 & \sin\vartheta_{\mathrm{y}} \\ 0 & 1 & 0 \\ -\sin\vartheta_{\mathrm{y}} & 0 & \cos\vartheta_{\mathrm{y}} \end{bmatrix} \tag{5.27}$$

$$\mathrm{R_z} = \begin{bmatrix} \cos\vartheta_{\mathrm{z}} & -\sin\vartheta_{\mathrm{z}} & 0 \\ \sin\vartheta_{\mathrm{z}} & \cos\vartheta_{\mathrm{z}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.28}$$

$$\mathrm{R} = \mathrm{R_x R_y R_z} \tag{5.29}$$

As for motion estimation, $N \geq 5$ correspondences are necessary for the quantitative assessment of the estimation accuracy. However, achieving higher accuracy requires employing as many accurate correspondences as we can establish in the two sonar views. Thus, one factor to be examined in our analysis is how the estimation robustness depends on the number of matched features over the sonar field of view, and the advantages in deploying a sonar with a larger FOV. If we have N pairs of correspondences between two images, then we have 2N equations, forming the equation yield to (5.30).

$$\begin{bmatrix} \alpha_{11}^{1} & \alpha_{12}^{1} & \cdots & \alpha_{19}^{1} \\ \alpha_{21}^{1} & \alpha_{22}^{1} & \cdots & \alpha_{29}^{1} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{11}^{N} & \alpha_{12}^{N} & \cdots & \alpha_{19}^{N} \\ \alpha_{21}^{N} & \alpha_{22}^{N} & \cdots & \alpha_{29}^{N} \end{bmatrix} M = \begin{bmatrix} {\mathrm{dx}'_{\mathrm{s}}}^{1} - \mathrm{\alpha}(\mathrm{q}_{11}\mathrm{dx}_{\mathrm{s}}^{1} + \mathrm{q}_{12}\mathrm{dy}_{\mathrm{s}}^{1}) \\ {\mathrm{dy}'_{\mathrm{s}}}^{1} - \mathrm{\alpha}(\mathrm{q}_{21}\mathrm{dx}_{\mathrm{s}}^{1} + \mathrm{q}_{22}\mathrm{dy}_{\mathrm{s}}^{1}) \\ \vdots \\ {\mathrm{dx}'_{\mathrm{s}}}^{\mathrm{N}} - \mathrm{\alpha}\left(\mathrm{q}_{11}\mathrm{dx}_{\mathrm{s}}^{\mathrm{N}} + \mathrm{q}_{12}\mathrm{dy}_{\mathrm{s}}^{\mathrm{N}}\right) \\ {\mathrm{dy}'_{\mathrm{s}}}^{\mathrm{N}} - \mathrm{\alpha}\left(\mathrm{q}_{21}\mathrm{dx}_{\mathrm{s}}^{\mathrm{N}} + \mathrm{q}_{22}\mathrm{dy}_{\mathrm{s}}^{\mathrm{N}}\right) \end{bmatrix} \tag{5.30}$$

Or simply:

$$\mathrm{A.M} = \mathrm{B} \tag{5.31}$$

Where A is $2\mathrm{N} \times 9$ matrix, M is 9×1 vector and B is $2\mathrm{N} \times 1$ vector, to find M, we used singular value decomposition [30] to calculate the pseudo-inverse of A i.e.

$$\mathrm{M} = \mathrm{V} * \mathrm{S}^{-1} * \mathrm{U}^{\mathrm{T}} * \mathrm{B} \tag{5.32}$$

Where:

$$\mathrm{A} = \mathrm{U} * \mathrm{S} * \mathrm{V}^{\mathrm{T}} \tag{5.33}$$

The pseudo-inverse method gives the solution for the M vector in a least squares sense. The inverse of S can be calculated by replacing each singular value by its reciprocal if it is bigger than zero, otherwise it will remain zero.

The condition number of A -which is the ratio of its largest singular value to its smallest singular value- was used to reject ill-conditioned cases, large condition number indicates an ill-conditioned case, where small condition number reveals well-contained linear system, the threshold is subjective, the highest threshold we used is 5500, the lowest is 2000.

5.4 Experiments

We have carried our experiments with synthetic data to assess the robustness and accuracy in motion estimation, and variations with number of features, size of the sonar field of view, and accuracy in localizing image features. We then present the results from an experiment with real data, to check agreement with our theoretical analysis.

5.4.1 Sensitivity analysis with synthetic data

We examine the solution for variations in M as a result of variations in the position of correspondences, encoded in B. We analyze the motion uncertainty in terms of various factors, including the number of features N, the size of the sonar field of view where these features are distributed, and the uncertainty in correspondence positions, represented by the standard deviation σ_s ($\epsilon\{\delta x_s^2\} = \epsilon\{\delta y_s^2\} = \epsilon\{\delta x'^2_s\} = \epsilon\{\delta y'^2_s\} = \sigma_s$ without loss of generality).

To do this, we assume arbitrary motions and planar scene surfaces, and create random correspondences; we choose N random points $\mathrm{p_s}$ in the first view, compute the correspondences $\mathrm{p'_s}$ according to (5.17). We randomly determine a zero-mean noise sample $\{\delta x_s^i, \delta y_s^i, \delta x'^i_s, \delta y'^i_s\}$ $(i = 1..N)$ from zero-mean Gaussian distribution with standard deviation σs. We compute M from (5.32) , then repeat the process 500 times, and determine the mean and standard deviation of M. In our analysis, we have eliminated a small number of outliers that often arise due to some degeneracies (e.g., most of the N points are clustered around a line). This is done by setting a threshold on the condition number of A, defined in terms of the ratio of the largest to the smallest non-zero singular value (threshold is adjusted according to the mean of the condition numbers in each case).

Figure 5.1and Figure 5.2 depict the variation in the mean and standard deviation of the motion estimation error for a range of feature location errors, Figure 5.3, Figure 5.4, Figure 5.5 and Figure 5.6 illustrate the relation between the size of the sonar field of view and the mean and standard deviation of the motion estimation error with different feature uncertainty settings, Figure 5.7 and Figure 5.8 relate the number of correspondences to the mean and standard deviation of the motion estimation error. In the experiments of the Figure 5.1 up to Figure 5.8, the planar scene features have been selected in the range of 2.5m- 5.0m and the motion is varied randomly in the range ± 0.4m in translation and $\pm 20°$ for each axis (recall that $\mathrm{R} = \mathrm{R_x R_y R_z}$) as stated in (5.29).

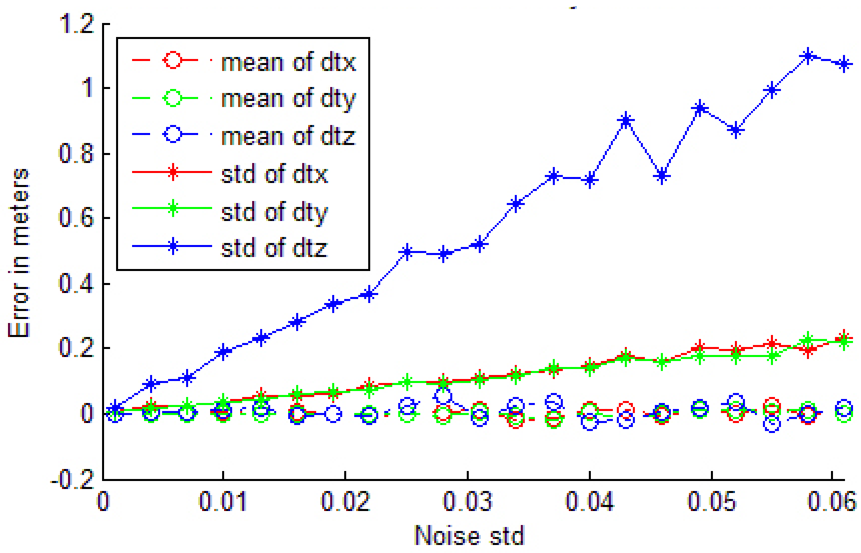


Figure 5.1: The relation between the feature location uncertainty [m] and dt_x, dt_y, dt_z [m]; No Features=25, Half-Angle FOV=65°; target features are assumed in the range 2.5-5 [m]. While the noise increases in the feature locations, The errors in estimating the translation increase, particularly, translations on the Z axis.

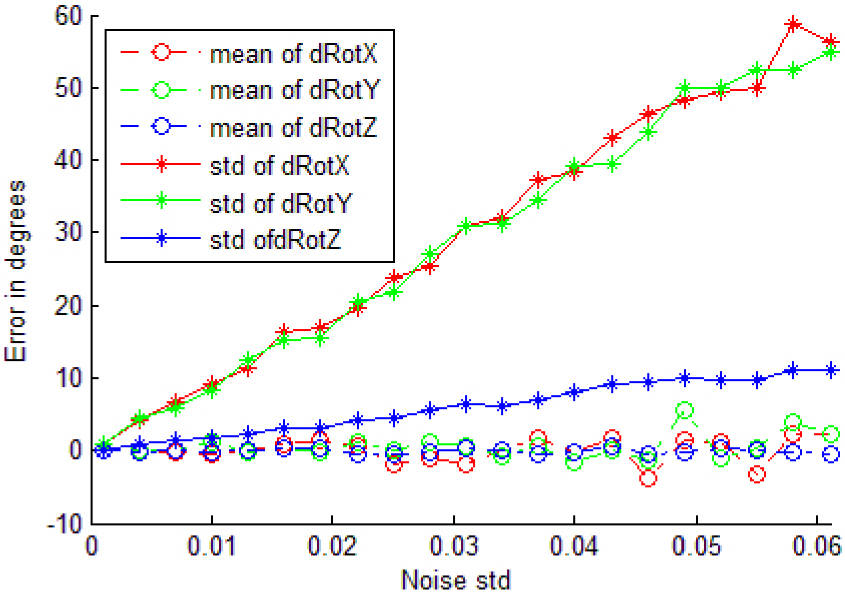


Figure 5.2: The relation between the feature location uncertainty [m] and $d\vartheta_x, d\vartheta_y, d\vartheta_z$ [deg]; No Features=25, Half-Angle FOV=65°; target features are assumed in the range 2.5-5 [m]. While the noise increases in the feature locations, The errors in estimating the rotation increase, particularly, rotations about the X and Y axes.

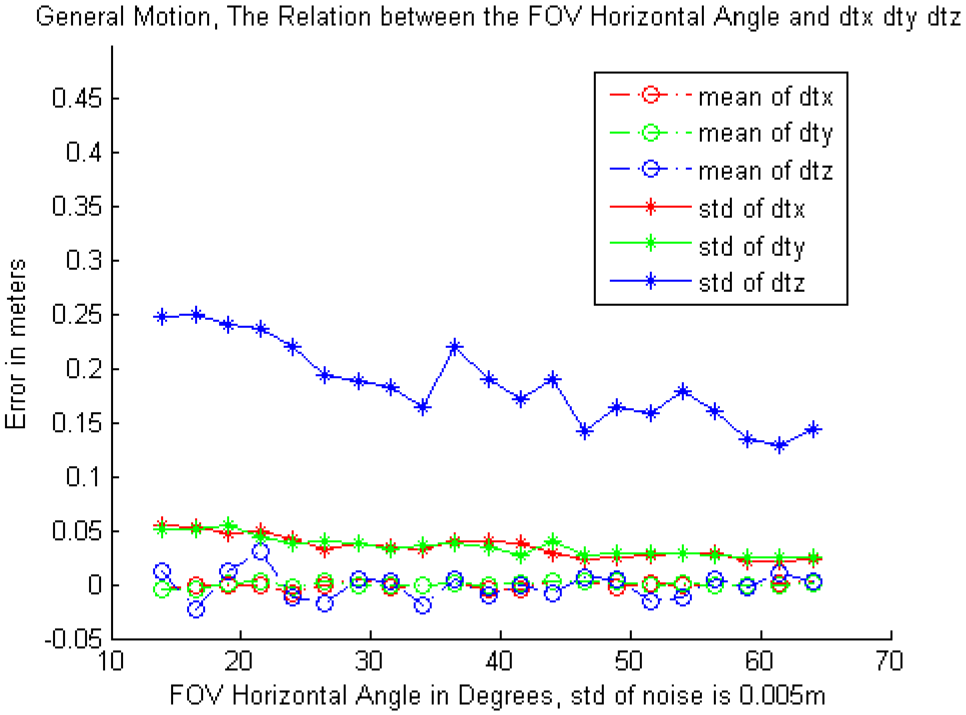


Figure 5.3: The relation between the field of view [deg] and $\mathrm{dt_x}, \mathrm{dt_y}, \mathrm{dt_z}$ [m] (No Features=25, Feature position uncertainty = 0.005m; target features are assumed in the range 2.5-5 [m]. The bigger the FOV angle, the better the estimation of the translation.

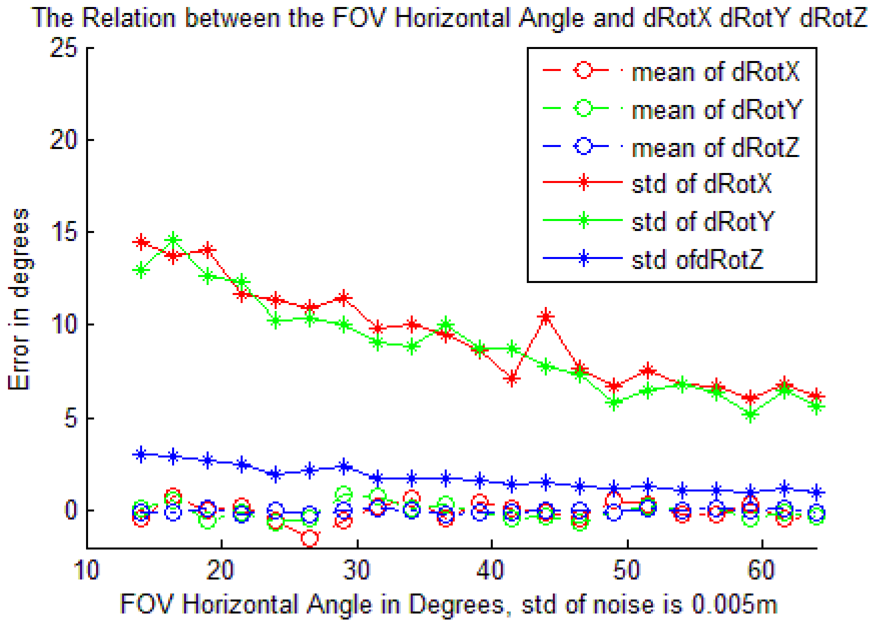


Figure 5.4: The relation between the field of view [deg] and $d\vartheta_x, d\vartheta_y, d\vartheta_z$ [deg] No Features=25, Feature position uncertainty = 0.005m; target features are assumed in the range 2.5-5 [m]. The bigger the FOV angle, the better the estimation of the rotation.

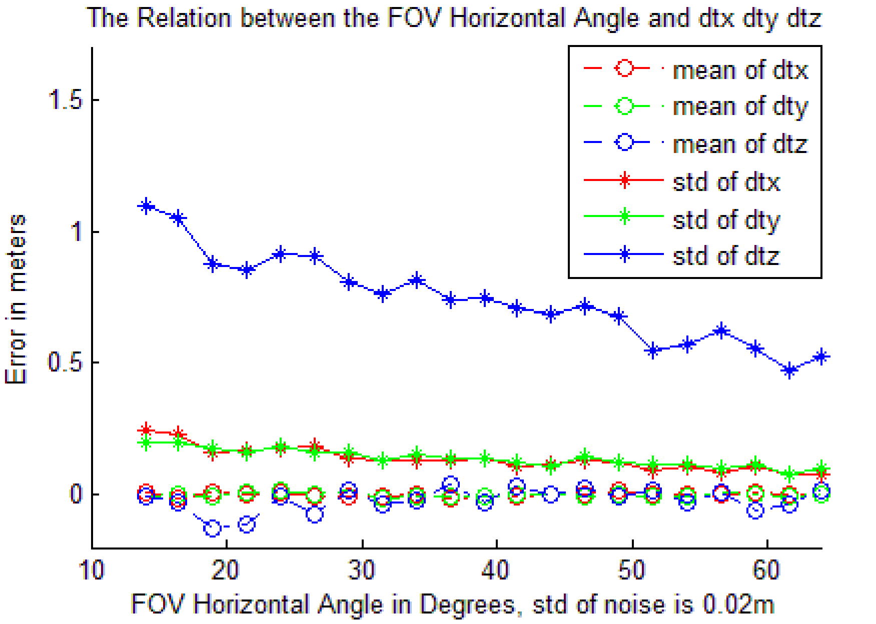


Figure 5.5: The relation between the field of view [deg] and dt_x, dt_y, dt_z [m] No Features=25, Feature position uncertainty = 0.02m; target features are assumed in the range 2.5-5 [m]. The bigger the FOV angle, the better the estimation of the translation. Also, in this figure the noise is higher compared to that in Figure 5.3 which increases the errors on the xyz axes compared to the earlier errors in Figure 5.3.

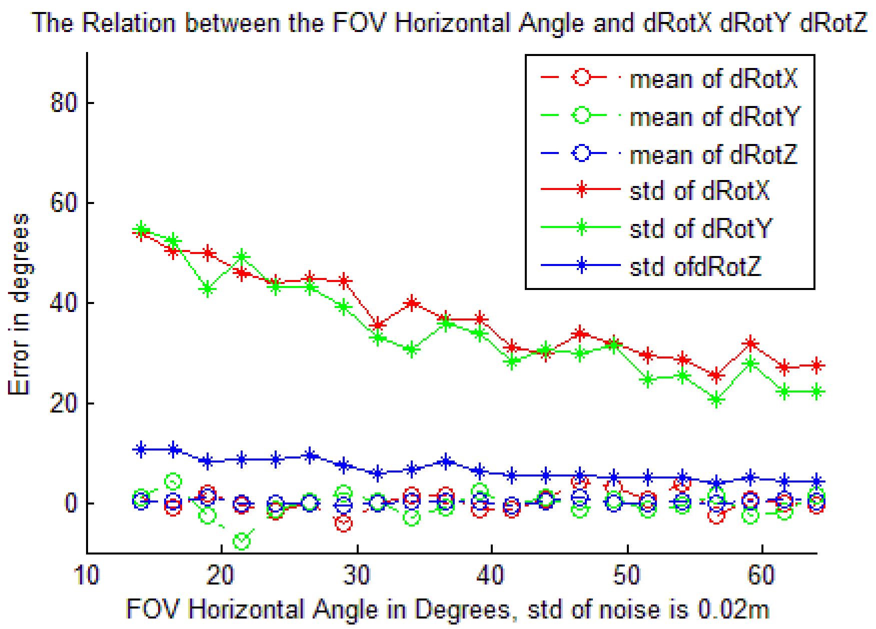


Figure 5.6: The relation between the field of view [deg] and $d\vartheta_x, d\vartheta_y, d\vartheta_z$ [deg] No Features=25, Feature position uncertainty = 0.02m; target features are assumed in the range 2.5-5 [m]. The bigger the FOV angle, the better the estimation of the rotation. Also, in this figure the noise is higher compared to that in Figure 5.4 which increases the errors on the xyz axes compared to the earlier errors in Figure 5.4.

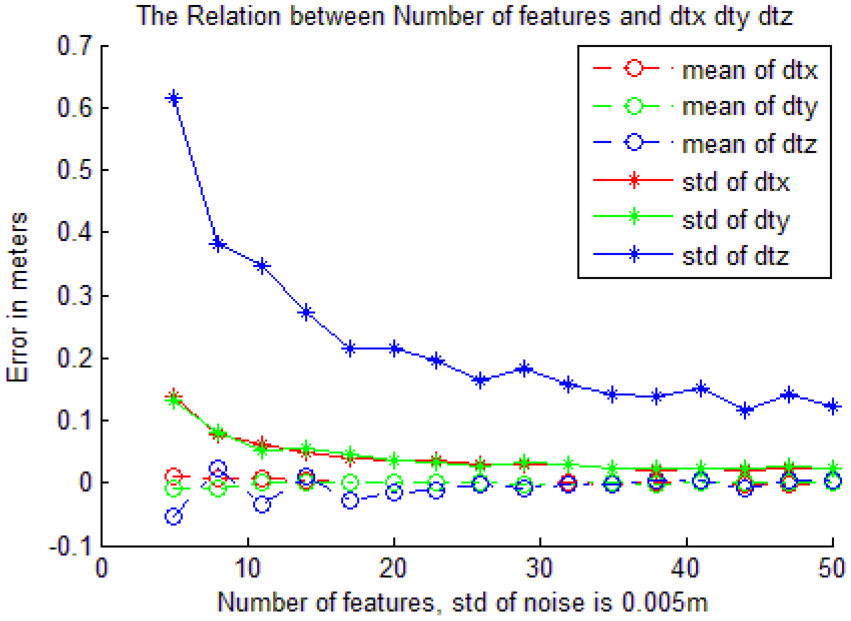


Figure 5.7: The relation between the number of features and dt_x, dt_y, dt_z [m] Feature position uncertainty = 0.005m, Half-Angle FOV=65°; target features are assumed in the range 2.5-5 [m]. As the number of features increases, the errors in estimating the translation decrease.

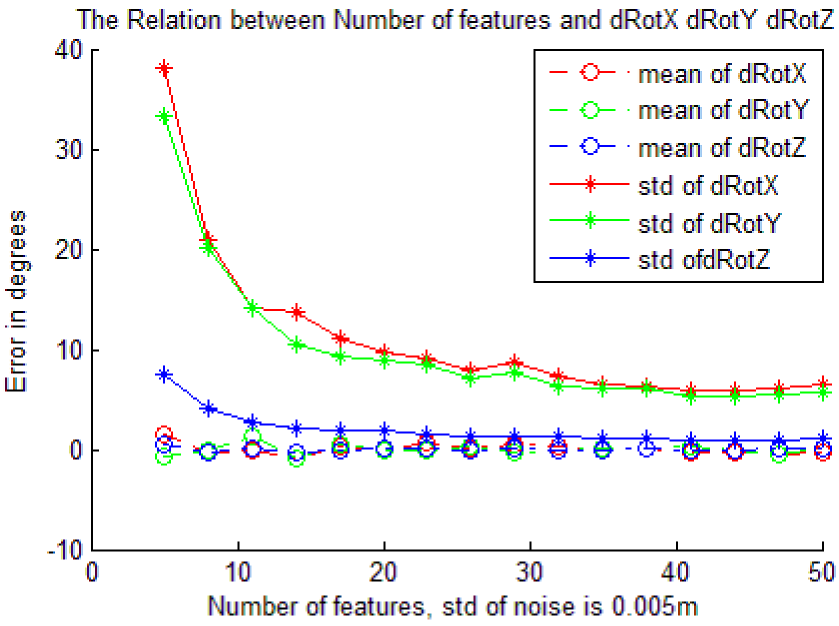


Figure 5.8: The relation between the number of features and $d\vartheta_x, d\vartheta_y, d\vartheta_z$ [deg] Feature position uncertainty = 0.005m, Half-Angle FOV=65°; target features are assumed in the range 2.5-5 [m]. As the number of features increases, the errors in estimating the rotation decrease.

The sensitivity to the field of view has been computed for two different pixel localization uncertainties of 0.5cm and 2cm. It is noted that there is a large discrepancy between translations and rotations in the *XY* and *Z* directions. Here, the sensitivity is higher for *Z* translations while rotation uncertainty is higher for *XY* rotations. This translation uncertainty is intuitive due to elevation ambiguity, and the fact that 3-D points maps onto the zero-elevation plane along circular arc (two points with different elevations along the same circular arc map onto the same image points). The less uncertainty in *Z* rotation can be recognized by noting that this impacts the azimuth (bearing) of the 3-D point which is directly measured with good resolution (0.3° beam-width in DIDSON).

It is noted that the number of feature correspondences is tied to the scene complexity and the feature position localization accuracy depends on a number of factors, including but not limited to the scene surface complexity, sonar properties, and environmental conditions. Unlike the optical cameras where the FOV can be readily adjusted with the lens properties over a range of angles, there is a little flexibility with the sonar cameras due to limitations of a small number existing technologies. Now, given that the field of view has a large impact on estimation accuracy/robustness, it is critical to use a camera with the largest field of view, e.g., a Blueview P900-130 blazed-array sonar with Half-angle FOV of 65° (extreme point on the FOV sensitivity graphs, and the assumed FOV in other plots).

We have repeated the sensitivity analysis with passive navigation over the seabed; see Figure 5.9 and Figure 5.10. The specific parameters are chosen according to the real data in our next experiment, collected with a Blueview P900-130 in the marina of the Florida Atlantic University- Sea Technology Centre, Dania Beach, FL. The feature positions vary randomly over the 2m-30m range. The sonar half-vertical beam width is 10° and half-angle horizontal field of view is 65°, as stated in [7].

