

Image registration for sonar applications

Benjamin Thomas Henson

Doctor of Philosophy

University of York
Electronic Engineering

August 2017

Abstract

This work develops techniques to estimate the motion of an underwater platform by processing data from an on-board sonar, such as a Forward Looking Sonar (FLS). Based on image registration, a universal algorithm has been developed and validated with in field datasets. The proposed algorithm gives a high quality registration to a fine (sub-pixel) precision using an adaptive filter and is suitable for both optical and acoustic images. The efficiency and quality of the result can be improved if an initial estimate of the motion is made. Therefore, a coarse (pixel-wide) registration algorithm is proposed, this is based on the assumption of local sparsity in the pixel motion between two images. Using a coarse and then fine registration, large displacements can be accommodated with a result that is to a sub-pixel precision. The registration process produces a displacement map (DM) between two images. From a sequence of DMs, an estimation of the sensor's motion is made. This is performed by a proposed fast searching and matching technique applied to a library of modelled DMs. Further, this technique exploits regularised splines to estimate the attitude and trajectory of the platform. To validate the results, a mosaic has been produced from three sets of in field data. Using a more detailed model of the acoustic propagation has the potential to improve the results further. As a step towards this a baseband underwater channel model has been developed. A physics simulator is used to characterise the channel at waymark points in a changing environment. A baseband equivalent representation of the time varying channel is then interpolated from these points. Processing in the baseband reduces the sample rate and hence reduces the run time for the model. A comparison to a more established channel model has been made to validate the results.

Table of contents

Abstract	2
List of tables	9
List of figures	11
Acknowledgements	18
Dedication	20
Declaration	22
Acronyms	25
1 Introduction	27
1.1 Introduction to underwater acoustic imaging	28
1.2 Algorithm development	30
1.3 Contributions	31
1.4 Thesis structure	32
2 Registration Algorithm Development	33
2.1 Introduction	33
2.2 The registration problem	39
2.2.1 Simple motion models	40
2.2.2 Strategy for research	41
2.3 Adaptive filter registration	42
2.3.1 Adaptive filter overview	42
2.3.2 Smoothness and diversity	45
2.3.3 Sparse promoting penalty	48
2.3.4 Evaluation methods for optical flow	49

Table of contents

2.3.5	Adaptive filter registration results	51
2.3.6	Adaptive filter registration discussion	63
2.3.7	Sonar type noise	67
2.4	Conclusion and discussion	69
3	Registration algorithm for FLS images	71
3.1	Coarse DM estimation	73
3.1.1	Sparse representation techniques	73
3.1.2	Pixel block matching techniques	75
3.1.3	Selection of measurement (sample) points	77
3.1.4	Mode filter	78
3.1.5	Forward-backward registration	79
3.2	Fine DM estimation	80
3.2.1	Re-centring and reordering	80
3.2.2	Adaptive filtering and centre estimation	81
3.2.3	Convolution kernel peak position estimation	83
3.3	Complexity	85
3.4	Conclusion and discussion	88
4	Sensor motion estimation	91
4.1	Introduction	91
4.2	Preprocessing of the fine DM	94
4.3	Motion model	94
4.4	Estimation of the motion vector	95
4.5	Attitude-trajectory estimation	97
4.5.1	P-splines	99
4.5.2	Attitude-trajectory generation	99
4.5.3	Mosaic generation	103
4.6	Conclusion and discussion	105
5	Experiment results	107
5.1	Middlebury dataset	107
5.2	Acoustic imaging datasets	110
5.2.1	Ship's hull example dataset	111
5.2.2	Dam inspection example dataset	116
5.2.3	Pipeline and pile dataset	120
5.3	Conclusion and discussion	124

6 Conclusion and further work	125
6.1 Conclusion	125
6.2 Further research	126
References	129
Appendix A Debris dataset example	139
A.1 Pitch and altitude estimation	139
A.2 Debris example dataset	140
Appendix B Underwater Acoustic channel model	143
B.1 Signal Distortions	144
B.2 Acoustic Channel Modelling	145
B.2.1 Modelling Techniques	147
B.3 Baseband Waymark Channel Model	153
B.3.1 Underwater Channel Simulation	153
B.3.2 Previous work	154
B.3.3 Baseband Waymark Channel Model	155
B.3.4 Comparison of Waymark and VirTEX channel models	157
B.3.5 Discussion	159
B.4 Channel modelling for imaging	161
B.5 Conclusion and discussion	162

List of tables

2.1	A summary of the registration results from the initial registration algorithm.	63
2.2	A summary of the registration results from the initial registration algorithm.	64
2.3	A summary of the registration results from the initial registration algorithm.	64
2.4	A summary of the best registration results from the initial registration algorithm.	66
2.5	A comparison of the RLS and Sparse filter with Normalised Cross Correlation (NCC) and Sum of Absolute Differenced (SAD). The Hydrangea image was uses with a single scan path and 0 iterations.	66
2.6	A comparison of the RLS and RLS-Sparse filter with different amounts of added noise. The Hydrangea image was uses with a single scan path and 0 iterations.	69
3.1	Complexity associated with the coarse displacement estimate	86
3.2	Complexity associated with the fine displacement estimate [95]	87
3.3	Complexity comparison for classical and advanced implementation of RLS algorithm [95]	87
5.1	A comparison of the RLS, RLS-Sparse filter and the final sonar registration algorithm with different amounts of added noise. The Hydrangea image was uses with a single scan path and 0 iterations.	110

List of figures

1.1	An illustration of a typical field of view of a forward looking sonar.	29
1.2	An example of a single forward looking sonar frame. Original data from [32, 33].	29
1.3	A block diagram showing how the research topics relate to each other. . . .	30
2.1	An illustration of a typical field of view of a forward looking sonar.	35
2.2	An illustration of the projection of the reflections onto the beam centre caused by a single wide vertical beam. Also shown is how the single point illumination casts shadows in the image.	35
2.3	The radial beam of the FLS with the area represented by each pixel increasing with range.	36
2.4	Illustrations of relative movements within a scene.	39
2.5	Proper ridged transform.	40
2.6	Affine transform.	40
2.7	A colour wheel to show the movement; hue shows the direction and the saturation represents the magnitude.	41
2.8	An adaptive filter used to identify the convolution kernel between two images.	42
2.9	A comparison of a zig-zag scan path with a Hilbert SFC scan path. It can be seen that the number of transitions between objects is greatly reduced with the SFC.	46
2.10	An example of a Moore SFC. This is related to the Hilbert curve but is closed allowing for a continuous path.	47
2.11	(a) location of the example filter, note that at this point there is a transition between two movements. (b) and (c) 3D and 2D plot, respectively, of the filter coefficients for the l_2 cost function. (d) and (e) 3D and 2D plot, respectively, of the filter coefficients for the $l_2 + l_1$ cost function.	50
	(a)	50
	(b)	50

List of figures

(c)	50
(d)	50
(e)	50
2.12	RLS filter: EE (a), AE (b), IE (c) for the example images against aperture size. The errors begin to plateau at a size of approx 23×23 pixels.	53
2.13	RLS-sparse promoting filter: EE (a), AE (b), IE (c) for the example images against aperture size. It can be seen that the errors change more slowly with aperture size. An aperture of 23×23 was selected as a compromise between all of the images.	53
(a)	53
(b)	53
(c)	53
(a)	53
(b)	53
(c)	53
2.14	(a) A diagram depicting the adaptive filter loop for the iterative registration. A , adaptive filter is used with four different scan path directions. B , the result from the scan path with the smoothest DM surface is selected. C , the total DM is accumulated for each iteration of the registration. D , a new “corrected” target is interpolated from the original target image and the reverse for the DM. E , the new target image is used in the next iteration of the registration. (b) a more detailed view if the adaptive filter. Image No. NCAP-000-000-081-843 [88].	55
(a)	55
(b)	55
(a)	Colour wheel key for the results of the adaptive filter registration. . .	57
(b)	Rubber Whale reference image and ground truth DM.	57
(c)	Hydrangea reference image and ground truth DM.	57
2.15	Test images Rubber Whale Hydrangea, Grove2 and Urban2 from the Middlebury dataset. In addition an analytically generated ‘lines’ test image was created.	58
(d)	Grove2 reference image and ground truth DM.	58
(e)	Urban2 reference image and ground truth DM.	58
(f)	Lines reference image and ground truth DM.	58
(a)	Reference image.	59
(b)	Ground truth displacement map (optical flow).	59

(c)	Estimated displacement map.	59
(d)	Difference between the Ground truth and the estimated displacement map.	59
2.16	An example of registration of the Rubber Whale image from the Middlebury dataset.	60
(e)	Initial error in pixel intensities between the reference and target images.	60
(f)	Final error in pixel intensities between the reference and target images.	60
(g)	Interpolated output image. The target image pixel interpolated back to the estimated position in the reference image.	60
(a)	zig-zag scan pattern top to bottom (Experiment 1).	61
(b)	Space filling curve scan pattern (Experiment 2).	61
(c)	Space filling curve scan pattern, with diversity in scan direction (Experiment 3).	61
2.17	Five experiment results using the Rubber Whale image from the Middlebury dataset.	62
(d)	Space filling curve scan pattern, with diversity in scan direction and iterative registration (Experiment 4). The best result was at zero iterations.	62
(e)	Space filling curve scan pattern, with diversity in scan direction, iterative registration and sparse promoting penalty (Experiment 5). The best result was at zero iterations.	62
2.18	(a), the hydrangea reference image. (a) the background and foreground separation using intensity level thresholding.	67
(a)	67
(b)	67
2.19	The noise distributions added for each of the three tests. Test 1 has the distribution Shadow 1 added to the background/‘shadow region’, and Reflectors 1 distribution added to the foreground/‘reflector’ region. Shadow and Reflectors 2 for test 2 and Shadow and Reflectors 3 for test 3. For the Gaussian distribution $\mu = 35$ and $\sigma = (4, 8, 12)$. For the Rayleigh noise the start point was 50 and the peak of the distribution was at (10, 20, 30).	68
2.20	(a) reference image, (b) target image. The two images to register with noise added as per test 3.	68
(a)	68
(b)	68
3.1	A block diagram showing the proposed registration algorithm for FLS images.	72

List of figures

3.2	An illustration of the interpolation using the Mode filter	79
3.3	A block diagram showing the forward and backward estimates and the sum of the vectors. The displacements where the difference is within a certain magnitude are retained.	80
3.4	An illustration (with the same image as used in Fig. 2.9) of how the SFC might be rearranged based on the results of the coarse DM estimate. In this example all three shapes would have different movements. The number of motion transitions can be seen in graph where the colour indicates each object (motion) boundary.	82
3.5	An illustration (with the same image as used in Fig. 2.9) of how the SFC might be rearranged based on the results of the coarse DM estimate. In this example all three shapes would have different movements.	83
3.6	An illustration of a single target pixel and the reference image aperture from which a convolution kernel is estimated.	84
3.7	A block diagram showing the proposed registration algorithm.	85
3.8	Example outputs at selected stages of the proposed inter-frame DM estimation algorithm.	90
4.1	An illustration of a typical field of view of a forward looking sonar. The sonar looks upon the surface being inspected, for instance the underside of a ship's hull.	92
4.2	A block diagram on the proposed algorithm for the attitude-trajectory estimation.	93
4.3	Position of the point $[u_x, u_y]$ on the seabed.	95
4.4	An example of the inter-frame registration weighting (first 400 frames of the dam dataset - see section 5.2.2).	97
4.5	An example match for the estimated DM and a modelled motion DM. . . .	98
4.6	Two examples of attitude-trajectory for the 3 DOF, unsmoothed is plotted in green, smoothed in blue (dam inspection dataset - Section 5.2.2).	101
	(a) $\mu = 10^{-5}$	101
	(b) $\mu = 10^2$	101
4.7	The result for the dam wall attitude-trajectory estimation with the smoothed parameters $\mu = 10^{-5}$ and $\mu = 10^2$. The sonar sensor trajectory is shown in red and the attitude at every 30 th frame is shown as a blue arrow.	102
	(a) A mosaic showing a track along a dam wall where the motion in minimally smoothed.	102

	(b)	A mosaic showing a track along a dam wall where the motion is more heavily smoothed.	102
4.8	(a)	is a single frame moved and interpolated onto the mosaic grid. (b) the map of the number of contributing frames for each pixel. (c) the mosaic with the example frame blended into it.	104
	(a)	104
	(b)	104
	(c)	104
5.1		Colour wheel key for the results of the coarse-fine registration.	108
5.1		The results from four experiments using selected images from the Middlebury dataset [77].	109
5.2		An example application for a FLS is the inspection of the underside of a ship. An inspection robot with an FLS would follow multiple tracks along the ship's hull (illustrated here as red arrows) to ensure that whole surface is inspected.	111
5.3		Zig-zag path of the inspection robot. The FLS path is shown as a red line and the attitude as a blue arrow. The FLS footprints through time are shown in grey scale.	112
5.4		The coordinate system for the ship's hull dataset.	113
5.5		A single frame from the ship's hull dataset, shown in Cartesian coordinates.	113
5.6		A mosaic of 560 frames showing a track along the ship's hull (track motion is in positive y direction). The sonar sensor trajectory is shown in red and the attitude at every 30 th frame is shown as a blue arrow.	114
5.7		A mosaic of 550 frames showing a second track along the ship's hull (track motion is in negative y direction). The sonar sensor trajectory is shown in red and the attitude at every 30 th frame is shown as a blue arrow.	114
5.8	(a), (b) and (c)	show the coarse forward, backward and complementing forward/backward registrations, respectively. (d) shows the reference image and (e) shows the reference image with the positions of the forward/backward registrations overlaid. It can be seen that the unreliable results in the shadow region is have been rejected and therefore are not part of the platform motion estimation.	115
	(a)	115
	(b)	115
	(c)	115
	(d)	115

List of figures

(e)	115
5.9 The coordinate system for the dam inspection dataset.	116
5.10 A photograph showing the dam wall after the dam had been drained, copied from [33]. Exposed re-enforcing bar can be seen in the foreground.	117
5.11 A single frame from the dam inspection dataset, shown in Cartesian coordinates. Exposed re-enforcing bars are shown faintly on the right of the image. Also, a number of fish can be seen at the top of the image, these cast an acoustic shown over the scene.	117
5.12 Inter-frame registration weighting for the dam inspection dataset.	118
5.13 The result for the dam wall attitude-trajectory estimation. The sonar sensor trajectory is shown in red and the attitude at every 30 th frame is shown as a blue arrow.	119
5.14 The coordinate system for the pipeline and pile dataset	120
5.15 A single frame from the pipeline and pile dataset.	121
5.16 Inter-frame registration weighting for the pipeline and pile dataset.	121
5.17 The result for the sea floor attitude-trajectory estimation. The sonar sensor trajectory is shown in red and the attitude at every 30 th frame is shown as a blue arrow.	122
5.18 A blended image of the 1 st (green) and 398 th (magenta) frames.	123
A.1 The illuminated area from the vertical beam. From this area an estimate of the altitude (z) and pitch angle (Φ) of the sensor is made.	139
A.2 Position of the FLS field of view and the seabed. r is the range, Ψ is the field of view, Φ the elevation and θ is the rotation around the z axis.	140
A.3 A mosaic of 486 frames showing a debris field. The sonar sensor trajectory is shown in red and the attitude at every 30 th frame is shown as a blue arrow.	141
A.4 The smoothed attitude-trajectory of 486 frames over the debris field.	141
B.1 An illustration of the multipath effect through refractive (top figure) and reflective (bottom two figures) paths, the speed of sound profile is also shown.	146
B.2 Examples of the source of the Doppler effect.	146
B.3 A point source with a wave front. The rays that represent it are discrete in the launch angle and perpendicular to the wave front, which is the direction of motion [121].	147
B.4 Ray tracing beam shapes.	149
B.5 An example of a ray diagram for the Munk SSP generated from BELLHOP. The model is able to incorporate a surface and sea floor model.	150

B.6	An example of a transmission loss diagram for the Munk SSP generated from BELLHOP.	151
B.7	The Speed of Sound profile for the Munk environment.	151
B.8	The speed of sound profile for the SWellEx-96 environment. Where the attenuation is a linear function of distance and frequency.	152
B.9	An example of a ray diagram for the SWellEx-96 SSP. This is an example of the environment used in the experiments detailed in Section B.3.4. In the experiment a much wider set of launch angles were used.	152
B.10	An example of a transmission loss diagram for the SWellEx-96 SSP. This is an example of the environment used in the experiments detailed in Section B.3.4.	153
B.11	A block diagram of the UWA simulator [1] as a development on the system presented in [109]. The link between the delay compensation for the impulse response and the upshifting of the signal can be seen represented with $\hat{\tau}(nT_s)$. 155	
B.12	The received signal spectrum from the Waymark and VirTEX models with a flat SSP environment and a flat surface model [1].	158
B.13	The received signal spectrum from the Waymark and VirTEX models with a flat SSP environment and a sinusoidal surface model.	158
B.14	The received signal spectrum from the Waymark and VirTEX models with the SWellEx environment and trajectory with a flat surface model [1]. . . .	160
B.15	The received signal spectrum from the Waymark and VirTEX models with the SWellEx environment and trajectory with a sinusoidal surface model. . .	160
B.16	An example of a modelled 3-D environment using the k-Wave simulator [133]. The seabed is represented in magenta and the transducer array is shown in yellow. Note the rough surface of the seabed results in diffuse reflections.	161

Acknowledgements

I would like to thank my supervisor Dr. Yuriy Zakharov for his guidance, patience and support throughout the PhD. I would also like to thank my second supervisor Dr David Halliday for his support and the members of the Communication Technologies Research Group and in particular Jianghui Li.