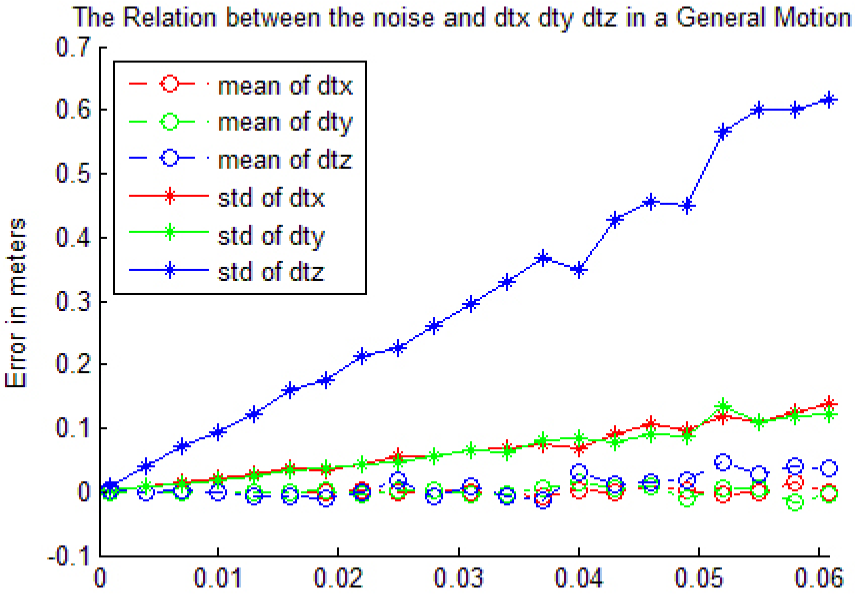


Figure 5.9: The relation between the feature uncertainty [m] and $\mathrm{dt_x, dt_y, dt_z}$ [m] No Features=25, Half-Angle FOV=65°; target features are assumed in the range 2-30 [m].

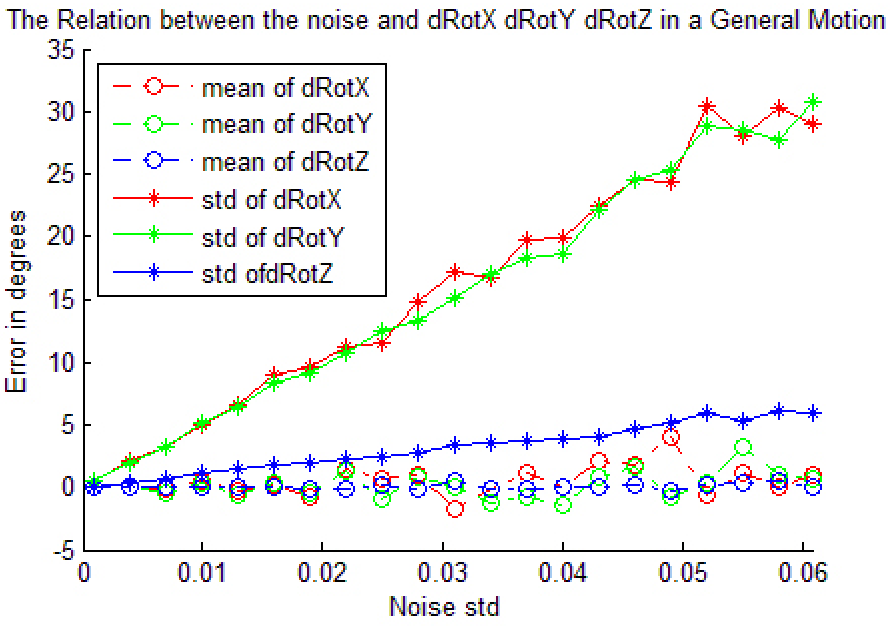


Figure 5.10: The relation between the feature uncertainty [m] and $d\vartheta_x, d\vartheta_y, d\vartheta_z$ [deg] No Features=25, Half-Angle FOV=65°; target features are assumed in the range 2-30 [m].

One notes a significant improvement in estimation uncertainty for any particular pixel localization uncertainty, primarily due the increase in the target range (corresponding to a larger SNR). As an example, the standard deviation is about 2cm for XY translation, 18cm for Z translation, 2° for Z rotation, and 10° for XY rotation, for a feature position uncertainty of 2cm (for either x_s or y_s).

5.4.2 Motion estimation with real data

Our sensitivity analysis can be compared with the results from an experiment with real data. This constitutes the estimation of motion for 5 frames of a video sequence, recorded at the Sea Tech Marina of the Florida Atlantic University in Dania Beach, FL. The sonar placement is near the end of a horizontal beam, roughly 10 [ft] (3.07m) from a vertical rotating pole, while tilted down slightly. Rotating the pole, two perpendicular rows of vertical dock pilings, the main scene features, were imaged. Figure 5.11 to 5.14 show two of the five images, also superimposed with features, used for motion estimation. Figure 5.15 illustrates the setup of the experiment.

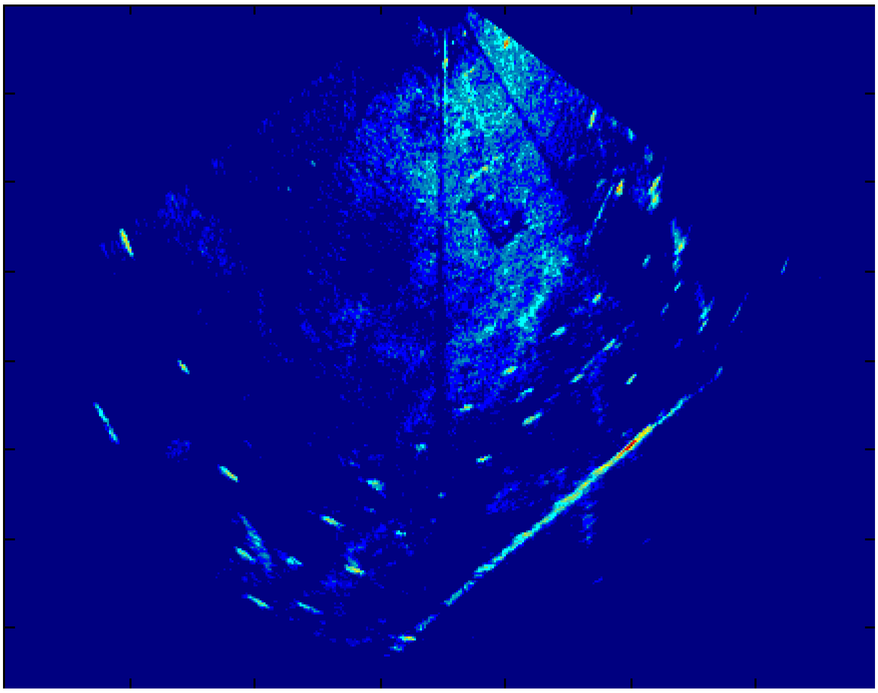


Figure 5.11: Frame 1 from FAU Marina in experiments with real data, also superimposed with features used for motion estimation in Figure 5.12

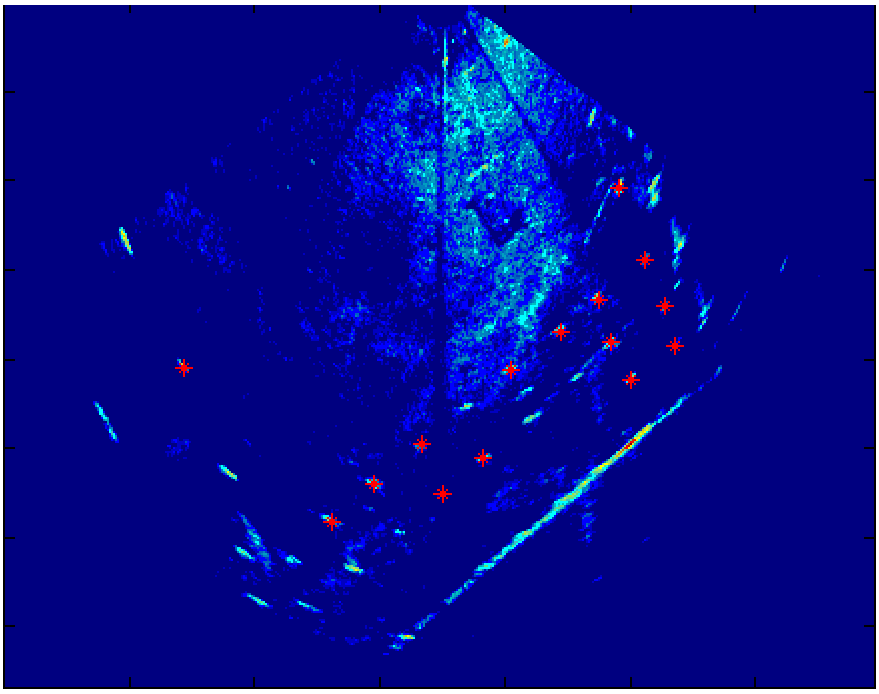


Figure 5.12: Frame 1 from FAU Marina in experiments with real data, also superimposed with features used for motion estimation.

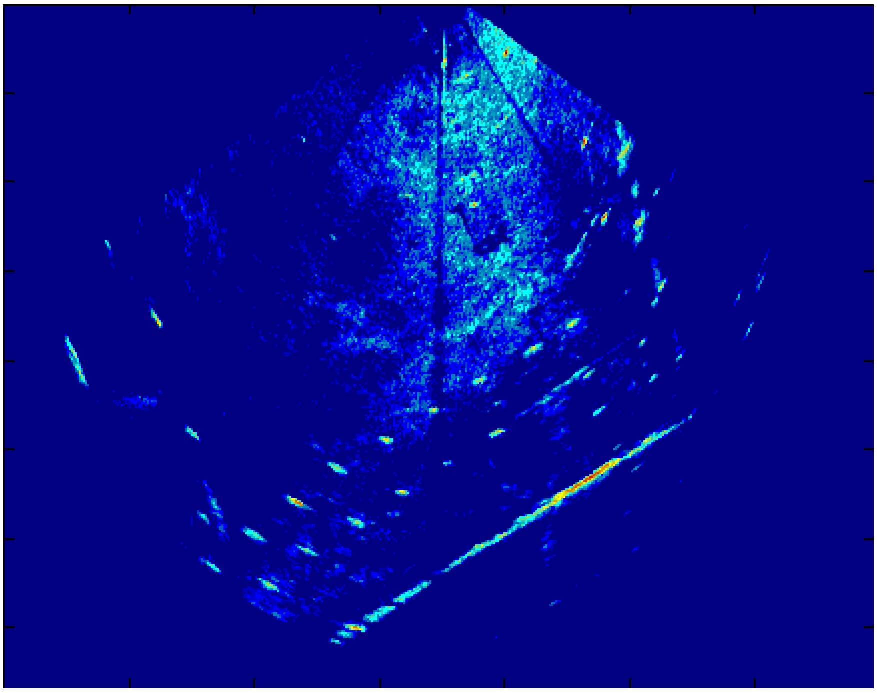


Figure 5.13: Frame 3 from FAU Marina in experiments with real data, also superimposed with features used for motion estimation in Figure 5.14.

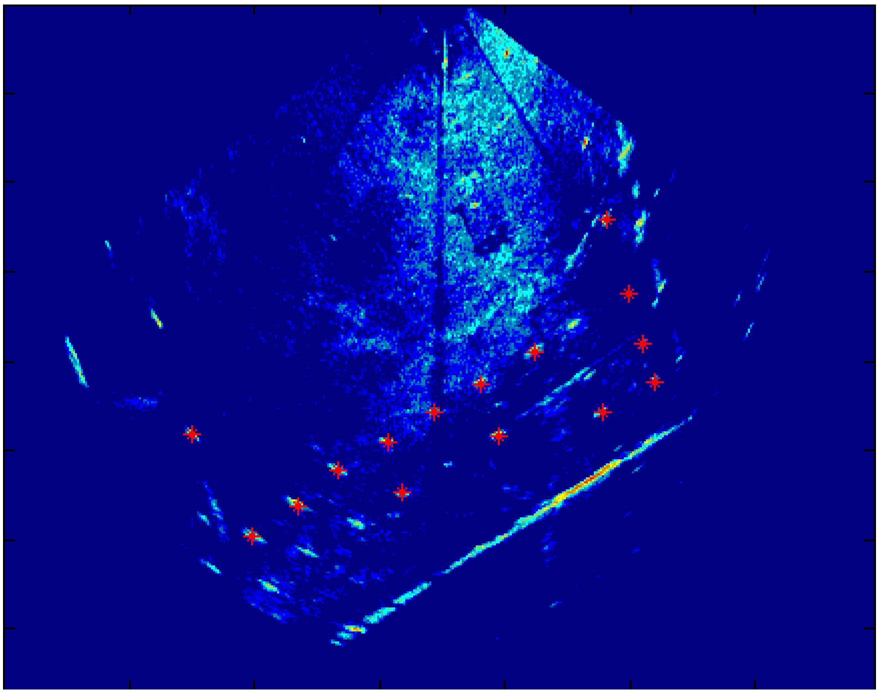


Figure 5.14: Frame 3 from FAU Marina in experiments with real data, also superimposed with features used for motion estimation.

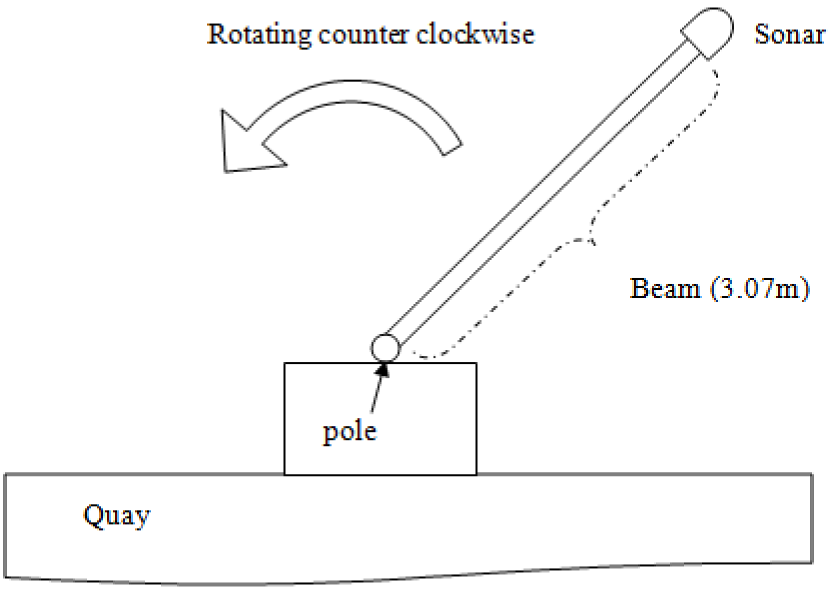


Figure 5.15: Sketch for the experiment setup.

Taking every 5th frame of the video, the image-to-image sonar rotations have been estimated at roughly 0.12 [rad] (or 7°), with the induced image-to-image translation of roughly 0.36m, dominantly in the XY plane (although neither is constant). The estimated motions and the planar normals have been summarized in Table 5.1.

Seq / Motion	1-2	2-3	3-4	4-5
tx	-0.3773	-0.3916	-0.3786	-0.4008
ty	0.0371	0.075	-0.1439	0.0276
tz	0.0029	0.0072	-0.008	0
ωx	0.0094	0.0101	-0.0215	0.0307
ωy	-0.0075	0.0099	-0.023	-0.0303
ωz	-0.1101	-0.1215	-0.1214	-0.1534
ny	0.0684	0.0111	0.056	0.0305
nz	-0.0489	-0.0009	-0.0287	0.0322

Table 5.1: Estimated motion components, translation $t = [tx, ty, tz]$ [m] and rotation $\omega = [\omega x, \omega y, \omega z]$ [rad] between 5 consecutive images, along the sonar trajectory. The surface normals are defined with respect to each sonar position.

For quantitative assessment, we need ground truth of the frame-to-frame motions, which is not available. Instead, we have constructed 6 sequences that form closed trajectories, starting and ending at the same position, roughly of the same length. This means that while we do not know the intermediate positions along the sonar track, the first and last positions should coincide. Therefore, the accumulated end position and angle can be used for quantitative accuracy assessment. Table 5.2 summarizes the results. As an example, the first closed trajectory is defined by positions 1, 2, 3, 4, 3, 2 and 1, with accumulated errors of et = [0.054, 0.18,−0.03] [m], and ew = [2.9, 1.6,−2.7] [deg] in position and orientation, respectively, while travelling a distance of 2.4 [m]. Based on the total errors in position and angle for the 5 trajectories, the average image-to-image translation and rotation errors are less than 3 [cm] and 1.7°.

Seq Motion	1234321	2345432	134321	12421	23532	45324
et_x	0.0544	-0.0026	-0.0354	0.092	-0.0824	0.0433
et_y	0.1782	0.1181	0.0742	0.2617	-0.0124	0.2161
et_z	-0.0318	-0.0042	-0.0572	0.0023	-0.1353	0.0121
$\|\|et\|\|$	0.19	0.12	0.1	0.28	0.16	0.22
$e\omega_x$	2.8661	1.9628	2.7042	7.9027	8.6307	6.245
$e\omega_y$	1.6099	4.3059	0.1152	4.6707	1.6081	4.2471
$e\omega_z$	-2.6691	-1.3981	-3.6185	-1.6876	-5.9191	-6.1983
$\|\|e\omega\|\|$	4.2	4.9	4.5	9.2	10.3	9.6
Distance	2.419	2.4444	2.4738	2.3911	2.4007	2.3696

Table 5.2: six closed trajectories formed from 5 sonar positions, with roughly the same travelled distance. Each row depicts the position numbers defining the loop, accumulated error in position $\mathrm{et} = [\mathrm{etx}, \mathrm{ety}, \mathrm{etz}]$ [m] and angle $\mathrm{e}\omega = [\mathrm{e}\omega_\mathrm{x}, \mathrm{e}\omega_\mathrm{y}, \mathrm{e}\omega_\mathrm{z}]$ [deg], and estimated total travled distance [m].

Referring to our theoretical analysis, recall that the translation and rotation magnitudes have been assumed at 0.4m and 20°, respectively. Compared to results from real data ($|| t || \approx$ 0.36m and $|| \omega || \approx$ 7°), there is agreement between motion error magnitudes, although the motion component errors differ. One explanation may be that our analysis is based on motions with all 6 degrees of freedom, while our experiment has involved sonar motions with mainly 2 degrees of freedom (although we have not made use of this fact in our computations).

Finally, we have depicted in Figure 5.16, Figure 5.17, Figure 5.18, and Figure 5.19 various views of three selected closed trajectories to visualize the earlier results. In addition to the accumulated error at the final position, another quantitative measure of accuracy is the discrepancy in the estimate sonar positions when visited more than once along each and across different trajectories. In this case, we can compare each position over different trajectories because the same position, namely position 1, is used as the reference for all of these trajectories. As expected, the accumulated errors increase with the length of distance travelled.

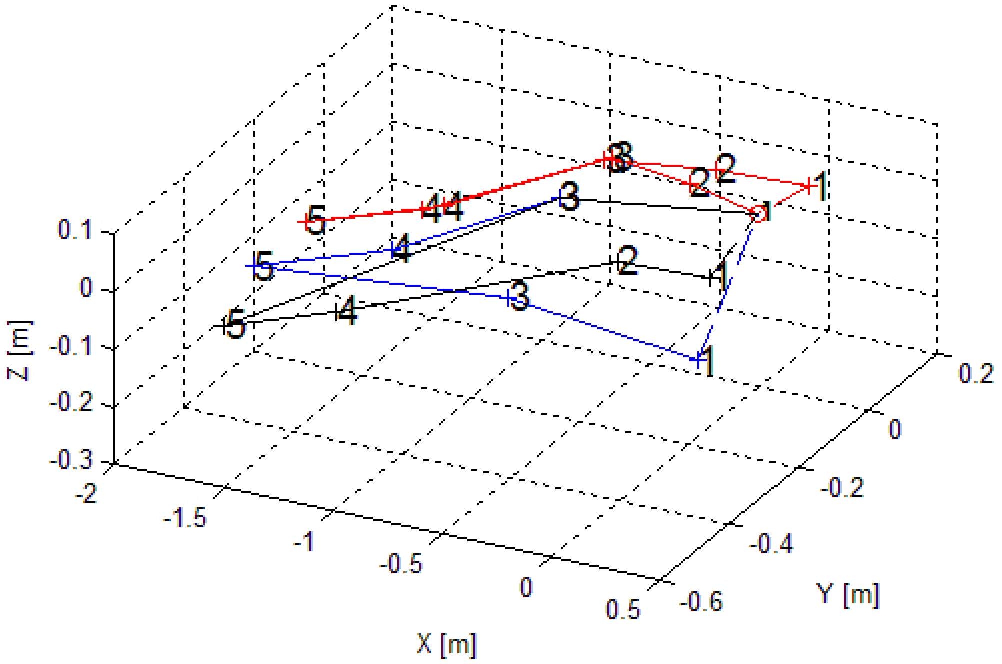


Figure 5.16: Three selected trajectories constructed from different pairs of frames. The discrepancies in various positions along each and across different positions reflect the accumulation error, blue: 134531, black: 135421, red: 123454321

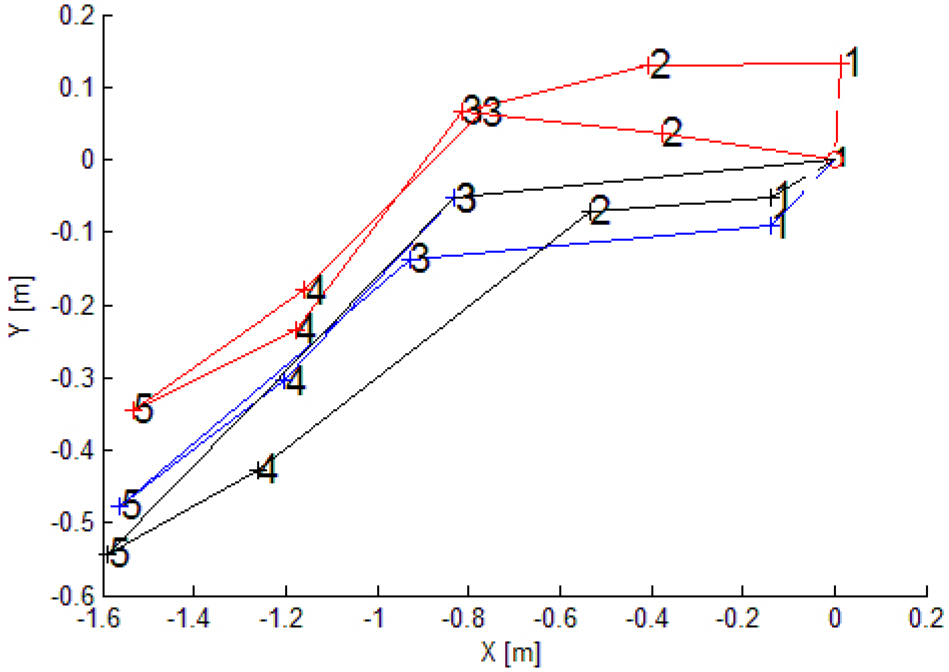


Figure 5.17: XY projection from Figure 5.16, blue: 134531, black: 135421, red: 123454321

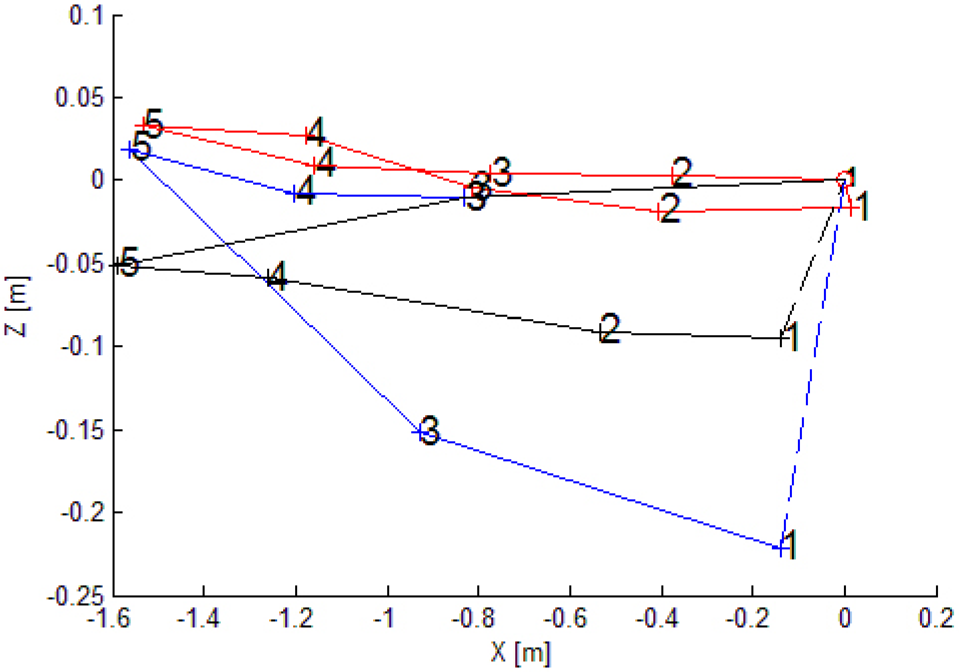


Figure 5.18: XZ projection from Figure 5.16, blue: 134531, black: 135421, red: 123454321

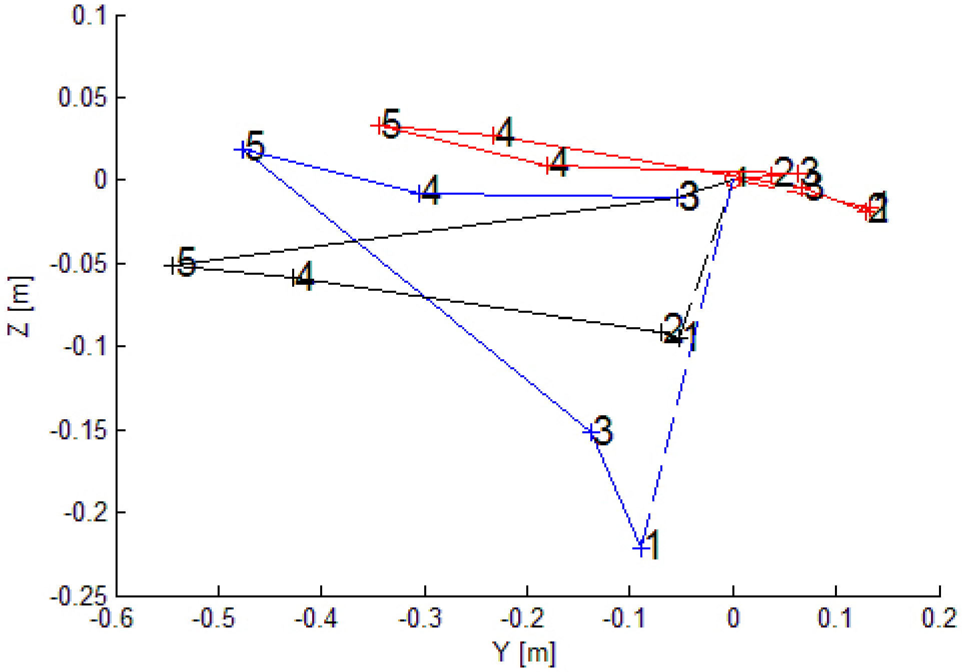


Figure 5.19: YZ projection from Figure 5.16, blue: 134531, black: 135421, red: 123454321

5.5 Conclusion

This chapter deals with the estimation of motion from sonar image sequences. One of the important objectives has been to determine how the estimation accuracy depends on a number of factors, comprising the uncertainty in the position of image features used as image-to-image correspondences, number of these features, and the size of the sonar camera's field of view. We have considered the ranges of these parameters for some existing sonar technologies. We have also presented results of experiments to evaluate performance with real data. Here, we have provided various measures to quantify the achieved accuracy, based on ground truth. This includes accumulated errors in closed trajectories and the position of the sonar along various tracks, with respect to a common reference position. Our results suggest merit in the application of motion vision approaches to sonar video. Some challenges comprise the development of 1) image enhancement methods to improve the signal-to-noise ratio; 2) robust algorithms for feature detection and matching, to maximize number of accurate correspondences which is discussed in the next Chapter.