I would like to thank Prof Yvan Petillot, School of EPS, Heriot-Watt University and Dr. Natàlia Hurtós, Girona Underwater Vision and Robotics, University of Girona, for providing the ship's hull inspection dataset.

I would also like to thank Dr. Luis A. Conti, University of São Paulo and Acquest Subaquatic Geology and Geophysics for providing the dam inspection dataset.

Finally, I would like to thank Jeff Chambers, Trittech International Ltd. for providing the pipeline and pile dataset.

To Rosie, Liz and Jon.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Publications

Publications directly associated with thesis work

- B. Henson, J. Li, Y. V. Zakharov, and C. Liu, “Waymark baseband underwater acoustic propagation model,” in *Underwater Communications and Networking (UComms)*. IEEE, 2014, pp. 1–5
- B. Henson and Y. V. Zakharov, “Adaptive filter based image registration,” in *European Modelling Symposium (EMS)*, Madrid, Spain, Oct., 2015
- B. Henson and Y. Zakharov, “Local optical-flow estimation for forward looking imaging sonar data,” in *MTS/IEEE OCEANS-Monterey*, 2016, pp. 1–8
- B. Henson and Y. Zakharov, “Estimating attitude and trajectory of forward looking imaging sonar using inter-frame registration,” in *International Conference and Exhibition on Underwater Acoustics (UACE)*, Skiathos, Greece, 2017
- B. Henson and Y. Zakharov, “Attitude-trajectory estimation for forward looking sonar based on acoustic image registration,” *IEEE Journal of Oceanic Engineering*, First submission May 2017, second submission Dec. 2017

Other publications

- L. Liao, Y. Zakharov, and B. Henson, “Grid waymark baseband underwater acoustic transmission model,” in *International Conference and Exhibition on Underwater Acoustics (UACE)*, Skiathos, Greece, 2017

Declaration

- J. Li, Y. V. Zakharov, and B. Henson, “Multibranch autocorrelation method for doppler estimation in underwater acoustic channels,” *IEEE Journal of Oceanic Engineering*, 2017

Benjamin Thomas Henson
August 2017

Acronyms

2D Two dimensions

3D Three dimensions

AE Angular Error

AUV Autonomous Underwater Vehicle

DCD Dichotomous Coordinate Decent

DFT Discrete Fourier Transform

DM Displacement Map

DOF Degrees Of Freedom

EE Endpoint Error

ERLS Exponentially weighted Recursive Least Squares

FLS Forward Looking Sonar

GPS Global Positioning System

HAUV Hovering Autonomous Underwater Vehicle

IE Interpolation Error

LASSO Least Absolute Shrinkage and Selection Operator

LMS Least Mean Squares

LS Least Squares

NCC Normalised Cross Correlation

Acronyms

OMP Orthogonal Matching Pursuit

PDF Probability Density Function

RLS Recursive Least Squares

SFC Space Filling Curve

SNR Signal to Noise Ratio

SSP sound speed profile

UWA Underwater Acoustic

Chapter 1

Introduction

The desire to explore and exploit the oceans, combined with a need to reduce the risk to humans, has led to the use of robotics underwater. A fundamental requirement for a robotic system, whether autonomous or piloted, is the need to navigate. While there are many technologies to assist in navigation for land or air based vehicles, these technologies do not necessarily transfer well to the underwater environment. Global Positioning System (GPS) may be referenced using underwater acoustic beacons [8], however a system may not always be available, or the accuracy good enough, for tasks such as exploration [9] or underwater inspection [10, 11]. Although inertial navigation systems are unaffected by being underwater, they are costly and subject to drift [12]. Systems that have sensors that detect and interpret the scene around the underwater platform have the advantage in that they calculate the position using fixed points in the environment. This is the basis of visual odometry [13]. Visual odometry that is based on the estimated motion between frames is subject to drift from the accumulated error in the estimation. If revisited areas or features can be identified the error in the estimate of a motion path can be bound by closing a loop in the motion path [14, 9]. Optical vision systems have been effectively used for visual odometry on land [15], in air [16] and underwater [17, 18]. However, underwater optical systems may suffer from poor visibility [19].

A practical alternative is acoustic imaging in place of an optical system [10, 14, 20]. With the advent of higher resolution forward looking sonars (FLSs) the use of optical processing techniques become more applicable. A fundamental part of a visual odometry system is examination of motion between images [21]. Image registration is the geometrical alignment of two images that are of similar subjects but differ due to being taken under different circumstances, either different times, viewpoints or sensor technologies [22]. The registration process is the evaluation of the change between images, with movement inferred from it. Image registration is useful in many application areas, for instance in image enhancement,

where the signal to noise ratio is improved by combining several images of the same scene [23, 14, 24]. Alternatively, a large mosaic can be created from multiple images to show a wider area, also with potentially a higher resolution [25, 23, 26]. Combining multiple images where the motion is known to a sub-pixel accuracy can increase the resolution of the combined image, this is termed super resolution [24, 27, 28].

Another application area where registration is used is to highlight the difference between two images taken of the same scene. The registration would co-locate the images, then a further comparison would emphasise any differences. One example is in detecting the change in land usage over time, where a satellite image might be taken of an area periodically [23]. Another example would be in the medical field where the effectiveness of treatment is assessed by examining physical changes that might be imaged, for instance, the growth or remittance of a tumour [29].

Image registration can also be used to automatically extract more abstract information about the scene. For instance, in imaging for robotic applications the difference might be measured to give a representation of the movement of a scene and from this, movement of the robot may be inferred [10, 17, 30].

The aim of the work in this thesis is to propose an algorithm that is able to estimate the motion of a sonar platform from inter-frame registration. Image registration is used to detect motion in the scene, and, when the platform is moving, build up a map that can be used by the robot for navigation [13, 14, 20, 31].

1.1 Introduction to underwater acoustic imaging

In underwater exploration sound has been widely used for communication and surveying, with more sophisticated techniques being employed as a result of advances in computing.

It is the surveying and imaging aspect that is the subject of this thesis, more specifically trying to improve the quality, and extract position information from an image obtained by a FLS. Fig. 1.1 shows a diagram with an illustration of the FLS field of view and an example FLS image can be seen in Fig. 1.2. The field of view is sampled in range (r) and in the azimuth angle (Ψ), however, the point of reflection in the vertical beam is ambiguous and therefore the image is a flat projection onto the vertical centre line.

The registration problem is the detection and measurement of motion in the field of view. The results of these measurements can be used in a number of ways. The images can be enhanced directly by blending co-aligned sections of the images; this improves the signal to noise ratio (SNR). Also the resolution can be improved if the accuracy of the motion estimation allows the pixels to be plotted onto a finer spacial grid. Another application is to

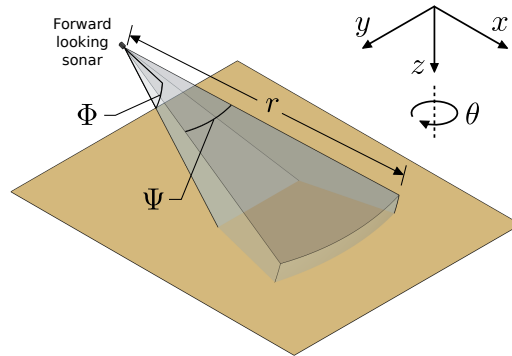


Fig. 1.1 An illustration of a typical field of view of a forward looking sonar.

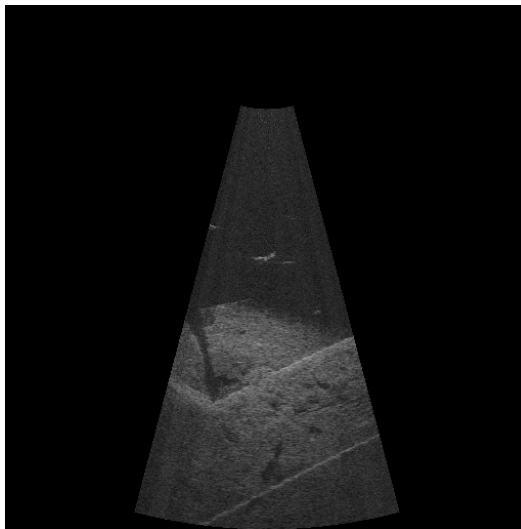


Fig. 1.2 An example of a single forward looking sonar frame. Original data from [32, 33].

consider the motion between images as a whole to estimate the motion of the sensor itself. Often the sensor is a part of a mobile platform, such as an underwater robot, and so an estimation of the motion can be incorporated into a navigation and mapping system.

In order to derive the movement of the sensor from the motion in the two dimensional image, a number of assumptions need to be made about its position relative to the scene and the propagation of the acoustic wave through the water. The approach taken to estimate the sensor attitude and position is to model how the scene would look after a range of possible motions. These modelled motions are then compared with what is observed from the sonar images. The quality of the attitude-trajectory estimation not only depends on the quality of the registration but also on the quality of the model. The movement modelling can be improved with a more accurate model of the propagation in the environment.

The physics associated with the propagation of sound through water may be well understood in theory, however, making useful tools from that theory is still a subject for

development. Advances in computer technology have meant that simulation in a reasonable time and to a useful accuracy may be performed. The results created are useful in direct applications such as channel estimation which describes the distortion of the acoustic signal with time. Fig. 1.3 shows an outline of the research topics and how they relate to one another.

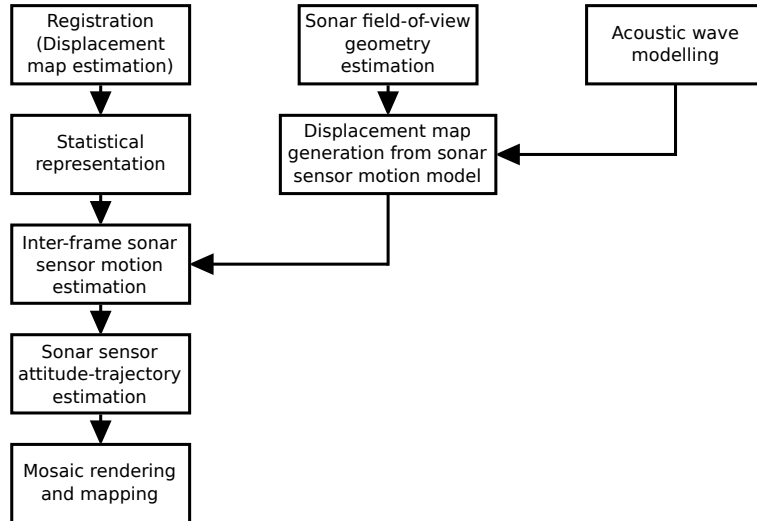


Fig. 1.3 A block diagram showing how the research topics relate to each other.

1.2 Algorithm development

The research for the measurement of the motion within a scene was started by examining the problem first in the context of optical images. This gave an insight into many of the problems associated with image registration; such as multiple motions, revealed and occluded pixels and variation in image intensity [34]. An adaptive filter based registration algorithm has been developed to give an evaluation using an optical dataset. The results from this development highlighted successful techniques and areas for improvement. From this the proposed algorithm has been developed further to address the more specific problems associated with FLS images. This later algorithm has been tested on a range of datasets showing some promising results for both registration of individual elements within the image, and the estimation of the sensor motion through the generation of whole frame DMs.

For underwater acoustic channel modelling a baseband Waymark model has been developed. Using the results of a complex amplitude and an arrival time from the BELLHOP acoustic wave propagation model [35], the algorithm uses a novel interpolation technique to generate a set of impulse responses and delays for the environment through time. From these environmental parameters the distortion on the signal is modelled and applied to a baseband

frequency range, resulting in more efficient processing of the signal. The implementation has also been modified to allow the calculation to be performed as a continuous process. This greatly improved the memory requirements and meant that the model could be run for very long signal simulations.

1.3 Contributions

In this work a method has been developed that is able to take a sequence of frames from a FLS and estimate the attitude-trajectory of the sonar platform. The method is based on the estimation of the inter-frame pixel DMs. These DMs are then compared with a set of maps generated from a motion model. A more accurate motion model may be obtained from the improved simulation of underwater acoustics. To that end further work has been done on the efficient modelling of an underwater channel.

More specifically the contributions are:

1. Inter-frame DM estimator - An algorithm has been proposed based on an adaptive filter to estimate the inter-frame motion of each pixel in the FLS frame. The proposed algorithm is improved with the inclusion of a coarse pixel displacement estimator that employs a sparse recovery technique. This two step process of a coarse and fine estimation allows a DM to be estimated for the whole image to a sub-pixel accuracy whilst accommodating large motions. Due to the flexibility of this approach the pixel motion estimation can be made in the native polar format of the FLS data, removing the need for interpolation to a Cartesian coordinate system. Another advantage of generating a full frame DM is that complicated motions for objects within the scene can be represented. This development work is detailed in Chapters 2 and 3.
2. Attitude-trajectory estimator for the sonar sensor - An efficient method for the comparison of estimated and modelled DMs is proposed, through reducing the DMs dimensions using order statistics, and the application of a dichotomous coordinate descent algorithm. This work is detailed in Sections 4.2 to 4.4. Regularised splines have been exploited to compile estimated inter-frame sensor motion into a smoothed attitude-trajectory for the sensor. Regularised splines also allow additional factors to be incorporated into the motion estimation such as the quality of the registration and the dynamics of the underwater platform. This work is detailed in Section 4.5.
3. Validation on in field datasets - The proposed algorithm has been validated through the application to in field datasets that show a range of image types. The results from the attitude-trajectory estimation using in field datasets are shown in Chapter 5.

4. Underwater channel model - A faster variant of the Waymark underwater channel model has been developed to allow processing of a signal in the baseband and thus take advantage of the computational savings. The improved underwater channel model is presented in Appendix B. Improvement in underwater channel modelling can be used to improve the accuracy of the modelled DMs.

1.4 Thesis structure

The remainder of this thesis is structured as follows:

- Chapter 2 discusses the problems involved with image registration and the estimation of a DM. The chapter shows the development of an initial image registration algorithm and the assessment of it using a widely used optical registration benchmark dataset.
- Chapter 3 reviews current image registration techniques and then goes on to detail the final algorithm developed for FLS images. This algorithm takes a pair of FLS frames and produces a DM which is an estimate of the motion of each pixel between frames.
- The interpretation of the DM is described in Chapter 4. First the generation of a set of DM derived from modelled sonar sensor motions is detailed, including the assumptions made about the environment and the sonar. Next, the comparison between the estimated DM and the modelled DMs is described, this is the interpretation of the estimated DM as a motion for the sonar sensor. Finally, the method for using regularised splines to convert the sequence of sensor motions into a smooth attitude-trajectory is described.
- Chapter 5 shows the results of applying the DM estimation algorithm to the original optical dataset. In addition, the DM and attitude-trajectory estimation algorithms are also applied to three in field FLS datasets.
- Chapter 6 has the conclusion for the thesis and details further research areas. In
- Appendix A the Debris dataset is presented with the addition of pitch estimation for the sonar platform.
- The work done on the underwater channel model is described in Appendix B. Results are shown for a set of test scenarios based on a underwater communications channel. Also discussed in this chapter is the way that the channel model may be used to enhance sonar image registration.

Chapter 2

Registration Algorithm Development

In this chapter the problem of image registration is examined. In Section 2.1 there is a review of a selection of the literature on the subject with an emphasis on what algorithms might be suitable for application to FLS data. Section 2.2 has a more detailed discussion on some of the more general problems associated with image registration and the more specific problems using FLS images. This is followed in Section 2.3 by the development of a fine (sub-pixel) registration algorithm based on an adaptive filter. Section 2.3.5 has the application of the proposed algorithm to a selection of optical images from a widely used benchmark dataset. The results are evaluated with a view to improving the algorithm for use with FLS images. The chapter concludes in Section 2.4 with a summary of what has been learnt, what parts of the algorithm are to be used further and where the areas for improvement are.

Part of the work in this chapter is presented in the conference paper: B. Henson and Y. V. Zakharov, “Adaptive filter based image registration,” in *European Modelling Symposium (EMS)*, Madrid, Spain, Oct., 2015.

2.1 Introduction

Image registration is the geometric alignment of two images that are of a similar scene but differ due to being taken under different circumstances; either different times, viewpoints or sensor technologies [22]. The registration process is the evaluation of the change between images, with movement inferred from it. This ability to detect and measure motion is useful in many application areas, because of this the subject has received a great deal of attention as a topic of research [36, 28, 22].

Image registration, either optical if sonar image, can be categorised into two main branches; feature based registration and area matching registration. Feature based registration is based on identifying the same point in the reference and target image through a distinctive

feature. The recognition of the feature needs to be robust against the expected transformation that might occur between the two images. Many types of features are used such as corners [37] and edges [38] which are relatively simple to extract but make the matching process more difficult. Algorithms that provide a sophisticated identification of points have been designed, for examples Scale Invariant Feature Transform (SIFT) [39] and Speeded Up Robust Features (SURF) [40]. These provide better discrimination over a larger range of reference to target image transforms. Area matching is the search for a similarity based on an area of pixels. There are also many methods to assess the similarity between the pixel patches such as mutual information [41], sum of absolute difference [42], normalised cross-correlation [43] and phase correlation [23]. More recently there has been work to use different techniques to address different aspects of the problem such as large movement or the detection of discontinuities [44–46]. A good example is the work in [44] where the registration of the whole image was treated as an optimisation problem.

The remainder of this discussion will be focused upon techniques that may be more suited to the registration of FLS images.

FLS images typically have a number of specific differences when compared with a optical images [47]. The most fundamental difference is in the way that the image is constructed. For optical images the light is focused on a 2-D sensor allowing the incident angle to be ascertained from where it falls in the array. There is, however, an ambiguity in depth (or range) in the image. This is because typically the source of the illumination is unknown, therefore making a calculation of the depth in the scene either through inferences based on the geometry or time-of-flight calculations is difficult. FLSs by contrast control the illumination (insonification) source and therefore can measure through the time-of-flight the range to any reflector. However, processing a 2-D sensor array is very computationally intensive and therefore typically FLSs only discern the azimuth angle for a sound return (Ψ , Fig. 2.1). This means that there is an ambiguity in the elevation (Φ , Fig. 2.1) and all of the sonar returns appear as a projection on to the centre of the vertical beam. The vertical beam is often relatively wide in order to illuminate a large field of view, examples range from 14° to 20° [48, 49]. This projection is illustrated in Fig. 2.2.

Another problem arises from the way that the image is formed from the returns over a fixed number of radial beams. This means that the ‘pixel’ is non-homogenous, that is, the size increases further away from the sensor, as shown in Fig. 2.3. This has implications in the way that the data is registered. The data could initially be interpolated onto a Cartesian grid but this introduces an interpolation error at the beginning of the processing pipeline, which is undesirable. Alternatively, the image can be processed in the native polar format. This is also a consideration when trying to determine the movement of the sensor from the movement

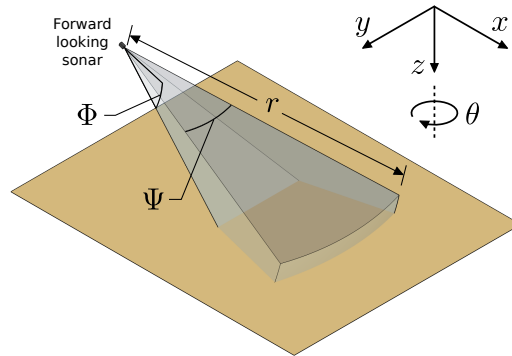


Fig. 2.1 An illustration of a typical field of view of a forward looking sonar.

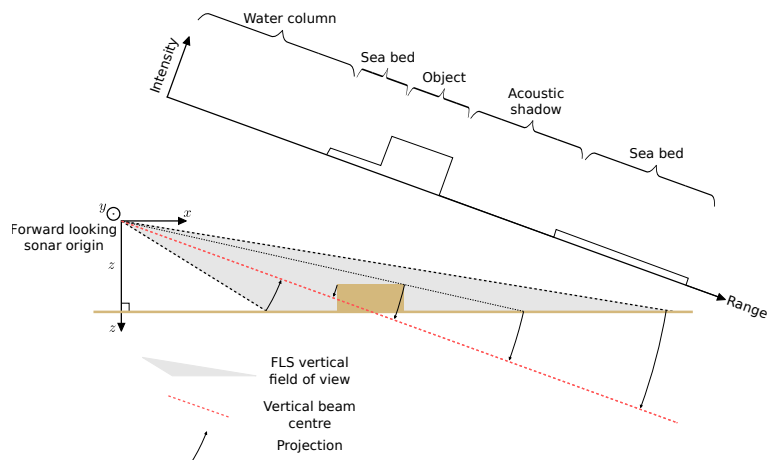


Fig. 2.2 An illustration of the projection of the reflections onto the beam centre caused by a single wide vertical beam. Also shown is how the single point illumination casts shadows in the image.

field in the image; considering range and bearing give a more complicated movement field for translation, albeit a simpler movement field for rotation. A distortion in the image might also be caused by rapid movement of the sonar. If the capture time is large then the motion of the sonar can cause the perceived position of the returns to change. This leads to movement distortions in the image as the range increases and in extreme cases large discontinuities.

Acoustic images usually have a low resolution; for instance, a DIDSON 300 sonar frame might be of 96×512 'pixels' [48], whereas, an inexpensive optical sensor can produce an image of much higher resolution, for example, 3280×2464 pixels (Sony IMX219PQ [50]).

FLS images have a lower SNR. The noise that is apparent in FLS images has two aspects to it. There is the noise that can be seen where there are no returns, for instance in open water or in an acoustic shadow. This noise comes from the electrical noise in the system and can be statistically modelled using a Gaussian curve [51]. The second source is where there is a return from an object or the seabed, this noise is created where the texture of the reflecting

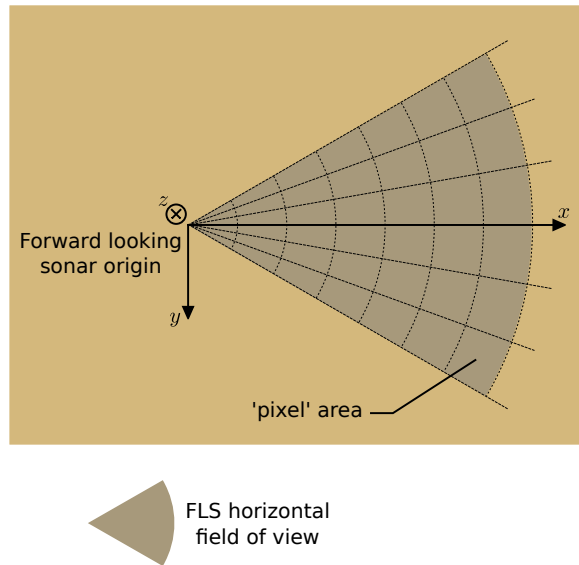


Fig. 2.3 The radial beam of the FLS with the area represented by each pixel increasing with range.

surface causes self interference on the returning wave front. This is termed speckle noise and is modelled with a Rayleigh curve [51].

There may be fast signal intensity changes due to variation to the sensor attitude [47]. There is a single point illumination, this means that pixels can disappear into, or appear from shadows rapidly.

For many state of the art optical processing techniques a starting point is an initial segmentation of the image [52]. In a real world scene this would be trying to associate particular areas of the image with objects. If corresponding objects can be found in the reference and target images, then the motion can be estimated. Depending on the application, the motion of the object might be considered separately or as a whole if the sensor motion is of interest.

FLS images typically have low signal to noise ratios (SNRs), and with a poor SNR, the segment boundaries are difficult to identify and track between frames. Indeed, this difficulty is often the focus of the processing for medical ultrasound imagery [53], which shares many of the problems with FLSs. However, in medical ultrasound often an idea of the structure in the image is known in advance, for example if the heart was being observed then the chamber structure can be expected [29, 54].

As with optical image registration previous work on FLS registration can be categorised into two main branches; feature based registration and area matching registration. Feature based registration where the aim is to identify distinct features that tracked between frames [10, 20, 31, 55–58, 40, 59, 60]. For example, for FLS images these can be sharp transition in

the image intensity where an acoustic shadow occurs [10]. More sophisticated registration can be based on modelling of the geometry of features [61, 62]. For FLS this is supported by segmentation specifically looking for high intensities followed by acoustic shadows [20]. Feature descriptors such as SIFT have been used with side-scan sonar images [58] and has shown potential in reducing the complexity of a registration.

Area based methods search for a similarity based on an area of pixels, this can be the whole image as in [14] or for smaller patches of the images as [25]. Another consideration is the type of registration transform that is the target for the registration, this can be a proper ridged transform applied to the whole image as in [14] or a dense DM that is used to enhance the appearance of each pixel [29].

One example algorithm presented in [14] was developed specifically for FLSs. The image was processed in the frequency domain using a phase correlation technique and optimised for a proper ridged transform between the images. A specialised spectral filtering algorithm was developed that made the registration more robust to the poor SNR associated with sonar imagery.

Another example uses a hybrid correlation and mutual information similarity measure to match parts of side-scan sonar images [25]. This retains the generality of low level patch matching with the flexibility to accommodate a wide range of motion models (such as local scaling due to changes in the water column). To represent the motion a pixel displacement map (DM) is generated for the whole image. It is this style of registration that is the focus of the research in this thesis.

One influential idea in image motion estimation is that of optical flow, the brightness change associated with a movement in the scene [63]. Horn and Schunck developed the optical flow constant equation which tracks a zero brightness gradient in time and space [64]:

$$E_x u + E_y v + E_t = \mathbf{0}, \quad (2.1)$$

where $u = dx/dt$, $v = dy/dt$, and E_x, E_y, E_t are partial derivatives of the image intensity with respect to the spatial dimensions x and y and time t , respectively, these values are measured from the reference to the target image. The optical flow constraint equation has a single constraint, that is the gradient of the intensity between the reference and the target is zero. This is to be solved for the motion in x , u and the motion in y , v . This means that the problem is ill-posed and needs additional smoothness constraint and regularisation [64, 34]. Lucas and Kanade [65] also sought to minimize an energy function where the pixel movement is assumed to be shown by a continuity in intensity. Optical flow is useful because it can be used to describe the motion of individual pixels. This is more versatile when working with different motion models.

The optical flow between two images may be represented as a convolution kernel [66, 67],

$$i_t(\xi) = \sum_{m=1}^M h(\xi_m, \xi) i_r(\xi - \xi_m) + \varepsilon(\xi), \quad (2.2)$$

where $h(\xi_m, \xi)$ is an unknown kernel at pixel location ξ , ξ_m is a displacement and $\varepsilon(\xi)$ a measurement noise. The displacements ξ_m are limited to a support X_r , $\xi_m \in X_r$, e.g. X_r is a rectangular window around the origin.

Elad and Feuer [68] later developed the recovery of optical flow as an adaptive filter problem. An adaptive filter is a good candidate for use on the optical flow problem because there are efficient methods of implementation. The optical flow was estimated with the evolution of the filter through a sequence of images of the scene over time. Elad and Feuer also used an adaptive filter approach for a superresolution restoration application [69]. In these systems, the adaptive filter encompassed the whole image area and the evolution was over an image sequence in time. The technique of using an adaptive filter to spatially filter the image was presented in [70], who used a least mean squares (LMS) adaptive filter to correct the movement between a pair of images. They proposed using a peak extraction process to estimate the displacement vector from the derived filter coefficients. The derived displacement vector was then used to interpolate a corrected image. The correction was refined by iterating over these steps. Recognising that the LMS adaptive algorithm showed poorer performance when the input data has a d.c. bias, and that this bias was inherent in imaging data, Lin, Nie and Unbehauen [71] sought to develop a local-mean estimator that could be used to dynamically compensate for the background brightness in the picture. An adaptive spatial filter was also used by Caner et al. [66]. They used a preprocessing step to estimate the integer pixel movement which was then used to centre the adaptive filter in its support. The central four filter coefficients were then used for bi-linear interpolation of the sub-pixel movement. More sophisticated estimations such as fitting a quadratic surface to the points, are described by Brauers et al. [72] and Bailey [73]. In order to try to evolve the filter over the smoothest path, Caner et al. [66] used a Hilbert space filling curve to cluster the local pixel movements and hence improve the performance of the adaptation processing.

A popular way of dealing with large movement is the use of search techniques to match up small sample areas or patches from the images to be registered [74, 75]. An additional tool that is used to find more information about the movement discontinuities in the image, is incorporating forward-backward registration; that is, registration from a reference image to a target image and from the target image back to the reference image. Pixel occlusion will only happen in one direction and so can be detected. This detection can be incorporated into

a single registration [44, 45], or may be used as a post processing comparison step that can be used as an extension to other registration techniques [76].

2.2 The registration problem

The following discussion is to illustrate some of the problems with registration for different image types and movements. In real applications there may be many objects that are subject to different relative movements. A wide range of movements need to be accommodated, from small continuous movement to large discrete blocks of movement with discontinuities between, see Fig 2.4. The independent movement within a scene might shadow or reveal pixels, which must either be ignored or recognised as missing. The revealed pixels cause less difficulties because they should be rejected by any matching process, however, occluded pixels are more difficult because the target equivalent is not present.



(a) Different parts of the scene move independently (example image from [77]).



(b) Moving viewpoint causes parallax effects (flower garden dataset [78]).

Fig. 2.4 Illustrations of relative movements within a scene.

Another added difficulty in image registration for real applications is local deformation where each object could change shape between images. A good example of this is in medical imagery, where, due to the patients movement and the non-rigid nature of soft tissue large local deformation might occur. This problem is also apparent in underwater acoustic imaging, where disturbances in the water column would cause local changes to the sound propagation

and therefore distort the image perceived at the receiver. This presents difficulties because each of the pixel movements need to be identified on a local scale.

2.2.1 Simple motion models

One simple motion model is the proper rigid transform which is a translation and rotation as applied to the whole image, as illustrated in Fig. 2.5. A simple example would be a single camera subject to small movement but looking at a scene in the distance. A more general

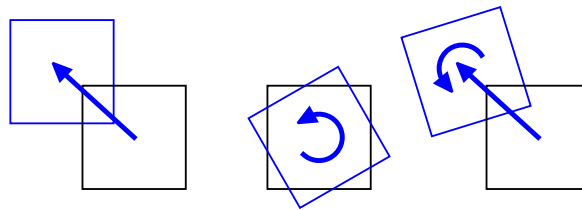


Fig. 2.5 Proper rigid transform.

transform that is able to model more complex movements is the affine transform, illustrated in Fig. 2.6. This transform can represent general movement of the viewpoint including perspective effects.