Chapter 6
Sorting the Correspondence Space

6.1 Introduction

Image registration is an indispensable component in computer vision; it represents the core problem in many tasks like shape recognition, motion estimation, 3D reconstruction, medical image analysis, mosaicing and stereo-matching to name but a few. The problem can be easily described as the task of assigning correspondences for a set of points in the first image to their counterparts in the second image. In other words, the problem is to find the transformation which maps the first set of points into the other. In Chapter 2, we have reviewed some important works for image registration, including point matching algorithms.

Herein, we introduce a novel, robust, and fast algorithm for affine point matching; in fact, the SCS algorithm cannot be categorized to spectral or to probabilistic methods discussed in Chapter 2. Our contributions in this algorithm can be outlined in the following points:

- Novel algorithm for affine point matching, the algorithm is intelligible and its core is derived using clear algebraic equations.
- Robust, the algorithm is very robust against outliers; in this context, it surpasses state of the art methodologies. Also, its execution time is naturally dependent on the outlier percentage in the point sets.
- Fast, compared to algorithms with similar robustness like RANSAC. The algorithm doesn't use iterative optimization in high dimensional space, it is neither probabilistic nor spectral; the algorithm also uses a novel error formula to enhance its performance.

6.2 Point matching problem.

Let's have two sets of features P and P′ extracted from two images, here, we seek the affine transformation T with 6 DOF (Degrees of Freedom) between P and P′.

$$T = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ 0 & 0 & 1 \end{bmatrix} \tag{6.1}$$

In other words, our goal is to know the correspondences between P and P′ so that $\mathrm{P}' = \mathrm{T.P}$ where:

$$P' = \begin{bmatrix} x'_1 & x'_2 & x'_3 & \dots & x'_n \\ y'_1 & y'_2 & y'_3 & \dots & y'_n \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \; ; \; P = \begin{bmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ y_1 & y_2 & y_3 & \dots & y_n \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix} \tag{6.2}$$

Obviously, P and P′ may have different sizes in the first instance due to outliers; however, for the simplicity of introducing the algorithm, we consider that our sets of features are outlier-free sets as we will discuss outliers later on; to show the essence of the algorithm, we will discuss an affine transformation with 5 degrees of freedom (5 DoF) as in equation (6.3), the algorithm is easily adapted for 6 DoF as explained later on.

$$T = \begin{bmatrix} Sx\cos\theta & -Sx\sin\theta & Tx \\ Sy\sin\theta & Sy\cos\theta & Ty \\ 0 & 0 & 1 \end{bmatrix} \tag{6.3}$$

6.3 Exploring the correspondence space

We start by rewriting T in equation (6.3) in a different form, after some triangulations we write T as follows:

$$T = \begin{bmatrix} \frac{\sqrt{S_x^2+S_y^2}}{2}[\cos(\theta-\varphi)+\cos(\theta+\varphi)] & -\frac{\sqrt{S_x^2+S_y^2}}{2}[\sin(\theta-\varphi)+\sin(\theta+\varphi)] & Tx \\ \frac{\sqrt{S_x^2+S_y^2}}{2}[\cos(\theta-\varphi)-\cos(\theta+\varphi)] & -\frac{\sqrt{S_x^2+S_y^2}}{2}[\sin(\theta-\varphi)-\sin(\theta+\varphi)] & Ty \\ 0 & 0 & 1 \end{bmatrix} \tag{6.4}$$

Where φ is dubbed the scale angle and defined as follows:

$$\cos\varphi = \frac{S_x}{\sqrt{S_x^2 + S_y^2}} \quad \text{and} \quad \sin\varphi = \frac{S_y}{\sqrt{S_x^2 + S_y^2}} \tag{6.5}$$

For any point $\mathrm{p}' = [\mathrm{x}'\ \mathrm{y}'\ 1]^{\mathrm{T}} \subset \mathrm{P}'$ which corresponds to $\mathrm{p} = [\mathrm{x}\ \mathrm{y}\ 1]^{\mathrm{T}} \subset \mathrm{P}$ we can write:

$$x' = \frac{\sqrt{S_x^2 + S_y^2}}{2} \{[\cos(\theta - \varphi) + \cos(\theta + \varphi)]\, x - [\sin(\theta - \varphi) + \sin(\theta + \varphi)]y\} + T_x \tag{6.6}$$

$$y' = \frac{\sqrt{S_x^2 + S_y^2}}{2} \{[\cos(\theta - \varphi) - \cos(\theta + \varphi)]\, x - [\sin(\theta - \varphi) - \sin(\theta + \varphi)]y\} + T_y \tag{6.7}$$

By adding the previous two equations (6.6) and (6.7) we get

$$\beta = x' + y' = \sqrt{S_x^2 + S_y^2}\, \{[\cos(\theta - \varphi)]\, x - [\sin(\theta - \varphi)]y\} + T_x + T_y \tag{6.8}$$

Or simply:

$$\boldsymbol{\beta} = \boldsymbol{S_{xy}}\, \boldsymbol{\alpha} + \boldsymbol{T_{xy}} \tag{6.9}$$

Where:

$$\boldsymbol{\alpha} = \boldsymbol{x} \cos \boldsymbol{\psi} - \boldsymbol{y} \sin \boldsymbol{\psi} \tag{6.10}$$

$$\psi = \theta - \varphi\, ; \quad S_{xy} = \sqrt{S_x^2 + S_y^2}\ ; \quad T_{xy} = T_x + T_y \tag{6.11}$$

The linear relation in (6.9) between α and β represents the correspondence space, in other words, points $\mathrm{C} = (\alpha, \beta)$ represents the correspondence relation between two points in P and P′. The algorithm exploits this relation to find the correspondences, to do that, we find β for all points in P′ and sort them in an ascending order:

$$x_1' + y_1' < x_2' + y_2' < x_3' + y_3' < \cdots \tag{6.12}$$

$$\beta_1 < \beta_2 < \beta_3 < \cdots \tag{6.13}$$

$$S_{xy}\,\alpha_1 + T_{xy} < S_{xy}\,\alpha_2 + T_{xy} < S_{xy}\,\alpha_3 + T_{xy} < \cdots \tag{6.14}$$

This yields to:

$$\alpha_1 < \alpha_2 < \alpha_3 < \cdots \tag{6.15}$$

$$x_1 \cos\psi - y_1 \sin\psi \;\; < x_2 \cos\psi - y_2 \sin\psi \;\; < x_3 \cos\psi - y_3 \sin\psi \;\; < \cdots \tag{6.16}$$

In the next example, we illustrate how to establish the correspondences between P and P'. This example demonstrates the basic concepts of the SCS algorithm and will defiantly raise some questions like filtering the outliers and the ambiguous order of the points where $\alpha_1 = \alpha_2$ or $\beta_1 = \beta_2$ (which are called trouble points). These questions will be answered in the sections after the example.

6.3.1 SCS example

Let's assume that we have two sets of points P and P′ as illustrated in the Figure 6.1 and their coordinates are listed in the Table 6.1. The true correspondence between the points is numbered and colour-coded for the ease of following the algorithm stages, so the red point ❷ in P corresponds to the red point ② in P′. We start with random order in the sets and attempt to find the correspondences by using the SCS algorithm.

$P\ (x\ ,y)$	(1.00, 2.00) ❷	(1.50, 1.20) ❸	(2.10, 1.30) ❹	(1.00, 1.80) ❶
$P'(\ x',\ y')$	(2.70, 2.79) ③	(1.97, 3.06) ①	(1.87, 3.23) ②	(3.17,3.18) ④

Table 6.1: The coordinates of two sets of points, The true correspondence between the points is numbered and colour-coded.

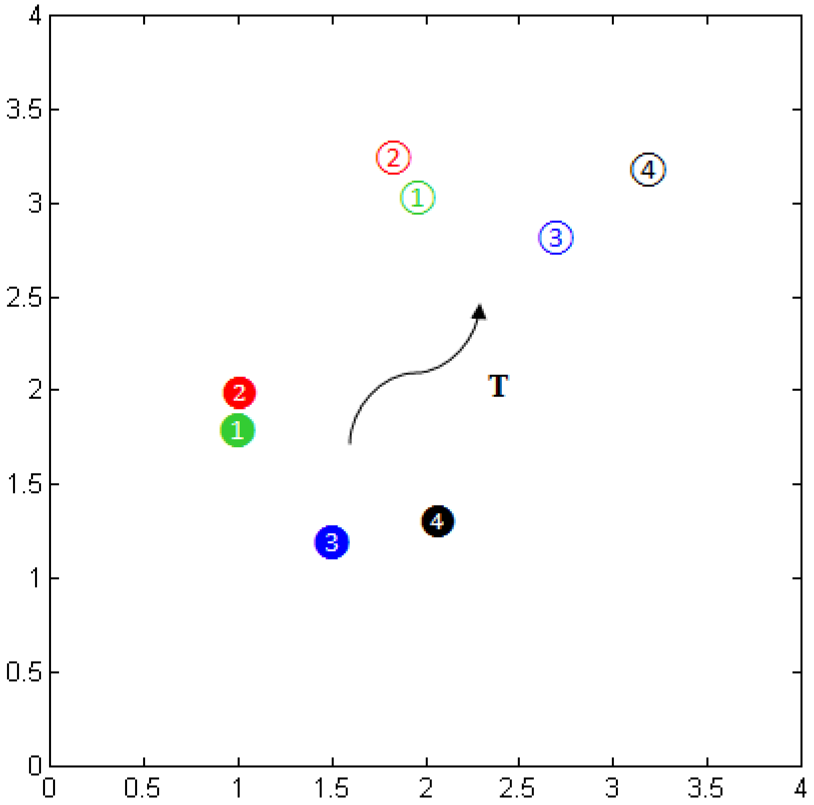


Figure 6.1: Two point sets P and P′, T is the missing transform matrix between them.

Firstly, we move P′ to β space by adding both coordinates for its points, and then we sort the β space in ascending order as illustrated in Table 6.2:

$P'(x', y')$	(2.70, 2.79) ③	(1.97, 3.06) ①	(1.87, 3.23) ②	(3.17,3.18) ④
β space $= x' + y'$	5.49 ③	5.03 ①	5.10 ②	6.36 ④
Sorted β vector	5.03 ①	5.10 ②	5.49 ③	6.36 ④

Table 6.2: Moving P' to β space and sorting the β space.

Secondly, to search for the correspondences, we search for the right ψ in the range $[-180°, 180°]$ by sampling the range in 20°, the sampling procedure will be discussed later on, for now, 20° is a heuristic number. For each ψ value, we calculate the value of the α vector and sort it in an ascending order. The right ψ will map the points in P to their correspondences in P′ as illustrated in the Table 6.3 and in the Table 6.4:

	$\alpha = x\cos\psi - y\sin\psi$			
$P\ (x\ ,y)$	(1.00, 2.00) ❷	(1.50, 1.20) ❸	(2.10, 1.30) ❹	(1.00, 1.80) ❶
$\psi_i = -180° \rightarrow \alpha =$	-1.00 ❷	-1.5 ❸	-2.1 ❹	-1.00 ❶
$\psi_i = -160° \rightarrow \alpha =$	-0.26 ❷	-0.10 ❸	-1.53 ❹	-0.32 ❶
…….	…….	…….	…….	…….
$\psi_i = -20° \rightarrow \alpha =$	1.62 ❷	1.82 ❸	2.42 ❹	1.56 ❶
$\psi_i = 0° \rightarrow \alpha =$	1.0 ❷	1.50 ❸	2.10 ❹	1.00 ❶
$\psi_i = 20° \rightarrow \alpha =$	0.26 ❷	0.10 ❸	1.53 ❹	0.32 ❶
…….	…….	…….	…….	…….
$\psi_i = 160° \rightarrow \alpha =$	-1.62 ❷	-1.82 ❸	-2.42 ❹	-1.56 ❶

Table 6.3: Moving the points in P to the α space depending on ψ values.

By sorting α vectors in an ascending order, we get potential correspondences for the point sets as illustrated in the Table 6.4:

Sorted β vector	5.03 ①	5.10 ②	5.49 ③	6.36 ④
	α vectors in ascending order			
$\psi_i = -180°$	-2.1 ❹	-1.5 ❸	-1.00 ❷	-1.00 ❶
$\psi_i = -160°$	-1.53 ❹	-0.32 ❶	-0.26 ❷	-0.10 ❸
…….	…….	…….	…….	…….
$\psi_i = -20°$	1.56 ❶	1.62 ❷	1.82 ❸	2.42 ❹
$\psi_i = 0°$	1.00 ❷	1.00 ❶	1.50 ❸	2.10 ❹
$\psi_i = 20°$	0.10 ❸	0.26 ❷	0.32 ❶	1.53 ❹
…….	…….	…….	…….	…….
$\psi_i = 160°$	-2.42 ❹	-1.82 ❸	-1.62 ❷	-1.56 ❶

Table 6.4: Sorting α vectors in an ascending order to match the points, when $\psi = -20°$ the suggested order in the α vector match the points correctly between P and P'.

Table 6.4 shows the suggested correspondences for each ψ value, for example, when $\psi = -160°$ the algorithm maps (2.10, 1.30) ❹ to (1.97, 3.06) ① and so on. We get the right correspondence when $\psi = -20°$ which maps correctly the points in P' to the points in P. It worth mentioning here that the true value of ψ is -15° for the used transformation, however when $\psi = -20°$, it is close enough to the right value of ψ so that it orders the points in a correct way. In this example, the SCS algorithm suggested $(360°/20) = 18$ different solutions out of $4! = 24$ possible solutions. While this reduction is not very attractive, the algorithm leads to a great reduction with large point sets where the number of possible solutions in RANSAC terms is n!.

In the previous example, it is clear that the search space is turned into one dimensional search space over (ψ) rather than five dimensional search space $(\mathrm{S_x}, \mathrm{S_y}, \mathrm{T_x}, \mathrm{T_y}, \theta)$, or six dimensional search space in the affine case as explained later on, this fact is one of the greatest advantages of the algorithm. As we have seen in the previous example, sampling ψ gives different correspondence solutions which we have to evaluate and choose the solution which best maps between the point sets. Thus, for each suggested solution (represented by a row in the Table 6.4) we calculate the transformation T in least squares terms using singular value decomposition and calculate the error associated with it, then we choose the transformation with the minimum error as the solution. In other words, to find correspondences between two sets of points P and P' with outlier-free assumption we follow these simple and fast steps:

1- Find the sorted $Beta_{Vector} = \{\beta_1, \beta_2, \beta_3 \dots \beta_n\}$ from P'

2- Sort points in P' according to their order in $Beta_{Vector}$ to get $P_s{}'$

3- Find $P_s'^{-1}$, the pseudoinverse of P_s' using single value decomposition (svd).

4- Find the sorted $Alpha\_i_{Vector} = \{\alpha_{i1}, \alpha_{i2}, \alpha_{i3} \dots \alpha_{in}\}$ based on a given ψ_i

5- Sort points in P according to their order in $Alpha\_i_{vector}$ to get Pi_s

6- Find the inverse of the affine Transformation $T_i^{-1} = Pi_s * P_s'^{-1}$

7- Find the Error value E_i associated with T_i using the error function in the correspondence space which is explained in the next section.

8- Repeat steps 4 to 7 for all $\psi_i \subset [-180° \;\; 180]$.

9- $T = T_k$ where $k = \arg \min_{i}\{E_i\}$

It is important to note that we use the inverse of P_s' rather than inversing Pi_s inside the loop, this arrangement decreases the computational burden significantly. It worth mentioning also that there is another correspondence space which can be inferred from subtracting equations (6.6) and (6.7); the only difference in this case is $\psi = \theta + \varphi$; we will use both spaces in the error function as explained later on.

6.4 Moving to complete affine transformation (6 DoF)

As mentioned earlier, the algorithm doesn't change in 6 DoF, for now let's have an affine transformation as described in equation (6.1), we can write x' and y' as follows:

$$x' = a_1 x + a_2 y + a_3 \tag{6.17}$$

$$y' = a_4 x + a_5 y + a_6 \tag{6.18}$$

As the earlier procedure, adding the equations (6.17) and (6.18) will give us:

$$\beta = x' + y' = (a_1 + a_4)x + (a_2 + a_5)y + a_3 + a_6 \tag{6.19}$$

$$\beta = \sqrt{(a_1 + a_4)^2 + (a_2 + a_5)^2}\ [x \cos\psi - y \sin\psi] + a_3 + a_6 \tag{6.20}$$

$$\boldsymbol{\beta = S_{xy}\, \alpha + T_{xy}} \tag{6.21}$$

Where:

$$\boldsymbol{\alpha = x \cos\psi - y \sin\psi} \tag{6.22}$$

$$S_{xy} = \sqrt{(a_1 + a_4)^2 + (a_2 + a_5)^2} \tag{6.23}$$

$$\cos\psi = \frac{(a_1+a_4)}{S_{xy}} \qquad \text{and} \qquad \sin\psi = \frac{-(a_2+a_5)}{S_{xy}} \tag{6.24}$$

$$T_{xy} = a_3 + a_6 \tag{6.25}$$

It is clear that both equations (6.21) and (6.9) are similar, so the algorithm doesn't change at all, we presented the 5 DOF case first of all to highlight the meaning of the scale angle φ and its relation to ψ and θ.

6.5 The error function

To differentiate between the suggested solutions by the sorted β vector and α vectors, we use an error function to evaluate each solution individually. In fact, the error function is an indispensable tool to pick the right affine transformation; it has to be fast and robust to outliers. It can be shown that slight amendments in the error function thresholds and/or mechanism will severely affect the performance of any point matching algorithm, herein we present the simple error function which is adopted by most point matching algorithms, also, we present a novel error function depending on the introduced correspondence space, in our new function, we have achieved relatively fast and robust performance by using two correspondence spaces simultaneously as explained later on.

6.5.1 Simple error function

One intuitive method to estimate the error for an affine transformation $\mathrm{T_i}$ is performed by accumulating the error associated with each point in $\mathrm{Pi}' = \mathrm{T_i} * \mathrm{Pi_s}$; the error for each point pi' in Pi' is defined by the smallest distance between pi' and each point in P', see Figure 6.2 ; this definition is simple and intuitive; however, this method is inefficient in terms of the computational load as it requires n^2distance-calculations to calculate the error associated with $\mathrm{T_i}$; we call this method SError for the simple error function.

$$SError_i = \sum_{j=1}^{n} \min_k \left| pi'_j - p'_k \right| \tag{6.26}$$

Where $\left|\mathrm{pi'_j} - \mathrm{p'_k}\right|$ is the Euclidean distance between $\mathrm{pi'_j} \subset \mathrm{Pi'} : \mathrm{j} = 1 \dots \mathrm{n}$ and $\mathrm{p'_k} \subset \mathrm{P'} : \mathrm{k} = 1 \dots \mathrm{n}$

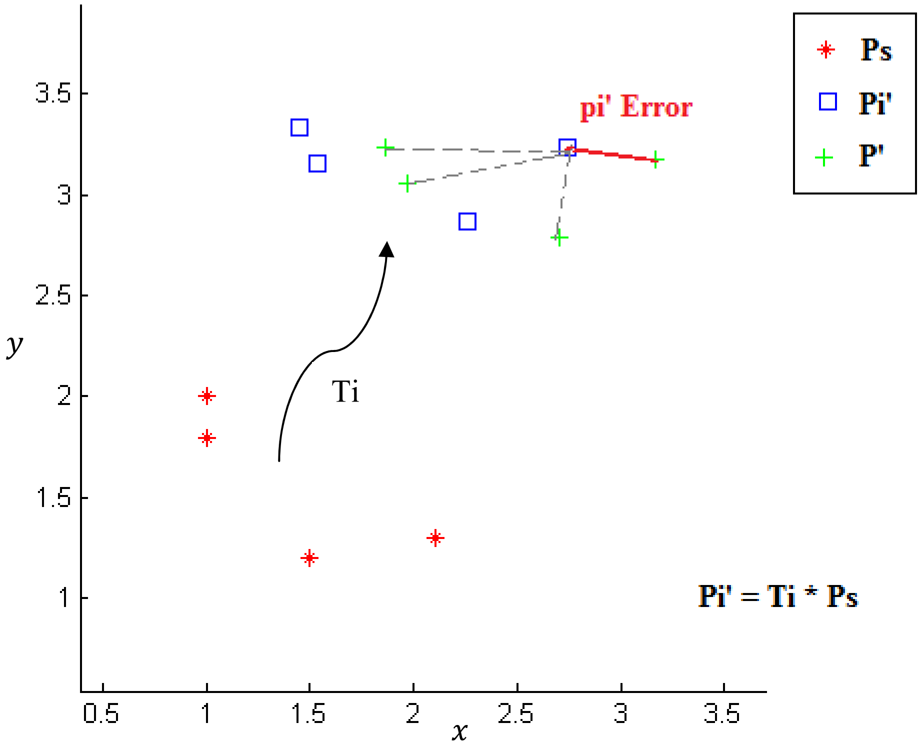


Figure 6.2: The error for each point pi′ in Pi′ is defined by the smallest distance between pi′ and each point in P′.

6.5.2 *The Error function in the correspondence space*

The error function has to measure how far the points of Pi′ are from the points of P′, this measurement will tell how good the calculated affine transformation T_i is. The best T_i will make the transformation from Pi′ to P′ a unity matrix; the error E_i in the correspondence space can be calculated as illustrated in Figure 6.3. Mathematically, E_i is defined as follows.

$$E_i = \sum_j |\beta_j - \beta i_j| + \sum_k |\beta_k - \beta i_k| \qquad (6.27)$$

Where:

$$\beta_j = x_j' + y_j' \quad : p_j' = [x_j' \;\; y_j' \;\; 1]^T \subset P' \;\&\; \beta_j < \beta_{j+1} \,\forall\, j = 1..n \qquad (6.28)$$

$$\beta i_j = xi_j' + yi_j' \quad : pi_j' = [xi_j' \;\; yi_j' \;\; 1]^T \subset Pi' \;\&\; \beta i_j < \beta i_{j+1} \,\forall\, j = 1..n \qquad (6.29)$$

β_k and βi_k are in the second correspondence space which is produced from subtracting equations (6.6) and (6.7) rather than adding them.

$$\beta_k = x'_k - y'_k \quad : p'_k = [x'_k \;\; y'_k \;\; 1]^T \subset P' \;\; \& \;\; \beta_k < \beta_{k+1} \; \forall \;\; k = 1..n \tag{6.30}$$

$$\beta i_k = xi'_k - yi'_k \; : pi'_k = [xi'_k \;\; yi'_k \;\; 1]^T \subset Pi' \;\; \& \; \beta i_k < \beta i_{k+1} \, \forall \, k = 1..n \tag{6.31}$$

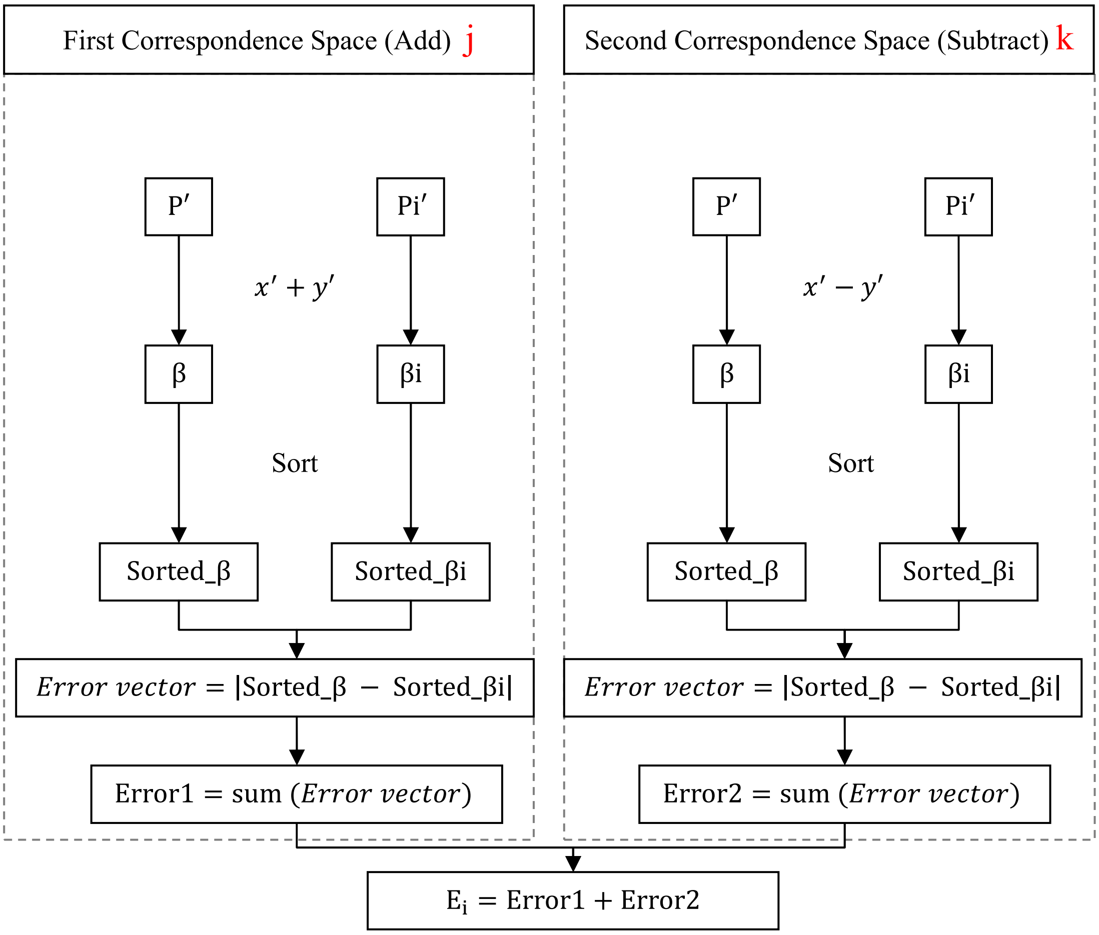

First Correspondence Space (Add) j
Second Correspondence Space (Subtract) k
P′
Pi′
x′ + y′
β
βi
Sort
Sorted_β
Sorted_βi
Error vector = |Sorted_β − Sorted_βi|
Error1 = sum (Error vector)
P′
Pi′
x′ − y′
β
βi
Sort
Sorted_β
Sorted_βi
Error vector = |Sorted_β − Sorted_βi|
Error2 = sum (Error vector)
E_i = Error1 + Error2

Figure 6.3: The error function in the correspondence space.

6.6 Filtering the outliers

The outlier is a point in P which has no correspondence point in P′ or vice versa; obviously, the outlier will distort the order of the $\text{Alpha}_{\text{vector}}$ and $\text{Beta}_{\text{vector}}$ which means assigning wrong correspondences between P and P′ as illustrated in the next relations:

$$\begin{tikzcd}[column sep=tiny]
\mathrm{Beta}_{\mathrm{vector}} = & \beta_1 \arrow[d, dashed, leftrightarrow] & < & \beta_2 \arrow[d, dashed, leftrightarrow] & < & \beta_3 \arrow[d, dashed, leftrightarrow] & < & \beta_4 \arrow[d, dashed, leftrightarrow] & < & \beta_5 \arrow[d, dashed, leftrightarrow] & < & \beta_6 \arrow[d, dashed, leftrightarrow] & < & \boldsymbol{\beta_o} \arrow[d, dashed, leftrightarrow] & < \cdots & (6.32) \\
\mathrm{Alpha}_{\mathrm{vector}} = & \alpha_1 & < & \alpha_2 & < & \alpha_3 & < & \boldsymbol{\alpha_o} & < & \alpha_4 & < & \alpha_5 & < & \alpha_5 & < \cdots & (6.33)
\end{tikzcd}$$

The outlier α_o has no correspondence in the sorted $\mathrm{Beta}_{\mathrm{vector}}$ which fools the aforementioned algorithm to assign the points: (p'_4, p_o) (p'_5, p_4) (p'_6, p_5) .... as correspondences. There are different methods to tackle the outliers here; for example, we may use guided RANSAC which selects the correspondence of the point p'_j from the neighbourhood of its direct correspondence i.e. $p_{j-2}, p_{j-1}, p_j, p_{j+1}, p_{j+2}$ with small neighbourhood range if the expected number of outliers is small and vice versa. Also, Hough Transform [156] can be applied to get a robust estimation for S_{xy} and T_{xy}.

Here, we filter the outliers by using the linear relation in the correspondence space, i.e. equation (6.21), for all β_j, α_j we can write:

$$Sj_{xy} = \frac{\beta_{j-1} - \beta_j}{\alpha_{j-1} - \alpha_j} \quad : j = 2 \ldots n \tag{6.34}$$

This relation doesn't hold if one of its components is an outlier, to filter the outlier out first we get an initial estimation ES_{xy} for S_{xy} by using the pseudoinverse of the first order vandermonde matrix as follows:

$$vand = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \cdots & \beta_n \\ 1 & 1 & 1 & \ldots & 1 \end{bmatrix} \tag{6.35}$$

$$\begin{bmatrix} \frac{1}{ES_{xy}} & \frac{-ET_{xy}}{ES_{xy}} \end{bmatrix} = [\alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \ldots \alpha_n] * vand^{-1} \tag{6.36}$$

Where $vand^{-1}$ is the pseudoinverse matrix of $vand$ calculated using the single value decomposition (svd) [157], another effective ways to compute the inverse of the vandermonde matrix can be found in [158], [159] and [160], we estimate ES_{xy} from the contaminated $\mathrm{Alpha}_{\mathrm{vector}}$ and $\mathrm{Beta}_{\mathrm{vector}}$, this estimation is very fast as we find the pseudoinverse of the vandermonde matrix only once (again it is not inside the loop). This procedure estimates the true S_{xy} with no more than 15% error even with 50%

outliers randomly distributed in each point set (this is of course applicable for the right $\mathrm{Alpha}_{\mathrm{vector}}$ which is calculated from the right ψ_i), Figure 6.4 illustrates the relation between the relative error in the S_{xy} estimation as described in (6.37) and the number of outliers. Next we describe S_{xy} filtering which is the first stage in filtering the outliers out.

$$Relative\ Estimation\ Error = \frac{\left|ES_{xy} - S_{xy}\right|}{S_{xy}} \tag{6.37}$$

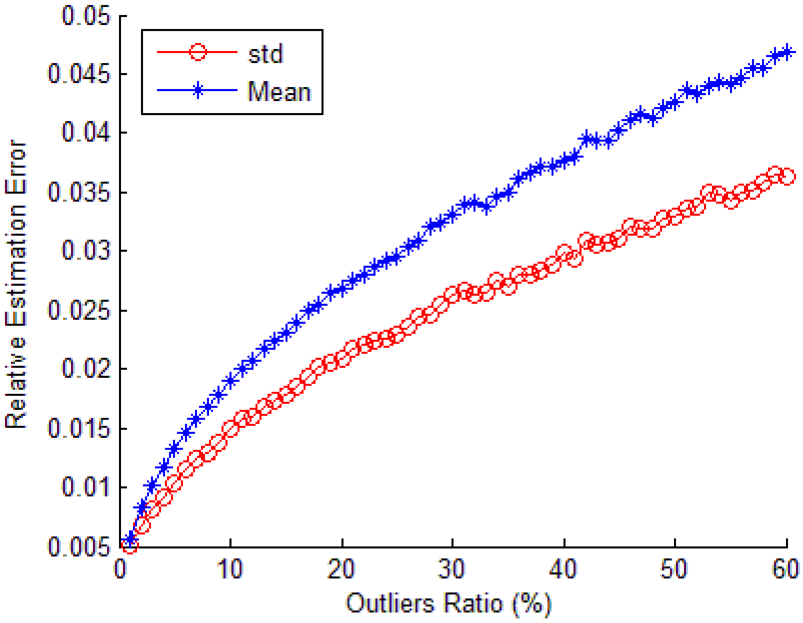


Figure 6.4: illustrates the relation between the relative estimation error and the outliers ratio in the point sets, for each outlier ratio we have repeated the experiment 5000 times with random affine transformation and random point sets, the 0.005 error when the outlier ratio is 0 is due to the ψ-sampling resolution .

6.6.1 S_{xy} filtering

When $\mathrm{Sj}_{\mathrm{xy}}$, described in (6.34), is considerably larger or smaller than $\mathrm{ES}_{\mathrm{xy}}$ i.e. the condition in (6.38) is not satisfied, it gives an indicator that one of the four terms in (6.34) is an outlier (or more than one), so we start calculating the effects of deleting each term (or more than one) and see if the new $\mathrm{Sj}_{\mathrm{xy}}$ survives the $\mathrm{ES}_{\mathrm{xy}}$ test, i.e. the condition in (6.38).

$$\left|\mathrm{Sj}_{\mathrm{xy}} - \mathrm{ES}_{\mathrm{xy}}\right| < \epsilon * ES_{\mathrm{xy}} \tag{6.38}$$

6.6.1.1 *One term tests ($\boldsymbol{\alpha_j}$, $\boldsymbol{\beta_j}$, $\boldsymbol{\alpha_{j-1}}$, $\boldsymbol{\beta_{j-1}}$)*

For each questioned term we perform two tests to see the effects of deleting that term forward and backward, for example if we question α_j then we calculate:

$$\mathrm{test1_b} = \frac{\beta_{j-1} - \beta_j}{\alpha_{j-1} - \alpha_{j+1}}; \qquad \mathrm{test1_f} = \frac{\beta_j - \beta_{j+1}}{\alpha_{j+1} - \alpha_{j+2}} \tag{6.39}$$

It worth mentioning here that $\mathrm{Alpha}_{\mathrm{vector}}$ and $\mathrm{Beta}_{\mathrm{vector}}$ are sorted and indexed as illustrated in (6.40) and (6.41), deleting the term in the vector will shift all other terms after its index to the left as illustrated in (6.42) and (6.43).

$$\begin{tikzcd}
\mathrm{Beta}_{\mathrm{vector}} = \ldots < & \beta_{j-2} \arrow[d, dashed, leftrightarrow] & < \beta_{j-1} \arrow[d, dashed, leftrightarrow] & < \beta_j \arrow[d, dashed, leftrightarrow] & < \beta_{j+1} \arrow[d, dashed, leftrightarrow] & < \beta_{j+2} \arrow[d, dashed, leftrightarrow] & < \beta_{j+3} \arrow[d, dashed, leftrightarrow] & < \cdots & (6.40) \\
\mathrm{Alpha}_{\mathrm{vector}} = \cdots < & \alpha_{j-2} & < \alpha_{j-1} & < \alpha_j & < \alpha_{j+1} & < \alpha_{j+2} & < \alpha_{j+3} & < \cdots & (6.41)
\end{tikzcd}$$

Before deleting α_j

$$\begin{tikzcd}
\mathrm{Beta}_{\mathrm{vector}} = \ldots < & \beta_{j-2} \arrow[d, dashed, leftrightarrow] & < \beta_{j-1} \arrow[d, dashed, leftrightarrow] & < \beta_j \arrow[d, dashed, leftrightarrow] & < \beta_{j+1} \arrow[d, dashed, leftrightarrow] & < \beta_{j+2} \arrow[d, dashed, leftrightarrow] & < \beta_{j+3} \arrow[d, dashed, leftrightarrow] & < \cdots & (6.42) \\
\mathrm{Alpha}_{\mathrm{vector}} = \cdots < & \alpha_{j-2} & < \alpha_{j-1} & < \alpha_{j+1} & < \alpha_{j+2} & < \alpha_{j+3} & < \alpha_{j+4} & < \cdots & (6.43)
\end{tikzcd}$$

After deleting α_j

If both of the backward and forward values pass the $\mathrm{ES}_{\mathrm{xy}}$ test i.e. $\left|\mathrm{test1_{b,f}} - \mathrm{ES}_{\mathrm{xy}}\right| < \epsilon * ES_{\mathrm{xy}}$ then we delete α_j and its associated point p_j in P_s and move to the next j value. Also, we delete α_j and its associated point p_j if its backward value is the only one passing the $\mathrm{ES}_{\mathrm{xy}}$ test among all other backward values in the one term tests, (even if the forward value doesn't pass the $\mathrm{ES}_{\mathrm{xy}}$ test). In Table 6.5 we summarize all test values with their questioned terms:

Term	Backward test	Forward test	Notes	
α_j	$test1_b = \frac{\beta_{j-1} - \beta_j}{\alpha_{j-1} - \alpha_{j+1}}$	$test1_f = \frac{\beta_j - \beta_{j+1}}{\alpha_{j+1} - \alpha_{j+2}}$		
β_j	$test2_b = \frac{\beta_{j-1} - \beta_{j+1}}{\alpha_{j-1} - \alpha_j}$	$test2_f = \frac{\beta_{j+1} - \beta_{j+2}}{\alpha_j - \alpha_{j+1}}$		
α_{j-1}	$test3_b = \frac{\beta_{j-2} - \beta_{j-1}}{\alpha_{j-2} - \alpha_j}$	$test3_f = \frac{\beta_{j-1} - \beta_j}{\alpha_j - \alpha_{j+1}}$	To delete α_{j-1} or β_{j-1} the next value should not pass the ES_{xy} test	$\frac{\beta_{j-2} - \beta_{j-1}}{\alpha_{j-2} - \alpha_{j-1}}$
β_{j-1}	$test4_b = \frac{\beta_{j-2} - \beta_j}{\alpha_{j-2} - \alpha_{j-1}}$	$test4_f = \frac{\beta_j - \beta_{j+1}}{\alpha_{j-1} - \alpha_j}$		

Table 6.5: backward and forward test values for the one term test.

6.6.1.2 Two terms tests:

If we cannot rectify the problem by deleting one term only from $\text{Alpha}_{\text{vector}}$ or $\text{Beta}_{\text{vector}}$, we perform two points test to delete two terms at once (with their associated points in P_s , P'_s), here we make one test only (and may add more tests for extra robustness), Table 6.6 summarizes the performed tests:

Term	Forward test
α_j & β_j	$test5 = \frac{\beta_{j-1} - \beta_{j+1}}{\alpha_{j-1} - \alpha_{j+1}}$
α_{j-1} & β_{j-1}	$test6 = \frac{\beta_{j-2} - \beta_j}{\alpha_{j-2} - \alpha_j}$
α_j & α_{j+1}	$test7 = \frac{\beta_{j-1} - \beta_j}{\alpha_{j-1} - \alpha_{j+2}}$
β_j & β_{j+1}	$test8 = \frac{\beta_{j-1} - \beta_{j+2}}{\alpha_{j-1} - \alpha_j}$
β_j & α_{j-1}	$test9 = \frac{\beta_{j-1} - \beta_{j+1}}{\alpha_j - \alpha_{j+1}}$
β_{j-1} & α_j	$test10 = \frac{\beta_j - \beta_{j+1}}{\alpha_{j-1} - \alpha_{j+1}}$

Table 6.6: test values for the two term test.

It is true that if α_i , β_i are outliers they will not distort the correspondences, but they will affect later stages in calculating the error and getting the best affine transformation, so we delete them.

6.6.1.3 Three terms test

Again, if we cannot rectify the problem by deleting tow terms from $\mathrm{Alpha}_{\mathrm{vector}}$ and/or $\mathrm{Beta}_{\mathrm{vector}}$, we perform three points test to delete three terms at once (with their associated points in P_s , P_s'), here we make two tests, backward and forward; both tests have to pass the ES_{xy} test to delete their associated points, Table 6.7 summarizes the performed tests:

Term	Backward test	Forward test
$\alpha_{j-1} , \alpha_j , \alpha_{j+1}$	$test11_b = \frac{\beta_{j-1} - \beta_j}{\alpha_{j+2} - \alpha_{j+3}}$	$test11_f = \frac{\beta_j - \beta_{j+1}}{\alpha_{j+3} - \alpha_{j+4}}$
$\beta_{j-1} , \beta_j , \beta_{j+1}$	$test12_b = \frac{\beta_{j+2} - \beta_{j+3}}{\alpha_{j-1} - \alpha_j}$	$test12_f = \frac{\beta_{j+3} - \beta_{j+4}}{\alpha_j - \alpha_{j+1}}$
$\alpha_{j-1} , \beta_{j-1} , \beta_j$	$test13_b = \frac{\beta_{j-2} - \beta_{j+1}}{\alpha_{j-2} - \alpha_j}$	$test13_f = \frac{\beta_{j+1} - \beta_{j+2}}{\alpha_j - \alpha_{j+1}}$
$\alpha_{j-1} , \alpha_j , \beta_{j-1}$	$test14_b = \frac{\beta_{j-2} - \beta_j}{\alpha_{j-2} - \alpha_{j+1}}$	$test14_f = \frac{\beta_j - \beta_{j+1}}{\alpha_{j+1} - \alpha_{j+2}}$
$\alpha_j , \beta_{j-1} , \beta_j$	$test15_b = \frac{\beta_{j-2} - \beta_{j+1}}{\alpha_{j-2} - \alpha_{j-1}}$	$test15_f = \frac{\beta_{j+1} - \beta_{j+2}}{\alpha_{j-1} - \alpha_{j+1}}$
$\alpha_{j-1} , \alpha_j , \beta_j$	$test16_b = \frac{\beta_{j-2} - \beta_{j-1}}{\alpha_{j-2} - \alpha_{j+1}}$	$test16_f = \frac{\beta_{j-1} - \beta_{j+1}}{\alpha_{j+1} - \alpha_{j+2}}$
$\beta_{j-1} , \beta_{j+1} , \alpha_j$	$test17_b = \frac{\beta_{j-2} - \beta_j}{\alpha_{j-2} - \alpha_{j-1}}$	$test17_f = \frac{\beta_j - \beta_{j+2}}{\alpha_{j-1} - \alpha_{j+1}}$
$\alpha_{j-1} , \alpha_{j+1} , \beta_j$	$test18_b = \frac{\beta_{j-2} - \beta_{j-1}}{\alpha_{j-2} - \alpha_j}$	$test18_f = \frac{\beta_{j-1} - \beta_{j+1}}{\alpha_j - \alpha_{j+2}}$

Table 6.7: backward and forward test values for the three term test.

After the end of the three terms test and deleing all outliers, we perform another Sj_{xy} scan and delete all α_j , β_j and their associated points in P_s , P_s' if Sj_{xy} doesn't pass the

$\mathrm{ES_{xy}}$ test. Also we make sure that both $\mathrm{P_s}$ and $\mathrm{P_s'}$ have the same size by deleting all points at the end of the larger one, as they undergo unbalanced treatment depending on the number of outliers in each one. In most cases we end up at this stage with many points deleted and few left, but this will not be the case with the right ψ_i value. So deleting too many points is a good indicator that we are working with the wrong ψ_i. It worth mentioning also that ϵ has no major effect on the $\mathrm{ES_{xy}}$ test, we found $\epsilon = 15\%$ to be a good heuristic value proved by Monte Carlo simulation in Figure 6.4 ; however, any value between 10% and 40% works well.

6.7 Trouble points

Before performing the $\mathrm{S_{xy}}$ filtering, we sort $\mathrm{Alpha_{vector}}$ in an ascending order; if two values in $\mathrm{Alpha_{vector}}$ are equal or very close to each other we may give them incorrect order due to the sampling of ψ. These points are called trouble points; we take a record of them before the $\mathrm{S_{xy}}$ filtering, and delete them from $\mathrm{P_s}$ after the $\mathrm{S_{xy}}$ filtering if they still exist (we delete their supposed correspondences in $\mathrm{P_s'}$ as well).

In fact, trouble points have the same projection into the correspondence space in which we search for the right ψ, for example, if we have a point $\mathrm{p'(x',y')} \subset \mathrm{P_s'}$ and its true correspondence is $\mathrm{p(x,y)} \subset \mathrm{P_s}$ the relation between them is described in equations (6.6) and (6.7) or (6.17) and (6.18); however, if there is a point $\mathrm{p_o(x_o\,,y_o)} \subset \mathrm{P_s}$ & $\mathrm{p_o} \neq \mathrm{p}$ which satisfies the following equations:

$$x' - \lambda \approx \frac{\sqrt{S_x^2 + S_y^2}}{2} \{[\cos(\theta - \varphi) + \cos(\theta + \varphi)]\, x_o - [\sin(\theta - \varphi) + \sin(\theta + \varphi)] y_o\} + T_x \tag{6.44}$$

$$y' + \lambda \approx \frac{\sqrt{S_x^2 + S_y^2}}{2} \{[\cos(\theta - \varphi) - \cos(\theta + \varphi)]\, x_o - [\sin(\theta - \varphi) - \sin(\theta + \varphi)] y_o\} + T_y \tag{6.45}$$

For any $\lambda \in \mathbb{R}^*$; adding (6.44) and (6.45) will show that both points $\mathrm{p} \neq \mathrm{p_o}$ will be projected to the correspondence space as $\alpha = \alpha_o$ so one point will be aligned with $\mathrm{p}'(\mathrm{x}',\mathrm{y}')$ as a correspondence and the other will be filtered with the $\mathrm{S_{xy}}$ filtering, both points (p and $\mathrm{p_o}$) have similar chances to survive the $\mathrm{S_{xy}}$ filtering, so it will be better to delete the point which survived the test with its associated correspondence p' as there is 50% chance to filter the correct point out.

Another issue can be raised here is the size difference between the point sets, so far we have considered that both point sets have the same size, if they don't, we simply add random points to the smaller point set to make it the same size as the larger one, they have to be in the same range as the smaller point set, then we include them in the trouble points record.

6.8 Calculating $\boldsymbol{T_i}$

After deleting the trouble points, $\mathrm{Pi_s}$ and $\mathrm{P_s'}$ will have many true correspondences and few incorrect correspondences; because of the mechanism of the $\mathrm{S_{xy}}$ filtering, true correspondences occur in blocks of successive points in $\mathrm{Pi_s}$ and $\mathrm{P_s'}$. To calculate the best possible $\mathrm{T_i}$ we infer the transformation $\mathrm{T_{ik}}$ from each successive three points in $\mathrm{Pi_s}$ with their correspondences in $\mathrm{P_s'}$, from each $\mathrm{T_{ik}}$ we can deduce $\mathrm{Sik_{xy}}$ which is another estimation for $\mathrm{S_{xy}}$. Then we quantize the absolute difference between $\mathrm{ESi_{xy}}$ and $\mathrm{Sik_{xy}}$, hereafter, we use all correspondences in $\mathrm{Pi_s}$ and $\mathrm{P_s'}$ which contributed to calculate the most frequent error to calculate the final $\mathrm{T_i}$. This step is very effective to tackle noise in the points coordinates.

6.9 Sampling $\boldsymbol{\psi}$ and calculating the error

For each ψ_i we infer $\mathrm{T_i}$ and calculate its error $\mathrm{E_i}$ using the filtered points $\mathrm{P_s}$ and $\mathrm{P_s'}$ only, to add extra robustness to the algorithm we revisit the Error function in the correspondence space to enhance its performance. First, as the number of the filtered points in $\mathrm{Pi_s}$ and $\mathrm{P_s'}$ will be different for different ψ_i values, we normalize the error depending on the number of correspondences survived the $\mathrm{S_{xy}}$ filtering and the trouble points deletion. Second, we use the smallest 50% errors in each correspondence space, as large errors are more likely to be caused by the leaked outliers, see Figure 6.5. In mathematical terms the error E_i will be calculated as follows:

$$E_i = \sum_{j=1}^{n/2} |\beta_j - \beta i_j| /n + \sum_{k=1}^{n/2} |\beta_k - \beta i_k| /n \tag{6.46}$$

Where $|\beta_j - \beta i_j| < |\beta_{j+1} - \beta i_{j+1}|$ $\forall j = 1..n$ in the first correspondence space ($\beta_j = x'_j + y'_j$) and $|\beta_k - \beta i_k| < |\beta_{k+1} - \beta i_{k+1}|$ $\forall k = 1..n$ in the second correspondence space ($\beta_k = x'_k - y'_k$). In other words, the error vector is sorted ascendingly before accumulating its first half.