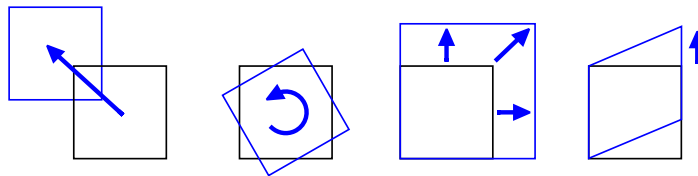


Fig. 2.6 Affine transform.

In this thesis the optical flow is represented as a displacement vector from each pixel at the reference image to the new position in the target image. For illustration purposes a display style developed for the Middlebury dataset [77] has been used. The displacement map is shown as a colour map with the hue representing the direction of movement and the colour saturation representing the magnitude, an example is shown in Fig 2.7. A selection from the dataset is shown in Figs. 2.15. The Middlebury dataset [77] is commonly used for the evaluation of registration algorithms and it shows many of the aspects of the problems discussed. The Middlebury dataset contains computer generated images, however the projection is a simulation of a camera viewpoint and therefore for the purposes of this thesis they are considered to be ‘optical’.

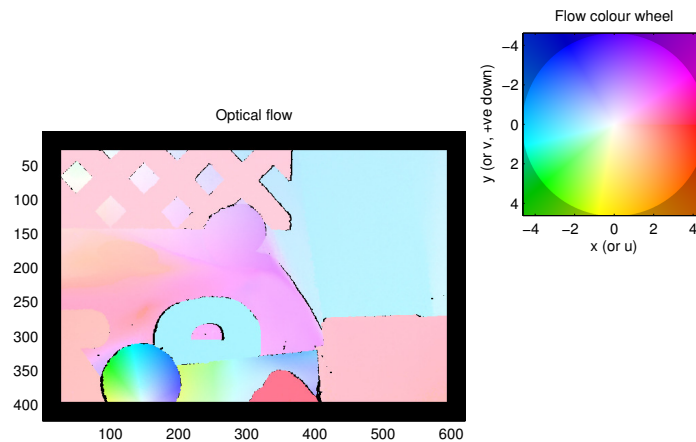


Fig. 2.7 A colour wheel to show the movement; hue shows the direction and the saturation represents the magnitude.

2.2.2 Strategy for research

The aim and contribution of this work is to research an algorithm that is able to identify motion between frames without explicitly identifying features. Also, the aim is to identify motion within a frame on a pixel-by-pixel basis. This allows greater flexibility for matching of a motion model. This puts the work as a midpoint between algorithms that detect features and allow for a very flexible motion model such as described in [79] and a full image registration algorithm that is restricted to a simpler model for the motion [14]. In terms of processing, the aim is to make a high quality inter-frame registration and therefore perform a sub-pixel displacement estimate for each pixel.

This work on the registration uses a publicly available dataset described in [77]. This dataset provides a set of image pairs that highlight many of the problems associated with image registration, e.g. revealed and occluded pixels, multiple independent movement, etc.. In addition the datasets are provided with a ground truth for the inter-frame displacement map (optical flow), this means a quantitative assessment is possible.

2.3 Adaptive filter registration

2.3.1 Adaptive filter overview

Adaptive filters have been widely used in signal processing applications for system identification [80], and so are a good candidate technique to apply to the optical flow problem [70, 68, 71, 66, 3]. The convolution kernel identified can describe different types of motion through the surface created by the coefficients. The kernel support can also be small enough so that it describes local changes giving good definition to the DM. Here an adaptive filter is used to identify the convolution kernel, related to the sonar sensor motion, to a sub-pixel precision. As with the work in [70] the final pixel displacement estimate is extracted from the estimated kernel by means of a peak extraction. This uses the assumption that there is a single movement associated with each pixel. This assumption is also used for the algorithm developed in this thesis.

Adaptive filter convolution kernel estimation

The adaptive filter is used to estimate the convolution kernel between the reference image and the target image, within a small area as identified by the aperture of the filter. It is by examining the distribution of the coefficients of the convolution kernel that an estimate of the pixel motion is made. Fig. 2.8 illustrates the adaptive filter configuration.

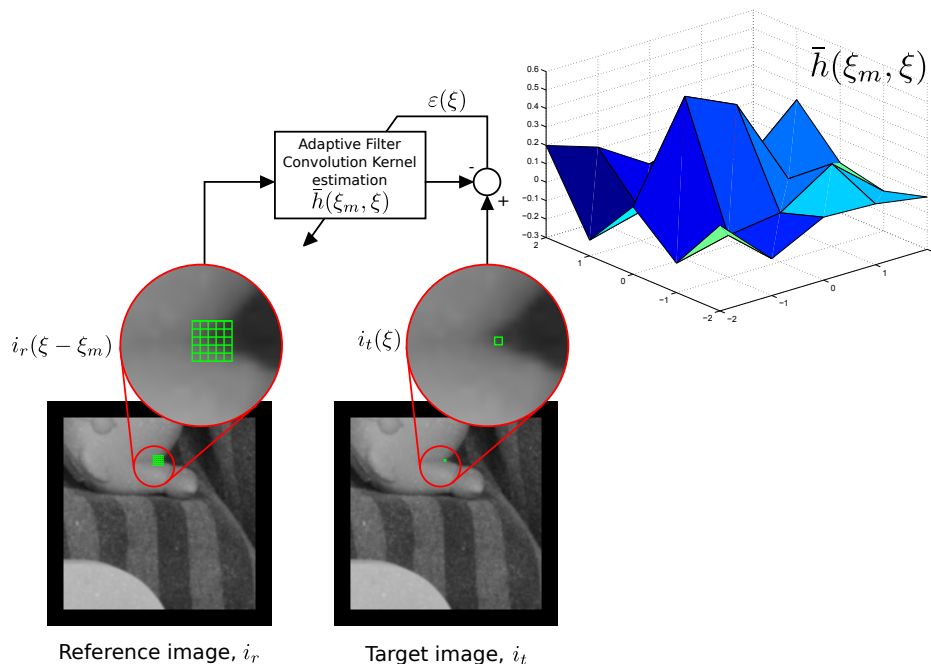


Fig. 2.8 An adaptive filter used to identify the convolution kernel between two images.

The known value of a pixel in the target image $i_t(\xi)$ is assumed to be the linear combination of a small aperture of known pixels from the reference image $i_r(\xi - \xi_m)$, centred around the same pixel location ξ . The coefficients of the combination are to be estimated giving the convolution kernel $\bar{\mathbf{h}}(\xi_m, \xi)$.

$$i_t(\xi) = \bar{\mathbf{h}}^T(\xi)\boldsymbol{\gamma} + e(\xi), \quad (2.3)$$

where the reference frame in the vicinity X_r of pixel ξ , where $\{\xi_m \in X_r\}$, is treated as a measurement vector (vect) $\boldsymbol{\gamma}$:

$$\boldsymbol{\gamma} = \text{vect}\{i_r(\xi - \xi_m), \xi_m \in X_r\}, \quad (2.4)$$

and $e(\xi)$ is the estimation error in the representation of the target pixel. More explicitly:

$$e(\xi) = i_t(\xi) - \bar{\mathbf{h}}^T(\xi)\boldsymbol{\gamma}, \quad (2.5)$$

The coefficients that minimise the sum of the square of the error is taken to be the estimate of the convolution kernel. The least squares problem is represented as minimising error cost function at location ξ [81],

$$J(\xi) = \sum_{k=0}^{\xi} e^2(k), \quad (2.6)$$

The minimum value of $J(\xi)$ is found with the optimised convolution kernel $\hat{\mathbf{h}}$,

When the filter aperture moves around the image then the input and desired output changes. A matrix (Γ) is defined with each of the aperture snapshots combined.

$$\Gamma = \begin{bmatrix} \boldsymbol{\gamma}_1 & \boldsymbol{\gamma}_2 & \dots & \boldsymbol{\gamma}_{\xi_m} \end{bmatrix}. \quad (2.7)$$

An equivalent vector can also be defined for the target pixels \mathbf{i}_t ,

$$\mathbf{i}_t = [i_t(1) \quad i_t(2) \dots i_t(\xi)], \quad (2.8)$$

and the error \mathbf{e}

$$\mathbf{e} = [e(1) \quad e(2) \dots e(\xi)]. \quad (2.9)$$

The cost function can now be defined in terms of the multiple aperture matrix an vectors defined above,

$$J = \mathbf{e}^T \mathbf{e} = (\mathbf{i}_t - \Gamma \bar{\mathbf{h}})^T (\mathbf{i}_t - \Gamma \bar{\mathbf{h}}), \quad (2.10)$$

where $(\mathbf{i}_t - \mathbf{\Gamma}\bar{\mathbf{h}})$ is the multiple aperture equivalent of (2.5). Expanding the terms and rearranging gives the cost function

$$J = \mathbf{i}_t^T \mathbf{i}_t - 2\mathbf{b}^T \bar{\mathbf{h}} + \bar{\mathbf{h}}^T \mathbf{R} \bar{\mathbf{h}}, \quad (2.11)$$

where

$$\mathbf{b} = \mathbf{\Gamma}^T \mathbf{i}_t, \quad (2.12)$$

and

$$\mathbf{R} = \mathbf{\Gamma}^T \mathbf{\Gamma}. \quad (2.13)$$

To find the optimised solution the lowest point in the error surface J is found. This is achieved by setting the gradient of J with respect to each dimension of $\bar{\mathbf{h}}$ to zero,

$$\mathbf{R}\bar{\mathbf{h}} = \mathbf{b}, \quad (2.14)$$

or as a solution for $\bar{\mathbf{h}}$,

$$\bar{\mathbf{h}} = \mathbf{R}^{-1} \mathbf{b}, \quad (2.15)$$

where \mathbf{R}^{-1} is the inverse matrix of \mathbf{R} .

The solution to the least-squares problem requires that all of the data from the input apertures and the target pixels be included in the optimum coefficient calculation. For an image of a useful size this makes the complexity high. A more efficient method is to calculate the convolution kernel coefficient for a new input reference aperture and target pixel based on the result of the previous calculation. This is the recursive least-squares algorithm (RLS). For the RLS algorithm we now redefine \mathbf{R} and \mathbf{b} to be recursive;

$$\mathbf{R}(\xi) = \lambda \mathbf{R}(\xi_{-1}) + \boldsymbol{\gamma}(\xi) \boldsymbol{\gamma}^T(\xi), \quad (2.16)$$

$$\mathbf{b}(\xi) = \lambda \mathbf{b}(\xi_{-1}) + i_t(\xi) \boldsymbol{\gamma}(\xi), \quad (2.17)$$

where λ is an additional weighting in the standard RLS algorithm that allows a balance to be made between the historic result and the adaptation new incoming data. This is termed the forgetting factor, where $\lambda \in (0, 1]$ giving the recursive relationship an exponential-weighting. Including λ in the cost function gives;

$$J(\xi) = \sum_{k=0}^{\xi} \lambda^{\xi-k} e^2(k). \quad (2.18)$$

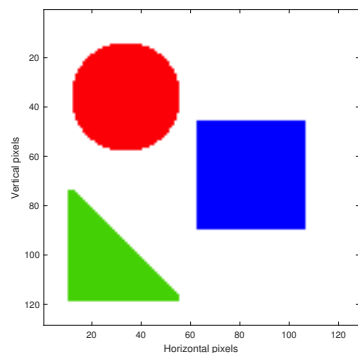
and the estimate for the exponentially-weighted recursive algorithm as;

$$\mathbf{R}(\xi)\hat{\mathbf{h}} = \mathbf{b}(\xi) \quad (\hat{\mathbf{h}} = \mathbf{R}(\xi)^{-1}\mathbf{b}(\xi)) \quad (2.19)$$

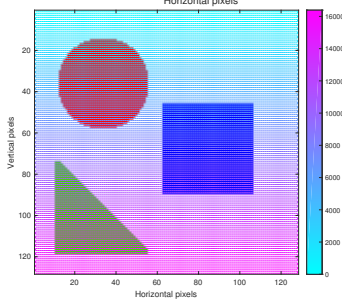
2.3.2 Smoothness and diversity

The adaptive filter works most effectively if the change in the convolution kernel from one adaptive iteration to the next is a slow evolution. With this aim, the order that the pixels are presented to the filter is chosen to try to group similar movements.

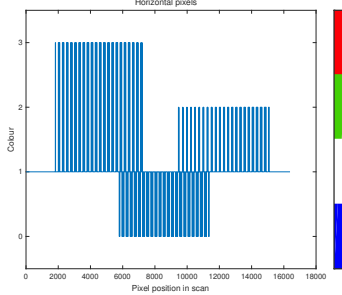
The aim of altering the pixel sequence that the filter moves through (the scan path) is to improve the input conditions for the adaptive filter. That is, the variation in the displacement vectors between samples should be as smooth as possible. The problem is that the content and change in the image is unknown, therefore diversity in the scan path can help the stability of the adaptation of the filter. A method for scanning a n-dimensional dataset with the aim of reducing the changes per scan step is to use a Space Filling Curve (SFC) [82], this is illustrated in Fig. 2.9. A Hilbert curve was used by Caner et al. [66] and is reported to have good results with regards to the adaptation in their filter design. This is because it clusters close locations in the same area along the scan path, reducing the number of transitions between regions of movement.



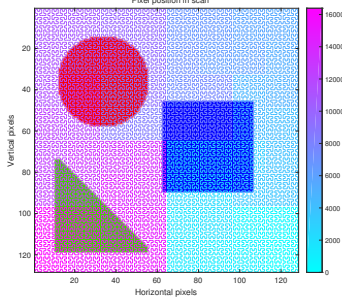
(a) Image showing three different objects which may have different motions associated with them.



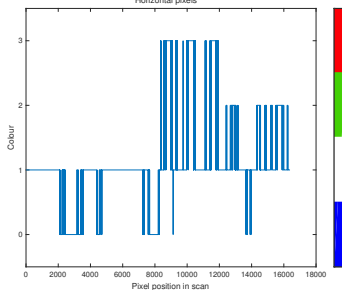
(b) Image with a zig-zag scan path superimposed.



(c) Plot of the colour transitions for the zig-zag path against the position in the scan path.



(d) Image with a Hilbert SFC scan path superimposed.



(e) Plot of the colour transitions for the Hilbert SFC path against the position in the scan path.

Fig. 2.9 A comparison of a zig-zag scan path with a Hilbert SFC scan path. It can be seen that the number of transitions between objects is greatly reduced with the SFC.

A recursive adaptive filter takes a number of iterations to evolve away from the start conditions to better represent the data. For a spatial filter, this means that a number of pixels at the beginning of the scan have a poor quality estimate of the convolution kernel. For a Hilbert SFC, where the beginning and endpoints are at different positions, it is problematic to remove the effects of the adaptive filter ‘start up’. A Moore curve [83] is a set of four Hilbert curves that are arranged so that the beginning and the end of the curve meet, making it a continuous loop, an example is shown in Fig. 2.10. The advantage of this is twofold, firstly it is easy to move the start/end point of the scan. The second advantage is that the adaptive filter is able to smoothly overrun the beginning of the scan path. In this way the portion of the output that was calculated with better convergence than when the filter was starting up may be recalculated without loss of continuity.

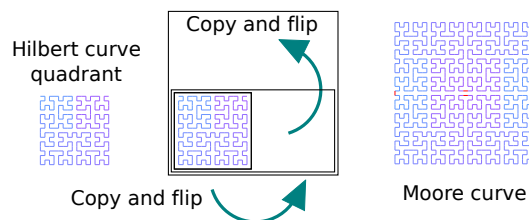


Fig. 2.10 An example of a Moore SFC. This is related to the Hilbert curve but is closed allowing for a continuous path.

A difficulty in the use of a Hilbert SFC (or a SFC derived from it) is the requirement that the image needs to be square with sides of an integer power of two [84]. A Moore curve is more restrictive requiring four smaller Hilbert curves. A system for generating a pseudo-Hilbert curve that could encode an area of arbitrary size is proposed in [84]. The same idea was used in this work, only the image area was first divided into quadrants, which were then divided into squares. A SFC was drawn in the square [84], and then joined up to give a variation in orientation.