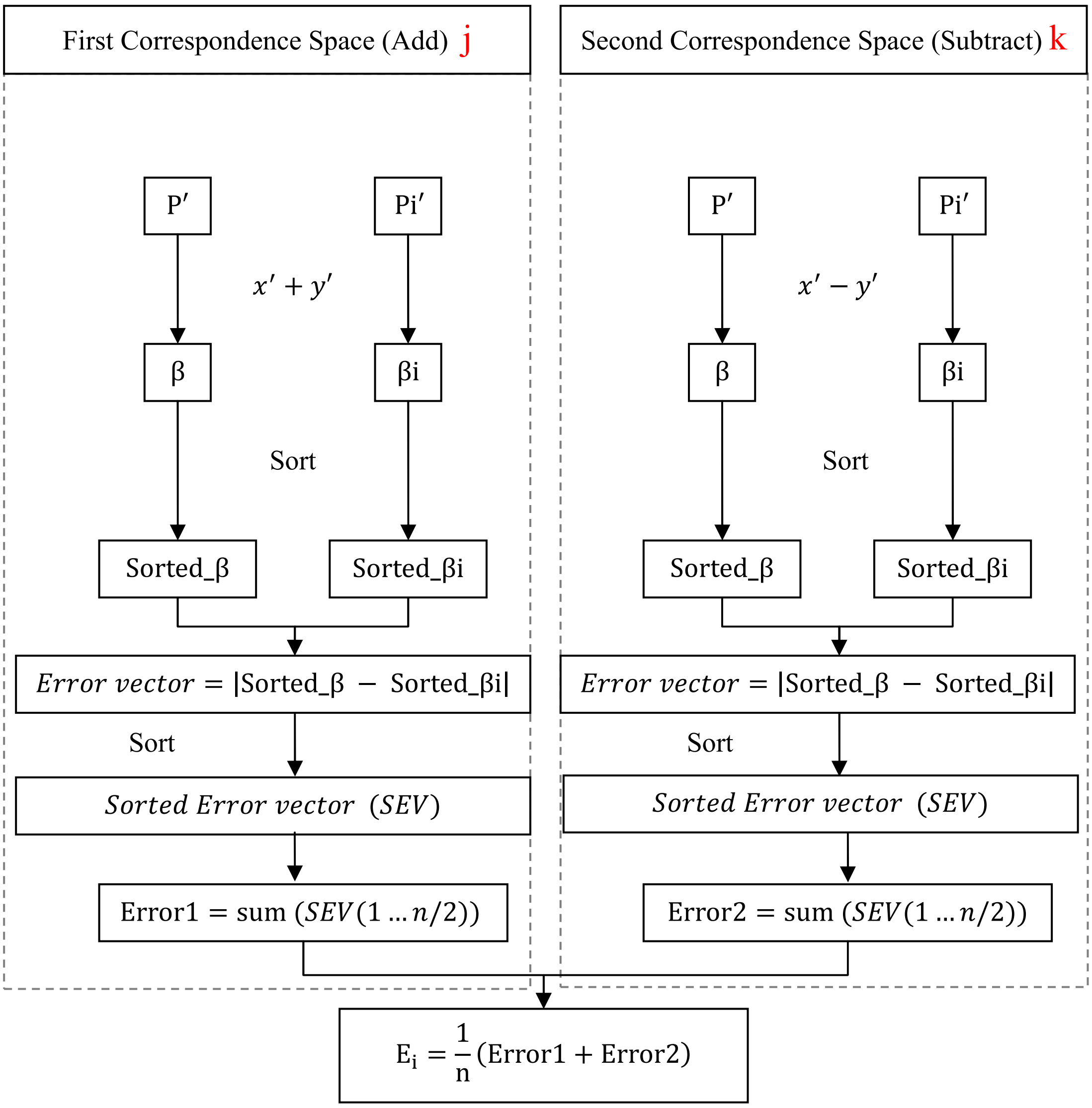

First Correspondence Space (Add) j
Second Correspondence Space (Subtract) k
P′
Pi′
$x' + y'$
P′
Pi′
$x' - y'$
β
βi
β
βi
Sort
Sort
Sorted_β
Sorted_βi
Sorted_β
Sorted_βi
$Error\ vector$ = |Sorted_β − Sorted_βi|
$Error\ vector$ = |Sorted_β − Sorted_βi|
Sort
Sort
$Sorted\ Error\ vector\ (SEV)$
$Sorted\ Error\ vector\ (SEV)$
Error1 = sum ($SEV(1 \ldots n/2)$)
Error2 = sum ($SEV(1 \ldots n/2)$)
$E_i = \frac{1}{n}$(Error1 + Error2)

Figure 6.5: The robust error function in the correspondence space.

In the same fashion, we can adjust the simple error function to use 50% smallest simple errors; in other words, we sort the errors in an ascending order and sum the first half of the vector. In mathematical terms we can write:

$$SError_i = \sum_{j=1}^{n/2} \min_k \left| pi'_j - p'_k \right| \tag{6.47}$$

Where:

$$\min_k \left| pi'_j - p'_k \right| \; < \; \min_k \left| pi'_{j+1} - p'_k \right| \tag{6.48}$$

Sampling ψ is done using different levels, in the first level we use large intervals between ψ_i and $\psi_\mathrm{i+1}$ and differentiate the calculated transformations $\mathrm{T_i}$ by using the enhanced error function in the correspondence space ($\mathrm{E_i}$), then we calculate the enhanced simple error $\mathrm{SError_L}$ of the best transformation $\mathrm{T_L} = \mathrm{T_k} : \mathrm{k} = \mathrm{arg}\ \min_\mathrm{i}\{\mathrm{E_i}\}$ by using $\mathrm{T_L} * \mathrm{P}$ and P' as inputs, if $\mathrm{SError_L}$ is small enough we stop and the best transformation will be $\mathrm{T_L}$, otherwise, we move to the next level and sample ψ using half of the previous interval, all previously visited ψ values are ignored. Our results show that 5 levels are very robust, starting from 1° degree interval in the first level up to 0.0625° interval in the fifth level which is only reached if the outliers ratio is very high. (40% to 50%). This sampling procedure gives the algorithm an advantage of making the execution time relative to the outliers' ratio; however, there is always an upper bound limit on the execution time.

6.10 Results

We have implemented the SCS algorithm in Matlab, the core of the algorithm is straightforward and doesn't require more than tens of lines to implement; however, the filtering function needs some efforts to be implemented in a "vectorized" style for best performance, in the next figures (Figure 6.6 to 6.25) we show how the algorithm restored the correct affine transformation for some challenging point sets. Figures 6.14 to 6.25 contain outliers in the point sets with different ratios (20% up to 50%), each point set contains 100 points, if the outlier ratio is 35% then only 65 points have correspondences and 35 points in each point set are merely outliers. In the following figures correspondences are connected with green dashed lines, the outliers in the first point set are blue stars, in the second point set the outliers are empty red circles.

The SCS algorithm doesn't depend on the proximity assumption to find correspondences as in most point matching algorithms [122], [125], but rather it depends on simple algebraic principles which makes it easy to implement and comprehend; moreover, SCS uses fast error function with $O(N\log(N))$ complexity rather than the conventional error function with $O(N^2)$ complexity, furthermore, the error function is naturally robust against outliers.

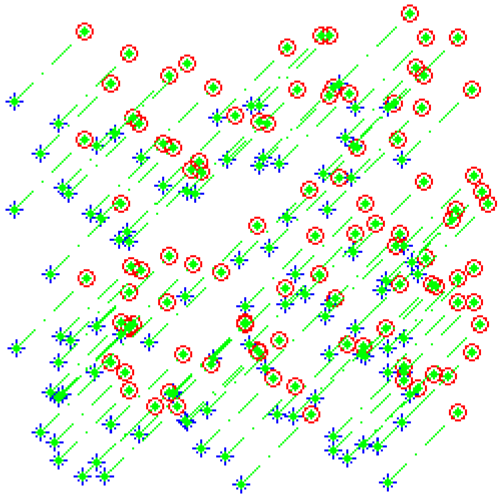


Figure 6.6: Translation only with no outliers, the transformation is restored exactly.

$$T = \begin{bmatrix} 1 & 0 & 35.11 \\ 0 & 1 & 35.04 \\ 0 & 0 & 1 \end{bmatrix}$$

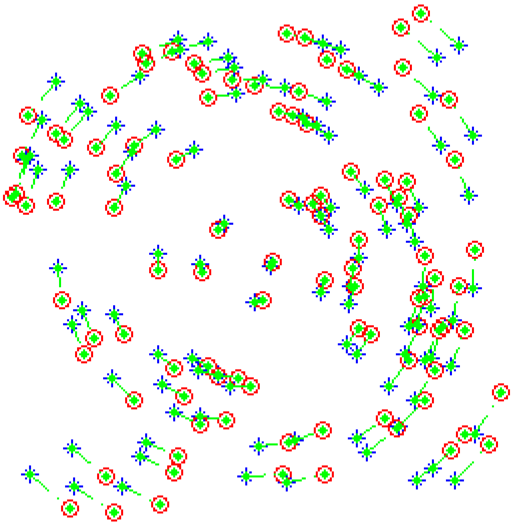


Figure 6.7: Rotation only with no outliers, this case is called "wanog wheeling". The transformation is restored exactly.

$$T = \begin{bmatrix} 0.98 & -0.16 & 0 \\ 0.16 & 0.98 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

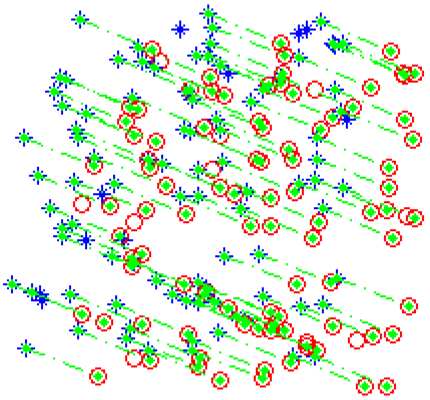


Figure 6.8: Translation only with %10 outliers, the transformation is restored exactly.

$$T = \begin{bmatrix} 1 & 0 & 41.77 \\ 0 & 1 & -17.14 \\ 0 & 0 & 1 \end{bmatrix}$$

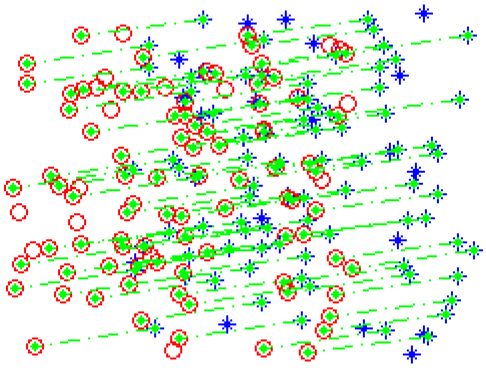


Figure 6.9: Translation only with %20 outliers, the transformation is restored exactly.

$$T = \begin{bmatrix} 1 & 0 & -70.98 \\ 0 & 1 & -10.18 \\ 0 & 0 & 1 \end{bmatrix}$$

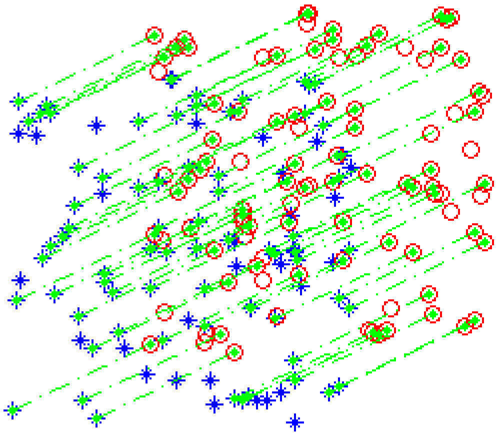


Figure 6.10: Translation only with %30 outliers, the transformation is restored exactly.

$$T = \begin{bmatrix} 1 & 0 & 80.22 \\ 0 & 1 & 38.42 \\ 0 & 0 & 1 \end{bmatrix}$$

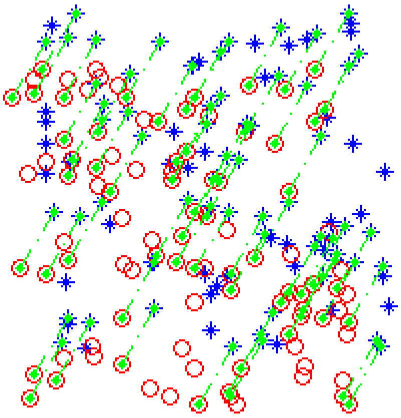


Figure 6.11: Translation only with %40 outliers, the transformation is restored exactly.

$$T = \begin{bmatrix} 1 & 0 & -19.34 \\ 0 & 1 & -33.09 \\ 0 & 0 & 1 \end{bmatrix}$$

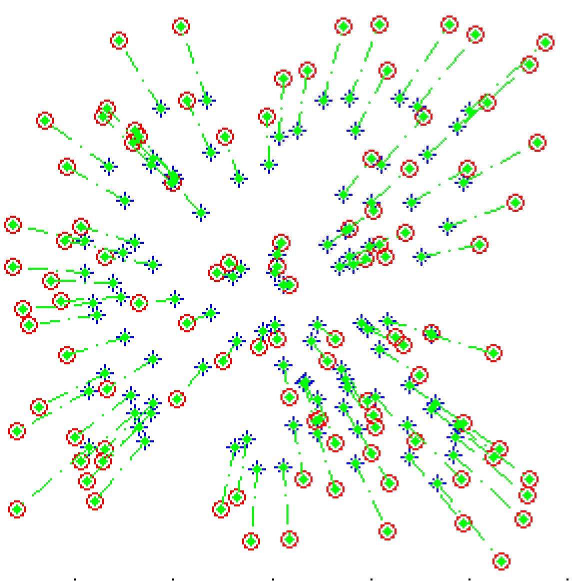


Figure 6.12: Scale only with no outliers, the transformation is restored exactly.

$$T=\begin{bmatrix}1.38 & 0 & 0\\ 0 & 1.38 & 0\\ 0 & 0 & 1\end{bmatrix}$$

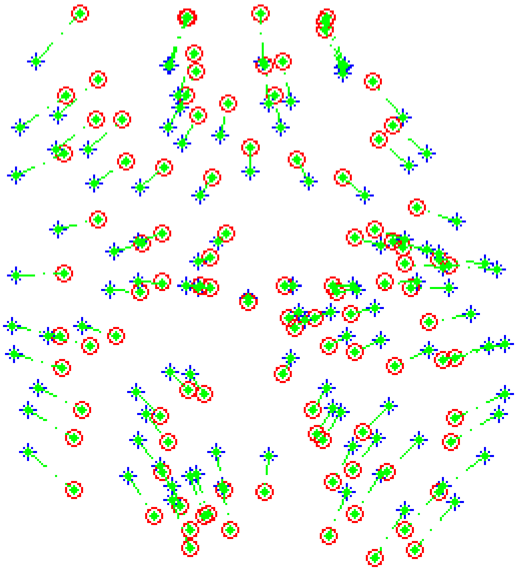


Figure 6.13: Sheer only with no outliers, the transformation is restored exactly.

$$T=\begin{bmatrix}0.78 & 0 & 0\\ 0 & 1.21 & 0\\ 0 & 0 & 1\end{bmatrix}$$

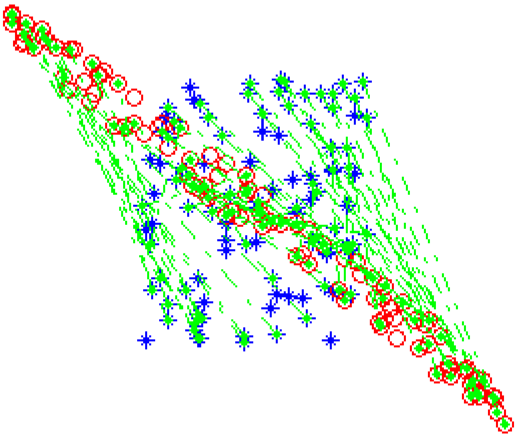


Figure 6.14: Outliers ratio is 35%. The algorithm restored the affine transformation exactly.

$$T=\begin{bmatrix}0.14 & 2.05 & 0.11\\ -0.30 & -1.37 & 0.09\\ 0 & 0 & 1\end{bmatrix}$$

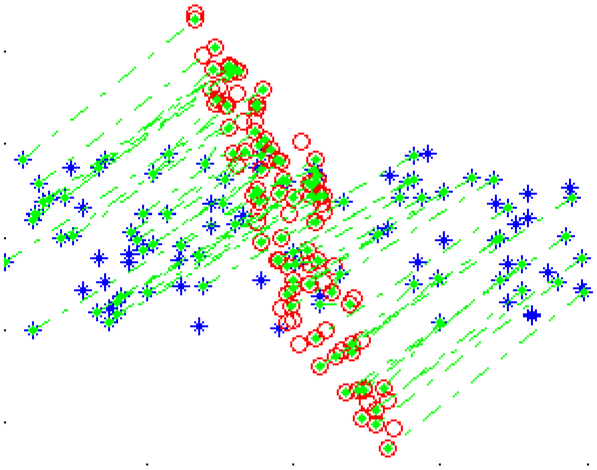


Figure 6.15: Outliers ratio is 35%; The algorithm restored the affine transformation exactly.

$$T=\begin{bmatrix}0.17 & -0.22 & 1.32\\ -1.81 & 0.78 & 0.43\\ 0 & 0 & 1\end{bmatrix}$$

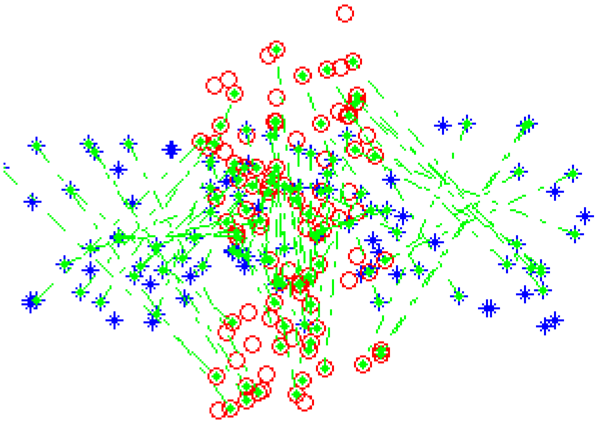


Figure 6.16: Outliers ratio is 35%. The algorithm restored the affine transformation exactly.

$$T = \begin{bmatrix} 0.29 & 0.06 & 0.18 \\ 0.52 & -1.72 & 1.35 \\ 0 & 0 & 1 \end{bmatrix}$$

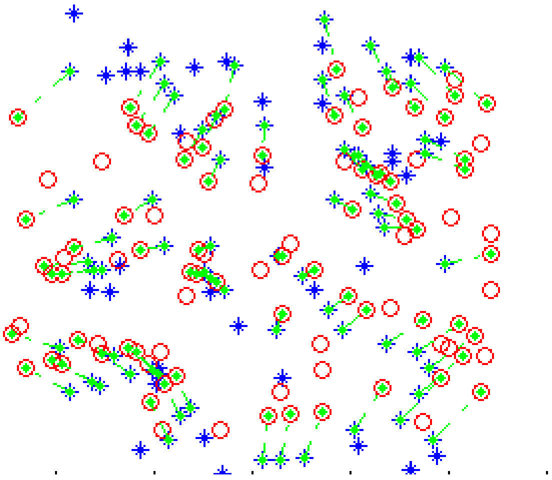


Figure 6.17: Outliers ratio is 35%. The algorithm restored the affine transformation exactly.

$$T = \begin{bmatrix} 1.25 & -0.02 & -1.01 \\ 0.03 & 0.77 & -1.09 \\ 0 & 0 & 1 \end{bmatrix}$$

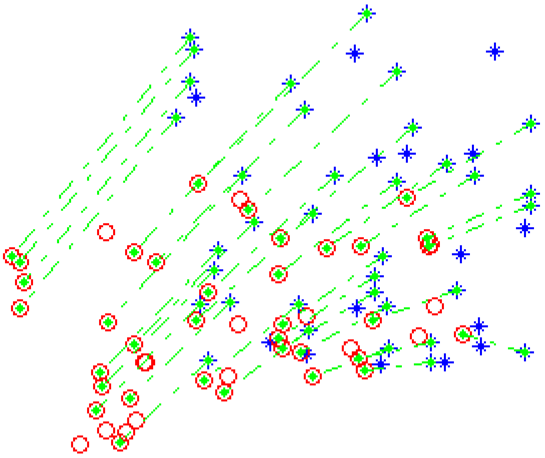


Figure 6.18: Outliers ratio is 30% in 50 points; the algorithm restored the affine transformation exactly.

$$T = \begin{bmatrix} 1.08 & -0.26 & -62.85 \\ 0.33 & 0.59 & -56.67 \\ 0 & 0 & 1 \end{bmatrix}$$

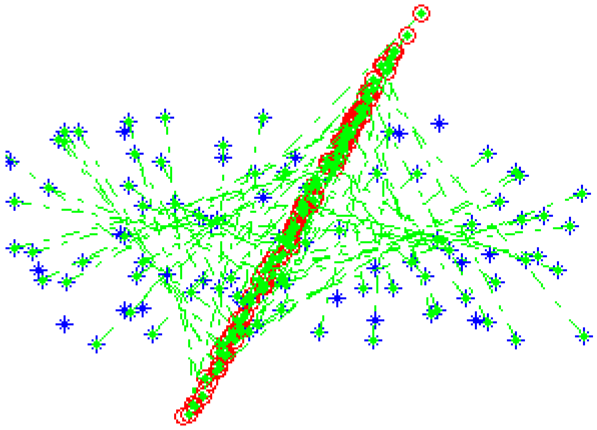


Figure 6.19: Outliers ratio is 20% ; the algorithm restored the affine transformation exactly.

$$\mathrm{T} = \begin{bmatrix} -0.08 & -0.38 & 0.10 \\ -0.26 & -1.69 & -0.71 \\ 0 & 0 & 1 \end{bmatrix}$$

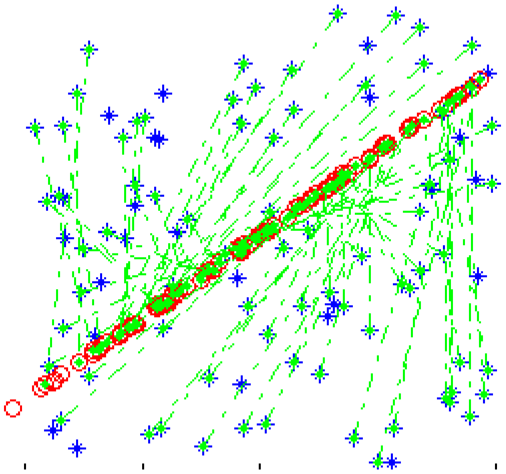


Figure 6.20: Outliers ratio is 25%. The algorithm restored the affine transformation exactly.

$$T=\begin{bmatrix}0.32 & -0.81 & -1.54\\ 0.22 & -0.58 & -0.26\\ 0 & 0 & 1\end{bmatrix}$$

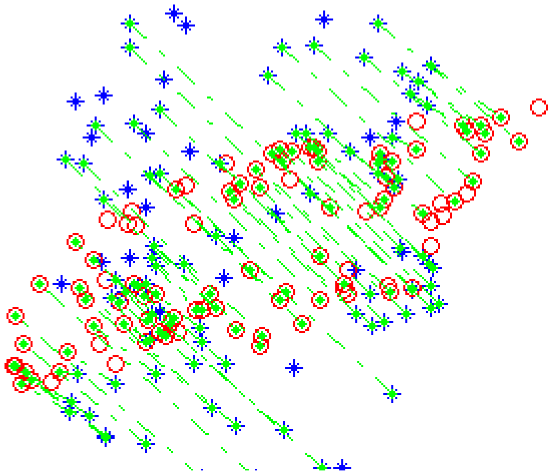


Figure 6.21: Outliers ratio is 25%. The algorithm restored the affine transformation exactly.

$$T=\begin{bmatrix}0.42 & 1.11 & 0.60\\ 0.49 & 0.15 & -0.60\\ 0 & 0 & 1\end{bmatrix}$$

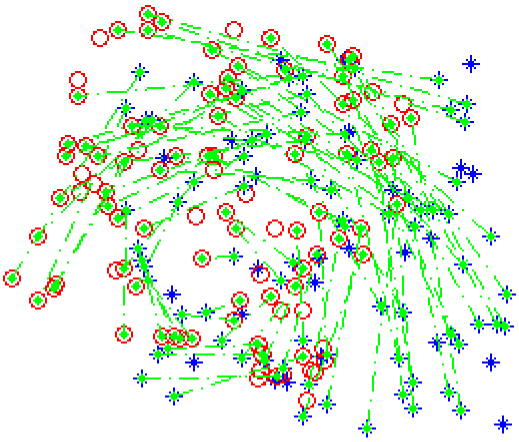


Figure 6.22: Outliers ratio is 25%; the algorithm restored the affine transformation exactly.

$$T=\begin{bmatrix}0.33 & -0.86 & -49.47\\ 0.83 & 0.32 & 31.01\\ 0 & 0 & 1\end{bmatrix}$$

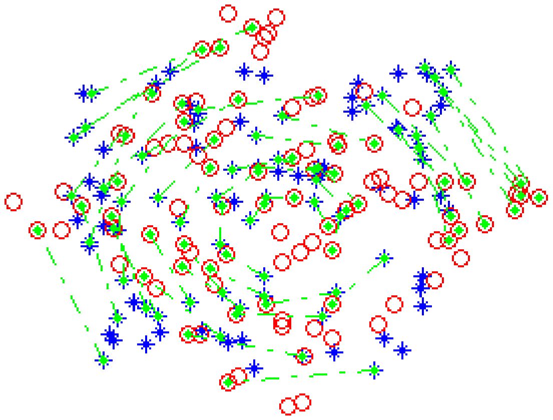


Figure 6.23: Outliers ratio is 50%; the algorithm restored the affine transformation exactly.

$$T=\begin{bmatrix}-0.98 & -0.96 & 0.35\\ 0.19 & -1.75 & -0.74\\ 0 & 0 & 1\end{bmatrix}$$

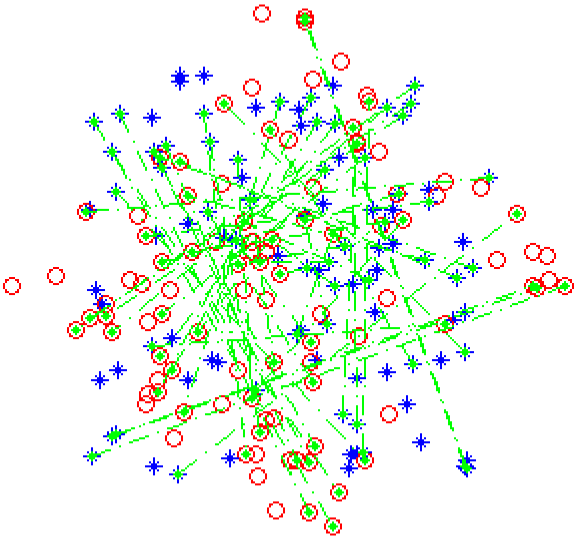


Figure 6.24: Outliers ratio is 50%; the algorithm restored the affine transformation exactly.

$$T = \begin{bmatrix} -0.72 & -0.69 & 0 \\ 0.69 & -0.72 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

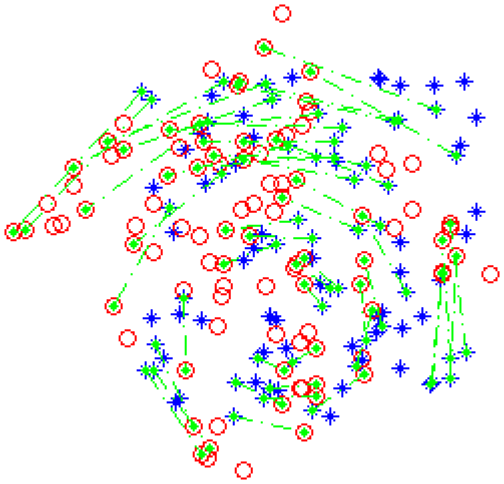


Figure 6.25: Outliers ratio is 50%; the algorithm restored the affine transformation exactly.

$$T = \begin{bmatrix} 0.81 & -0.66 & -33.97 \\ 0.66 & 0.81 & 2.42 \\ 0 & 0 & 1 \end{bmatrix}$$

6.11 Comparing the SCS algorithm with other point matching algorithms

Next we report the performance of 4 different point matching algorithms running the same experiment in the same environment and against identical success/fail evaluation. The studied algorithms are Scott Longuet-Higgins algorithm followed by RANSAC for outlier removal [122], iterative closest point ICP [125] [126], Coherent Point Drift [161], and finally the SCS algorithm. The first three algorithms fairly represent the spectral, iterative and probabilistic point matching algorithms. We measured the ability of each point matching algorithm to recover the transformation between two point sets, each point set has 100 points and the first point set is uniformly distributed in the range [-100 100]. In each experiment, we have started with outlier free sets and monitored the performance of the algorithms, then added outliers to the sets and removed inliers so the point set size is always 100 points. For example, the experiment with 20% outliers means that each point set has 80 points inliers and 20 points outliers. We have tested different transformations including translation only, rotation only, translation, rotation and scaling, 5 DOF, and 6 DOF. However, we configured all algorithms to the affine settings so that we don't assume that we know the transformation model as a priori.