Ultimately the evolution of the adaptive filter and this registration system as a whole depends on the content of the image to be filtered. For instance, large areas with little detail gives only a small amount of information for the filter to adapt to. When the filter then encounters a more textured area and is able to make an improved estimate of the kernel the displacement vector of the pixel could have changed significantly.

By changing the way that the image is scanned, diversity is introduced into the evolution of the filter. One way to achieve this is to scan the image with multiple paths. Each adaptive filter follows a scan path that is a pseudo-Moore SFC that will approach each pixel from 0° , 90° , 180° and 270° , respectively. In each of the scan paths the adaptive filter may have difficulties at different sections in the image, this changes the way that the adaptive filter

evolves thus altering the results. The aim is to select the best scan path available. To that end the scan with the smallest maximum displacement (magnitude) is selected, in order to remove extreme motion estimates.

2.3.3 Sparse promoting penalty

We have made the assumption that there is only a single displacement associated with each pixel. Also that for a smooth displacement map there will be a small number of similar motions represented in a given adaptive filter aperture size. Therefore, to represent those motions it is reasonable to assume that there will be a small number of coefficients in the estimated convolution kernel. Ideally, there would be a single coefficient but in practice there would be a small number representing a sub-pixel motion. However, where there is a discontinuity in the displacement map, for instance at an object boundary, then there may be two or more distinct motions represented in the adaptive filter aperture. In this case it is desirable to have the adaptive filter try to promote a smaller number of coefficients with the aim of identifying a single motion that is most dominant, and thus better define a boundary. The penalty term for the adaptive filter can be changed to promote this sparsity. The filter used is a Homotopy RLS-DCD adaptive filter as described in [85]; an RLS-like adaptive filter with a sparse promoting penalty term. With this filter there is not only a forgetting factor, which weights previous results and therefore how fast the filter adapts to change, but also the weighting of the sparse promoting penalty. The weight of this penalty is a trade off between having enough coefficients to represent sub-pixel motion and reducing them so that object boundaries are better defined.

The l_1 norm or LASSO penalty [86] is a common way of promoting sparsity in the calculated convolution kernel [85]. For registration, a single position in the adaptive filter input pixel grid is promoted as a solution and can therefore be used as the displacement estimate. From [85], and using the same notation as (2.3), the adaptive filter setup is given as follows: Assuming a linear system on a local scale, the output pixel,

$$i_t(\xi) = \bar{\mathbf{h}}^T(\xi)\boldsymbol{\gamma} + \varepsilon(\xi), \quad (2.20)$$

The aim of the adaptive filter is to minimize an error cost function that promotes sparsity with respect to \mathbf{h} ,

$$\min_{\mathbf{h}(\xi)} J[\mathbf{h}(\xi)], \quad (2.21)$$

where:

$$J[\mathbf{h}(\xi)] = \frac{1}{2}\mathbf{h}^T(\xi)\mathbf{R}(\xi)\mathbf{h}(\xi) - \mathbf{h}^T(\xi)\mathbf{b}(\xi) + \tau\mathbf{w}^T(\xi)|\mathbf{h}(\xi)|, \quad (2.22)$$

where $\mathbf{w}(\xi)$ is a set of N weights that are used to express additional information about the distribution of the sparse elements in $\bar{\mathbf{h}}$, τ is a regularisation parameter that controls the emphasis between the least squares and sparse optimisation. With the inclusion of a forgetting factor $\lambda \in (0, 1]$ the recursive relationship becomes an exponentially-weighted RLS problem as set out in Eq. 2.16 and Eq. 2.17. The forgetting factor (λ) is the weighting given to previous input data in relation to the current data [81]. The range of values for λ are 0.995 to $1 - (2/N)$ [87], where N is the filter aperture the forgetting factor is reduced at each iteration. Reducing λ in this way leads to a more stable evolution of the filter whilst still allowing faster changes in the impulse response to be found.

To illustrate the difference in the results from the least squares (l_2) only cost function and that of the least squares with the addition of a sparse promoting l_1 penalty. Fig. 2.11 shows a snapshot of the estimated filter coefficients for a single location in the Rubber Whale test image. The precise point is shown in Fig. 2.11a, the aperture for the filter covers the transition between two different motions, the background (striped cloth) and the foreground object (light coloured arch). Fig. 2.11b and Fig. 2.11c show the 3D and 2D plot, respectively, of the filter coefficients for the l_2 cost function. It can be seen that the coefficients represent both motions and two large peaks can be seen in the centre of the aperture. Fig. 2.11d and Fig. 2.11e showing the 3D and 2D plot, respectively, of the filter coefficients for the $l_2 + l_1$ cost function. Again two areas of motion can be seen, however, one is more dominant allowing a single motion to be identified with more confidence.

2.3.4 Evaluation methods for optical flow

The evaluation methods for the estimated displacement map are based on the error vector for the displacement and the error in the pixel intensity for a re-constructed reference image. These measures are defined in [34] for the Middlebury dataset .

The error vector is defined as the difference between the estimated pixel displacement and the ground truth pixel displacement. Over the whole image this produces an error displacement map. The endpoint error (EE) is the error vector magnitude:

$$EE = \sqrt{\left(d^{(1)} - d_{\text{gt}}^{(1)}\right)^2 + \left(d^{(2)} - d_{\text{gt}}^{(2)}\right)^2}, \quad (2.23)$$

where $d^{(j)}$ is the estimated displacement for the pixel with the coordinate j , d_{gt} is the ground truth displacement. The total endpoint error is the sum of the endpoint errors over the whole displacement map.

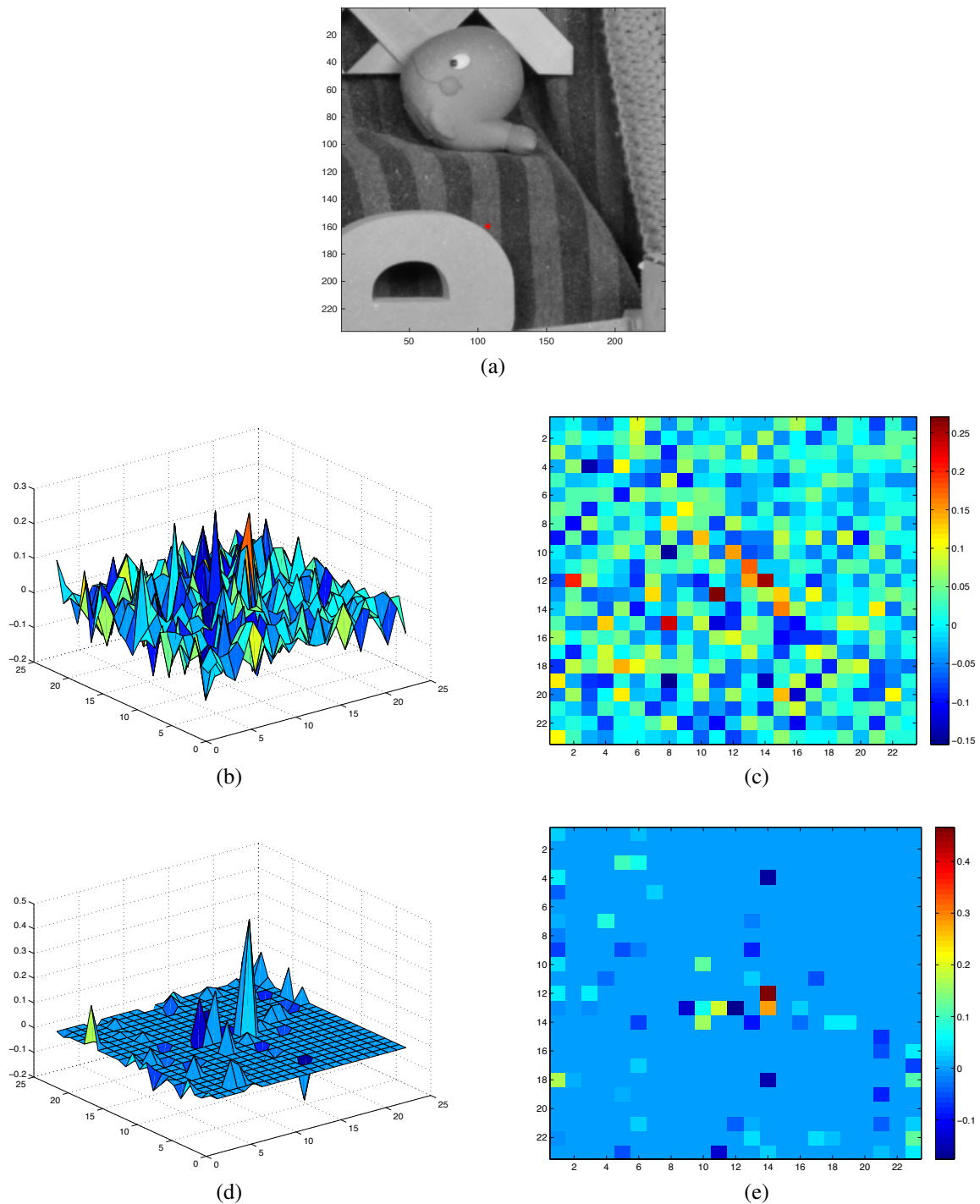


Fig. 2.11 (a) location of the example filter, note that at this point there is a transition between two movements. (b) and (c) 3D and 2D plot, respectively, of the filter coefficients for the l_2 cost function. (d) and (e) 3D and 2D plot, respectively, of the filter coefficients for the $l_2 + l_1$ cost function.

The pixel angular error (AE), also set out in [34], in the direction of the estimated displacement vector compared with the ground truth:

$$AE = \arccos \left(\frac{1.0 + d_{gt}^{(1)} d^{(1)} + d_{gt}^{(2)} d^{(2)}}{\sqrt{1.0 + (d^{(1)})^2 + (d^{(2)})^2} \sqrt{1.0 + (d_{gt}^{(1)})^2 + (d_{gt}^{(2)})^2}} \right). \quad (2.24)$$

The total angular error being the sum of the pixel angular error over the whole image.

The interpolation error (IE) is the error in the pixel intensity for a target image interpolated back to the reference image positions. The IE is the sum of the intensity differences over the whole image.

2.3.5 Adaptive filter registration results

The Developed System - Overview

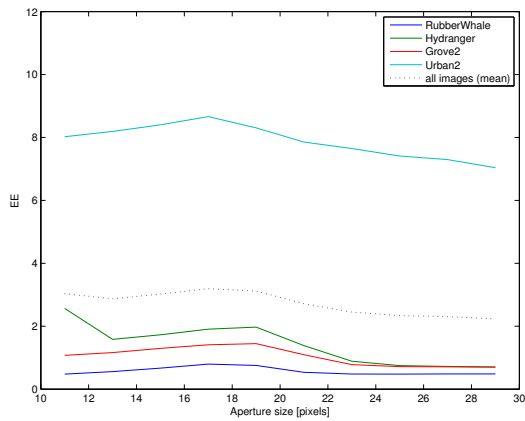
The adaptive filter registration algorithm proposed, depicted in Figure 2.14a, is based around an adaptive filter that is used to estimate the convolution kernel. The filter takes a square array of pixel intensities from a reference image and produces an estimate of the corresponding output pixel. The filter coefficients evolve and track the changes as the adaptive filter aperture is moved through the image. Rather than trying to simply correct the target image, information about the movement is captured by extracting the position of the peak of the convolution kernel. On a local scale the position of this peak is assumed to represent the pixel displacement. This process is improved by using an adaptive filter that promotes a sparse solution. For stability, one of four different versions of the Moore curve are used as a scan path and the smoothest is selected. An output image is then created from the estimated displacement field. Using the newly created image as the new target image the registration process can be iterated, with the final DM being the sum of the DM obtained at each iteration. For each iteration the adaptive filter parameters (λ - the forgetting factor) are adjusted so that large movement discrepancies are accounted for first and smoother motions are compensated for in subsequent iterations.

Filter aperture size selection

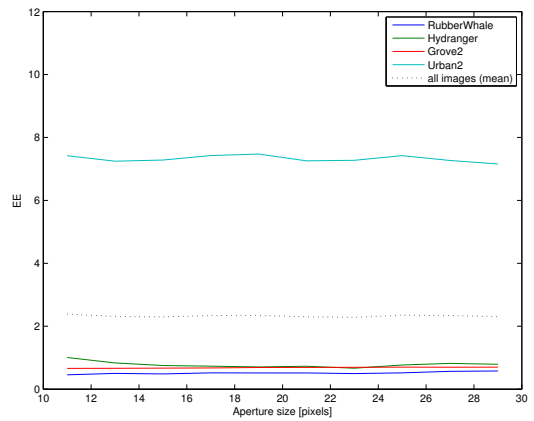
Fig. 2.12 and Fig. 2.13 show the EE, AE and IE for the each of the images as the aperture size changes. Fig. 2.12 shows the error metrics for the RLS filter. It can be seen that the mean average of the four example images (black dotted line) reduces with aperture size and plateaus out after an aperture size of approximately 23×23 pixels. Although the error

reduces further there is a trade of with processing time and therefore the aperture of 23×23 pixels was selected. Fig. 2.13 shows the results of aperture size change for the RLS-style filter with a sparse promoting penalty. It can be seen that the error is more consistent with aperture size. An aperture of 23×23 pixels was selected as a compromise between between the error curves for each image.

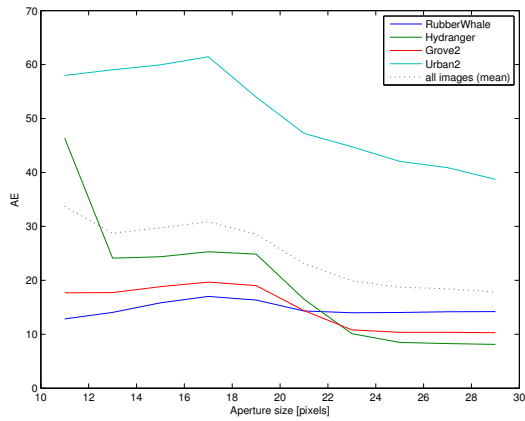
2.3 Adaptive filter registration



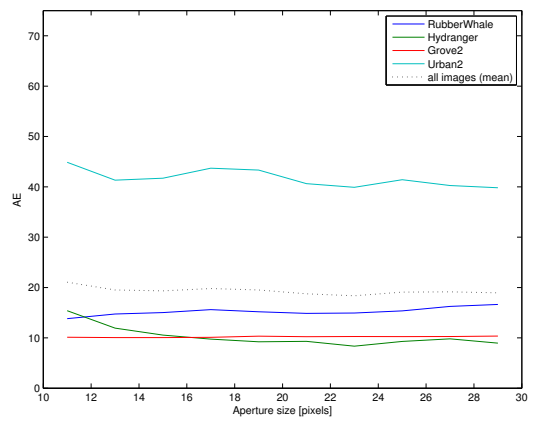
(a)



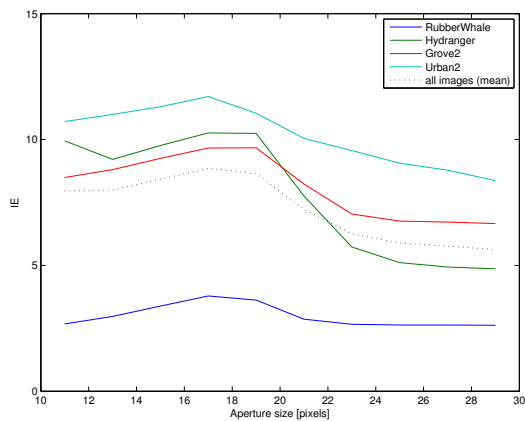
(a)



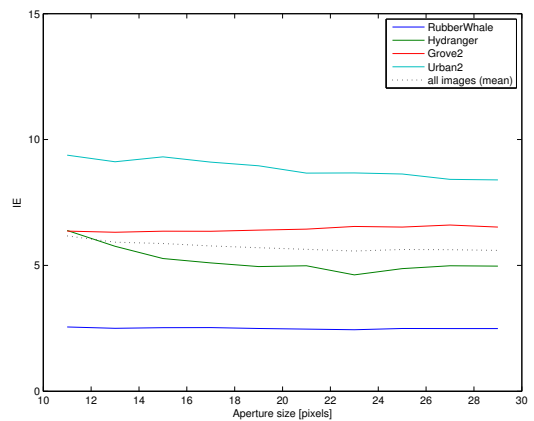
(b)



(b)



(c)



(c)

Fig. 2.12 RLS filter: EE (a), AE (b), IE (c) for the example images against aperture size. The errors begin to plateau at a size of approx 23×23 pixels.

Fig. 2.13 RLS-sparse promoting filter: EE (a), AE (b), IE (c) for the example images against aperture size. It can be seen that the errors change more slowly with aperture size. An aperture of 23×23 was selected as a compromise between all of the images.

Experimental setup

To test aspects of the registration discussed in this Chapter a system was set up as shown in Fig. 2.14. The adaptive filter in identification mode is shown Fig. 2.14a- **A** and in more detail in Fig. 2.14b. For the experiments, the adaptive filter is an RLS algorithm or an RLS style algorithm incorporating an l_1 optimisation constrain. Fig. 2.14a-**A** also shows multiple registrations with different scan paths (four in this numerical example). 2.14a-**B** is where the result from the scan path with the smoothest DM surface is selected. 2.14a-**C** is an accumulator recording the total DM for each iteration of the registration. 2.14a-**D**, a new “corrected” target is interpolated using cubic spline with not-a-knot end conditions. The interpolation is from the original target image and the reverse for the DM. 2.14a-**E**, the new target image is used in the next iteration of the registration.

The experiments show the aspects of the algorithm discussed in previous sections, applied to a selection of images from the Middlebury dataset [77]. The Middlebury dataset was selected for a number of reasons, firstly the selected images are designed to cover many of the problems that are encountered with registration. These problems include revealed and occluded pixels, sharp object boundaries, multiple independent areas of motion, smoothly varying areas of motion, repetitive patterns, areas with a large amount of detail, areas with little detail. Many of these problems are encountered with FLS images and therefore these images provided a good starting point to develop and algorithm. Also a reliable ground truth for the optical flow (DM) was available for each image pair. Finally, there was a basis for comparison with other registration algorithms. A 23×23 square aperture was set for the adaptive filter.

Five experiments were set up as follows:

1. RLS adaptive filter with a zig-zag scan path. This is the starting point for using an adaptive filter for registration.
2. RLS adaptive filter with a pseudo-Moore SFC scan path.
3. RLS adaptive filter with a pseudo-Moore SFC scan path, scanning the image in four directions to add diversity to the way that the adaptive filter evolves.
4. RLS adaptive filter with a pseudo-Moore SFC scan path, scanning the image in four directions. In addition the registration is performed multiple times, using the corrected image from the previous iteration as the new input image.
5. An adaptive filter with a pseudo-Moore SFC scan path, scanning the image in four directions, registration performed multiple times. In this experiment a RLS-style adaptive filter with a sparse promoting penalty is used as described in Sec. 2.3.3. The

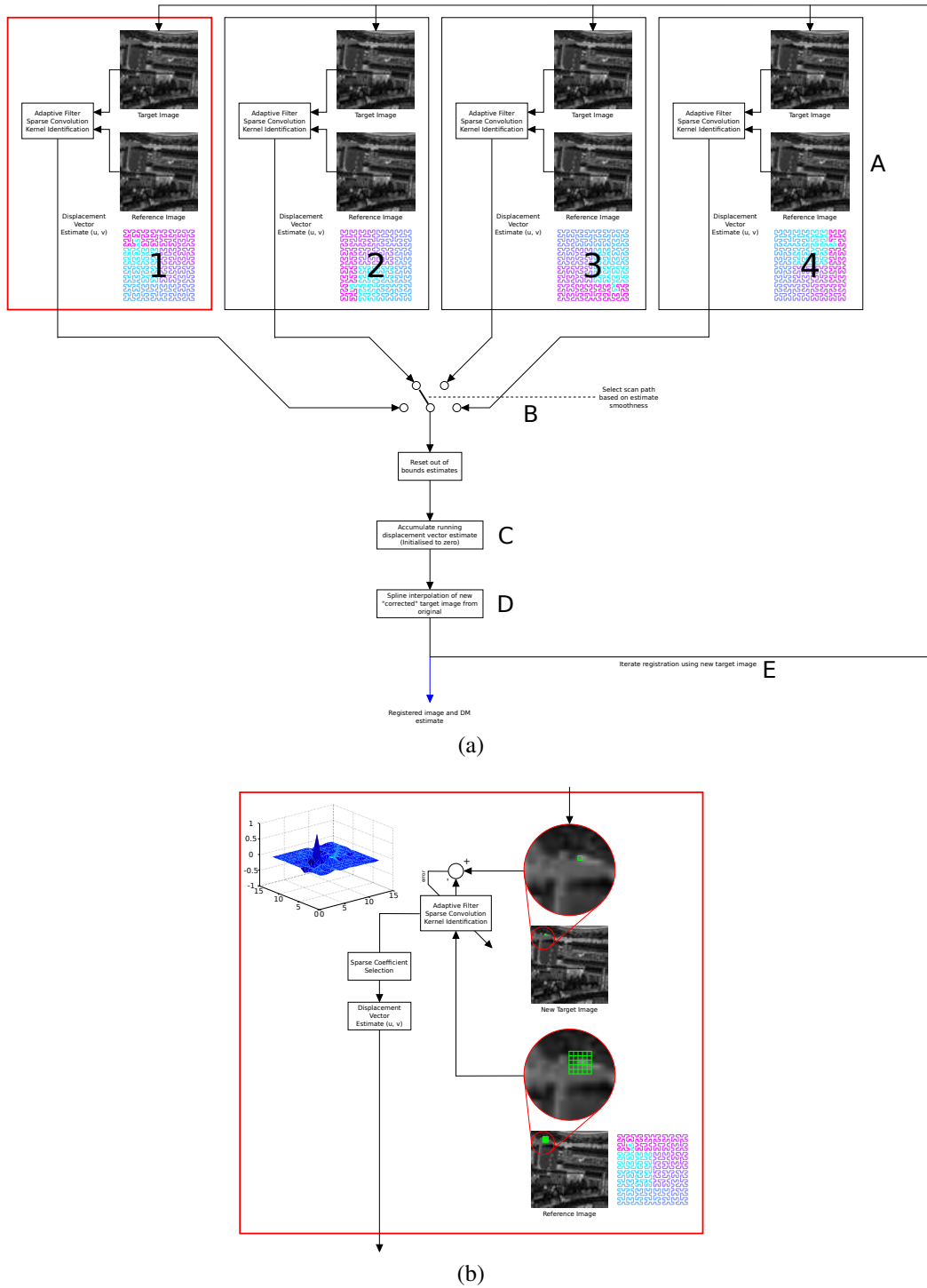
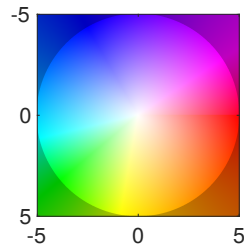


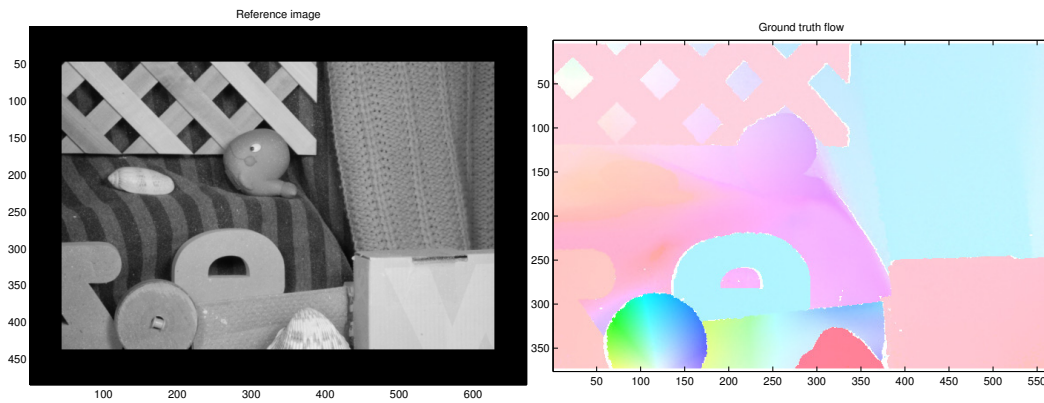
Fig. 2.14 (a) A diagram depicting the adaptive filter loop for the iterative registration. **A**, adaptive filter is used with four different scan path directions. **B**, the result from the scan path with the smoothest DM surface is selected. **C**, the total DM is accumulated for each iteration of the registration. **D**, a new “corrected” target is interpolated from the original target image and the reverse for the DM. **E**, the new target image is used in the next iteration of the registration. (b) a more detailed view of the adaptive filter. Image No. NCAP-000-000-081-843 [88].

aim of using a sparse promoting penalty is to better define the boundaries between objects.

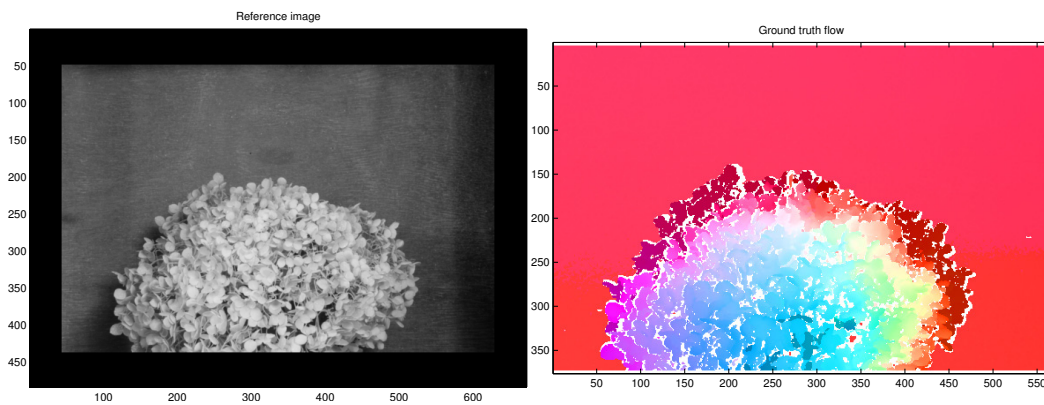
Shown in Fig. 2.15 are the selected images from the Middlebury dataset along with an analytically generated ‘lines’ image. Monochrome images were used for the experiments. In the evaluation experiments an initial value of $\lambda = 0.95$ was set. For the experiment where the registration was iterated the forgetting factor was reduced at each iteration as $\lambda_n = \lambda_{\max} - (\lambda_{\tau}^n (\lambda_0 - \lambda_{\max}))$, where $\lambda_{\tau} = 0.5$ is the λ decay time constant, and λ_{\max} is the maximum value λ can be. Figure 2.15a shows the colour scale for the adaptive filter registration results.



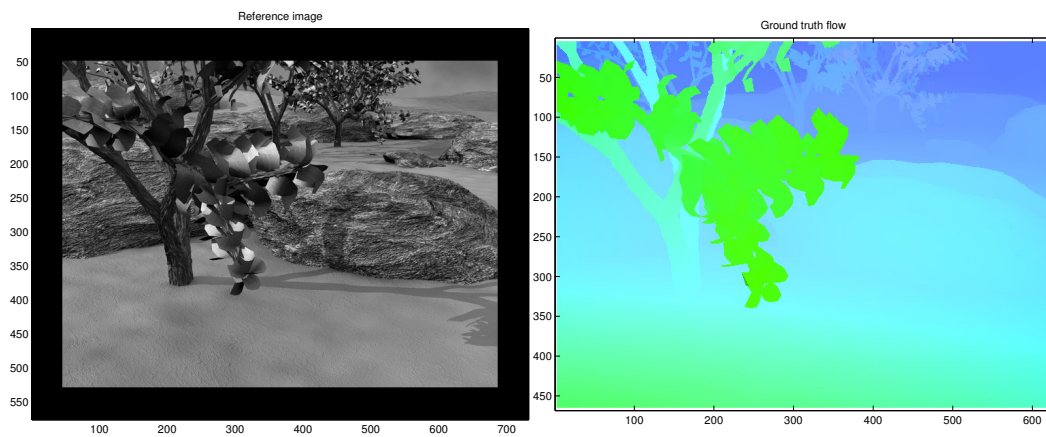
(a) Colour wheel key for the results of the adaptive filter registration.



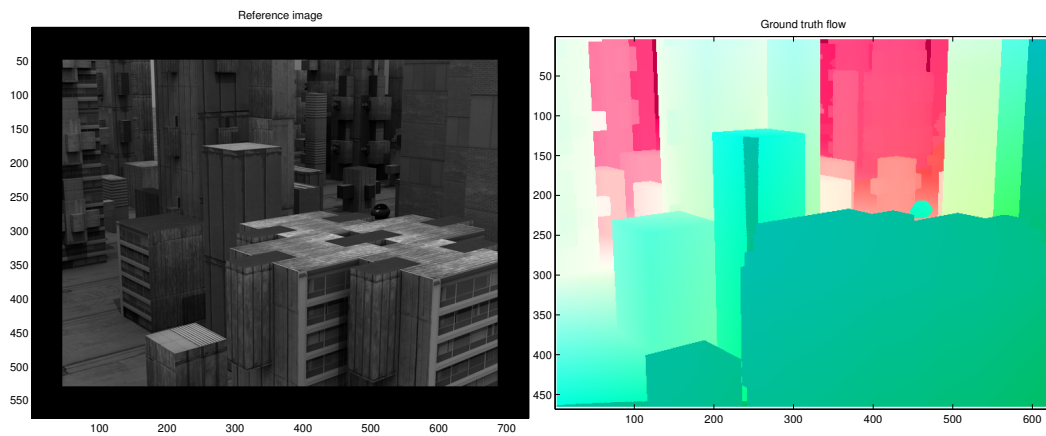
(b) Rubber Whale reference image and ground truth DM.



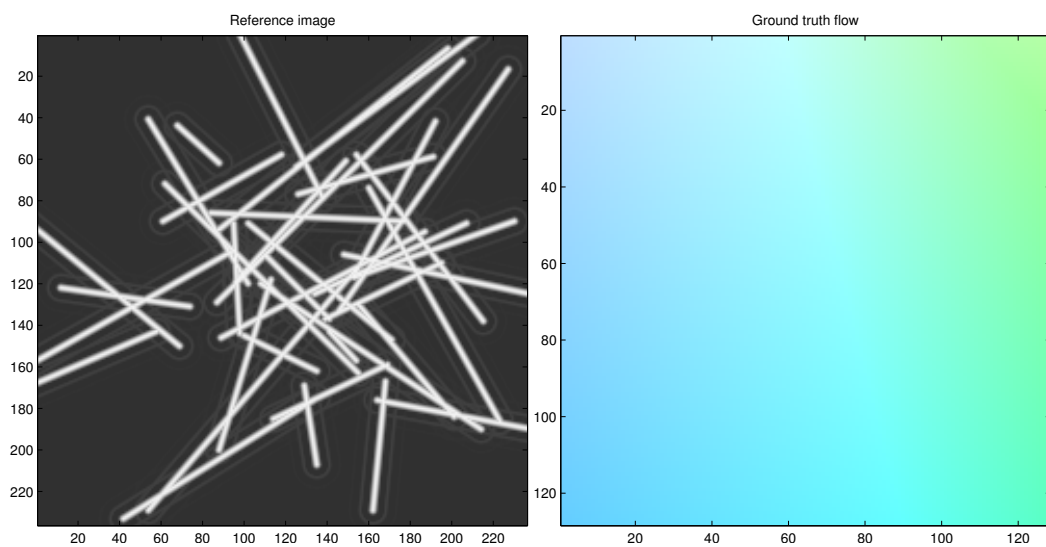
(c) Hydrangea reference image and ground truth DM.



(d) Grove2 reference image and ground truth DM.