Success out of 1000 runs, translation only						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	1000	1000	989	938	804	557
ICP	935	193	55	11	3	2
CPD	1000	1000	1000	979	214	0
SCS	1000	1000	996	952	791	507
Average elapsed time in seconds for each run- translation only						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	0.074	0.124	0.244	0.464	0.781	1.106
ICP	0.0011	0.0012	0.0012	0.0013	0.0013	0.0014
CPD	0.0190	0.0221	0.0252	0.0326	0.0393	0.0351
SCS	0.4661	0.4702	0.5902	1.1535	2.1957	3.6989

Table 6.8: The performance of four different algorithms, see text for details.

Success out of 1000 runs, rotation only						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	164	168	161	135	132	120
ICP	335	74	17	7	2	0
CPD	627	637	624	570	135	0
SCS	1000	1000	996	933	710	444
Average elapsed time in seconds for each run- rotation only						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	1.4866	1.4186	1.4394	1.4639	1.4655	1.5083
ICP	0.0085	0.0094	0.0090	0.0093	0.0095	0.0098
CPD	0.0288	0.0305	0.0329	0.0374	0.0417	0.0387
SCS	0.4722	0.5155	0.7249	1.4916	2.8869	4.1285

Table 6.9: The performance of four different algorithms, see text for details.

Success out of 1000 runs, translation, rotation and scaling						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	267	234	204	170	116	85
ICP	2	0	1	0	0	0
CPD	632	626	620	565	123	0
SCS	1000	1000	997	932	747	415
Average elapsed time in seconds for each run, translation, rotation and scaling						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	1.2666	1.3179	1.3734	1.4401	1.5203	1.5595
ICP	0.0013	0.0013	0.0013	0.0013	0.0014	0.0013
CPD	0.0296	0.0307	0.0334	0.0374	0.0415	0.0389
SCS	0.4654	0.5129	0.7174	1.4647	2.6961	4.2332

Table 6.10: The performance of four different algorithms, see text for details

Success out of 1000 runs, 5 DOF (Translation, rotation, scale and sheer).						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	278	222	195	164	124	74
ICP	0	0	0	0	0	0
CPD	632	639	595	590	126	1
SCS	1000	1000	996	943	730	421
Average elapsed time in seconds for each run, 5 DOF						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	1.2447	1.3483	1.4021	1.4557	1.5188	1.5661
ICP	0.0013	0.0014	0.0013	0.0014	0.0014	0.0014
CPD	0.0298	0.0314	0.0339	0.0379	0.0418	0.0391
SCS	0.4734	0.5091	0.7404	1.3914	2.8625	4.2260

Table 6.11: The performance of four different algorithms, see text for details

Success out of 1000 runs, 6 DOF.						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	41	33	38	26	22	9
ICP	0	0	0	0	0	0
CPD	138	135	140	113	32	0
SCS	1000	999	992	927	686	336
Average elapsed time in seconds for each run, 6 DOF.						
Outliers	0%	10%	20%	30%	40%	50%
SL- RANSAC	1.5723	1.5936	1.5878	1.6021	1.6083	1.6246
ICP	0.0015	0.0016	0.0016	0.0018	0.0017	0.0016
CPD	0.0383	0.0398	0.0382	0.0393	0.0404	0.0408
SCS	0.4652	0.5120	0.7335	1.5001	3.0026	4.4418

Table 6.12: The performance of four different algorithms, see text for details

6.11.1 Discussion

From the tables 6.80 to 6.12, it can be argued that the SCS algorithm is the slowest algorithm between all other algorithms. However, it is clear also that the SCS is the most robust algorithm. While execution time is really important, we think that there is little point in comparing the execution time between two algorithms if they do not deliver the same robustness. In this context, the SCS algorithm fails if the target transformation is translation only and the outliers' ratio is within 30% (see Table 6.8) in this case CPD [161] is a better solution. However, the SCS algorithm wins elsewhere as it recovers transformations successfully even with high outliers rates where other algorithms fail. Moreover, the SCS algorithm is agnostic towards transformations, in other words, the SCS algorithm has very similar success rates whether it solves for translation, rotation, 5 DOF or 6 DOF transformations. Furthermore, the SCS algorithm has approximately the same execution time regardless of the transformation it searches for, in fact, the execution time depends mostly on the outliers' ratio. Also, it is possible to decrease the execution time by adopting parallel implementation. Figure 6.26 and Figure 6.27 both illustrate the performance of the SCS and CPD algorithms against outliers respectively.

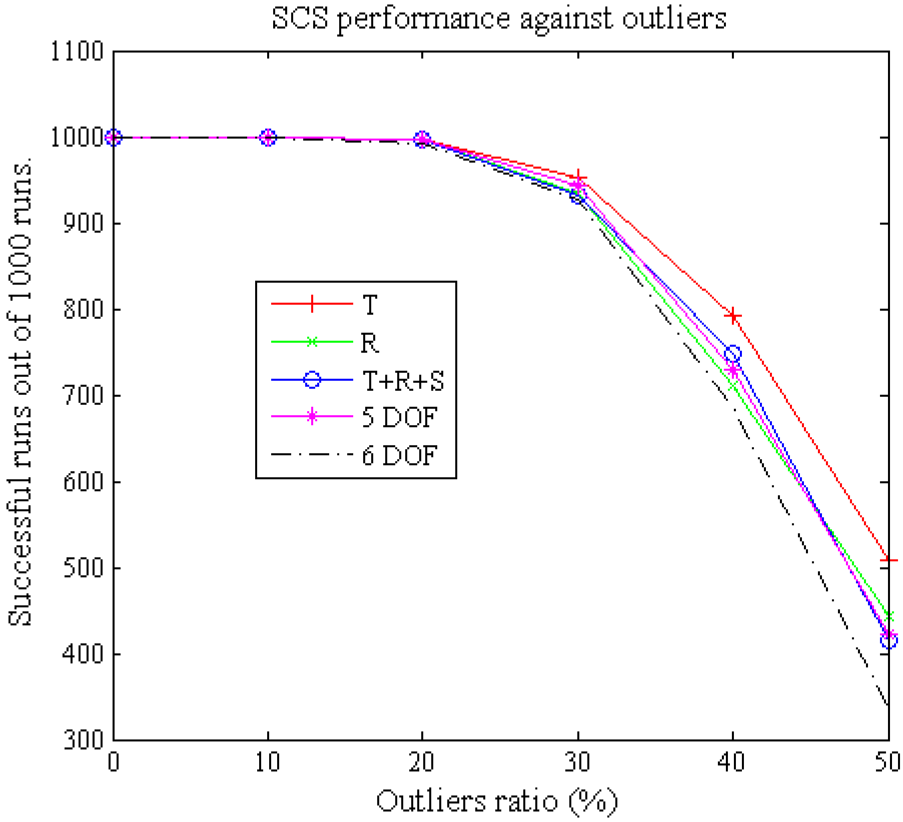


Figure 6.26: SCS performance against outliers for different transformations. T: translation, R: Rotation, S: scale.

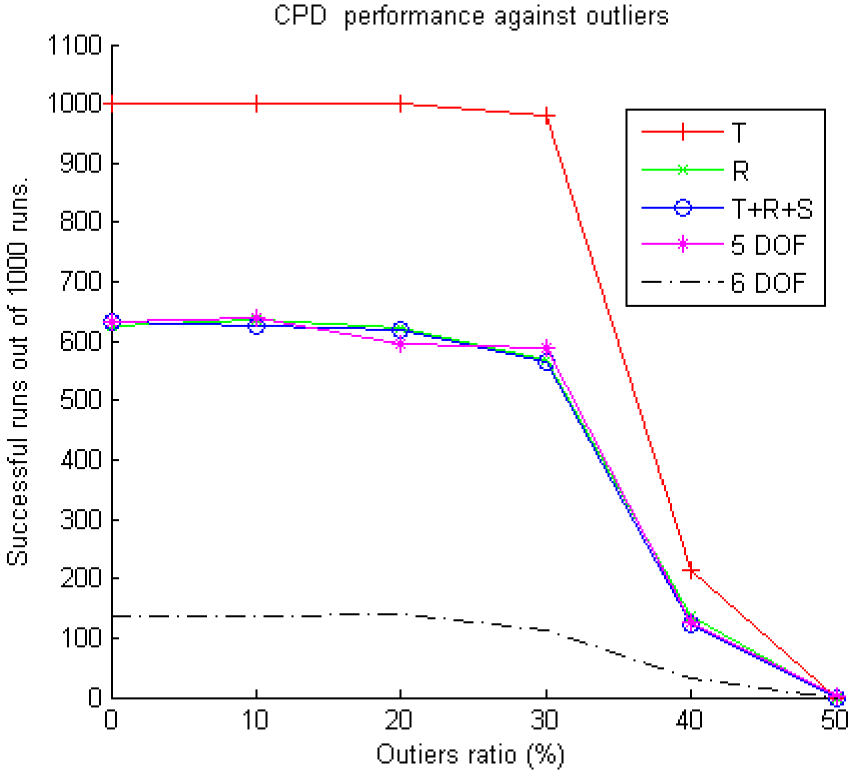


Figure 6.27: CPD [161] performance against outliers for different transformations. T: translation, R: Rotation, S: scale.

6.11.2 Where the SCS fails

Although the SCS algorithm is quite robust against outliers, there are some cases where it fails completely even with no outliers. These cases occur when there is a strong dependency between x and y coordinates in the point sets. For example, let us consider the case when there is a linear relation between x and y for all points in P i.e. $y =\propto x$ for some constant $\propto$. The algorithms fails to pick three pairs to infer the transformation as the pairs should not be collinear; however, all points in this case are collinear so the algorithm fails to infer the transformation. This case is a common problem shared between all point matching algorithms as it is ill-conditioned problem.

This problem doesn't exist if the relation is not completely linear as illustrated in Figure 6.28. This problem doesn't exist in acoustic images as features are fairly modelled with uniform distribution in the acoustic image.

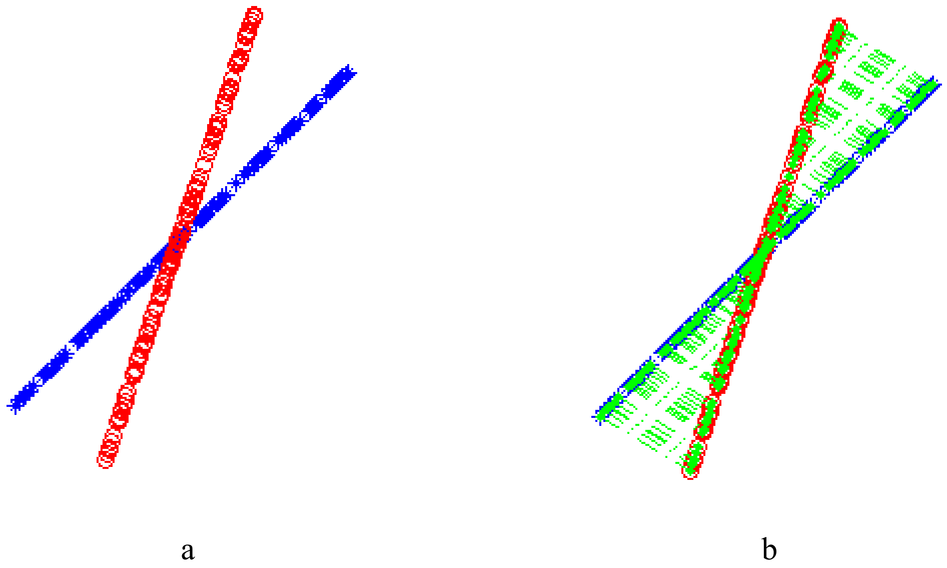


Figure 6.28: a) When $y =\propto x$ the SCS algorithm will not be able to recover the transformation. b) Adding Gaussian noise to the y coordinates with 10^{-6} std enables the SCS to recover the transformation.

6.12 Applying the SCS algorithm on acoustic images

The correspondence problem between two acoustic images is different from the correspondence problem between images obtained from optical cameras, mainly due to the nature of sensing the environment and building the images. The difference can be summarized in two points 1) optical images have abundant features in contrast to scarce features in acoustic images 2) SNR in optical images is much higher than SNR in acoustic images which affects the locus of the features. We use the aforementioned SCS methodology to find the best 2D-affine transform between two acoustic images and assign correspondences between them.

Dominant feature extraction algorithms for optical images do not usually select features close the edges or with high contrast; however, pixels with high contrast in acoustic images should be considered as features because they correspond to objects with high acoustic impedance in the imaged scene, therefore, we use simple and intuitive way to select pixels with high contrast relative to the image background.

Next, we explain the algorithm to extract features and find the correspondences between two acoustic images I_1 and I_2.

1. Divide the first image I_1 to small square blocks (we used 25x25 pixels in an image obtained from BlueView P900-130 sonar).
2. Calculate the standard deviation (std) of each block and select blocks exceeding pre-determined threshold.
3. Select the point with the maximum intensity in each block selected in the previous step, these points will serve as a core for candidate features.
4. Apply steps 1 to 3 for the second image I_2
5. Initialize candidate correspondences for each feature in the first image I_1 with features in the second image I_2; each feature in the first image I_1 will have a set of candidates so we can write:

$$F_1^i \text{Candidates} = \left\{F_2^j : F_{1x,y}^i - \gamma \leq F_{2x,y}^j \leq F_{1x,y}^i + \gamma\right\}$$

Where i, j are indexes for the features in the first and second images respectively, γ is the maximum distance (measured in pixels) a feature can move in any direction in the acoustic image due to the vehicle movement between two successive scans, thus γ is related to the maximum expected speed of the submerged vehicle and to the applied frame rate.

6. Apply SCS algorithm on the features F_1^i and F_2^jtaking into account the previous constraint in the filtering process.

The Std threshold affects the number of extracted features, which can be dynamically adjusted to limit the number of extracted features, this is particularly important as the number of features in both images should be the same (or close to each other) when we apply the SCS algorithm, Figures 6.29 , 6.30 , 6.31 , and 6.32 illustrate the extracted features with different std threshold.

SCS algorithm surpass SIFT and SURF in matching acoustic images, we have seen in Chapter 2, how SURF and SIFT gave more than 30% mismatches (Figure 2.14 and Figure 2.16), in contract, SCS algorithm matched the same frames with no outliers. This is mainly because of the intrinsic methodology of the SCS. While SURF and SIFT search for feature to feature best match, SCS searches for the best transformation which maps the first features set to the other.

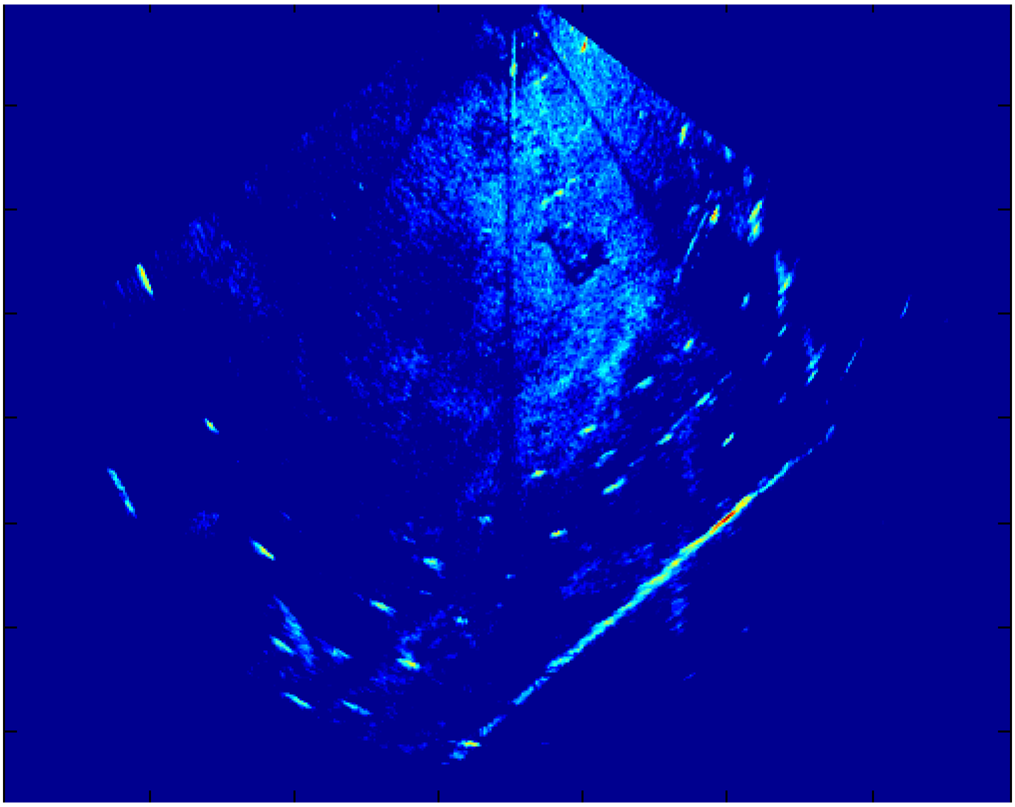


Figure 6.29: The original sonar image obtained from BlueView P900 in FAU marina

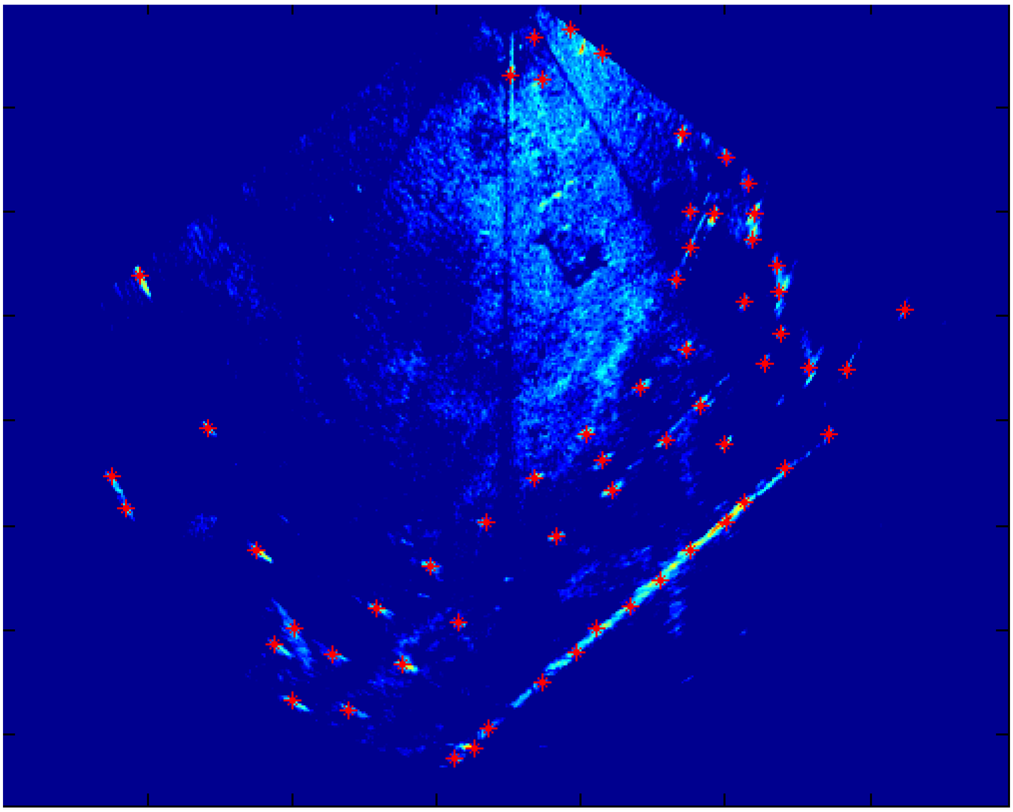


Figure 6.30: selected features from Figure 6.29 using std threshold 5.

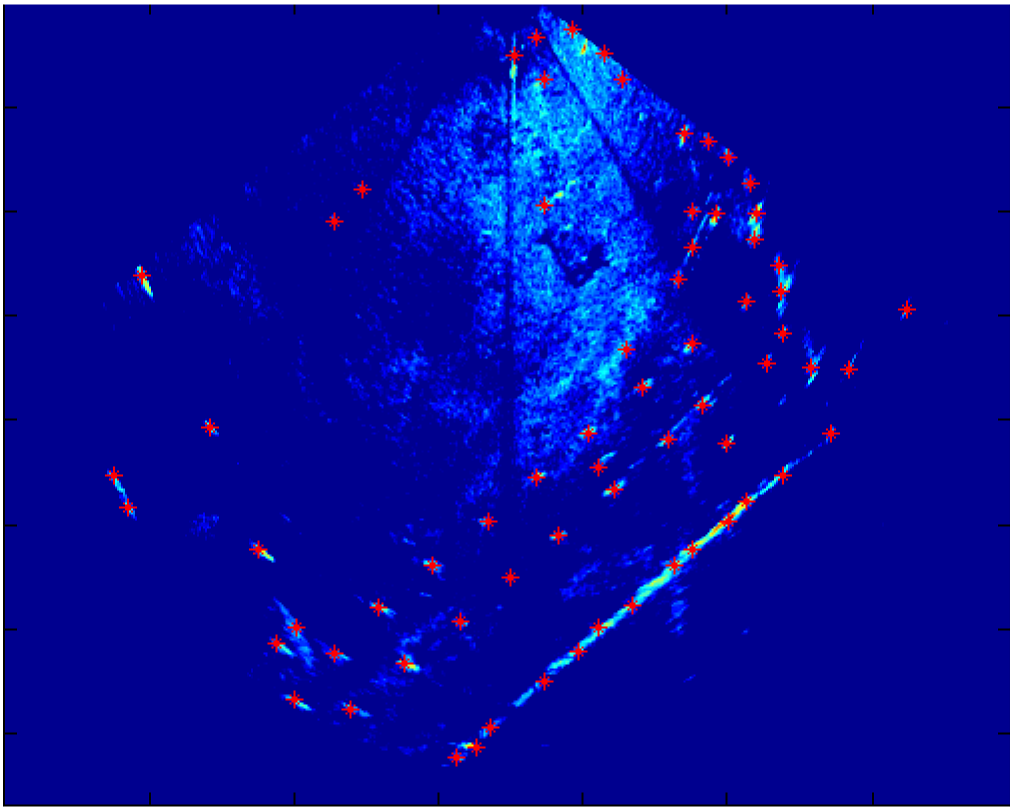


Figure 6.31: selected features from Figure 6.29 using std threshold 4.

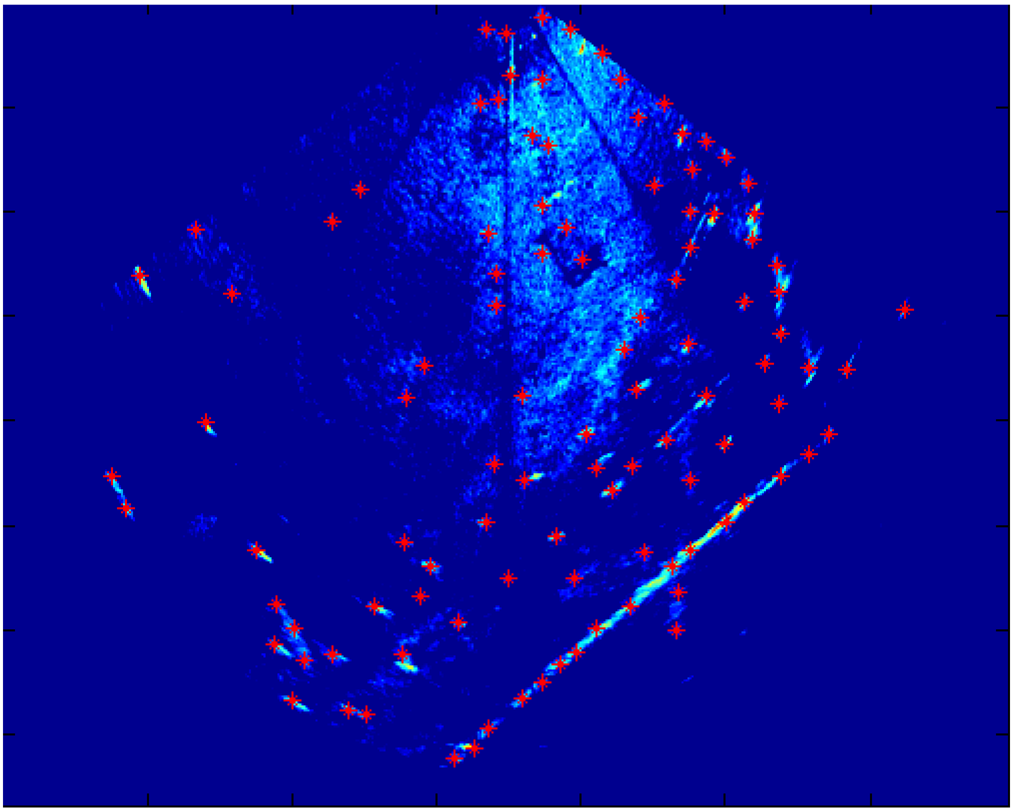


Figure 6.32: selected features from Figure 6.29 using std threshold 3.

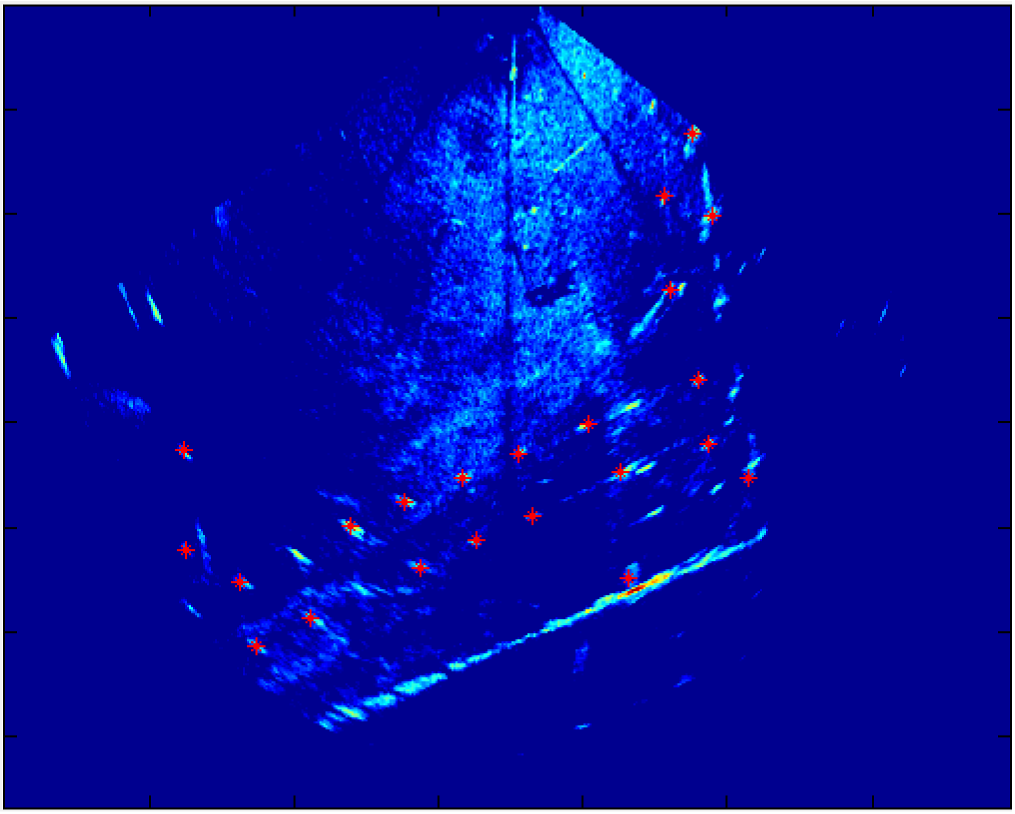


Figure 6.33: SCS features matched perfectly with the features in Figure 6.34

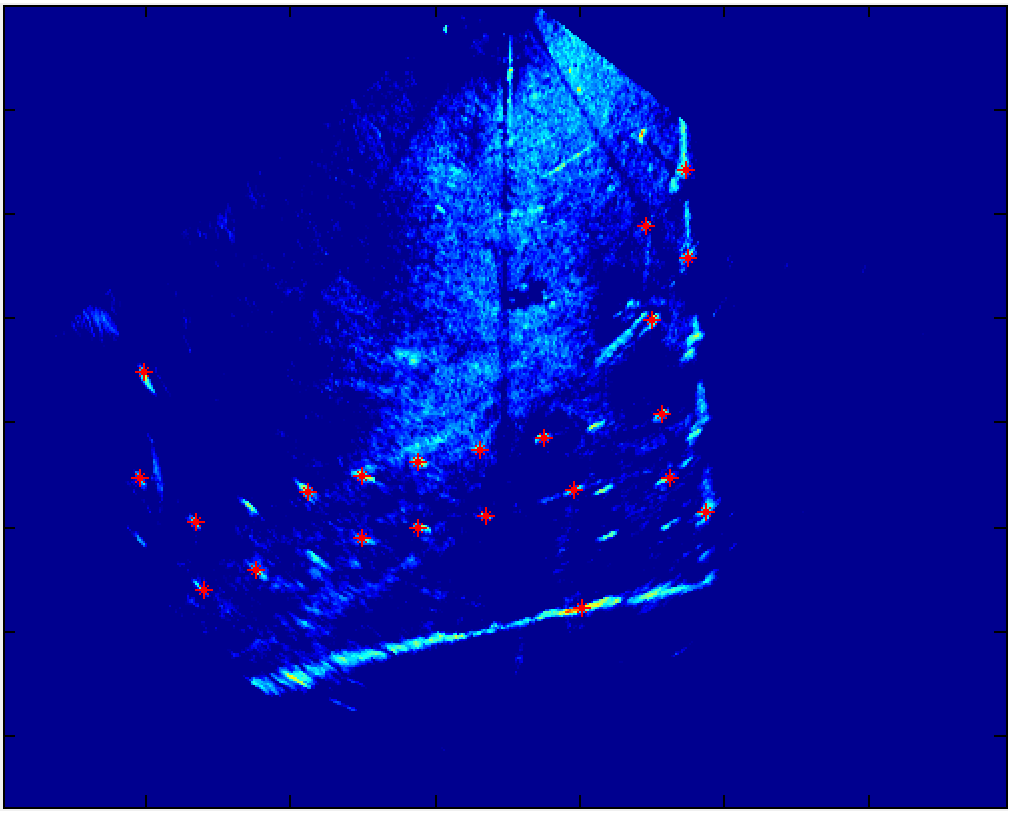


Figure 6.34: SCS features matched perfectly with the features in Figure 6.33.

The inferred transformation is : $T = \begin{bmatrix} 0.9953 & -0.1241 & 3.0365 \\ 0.1534 & 1.0250 & -123.4264 \\ 0 & 0 & 1 \end{bmatrix}$

6.13 Conclusion

In this Chapter, we have introduced a new algorithm, named Sorting the Correspondence Space (SCS), for point matching. SCS performs well in reducing the search space from 6 degrees of freedom in the transformation matrix to only one variable named ψ. Also, the algorithm is robust against outliers and can search for the best match in noisy feature sets. We have shown that SCS can be utilized to register acoustic images and find the best transformation matrix between successive sonar images. This can be employed in motion estimation and 3D reconstruction algorithms serving AUV platforms.

Chapter 7
Conclusion

7.1 Summary

In this thesis, we presented a new method for 3D reconstruction of underwater environment dubbed Acoustic Stereo Imaging (ASI) system. Two forward-looking sonars are used in the ASI system to reconstruct the scanned scene in a similar manner to classic stereo vision in terrestrial applications. The vertical ASI configuration is discussed and the according geometry is inferred to solve for the correspondence problem in two sonar views. The performance of the system is examined depending on different sonar parameters and the geometrical relation between both sonars. ASI system has the merits found in the T configured sonars regarding the cheap implementation, yet it solves its main demerit in terms of long delay scanning 3D volumes.

Motion estimation for Autonomous Underwater Vehicles (AUV) has a close relation to 3D reconstruction of the scanned scene. In this thesis we introduced a new motion estimation method based on two concepts of sampling the sonar arc and the Modified Discrete Uniform Distribution (MDUD). The motion estimation algorithm utilizes weighted-Hough array to solve for the ego-motion of the vehicle using the aforementioned concepts. We explore the impact of selected key factors on the accuracy of motion estimation from 2-D sonar cameras. These factors comprise the field of view of the camera, the number of feature matches in a motion sequence, and the inaccuracy in the knowledge of their image positions. The significance of each factor is analyzed based on the variances of the estimated motion parameters.

Autonomous Underwater Vehicles (AUVs) utilizing the proposed motion estimation framework and equipped with the ASI system for short-range 3D imaging can be used for enhanced object detection and classification. Numerous applications may benefit from this system including mine countermeasure missions, harbour monitoring, routinely dam inspection, search and rescue tasks, and coral reef monitoring to name but a few.

3D reconstruction and motion estimation both require image registration as a prerequisite, acoustic image registration is a challenging problem compared to the well-known correspondence problem in optical images. A new algorithm called *Sorting the Correspondence Space* (SCS) is introduced to tackle acoustic image registration; it serves 3D reconstruction and motion estimation for AUVs alike.

SCS algorithm can be easily adopted in different domains other than acoustic image registration; the SCS algorithm is a strong alternative for RANSAC and RANSAC-based algorithms introduced for 2D point matching. It can be used in medical image analysis, blurred image registration, and template recognition.

7.2 Further Work

There are different paths for future developments based on the methods which have been investigated in this thesis. Acoustic Stereo Imaging can be developed further in the horizontal configuration which may serve torpedo-shape vehicles. Also arbitrary configuration is of an interest for AUVs with complex geometry. ASI system calibration is an important project in the context of the ASI design and implementation, previous work can be a starting point for this project [162]. Integrating the ASI system with tracking methods similar to those proposed in [147] is an interesting topic. Also, 3D point registration, mesh building and outliers filtering are immediate aspects to consider in future ASI developments.

In motion estimation, acoustic image enhancement is an important step to improve shadow, background and reflection segmentation. This step has a major effect on motion estimation accuracy and can be considered for future developments. Different statistical methodologies may be investigated to exploit the arc sampling concept and the modified discrete uniform distribution (MDUD), these methods may interpret variations in the acoustic images to infer more distinctive features in the scanned environment. These features can be employed in proceeding motion estimation algorithms. Furthermore, Kalman filters and particle filters can be integrated with the proposed algorithms to obtain better measurements from the sensors on board of the AUVs.

As we have stated throughout the thesis that the correspondence problem is a prerequisite to implement 3D reconstruction or motion estimation using acoustic imaging. We have introduced the SCS algorithm for robust point matching and applied the SCS algorithm to register acoustic images. Further SCS developments may include investigating different outlier rejection approaches, extending the SCS algorithm to accommodate projective transformation (8 degrees of freedom), and implementing the SCS algorithm for 3D points registration. Also, enhanced SCS implementation can speed up the algorithm especially in Sxy filtering. Detailed analysis on the parameters of the SCS algorithm and its performance may open the door for different applications in the optical and/or medical domains.

Another important subject in the acoustic image registration is robust feature extraction. This step has a profound impact on the performance of the image registration algorithm and other subsequent algorithms for 3D reconstruction or motion estimation. In other words, developing robust feature extraction algorithms specifically designed for acoustic images is an emerging necessity for successful underwater 3D reconstruction and motion estimation algorithms.

Appendix A DIDSON Specifications

Detection Mode	
Operating Frequency	1.1 MHz
Beamwidth (two-way)	0.4° H by 14° V
Number of Beams	48
Beam Spacing	0.6°
(Extended) Window Start	0.83m to 52.3m in 0.83m steps
Frequency (Extended) Window Length	5m, 10m, 20m, 40m
Bin Size (relative to window length)	10mm, 20mm, 40mm, 80mm
Operating Range Pulse Length (relative to window length)	18µs, 36µs, 72µs, 144µs
Identification Mode	
Operating Frequency	1.8 MHz
Beamwidth (two-way)	0.3° H by 14° V
Number of Beams	96
Beam Spacing	0.3°
(Extended) Window Start	0.42m to 26.1m in 0.42m steps
Frequency (Extended) Window Length	1.25m, 2.5m, 5m, 10m
Bin Size (relative to window length)	2.5mm, 5mm, 10mm, 20mm
Operating Range Pulse Length (relative to window length)	$4.5\mu s, 9\mu s, 18\mu s, 36\mu s$
Both Modes	
Max Frame Rate (range dependent)	4-21 frames/s
Field-of-view	29°
Remote Focus	1m to Infinity
Control & Data Interface	UDP Ethernet
Aux Display	NTSC Video
Max cable length	(100/10BaseT) 61m/152m (200ft/500ft)
Max cable length	(twisted pair, Patton Extender) 1220m (4000ft)
Power Consumption	25 Watts typical
Weight in Air	7.9 kg (17.4 lb)
Weight in Sea Water	1.0 kg (2.2 lb)
Dimensions	31.0cm x 20.6cm x 17.1cm

Appendix B BlueView P900-130

Sonar Specification	
Operating Frequency	900 kHz
Update Rate	Up to 15 Hz
Field-of-View	130°
Max Range	100 m
Optimum Range	2 - 60 m
Beam Width	1° x 20°
Number of Beams	768
Beam Spacing	0.18°
Range Resolution	0.0256 m
Interface	
Supply Voltage	12 - 48 VDC
Power Consumption	19 W/23 W max
Connectivity	Ethernet/VDSL
Mechanical	
Weight in Air	2.58 Kg
Weight in Water	0.63 Kg
Depth Rating	1,000 m
Size L x W	0.29 x 0.13 m

Appendix C Eclipse

Eclipse is a highly versatile multibeam sonar which has the ability to clearly visualise the underwater environment in 3D.

The Eclipse carries out horizontal and vertical measurements through the use of Tritech's true time-delay beamforming and electronic beam steering technology. This coupled with the 3D underwater visualization make Eclipse the most flexible multibeam sonar on the market and the solution for mattress lay in zero visibility and also in search and salvage operations and pipeline inspections.

Eclipse can be deployed on an ROV at depths down to 1000m both in Forward Looking Navigation and 3D Model View modes.

Forward Looking Navigation

In forward-looking or search mode, Eclipse produces 2D imagery which can be used to aid navigation and obstacle avoidance when mounted on a ROV.

3D Model View

Eclipse's 3D Model View allows imaging up to 40m range, with 0.5º sweep steps. By electronically sweeping the 1.5º x 120º profiling beam, a 120º (horizontal) by 45 º (vertical) volumes can be produced ahead of the sonar. Depending on range setting, the Eclipse can image a complete a volume scan in less than one second.

Measurements possible with Eclipse's 3D model view include; range, bearing, horizontal and vertical distance and the slope angle between two points of interest. The 3D volume image can also be digitised onto a points cloud for export to third party applications for further processing.

Operating Frequency	240 kHz
Acoustic Angular Resolution	1.5°
Beam Width	120°
Number of Beams	256
Effective Angular Resolution	0.5°
Range Resolution	2.5cm (0.98”)
Typical Ranges	2D forward looking = 60m (197ft)
	2D search mode = 120m (393ft)
	3D = 40m (131ft)
Minimum Focus Distance	0.4m (1.31ft)
Scan Rate	140Hz @ 5m, 7Hz @ 100m

Appendix D Author Publications

- Assalih, H. Petillot, Y. and Bell, J. "Acoustic Stereo Imaging (ASI) system" in *OCEANS 2009 IEEE – Germany*, pp 1-7, 11-14 May 2009

- Assalih, H. Negahdaripour, S. and Petillot, Y., J. " 3-D Motion Estimation in passive navigation by acoustic imaging" in *OCEANS 2010 IEEE – WA Seattle*, pp 1-6, 20-23 Sept. 2010

- Negahdaripour, S. Assalih, H. and Petillot, Y., J. " Performance and accuracy in visual motion computation from FS sonar video sequences" in *OCEANS 2010 IEEE – WA Seattle*, pp 1-7, 20-23 Sept. 2010

- Assalih, H. Negahdaripour, S. and Petillot, Y., J. "Sorting the Corespondence Space" - *to be published soon.*

References

[1] S. Galli et al., "Orientation of migrating leatherback turtles in relation to ocean currents," *Animal Behaviour*, vol. 84, no. 6, pp. 1491-1500, 2012.

[2] H. Yoshida et al., "The high efficiency multi-less (HEML) fuel cell; A high energy source for underwater vehicles, buoys, and stations," in *OCEANS, 2011 IEEE - Spain*, 1-6, June 2011.

[3] O. Hasvold and K.H. Johansen, "The alkaline aluminium hydrogen peroxide semi-fuel cell for the HUGIN 3000 autonomous underwater vehicle," in *Autonomous Underwater Vehicles, 2002. Proceedings of the 2002 Workshop on*, 89-94, 2002.

[4] T. Hyakudome, H. Yoshida, S. Ishibashi, T. Sawa, and M. Nakamura, "Development of advanced Lithium-ion battery for underwater vehicle," in *Underwater Technology (UT), 2011 IEEE Symposium on and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC)*, 1-4, April 2011.

[5] E. Belcher, B. Matsuyama, and G. Trimble, "Object identification with acoustic lenses," in *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, vol. 1, 6-11 vol.1.

[6] Sound Metrics Corp, DIDSON Product Options, 2008.

[7] Inc. BlueView Technologies. (2010, Sep) P900 Series Data Sheet. [Online]. http://www.blueview.com/media/Data_Sheet_P900_Series_v2.pdf

[8] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision.*: Cambridge Univ. Press, 2001.

[9] Carlo Tomasi and Takeo Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, pp. 137-154, 1992.

[10] C.J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 3, pp. 206-218, Mar 1997.

[11] Piliang Gong, Qifeng Zhang, and Aiqun Zhang, "Stereo Vision Based Motion Estimation for Underwater Vehicles," in *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, vol. 3, 745-749, Oct. 2009.

[12] B. Kalyan, A. Balasuriya, H. Kondo, T. Maki, and T. Ura, "Motion estimation and mapping by autonomous underwater vehicles in sea environments," in *Oceans 2005 - Europe*, vol. 1, 436 - 441 Vol. 1, June 2005.

[13] S. Negahdaripour, P. Firoozfam, and P. Sabzmeydani, "On processing and registration of forward-scan acoustic video imagery," in *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*, 452-459, May 2005.

[14] H. Sekkati and S. Negahdaripour, "3-D Motion Estimation for Positioning from 2-D Acoustic Video Imagery," in *Pattern Recognition and Image Analysis*, Joan Martí et al., Eds.: Springer Berlin Heidelberg, 2007, vol. 4478, 80-88.

[15] Barbara Zitová and Jan Flusser, "Image registration methods: a survey," Image *and Vision Computing,* vol. 21, no. 11, pp. 977-1000, 2003.

[16] C. Chailloux, J.-M. Le Caillec, D. Gueriot, and B. Zerr, "Intensity-Based Block Matching Algorithm for Mosaicing Sonar Images," *Oceanic Engineering, IEEE Journal of*, vol. 36, no. 4, pp. 627-645, Oct. 2011.

[17] H. Sekkati, S. Negahdaripour, "3-D Motion estimation for positioning from 2-D acoustic video imagery," in *Proc. 3rd Iberian Conf. on Pattern Recog. Image Analysis*, Girona, Spain, June, 2007.

[18] Maria Palmese and Andrea Trucco, "An Efficient Digital CZT Beamforming Design for Near-Field 3-D Sonar Imaging," *IEEE Journal of Oceanic Engineering*, vol. 35, no. 3, pp. 584 - 594 , July 2010.

[19] A. Trucco, "Devising an Affordable Sonar System for Underwater 3-D Vision," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, no. 10, pp. 2348 - 2354 , Oct 2008.

[20] V. Murino and A. Trucco, "Three-dimensional image generation and processing in underwater acoustic vision," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1903 - 1948, Dec 2000.

[21] Peng Chen, Wei Zhu and Yayu Zhen, "Research of multi-FPGA signal processor for underwater 3-D imaging sonar system," in *2011 International Conference on Electrical and Control Engineering (ICECE)*, 4666 - 4669 , Yichang, China , 2011.

[22] Peng Chen; YaYu Zheng; Wei Zhu, "Optimized Simulated Annealing Algorithm for Thinning and Weighting Large Planar Arrays in Both Far-Field and Near-Field," *IEEE Journal of Oceanic Engineering*, vol. 36, no. 4, pp. 658 - 664, Oct 2011.

[23] Zhaofu Chen; Jian Li; P. Stoica; K.W. Lo, "Iterative Adaptive Approach for wide-band active sonar array processing," in *OCEANS 2010 IEEE* , 1 - 10 , Sydney , 2010.

[24] C. C. Gaudes, "Robust Array Beamforming With Sidelobe Control Using Support Vector Machines," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 574 - 58, Feb 2007.

[25] Maria Palmese and Andrea Trucco, "Three-Dimensional Acoustic Imaging by Chirp Zeta Transform Digital Beamforming," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 7, pp. 2080 - 2086 , July 2009.

[26] Blair Cunningham, Rolf Kahrs Hansen, Angus McFadzean, Martyn Sloss, "The Use of Advanced True 4D Sonar for Realtime Complex Mosaicing," CodaOctopus Research and Development Ltd, 26 August 2008.

[27] J. L. Sutton, "Underwater acoustic imaging," *Proceedings of the IEEE*, vol. 67, no. 4, pp. 554–566, Apr 1979.

[28] R. O. Nielsen, *Sonar Signal Processing*.: Artech , 1991.

[29] Mingsian R. Bai, Jia-Hong Lin and Kwan-Liang Liu, "Optimized microphone deployment for near-field acoustic holography: To be, or not to be random, that is the question," *Journal of Sound and Vibration*, vol. 329, no. 14, pp. 2809–2824, March 2010.

[30] Thomas S. Shores, *Applied Linear Algebra and Matrix Analysis*, S. Axler and K.A. Ribet, Ed. New York, USA: Springer Science+Business Media, LLC, 2007.

[31] L. Rosenblum, B. Kamgar-Parsi, E. Belcher, and O. Engelsen, "Acoustic imaging: the reconstruction of underwater objects," in *Visualization, 1991. Visualization '91, Proceedings., IEEE Conference on*, 94 -101, 414, Oct 1991.

[32] Peng Chen, Xiang Tian, and Yaowu Chen, "Optimization of the Digital Near-Field Beamforming for Underwater 3-D Sonar Imaging System," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 2, pp. 415 - 424 , Feb. 2010.

[33] S. Repetto, M. Palmese, and A. Trucco, "Design and Assessment of a Low-Cost 3-D Sonar Imaging System Based on a Sparse Array," in *Proceedings of the IEEE Instrumentation and Measurement Technology Conference, 2006. IMTC*, 410 - 415 , Sorrento, Italy, 2006.

[34] Y. Takasea, T. Anadaa, T. Tsuchiyaa, N. Endoha, N.Nakamwab and T. Tukiokac, "Real-time sonar system using acoustic lens and numerical analysis based on 2D/3D parabolic equation method," in *Oceans 2005 - Europe*, 23 - 28 Vol. 1 , Brest - France, 2005.

[35] Ju Wu, "Beam-forming and imaging using acoustic lenses: Some simulation and experimental results," in *Signal Processing Systems (ICSPS), 2010 2nd International Conference on*, V2-764 - V2-768 , Hongyu Bian, July 2010.

[36] K. M. Houston, "Three-dimensional acoustical imaging using micromechanical hydrophones," in *Conference Proceedings of OCEANS '95. MTS/IEEE. Challenges of Our Changing Global Environment.*, 1174 - 1182 vol.2 , San Diego, 1995.

[37] S. Repetto, M. Palmese, and A. Trucco, "High-resolution 3-D imaging by a sparse array: array optimization and image simulation," in *Oceans 2005 - Europe*, vol. 2, 763-768 Vol. 2, June.

[38] S. Holm and B. Elgetun, "Properties of the beampattern of weight- and layout-optimized sparse arrays," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 44, no. 5, pp. 983-991, Sept. 1997.

[39] M. Karaman, M.A. Kutay, and G. Bozdagi, "An adaptive speckle suppression filter for medical ultrasonic imaging," *Medical Imaging, IEEE Transactions on*, vol. 14, no. 2, pp. 283-292, Jun 1995.

[40] J.H. Hokland and T. Taxt, "Ultrasound speckle reduction using harmonic oscillator models," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, vol. 41, no. 2, pp. 215-224, March 1994.

[41] Chunhui XU, Akira Asada and Kazuki Abukawa, "A method of generating 3D views of aquatic plants with DIDSON," in *2011 IEEE Symposium on Underwater Technology (UT)*, 1 - 5, Tokyo, 2011.

[42] E. Chesters, M.J. Chantler, and V.J.C. Wright, "The use of a high-frequency pencil beam sonar to determine the position of tubular members of underwater structures," in *OCEANS '94. 'Oceans Engineering for Today's Technology and Tomorrow's Preservation.' Proceedings*, vol. 1, I/199 -I/204 vol.1, Sep 1994.