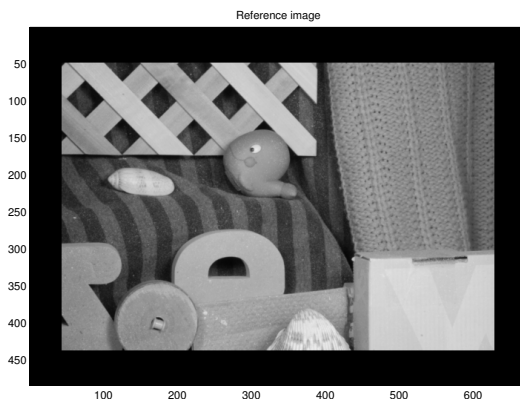


(e) Urban2 reference image and ground truth DM.

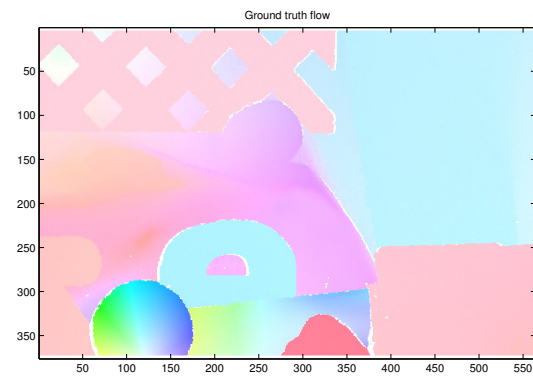


(f) Lines reference image and ground truth DM.

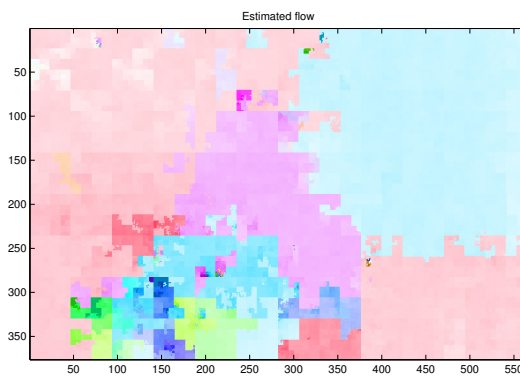
Fig. 2.15 Test images Rubber Whale Hydrangea, Grove2 and Urban2 from the Middlebury dataset. In addition an analytically generated 'lines' test image was created.



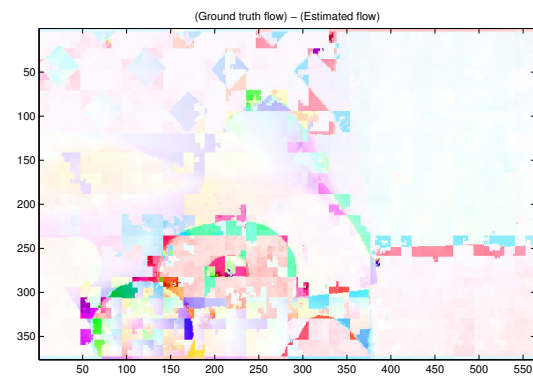
(a) Reference image.



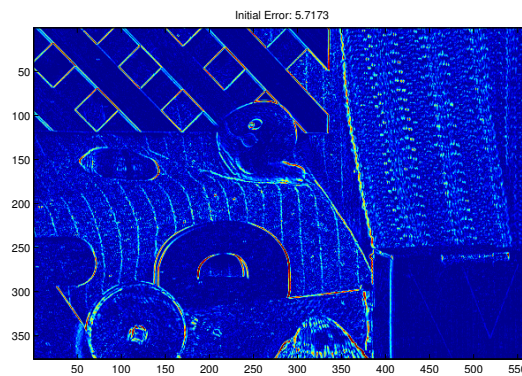
(b) Ground truth displacement map (optical flow).



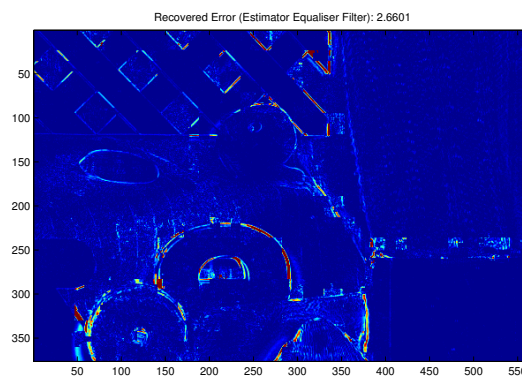
(c) Estimated displacement map.



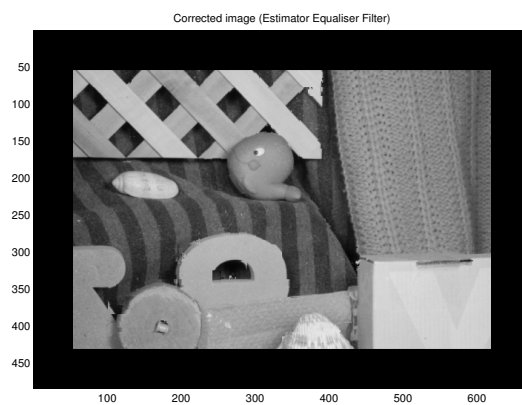
(d) Difference between the Ground truth and the estimated displacement map.



(e) Initial error in pixel intensities between the reference and target images.



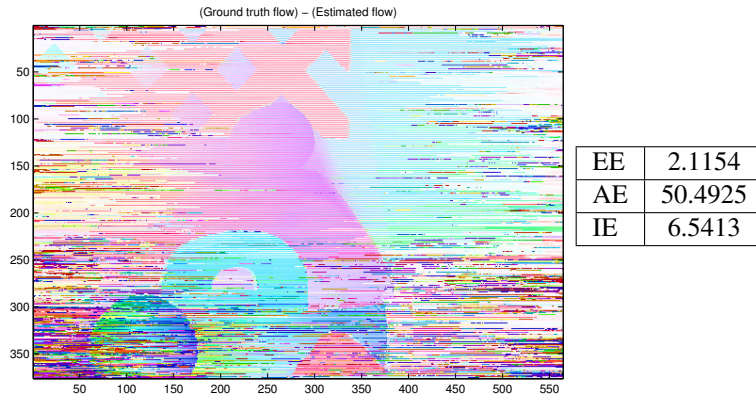
(f) Final error in pixel intensities between the reference and target images.



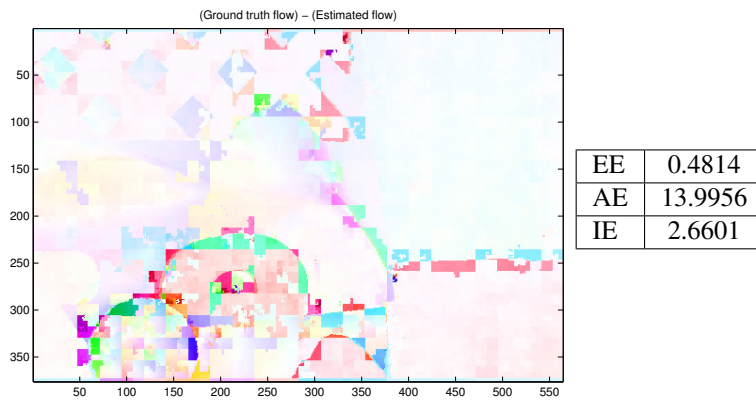
(g) Interpolated output image. The target image pixel interpolated back to the estimated position in the reference image.

Fig. 2.16 An example of registration of the Rubber Whale image from the Middlebury dataset.

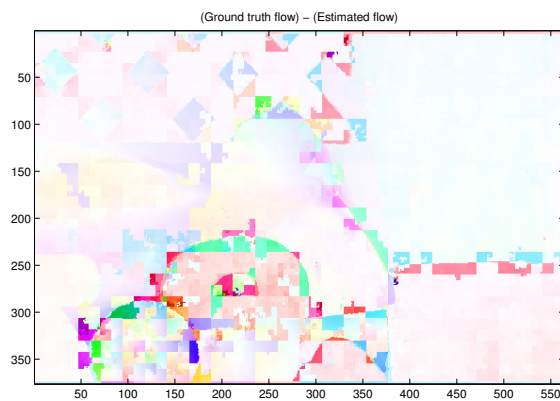
Registration results



(a) zig-zag scan pattern top to bottom (Experiment 1).

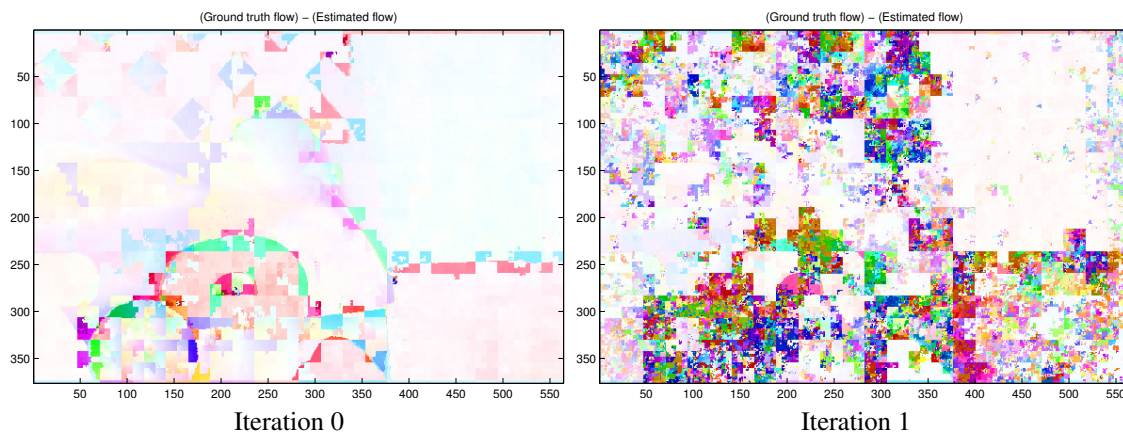


(b) Space filling curve scan pattern (Experiment 2).



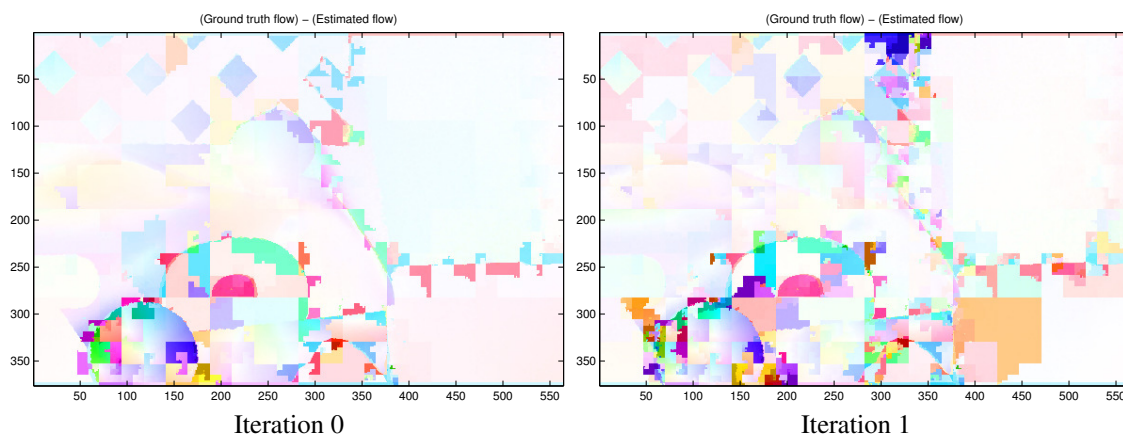
SFC orientation	1	2	3	4
EE	0.4814	0.4778	0.4785	0.4620
AE	13.9956	13.9737	13.9905	13.3801
IE	2.6601	2.6285	2.6105	2.6869

(c) Space filling curve scan pattern, with diversity in scan direction (Experiment 3).



Iteration	0	1	2	3	4
EE	0.4814	2.0169	4.9961	6.1787	6.4864
AE	13.9956	35.6733	64.029	72.9119	76.6382
IE	2.6601	7.186	13.9734	15.4634	15.8557

(d) Space filling curve scan pattern, with diversity in scan direction and iterative registration (Experiment 4). The best result was at zero iterations.








Iteration	0	1	2	3	4
EE	0.5474	0.7244	0.99642	1.5957	2.1275
AE	16.3263	19.3179	22.566	28.1722	38.61
IE	2.6243	2.9906	3.5545	5.07037	6.7809

(e) Space filling curve scan pattern, with diversity in scan direction, iterative registration and sparse promoting penalty (Experiment 5). The best result was at zero iterations.

Fig. 2.17 Five experiment results using the Rubber Whale image from the Middlebury dataset.

Table 2.1 A summary of the registration results from the initial registration algorithm.

Test image	Rubber Whale	Hydrangea	Grove2	Urban2	Lines	
						
EE	zig-zag	2.1154	3.4346	1.5546	10.5779	0.9557
	SFC	0.4814	0.8871	0.7801	7.6493	0.5640
	SFC, multi-direction (directions 1-4)	0.4814	0.8871	0.7801	7.6493	0.5640
		0.4778	0.8680	0.7708	7.6484	0.5321
		0.4785	0.8896	0.8235	7.4865	0.53981
		0.4620	0.8835	0.8243	7.5215	0.5431
	SFC, multi-direction, iteration (iterations 0-4)	0.4814	0.8871	0.7801	7.5215	0.5431
		2.0169	2.9546	2.4812	9.9757	3.8694
		4.9961	5.2703	4.488	11.2467	6.9079
		6.1787	6.6632	5.7153	11.3624	7.0413
	SFC, multi-direction, iteration, sparse (iterations 0-4)	6.4864	7.128	6.4241	11.4326	7.0947
		0.5474	0.7430	0.7000	7.2723	0.46188
		0.7244	1.0379	0.6780	7.6844	0.40401
		0.99642	1.4733	0.9032	8.323	0.43446
		1.5957	2.0339	1.2114	8.7821	0.5426
	2.1275	2.5428	1.474	9.1952	0.5581	

2.3.6 Adaptive filter registration discussion

With reference to the first experiment with the zig-zag scan path, the largest improvement comes with the use of the SFC as the scan path. This gave a marked improvement in all of the test images but particularly where there are many independent motions. This is most apparent in the Hydrangea dataset (see Fig. 2.15c). The SFC gives a large improvement with no runtime overhead, however implementation is made more difficult by the more complicated addressing for each pixel. Also, because the SFC traversed the image horizontally and vertically, processing can not be performed on a line by line basis (as may be captured by a FLS).

The variation in scan direction gave improvements in two of the datasets (although this is dependent on the best direction not being the default). The improvement was small for the large overhead of scanning the image four times.

The aim of iteratively registering the image was to try to smooth the image, taking into account rapid changes in the displacement map with a large forgetting factor. Then, refine the estimation by reducing the forgetting factor at each iteration. This was aimed at images such as Hydrangea (Fig. 2.15c) and Grove2 (Fig. 2.15d). It can be seen that for the classical RLS algorithm this did not improve the results for any of the further iterations. In the results, the Hydrangea and Grove2 images show the poorest results because most of the inaccuracies in

Table 2.2 A summary of the registration results from the initial registration algorithm.







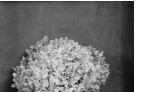

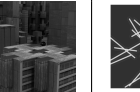

Test image		Rubber Whale 	Hydrangea 	Grove2 	Urban2 	Lines 
AE	zig-zag	50.4925	41.6283	20.9198	80.2345	17.732
	SFC	13.9956	10.1126	10.8291	44.7569	9.0942
	SFC, multi-direction (directions 1-4)	13.9956	10.1126	10.8291	44.7569	9.0942
		13.9737	10.2328	10.664	44.5662	8.6799
		13.9905	10.4174	11.3945	43.3466	8.8962
		13.3801	10.5847	11.4436	43.3457	8.8822
	SFC, multi-direction, iteration (iterations 0-4)	13.9956	10.1126	10.8291	43.3457	8.8822
		35.6733	31.4644	28.4239	69.4607	54.5184
		64.029	56.2084	49.3842	80.3846	80.103
		72.9119	71.1913	63.4162	82.2755	83.3715
	SFC, multi-direction, iteration, sparse (iterations 0-4)	76.6382	78.111	71.7629	83.5998	82.9427
		16.3263	9.2028	10.3712	40.8365	8.067
		19.3179	12.6196	9.6567	45.1447	7.317
		22.566	17.5259	11.9763	51.7198	7.9766
	28.1722	23.3312	14.8686	55.9602	9.8689	
	38.61	28.3371	18.1801	61.74	10.011	

Table 2.3 A summary of the registration results from the initial registration algorithm.