[43] M.H. Gross, O.G. Staadt, and R. Gatti, "Efficient triangular surface approximations using wavelets and quadtree data structures," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 2, no. 2, pp. 130-143, June 1996.

[44] G. Canepa, O. Bergem, and N.G. Pace, "A new algorithm for automatic processing of bathymetric data," *Oceanic Engineering, IEEE Journal of*, vol. 28, no. 1, pp. 62-77, Jan 2003.

[45] G. Canepa and O. Bergem, "Testing a new algorithm for automatic processing of bathymetric data," in *OCEANS '98 Conference Proceedings*, vol. 1, 313 -317 vol.1, Oct 1998.

[46] X. Lurton, "Precision analysis of bathymetry measurements using phase difference," in *OCEANS '98 Conference Proceedings*, vol. 2, 1131-1134 vol.2, 1998.

[47] CodaOctopus Products Ltd. (2009, April) Coda Echoscope Product Sheet. pdf Document. [Online].

http://www.codaoctopus.com/echoscope-3d-sonar/documents

[48] CodaOctopus Products Ltd. (2012, Sep) Echoscope 3D Sonar Images. Image. [Online]. http://www.codaoctopus.com/echoscope-3d-sonar/images

[49] Tritech International Limited. (2012, August) Eclipse Multibeam Imaging Sonar. pdf Document. [Online]. http://www.tritech.co.uk/media/products/multibeam-sonar-for-3d-model-view-of-sonar-imagery-eclipse.pdf

[50] Marine Electronics Ltd. (2012, August) PIN-POINT3D INTRUDER DETECTION SONAR. pdf Document. [Online]. http://www.marine-electronics.co.uk/brochures/3DSonar/PIN%20POINT%20V1.pdf

[51] Marine Electronics Ltd. (2010, 15th March) 3D Remote Profiling Sonar User Manual. pdf Document. [Online].

http://www.marine-electronics.co.uk/Manuals/3DRemote%20OM.pdf

[52] Inc BlueView Technologies. (2012, August) 3D Mechanical Scanning Sonar. pdf Document. [Online].

http://www.blueview.com/media/Data_Sheet_-_BV-5000_v8.pdf

[53] B. Zerr and B. Stage, "Three-dimensional reconstruction of underwater objects from a sequence of sonar images," in Image Processing, 1996. *Proceedings., International Conference on, vol. 3, 927-930 vol.3, 1996.*

[54] Ning Sun, Taebo Shim, and Maoyong Cao, "3D reconstruction of seafloor from sonar images based on the multi-sensor method," in Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE *International* Conference on, 573-577, Aug. 2008.

[55] Tae-Won Kim, A. Asada, S. Weatherwax, B. Collins and Junku Yuh Son-Cheol Yu, "Development of High-Resolution Acoustic Camera based Real-Time Object Recognition System by using Autonomous Underwater *Vehicles," in OCEANS 2006, 1 - 6 , Boston*, Massachusetts USA, 2006.

[56] P.N. Andono, E.M. Yuniarno, M. Hariadi, and V. Venus, "3D reconstruction of under water coral *reef images* using low cost multi-view cameras," in *Multimedia Computing and Systems (ICMCS), 2012 International Conference on*, 803-808, May 2012.

[57] A. Meline, J. Triboulet, and B. Jouvencel, "A camcorder for 3D underwater reconstruction of archeological objects*," in OCEANS 2010, 1-9, Sept.* 2010.

[58] F.-X. Espiau and P. Rives, "Extracting robust features and 3D reconstruction in underwater images," in OCEANS, 2001. MTS/IEEE Conference and Exhibition, vol. 4, 2564 -2569 vol.4, 2001.

[59] T. Nicosevici, R. Garcia, S. Negahdaripour, M. Kudzinava, and J. Ferrer, "Identification of Suitable Interest Points *Using Geometric and Photometric Cues in Motion Video for Efficient 3-D Environmental* Modeling," in Robotics and *Automation, 2007 IEEE International Conference on*, 4969-4974, April 2007.

[60] C. Beall, B.J. Lawrence, V. Ila, and F. Dellaert, "3D reconstruction of underwater structures," in Intelligent Robots and *Systems (IROS), 2010 IEEE/RSJ International Conference on, 4418-4423, Oct*. 2010.

[61] P. Firoozfam and S. Negahdaripour, "A multi-camera conical imaging system for robust 3D motion estimation, positioning and mapping from UAVs," in Advanced Video *and Signal Based Surveillance*, 2003. Proceedings. IEEE *Conference on*, 99-106, July 2003.

[62] A. Sedlazeck, K. Koser, and R. Koch, "3D reconstruction based on underwater video from ROV Kiel 6000 *considering underwater imaging conditions,"* in OCEANS 2009 - EUROPE, 1-10, May 2009.

[63] O. Pizarro, R.M. Eustice, and H. Singh, "Large Area 3-D *Reconstructions From Underwater Optical Surveys," Oceanic* Engineering, IEEE *Journal of*, vol. 34, no. 2, pp. 150-169, April 2009.

[64] Chris Harris and Mike Stephens, "A combined corner and edge detector," in In Proc. of *Fourth Alvey Vision Conference,* 147-151, 1988.

[65] Tinne Tuytelaars and Luc Van Gool, "Matching Widely Separated Views Based on Affine Invariant Regions," International Journal of Computer Vision, *vol. 59, pp. 61-85, 2004*.

[66] Berthold K. P. Horn, H.M. Hilden, and Shariar Negahdaripour, "Closed-Form Solution of Absolute Orientation using Orthonormal Matrices," JOURNAL OF *THE OPTICAL* SOCIETY AMERICA, vol. 5, no. 7, pp. 1127-1135, 1988.

[67] A. Sarafraz, S. Negahdaripour, and Y.Y. Schechner, "Performance assessment in solving the correspondence problem *in underwater stereo imagery," in OCEANS 2010, 1-7, Sept. 2010.*

[68] J.P. Queiroz-Neto, R. Carceroni, W. Barros, and M. Campos, "Underwater stereo," in Computer *Graphics and Image Processing, 2004. Proceedings. 17th Brazilian Symposium on*, 170-177, Oct. 2004.

[69] Kaiming He, Jian Sun, and Xiaoou Tang, "Single image *haze removal using dark channel prior," in* Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference *on*, 1956-1963, June 2009.

[70] S. Negahdaripour, "Epipolar Geometry of Opti-Acoustic Stereo Imaging," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 29, no. 10, *pp. 1776-1788, Oct 2007.*

[71] S. Negahdaripour, H. Pirsiavash, and H. Sekkati, "Integration of Motion Cues in Optical and *Sonar Videos for 3-D Positioning," in Computer Vision and* Pattern Recognition, 2007. *CVPR '07. IEEE Conference on*, 1-8, June.

[72] E. Belcher, W. Hanot, and J. Burch, "Dual-Frequency Identification Sonar (DIDSON)," in Underwater Technology, 2002. Proceedings of the 2002 International Symposium on*, 187-192, 2002.*

[73] A. Noureldin, T.B. Karamat, M.D. Eberts, and A. El-Shafie, "Performance Enhancement of MEMS-Based INS/GPS Integration for Low-Cost *Navigation Applications," Vehicular Technology, IEEE Transactions on, vol. 58,* no. 3, pp. 1077-1096, March 2009.

[74] Qi Nie, Wei Gao, Lin Zhao, and Wei Zu, "High Accuracy INS based on Fiber Optical Gyroscope for AUV Application," in Industrial Electronics and Applications, 2007. *ICIEA 2007. 2nd IEEE Conference* on, 2617-2622, May 2007.

[75] G. Grenon, P.E. An, S.M. Smith, and A.J. Healey, *"Enhancement of the inertial* navigation system for the Morpheus autonomous underwater vehicles," *Oceanic Engineering, IEEE Journal of*, vol. 26, no. 4, pp. 548-560, Oct 2001.

[76] M.J. Stanway, "Water profile navigation with an Acoustic Doppler Current Profiler," in OCEANS *2010 IEEE - Sydney*, 1-5, May 2010.

[77] Kearfott Corporation. (2012, Oct) KN-6050 SEADeViL INS/DVL/GPS Family Product Sheet. [Online].

http://www.kearfott.com/images/stories/pdf/DATASHEETS_KGN_NJ/SEA/kn-6050_seadevil.pdf

[78] R. McEwen, H. Thomas, D. Weber, and F. Psota, "Performance of an AUV navigation system at Arctic latitudes," Oceanic Engineering*, IEEE Journal of, vol. 30,* no. 2, pp. 443-454, April 2005.

[79] Kwang Hoon Kim, Jang Gyu Lee, and Chan Gook Park, "Adaptive Two-Stage *Extended Kalman Filter for a Fault-Tolerant INS-GPS* Loosely Coupled System," Aerospace and Electronic Systems, IEEE Transactions *on*, vol. 45, no. 1, pp. 125-137, Jan. 2009.

[80] A. Soloviev, "Tight Coupling of GPS and INS for Urban Navigation," Aerospace and Electronic Systems, IEEE Transactions on, vol. 46, no. 4, pp. 1731-1746, Oct. 2010.

[81] D.M. Bevly, J. Ryu, and J.C. Gerdes, "Integrating INS Sensors With GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tire Cornering *Stiffness," Intelligent Transportation* Systems, IEEE Transactions *on*, vol. 7, no. 4, pp. 483-493, Dec. 2006.

[82] Y. Watanabe, H. Ochi, T. Shimura, and T. Hattori, "A tracking of AUV with integration of SSBL *acoustic positioning and transmitted INS data*," in OCEANS 2009 - EUROPE, 1-6, May 2009.

[83] T. Ura and Kangsoo Kim, "On-site INS update of an AUV "r2D4" by SSBL based position estimation," in *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04, vol. 3, 1606* -1611 Vol.3, Nov. 2004.

[84] Shao Li, Li Sihai, and Pei Runsen, "Multiple sets of INS fault detection based on information fusion," in *Electronic Measurement Instruments (ICEMI), 2011 10th International Conference on, vol. 4, 81-85,* Aug. 2011.

[85] L. Stutters, Honghai Liu, C. Tiltman, and D.J. Brown, "Navigation Technologies for Autonomous Underwater Vehicles*,"* Systems, Man, and *Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 4, pp. 581-589, July 2008.

[86] R. Almeida, N. Cruz, and A. Matos, "Synchronized intelligent buoy network for underwater positioning," in OCEANS 2010, 1-6, Sep 2010.

[87] Y. Watanabe, H. Ochi, and T. Shimura, "A study of inverse SSBL acoustic positioning with data transmission for multiple AUV navigation," in OCEANS*, 2012 - Yeosu, 1-6,* May 2012.

[88] A. Caiti, A. Garulli, F. Livide, and D. Prattichizzo, "Localization of autonomous underwater vehicles by floating acoustic buoys: a set-membership approach," Oceanic Engineering, *IEEE Journal of, vol.* 30, no. 1, pp. 140-152, Jan. 2005.

[89] Fanlin Yang, Xiushan Lu, Yamin Dang, and Zhimin Liu, "Accurate and rapid localization of an AUV in an absolute reference frame using the iterative *resection," in OCEANS 2010 IEEE - Sydney,* 1-6, May 2010.

[90] Xianbo Xiang, Guohua Xu, Qin Zhang, Ying Guo, and Xinhan Huang, "A Novel Acoustic Navigation Scheme for Coordinated Heterogenous Autonomous Vehicles," in Mechatronics and Automation, 2007. *ICMA 2007. International Conference on, 2801-2806,* Aug. 2007.

[91] F.R. Driscoll, P.-P. Beaujean, W. Venezia, and L. Freitag, "Development and testing of an A-sized rapidly deployable navigation and *communication GATEWAY buoy," in OCEANS,* 2005. Proceedings of MTS/IEEE, 1234 -1241 Vol. 2, Sep 2005.

[92] T.C. Austin, R.P. Stokey, and K.M. Sharp, "PARADIGM: a buoy-based system for AUV navigation and tracking," in OCEANS 2000 MTS*/IEEE* Conference and Exhibition, vol. 2, 935 -938 vol.2, 2000.

[93] R. Sousa, A. Alcocer, P. Oliveira, R. Ghabcheloo, and A. Pascoal, "Joint positioning *and navigation aiding system for underwater robots," in OCEANS 2008,* 1-8, Sept. 2008.

[94] H.G. Thomas, "GIB buoys: an interface between space and depths of the oceans," in Autonomous Underwater Vehicles, 1998. AUV'98. Proceedings Of The 1998 Workshop *on*, 181-184, Aug 1998.

[95] James W. Youngberg, "A NOVEL METHOD FOR EXTENDING GPS TO UNDERWATER APPLICATIONS," NAVIGATION, vol. Vol. 38, No. 3, pp. *263* - 272, 1991.

[96] S. Carreno, P. Wilson, P. Ridao, and Y. Petillot, "A survey on Terrain *Based Navigation for AUVs,"* pp. 1-7, Sep. 2010.

[97] Feizhou Zhang, Xiuwan Chen, Min Sun, Ming Yan, and Dongkai Yang, "Simulation study of underwater passive navigation system based *on gravity gradient," in Geoscience and Remote Sensing* Symposium, 2004. IGARSS '04. Proceedings. 2004 *IEEE International*, vol. 5, 3111-3113 vol.5, 2004.

[98] C. Tyren, "Magnetic terrain navigation," in Unmanned Untethered Submersible Technology, Proceedings of the 1987 5th International Symposium on, vol. 5, 245-256, 1987.

[99] I. Nygren and M. Jansson, "Terrain navigation for underwater vehicles using the correlator method," *Oceanic Engineering, IEEE Journal* of, vol. 29, no. 3, pp. 906-915, July 2004.

[100] Peijuan Li, Xiaofei Zhang, and Xiaosu Xu, "Novel terrain integrated navigation system using neural network aided Kalman filter," in Natural *Computation (ICNC), 2010 Sixth International Conference on, vol.* 1, 445-448, Aug. 2010.

[101] B. Armstrong, E. Wolbrecht, and D.B. Edwards, "AUV navigation in the presence of a magnetic disturbance *with an extended Kalman filter," in OCEANS 2010* IEEE - Sydney, 1-6, May 2010.

[102] J. Mendel, "Extension of Friedland's bias filtering technique to a class of nonlinear systems," Automatic Control, IEEE Transactions on, *vol. 21, no. 2, pp. 296-298, Apr 1976.*

[103] Bo He et al., "Localization and map building based on particle filter and unscented Kalman Filter for an AUV," in Industrial Electronics and Applications, 2009. *ICIEA 2009. 4th* IEEE Conference on, 3926-3930, May 2009.

[104] M. Barisic, A. Vasilijevic, and D. Nad, "Sigma-point Unscented Kalman Filter used for AUV navigation," in Control Automation *(MED), 2012 20th Mediterranean* Conference on, 1365-1372, July 2012.

[105] Jianhong Wu and HongCai Zhang, "Data Fusion Algorithm Design of GPS/IMU Based on Fuzzy Adaptive Federated Kalman Filter," in Image and *Signal* Processing, 2009. CISP '09. *2nd International Congress on*, 1-4, Oct. 2009.

[106] F. Maurelli, Y. Petillot, A. Mallios, P. Ridao, and S. Krupinski, "Sonar-based AUV localization using *an improved particle* filter approach," in OCEANS *2009 - EUROPE*, 1-9, May 2009.

[107] G.T. Donovan, "Position Error Correction for an Autonomous Underwater Vehicle Inertial Navigation System (INS) Using a Particle Filter," Oceanic Engineering, IEEE Journal *of, vol. 37, no. 3, pp. 431-445, July 2012.*

[108] A.K. Lammas, K. Sammut, and F. He, "Improving navigational accuracy for AUVs *using the MAPR Particle Filter,"* in OCEANS 2008, 1-8, Sept. 2008.

[109] Y. Petillot and F. Maurelli, "A tree-based planner for active localisation: Applications to Autonomous Underwater Vehicles," in ELMAR, 2010 PROCEEDINGS, *479-483, Sept*. 2010.

[110] A. Mallios, P. Ridao, D. Ribas, F. Maurelli, and Y. Petillot, "EKF-SLAM for AUV navigation under probabilistic sonar scan-matching," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 4404-4411, Oct. 2010.

[111] E. Olson, J. Leonard, and S. Teller, "Robust *range-only beacon localization,*" in Autonomous Underwater Vehicles*, 2004 IEEE/OES*, 66-75, June 2004.

[112] Hong jian Wang, Jing Wang, Le Yu, *and Zhen ye Liu*, "A new SLAM method based on SVM-AEKF for AUV," in *OCEANS 2011*, 1-6, Sept. 2011.

[113] ITK Organization. (2013) ITK Software Guide. [Online]. *http://*www.itk.org

[114] C. E. Shannon, "A mathematical theory of communication," Bell system *technical journal, vol. 27, 1948.*

[115] J. A. Thomas T. M. Cover, Elements of information *Theory. USA: Hohn Wiley* & Sons, 1991.

[116] W. M. Wells P. A. Viola, "Alignment by maximization of mutual information," *International Journal of* Computer Vision, vol. 24, no. 2, pp. 137–154, 1997.

[117] D.G. Lowe, "Object recognition from local scale-invariant features," in Computer Vision*, 1999. The Proceedings of* the Seventh IEEE *International Conference on*, vol. 2, 1150-1157 vol.2.

[118] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Int. J. Comput. Vision, *vol.* 60, no. 2, pp. 91-110, Nov 2004.

[119] Tony Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," Journal of Applied *Statistics, pp. 224-270, 1994.*

[120] Matthew Brown and David Lowe, "Invariant Features from Interest Point Groups," in In British Machine Vision Conference, *656-665, 2002.*

[121] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in In ECCV, 404-417, 2006.

[122] G. L. Scott and C. Longuet-Higgins, "An algorithm for associating the features of two *images," Proceedings of the Royal Society London*: Biological Sciences, vol. 244, no. 1309, pp. 21–26, Apr 1991.

[123] H. Li and R. Hartley, "A new and compact algorithm for simultaneously *matching and estimation,"* in IEEE International Conference on Acoustics, Speech, *and Signal Processing, Proceedings (ICASSP '04)*, 5-8 vol.3, 2004.

[124] Xiabi Liu, Yunde Jia, and Yanjie Wang, "An Eigenvector Approach Based on Shape *Context Patterns for Point Matching," in Communications and Information Technologies*, 455 -458, 2006.

[125] P J Besl and H D McKay, "A method for registration of 3-D *shapes," Pattern Analysis and Machine Intelligence,* IEEE Transactions on, vol. 14, pp. 239 -256, Feb 1992.

[126] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," Image and Vision Computing, vol. 21, pp. 1145–1153, 2003.

[127] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in in International Conference on *3D Digital Imaging and Modeling (3DIM), 145*–152, 2001.

[128] A Myronenko and Xubo Song, "Point Set Registration: Coherent Point Drift," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, pp. 2262 -2275, Dec. 2010.

[129] S. Gold, C. P. Lu, A. Rangarajan, S. Pappu, and E. Mjolsness, "New algorithms for 2D *and 3D point matching: Pose* estimation and corresp.," NIPS, The MIT Press, vol. 7, pp. 957–964, 1995.

[130] Bin Luo and E R Hancock, "Structural graph matching using the *EM algorithm* and singular value decomposition," Pattern Analysis and Machine Intelligence, IEEE *Transactions on*, vol. 23, pp. 1120 -1136, Oct 2001.

[131] A. Rangarajan, H. Chui, E. Mjolsness, L. Davachi, P. S. Goldman-Rakic, *and J. S. Duncan, "A robust* point matching algorithm for autoradiograph alignment," MIA, vol. 1, no. 4, pp. 379–398, 1997.

[132] A Joshi and C Lee, "On the problem of correspondence in range data and some inelastic uses for elastic nets," Neural Networks, IEEE Transactions on, vol. 6, pp. 716 -723, May 1995.

[133] W. M. Wells, "Statistical approaches to feature-based object recognition," IJCV, , vol. 22, no. 1-2, pp. 63–98, Jan 1997.

[134] A D Cross and E R Hancock, "Graph matching with a dual-step EM algorithm," *Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol.* 20, pp. 1236 -1253, Nov 1998.

[135] B. Luo and E. R. Hancock, "A unified framework *for alignment and correspondence," CVIU, vol.* 92, no. 1, pp. 26–55, 2003.

[136] Martin A Fischler and Robert C Bolles, "Random sample consensus: a paradigm for model fitting with *applications to image analysis and automated cartography," Communications of* the ACM, vol. 24, no. 6, pp. 381-395, 1981.

[137] J. Matas and O. Chum, "Randomized RANSAC with sequential probability ratio test," in Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, *vol.* 2, 1727-1732 Vol. 2, Oct.

[138] O. Chum and J. Matas, "Optimal Randomized RANSAC," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 30, *no.* 8, pp. 1472-1482, 2008 Aug.

[139] M. Carcassoni and E.R. Hancock, "Point pattern matching with robust spectral *correspondence," in Computer Vision and Pattern Recognition, 2000.* Proceedings. IEEE Conference on, vol. 1, 649-655 vol.1.

[140] P.B. Van Wamelen, Z. Li, and S.S. Iyengar, "A fast expected time algorithm for the 2-D *point pattern matching problem,"* Pattern Recognition, vol. 37, no. 8, pp. 1699-1711, 2004.

[141] Jason A. Denton and J. Ross Beveridge, "An algorithm for projective point matching in the presence of spurious *points," Pattern Recognition, vol. 40, no*. 2, pp. 586-595, 2007.

[142] D. Nister, "Preemptive RANSAC for live structure and motion estimation," in Computer Vision, 2003. Proceedings. Ninth IEEE *International Conference on, 199-206* vol.1, Oct.

[143] P.H.S. Torr and A. Zisserman, "MLESAC: A New Robust Estimator with Application *to Estimating Image Geometry," Computer Vision and Image Understanding, vol. 78,* no. 1, pp. 138-156, 2000.

[144] B.J. Tordoff and D.W. Murray, "Guided-MLESAC: faster image transform estimation by using *matching priors," Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 27, no. 10, pp. 1523-1535, 2005 Oct.

[145] J. Rabin, J. Delon, and Y. Gousseau, "A Statistical Approach to the *Matching of Local Features,"* SIAM Journal on Imaging Sciences, vol. 2, no. 3, pp. 931-958, 2009.

[146] O. Chum and J. Matas, "Matching with PROSAC - progressive sample consensus," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. *IEEE Computer Society* Conference on, vol. 1, 220-226 vol. 1, June.

[147] A. Trucco and S. Curletto, "Extraction of 3D information from sonar image sequences," Systems, Man*, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 33, no. 4, pp. 687-699, 2003 Aug.

[148] Pan-Mook Lee et al., "Discrete-time quasi-sliding mode control of an autonomous underwater vehicle," Oceanic Engineering, IEEE Journal of, vol. 24, no. 3, pp. *388-395, Jul 1999.*

[149] Ji Hong Li and Pan Mook Lee, "A neural network adaptive controller design for free-pitch-angle diving behavior of an autonomous underwater vehicle," Robotics and Autonomous Systems, *vol. 52, no.* 2-3, pp. 132-147, 2005.

[150] H. Sekkati and S. Negahdaripour, "Direct and Indirect 3-D Reconstruction from *Opti-Acoustic Stereo Imaging," in 3D Data Processing, Visualization, and Transmission, Third* International Symposium on, 615-622, June 2006.

[151] Donna M. Kocak and Frank M. Caimi, "The Current Art of Underwater Imaging With *a* Glimpse of the Past and Vision of the Future," *Marine Technology Society Journal*, vol. 39, no. 3, pp. 5-26, 2005.

[152] Donna M. Kocak, Fraser R. Dalgleish, Frank M. Caimi, and Yoav Y. Schechner, "A Focus on *Recent Developments and Trends in* Underwater Imaging," Marine Technology Society Journal, vol. 42, no. 1, pp. 52-67, 2008.

[153] H. C. Longuet-Higgins, "The visual ambiguity of a moving plane," Proceedings *of The Royal Society* of London. Series B, Biological Sciences (1934-1990), vol. 223, pp. 165-175, 1984.

[154] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two *projections,"* Nature, vol. 293, no. 5828, pp. 133-135, Sep 1981.

[155] S. Negahdaripour and Ali Taatian, "3-D motion estimation by integrating visual cues in 2-D multi-modal *opti-acoustic stereo sequences," Computer Vision and Image Understanding*, vol. 114, no. 8, pp. 928-941, 2010.

[156] P E Hart, "How the Hough transform was invented [DSP History]," *Signal Processing Magazine, IEEE*, vol. 26, pp. 18 -22, November 2009.

[157] James E. Gentle and James E. Gentle, "Numerical Linear Algebra," in Computational Statistics.*: Springer New York,* 2009, 203-240.

[158] Shaohong Yan and Aimin Yang, "Explicit algorithm to the inverse of Vandermonde Matrix,*" in Test and Measurement, 2009. ICTM '09.* International Conference on, vol. 2, 176-179, Dec.

[159] L Lupas, "On the computation of the generalized Vandermonde matrix inverse," *Automatic Control, IEEE Transactions on,* vol. 20, pp. 559 - 561, Aug 1975.

[160] J A Heinen and R J Niedejohn, "Comments on Inversion of the *VanderMonde matrix," Signal Processing Letters*, IEEE, vol. 4, p. 115, April 1997.

[161] A. Myronenko and Xubo Song, "Point Set Registration: Coherent Point Drift*,"* *Pattern Analysis and Machine Intelligence*, IEEE Transactions on, vol. 32, no. 12, pp. 2262-2275, 2010.

[162] S. Negahdaripour, "Calibration of DIDSON forward-scan acoustic video camera," in *OCEANS, 2005. Proceedings of MTS/IEEE*, 1287-1294 Vol. 2, Sept.