Test image		Rubber Whale 	Hydrangea 	Grove2 	Urban2 	Lines 
IE	zig-zag	6.5413	12.5116	11.4576	13.7105	18.1601
	SFC	2.6601	5.7291	7.0405	9.5632	10.1
	SFC, multi-direction (directions 1-4)	2.6601	5.7291	7.0405	9.5632	10.1
		2.6285	5.5725	6.9632	9.5426	10.0512
		2.6105	5.6477	7.16	9.4856	9.7557
		2.6869	5.6467	7.169	9.4557	10.4567
	SFC, multi-direction, iteration (iterations 0-4)	2.6601	5.7291	7.0405	9.4557	10.4567
		7.186	12.7923	13.4487	13.5887	45.4328
		13.9734	14.8802	19.1785	14.6309	62.2096
		15.4634	15.5757	21.4774	14.7413	60.4212
	SFC, multi-direction, iteration, sparse (iterations 0-4)	15.8557	15.8331	22.218	14.5121	61.4261
		2.6243	4.9117	6.4341	8.4918	9.4892
		2.9906	6.1316	6.0075	9.2304	8.0466
		3.5545	7.7324	7.2109	10.5755	8.5259
	5.07037	9.5802	8.5353	11.4396	10.3716	
	6.7809	10.9114	9.5932	12.4443	10.907	

the estimation are at the motion transitions. Therefore, when the new target image is created it has diverged from the reference image.

In the final experiment a sparse promoting penalty was used for the adaptive filter. The aim was to better define the motion boundaries. For all of the images, apart from the Rubber Whale image, the registration was improved. In addition two images showed an improvement in the first iteration of the registration. The boundaries were more stable and the improvement came in the refining of the estimation in the smoother sections of the image.

The aperture size for the adaptive filter was set to 23×23 for all of the images. The estimate of the maximum movement, that in turn dictates the size of the adaptive filter aperture, is critical for the correct registration. This is apparent in the Urban2 (Fig. 2.15e) image registrations where the aperture was not large enough to accommodate the movement in the foreground and the registration fails. For FLS images where navigation is of interest then this maximum motion would be constrained by the physical dynamics of the platform.

So that many images could be tested, analytical generated images were used. This consisted of a series of lines that had a random size and orientation. An example of the registration for one Lines image is shown in Fig. 2.15f. The registration shows an improvement at each stage.

The adaptive filter is able to give good registration results where the motion change is smooth; this is enhanced by scanning the image with a SFC. The difficulties arise where there are boundaries in the displacement map. This is also a problem in FLS images where there is a moving acoustic shadow, or indeed where the poor SNR means that a good match for the pixel cannot be found. The sparse promoting penalty showed a general improvement in defining these boundaries.

The possible improvements with the diverse scan direction and the multiple iterations of the registration, in practice do not warrant the large computational overhead. Indeed, in many of the experiments the multiple iteration caused the registration error to increase.

In light of these conclusions an algorithm was developed for FLS specifically. The algorithm was to include sparse approximation techniques and needed a greater emphasis on the detection and rejection of poor registrations, particularly round motion boundaries.

To make a comparison with more common methods, the adaptive filter part of the algorithm was replaced with a Normalised Cross Correlation (NCC) algorithm and a Sum of Absolute Differenced (SAD) algorithm. Each algorithm produced a square aperture of results from the similarity measure and the peak was selected in the same way as for the adaptive filter experiments. The results are shown in Table 2.5. It can be seen from the results the improved performance of the proposed algorithms with the RLS-style filter with an l_2 and l_1 cost function giving the best results.

Table 2.4 A summary of the best registration results from the initial registration algorithm.

Test image	Rubber Whale	Hydrangea	Grove2	Urban2	Lines
EE					
zig-zag	2.1154	3.4346	1.5546	10.5779	0.9557
SFC	0.4814	0.8871	0.7801	7.6493	0.5640
SFC, multi-direction	0.4814	0.8871	0.7801	7.5215	0.5431
SFC, multi-direction, iteration	0.4814	0.8871	0.7801	7.5215	0.5431
SFC, multi-direction, iteration, sparse	0.5474	0.7430	0.6780	7.2723	0.40401
AE					
zig-zag	50.4925	41.6283	20.9198	80.2345	17.732
SFC	13.9956	10.1126	10.8291	44.7569	9.0942
SFC, multi-direction	13.9956	10.1126	10.8291	43.3457	8.8822
SFC, multi-direction, iteration	13.9956	10.1126	10.8291	43.3457	8.8822
SFC, multi-direction, iteration, sparse	16.3263	9.2028	9.6567	40.8365	7.317
IE					
zig-zag	6.5413	12.5116	11.4576	13.7105	18.1601
SFC	2.6601	5.7291	7.0405	9.5632	10.1
SFC, multi-direction	2.6601	5.7291	7.0405	9.4557	10.4567
SFC, multi-direction, iteration	2.6601	5.7291	7.0405	9.4557	10.4567
SFC, multi-direction, iteration, sparse	2.6243	4.9117	6.0075	8.4918	8.0466

Table 2.5 A comparison of the RLS and Sparse filter with Normalised Cross Correlation (NCC) and Sum of Absolute Differenced (SAD). The Hydrangea image was uses with a single scan path and 0 iterations.

Test image	SFC, RLS	SFC, RLS-sparse	SFC, NCC	SFC, SAD
EE				
	0.88711	0.66659	1.2995	1.5919
AE				
	10.1126	8.3509	12.4032	14.3289
IE				
	5.7291	4.6241	6.4831	6.7198

2.3.7 Sonar type noise

To investigate the effects of noise on the registration the Hydrangea image from the Middlebury dataset was selected and noise added. Noise in sonar images has two characteristics [51]; where there is no return from a reflector, for instance where there is an acoustic shadow, the noise follows a Gaussian distribution. Where there is a return from a reflector the noise has a Rayleigh distribution. To reflect this the Hydrangea image was modified with added noise. The Hydrangea image was selected because it had a well defined foreground on a dark background. This allowed the dark background to be treated as a shadow region and therefore Gaussian noise was added, and the lighter foreground to be treated like a reflector and so Rayleigh noise was added. Fig. 2.18 shows the reference Hydrangea image and a mask showing the separation between the background and the foreground. This separation was made using a threshold on the pixel intensity. For this image an intensity ≥ 110 is foreground, < 110 background, the full intensity range is 0-255. Three experiments were conducted with increasing noise levels. Fig. 2.19 shows noise distributions added to the shadow and reflector regions for each of the experiments. The two frames for experiment 3 are shown in Fig. 2.20. The mean for the Gaussian noise ($\mu = 35$) and the standard deviation ($\sigma = [4, 8, 12]$) was set based on measurements from the Ship's hull dataset (see Section 5.2.1). The Rayleigh noise distribution was based on the image threshold level of 110, where the widest distribution of test 3 when added to the image would give values up to the full range.

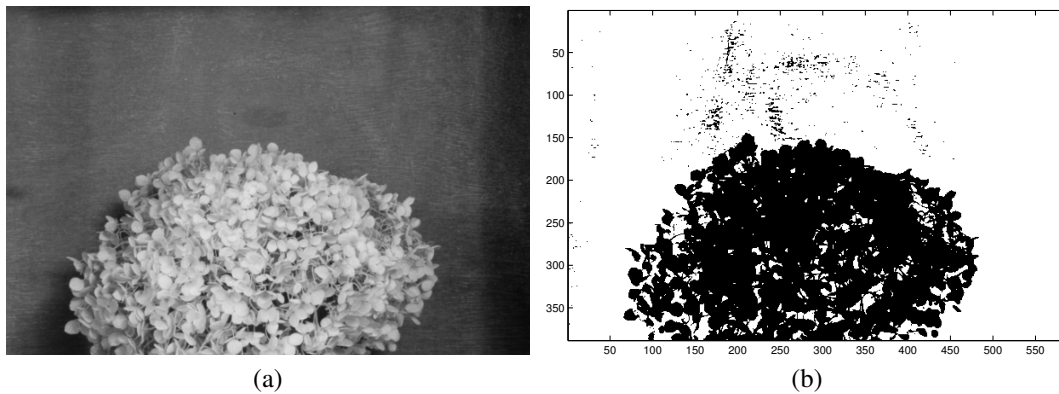


Fig. 2.18 (a), the hydrangea reference image. (a) the background and foreground separation using intensity level thresholding.

The results of the experiments are shown in Table 2.6. It can be seen that RLS-style filter with a sparse promoting penalty consistently out performs the classic RLS algorithm.

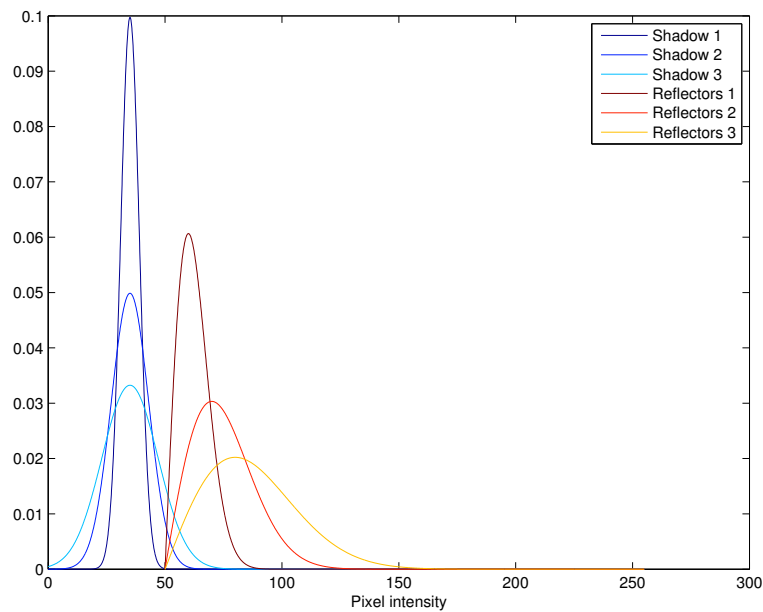


Fig. 2.19 The noise distributions added for each of the three tests. Test 1 has the distribution Shadow 1 added to the background/‘shadow region’, and Reflectors 1 distribution added to the foreground/‘reflector’ region. Shadow and Reflectors 2 for test 2 and Shadow and Reflectors 3 for test 3. For the Gaussian distribution $\mu = 35$ and $\sigma = (4, 8, 12)$. For the Rayleigh noise the start point was 50 and the peak of the distribution was at (10, 20, 30).

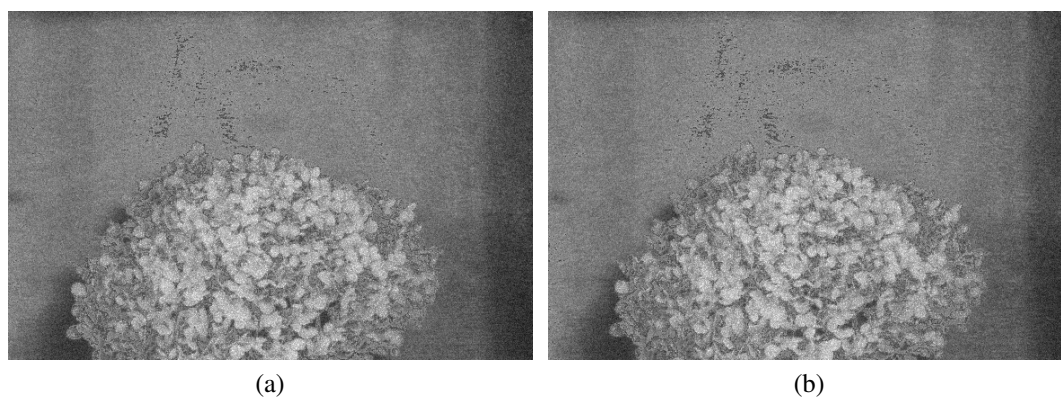


Fig. 2.20 (a) reference image, (b) target image. The two images to register with noise added as per test 3.

Table 2.6 A comparison of the RLS and RLS-Sparse filter with different amounts of added noise. The Hydrangea image was uses with a single scan path and 0 iterations.

Test image	SFC, RLS	SFC, sparse
EE		
No added noise	0.8871	0.6666
Noise test 1	2.6782	1.5303
Noise test 2	5.1588	4.1251
Noise test 3	6.3069	6.1393
AE		
No added noise	10.1126	8.3509
Noise test 1	29.0911	15.8455
Noise test 2	58.8554	43.1149
Noise test 3	71.9067	64.7966
IE		
No added noise	5.7291	4.6241
Noise test 1	10.9635	9.4866
Noise test 2	15.7257	14.7276
Noise test 3	20.0736	19.8731

2.4 Conclusion and discussion

This chapter discusses some of the problems associated with image registration and proposes a fine (sub-pixel) registration algorithm that is based on an adaptive filter. Many of the problems associated with optical image registration are also applicable to FLS data. These problems are made worse because of the characteristics of FLS images: poor SNR, low resolution and distortions due to the projection of a 3-D scene onto a 2-D plane. Common techniques that are used for optical registration are more challenging for FLS because object boundaries and features are more difficult to define.

The aim of the work in this chapter has been to examine some techniques that could be used in a registration algorithm that is able to produce a full frame DM. This places the algorithm between a full frame affine transform estimator as in [14] and feature detection as in [20], the work being closer to the side-scan registration proposed in [25]. An adaptive filter with a SFC formed the basis for the registration algorithm, having good results reported in [66] and [68].

An RLS adaptive filter has been implemented and the DM estimation investigated with a dataset that highlighted many of the problems associated with registration. This dataset also included a ground truth of the DM so a quantitative assessment could be made. This assessment showed problems with the adaptive filter registration, particularly at the motion boundaries. The use of a SFC improved this by reducing the number of boundary transitions. Also, a sparse promoting penalty has been used with the aim of making the boundary

transitions more defined. Table 2.4 shows the results from the proposed adaptive filter registration algorithm. It can be seen that the SFC and the sparse promoting penalty showed the largest improvement to the DM estimation. The multi-directional scan paths had little effect which showed that the adaptive filter has a robust performance regardless of how each of the motion boundaries are approached. Therefore, the less complex single scan path is sufficient without a loss to the accuracy of the registration. The results from iterating the registration gave poor results, with the errors for all of the images increasing after the second iteration.

These experiments show the importance of evolving the adaptive filter within each area of motion and trying to reduce the number of times the adaptive filter crosses a motion boundary. An initial estimate of the motion would assist in the detection of the motion boundaries. Also, it would allow the aperture of the adaptive filter to be reduced, this would make the adaptive filter more efficient and improve the accuracy. It is therefore proposed that a coarse registration algorithm is developed.

Chapter 3

Registration algorithm for FLS images

The experiments detailed in Chapter 2 showed the benefit of evolving the adaptive filter within each area of motion and minimising the number of times the adaptive filter crosses a motion boundary. Therefore the development of a coarse registration algorithm is proposed, addressing the following problems:

- A poor identification of the motion boundaries. A coarse estimate allows a classification of the different areas of motion.
- The desire to reject poor quality registrations. If a sensor motion is to be inferred from the DM estimate then erroneous results would result in a poor estimation. Therefore, it would be desirable to be able to detect and remove poor estimates from the DM. A common method to validate the DM is to make a comparison between the registration in the forward direction (from the reference to the target image) and the backward direction (from the target to the reference image) [76, 25]. The aim of this chapter is to incorporate this validation technique into a registration algorithm. If the forward/backward registration is made to a discrete (pixel-wide) grid then the motion estimates can be easily compared. Using the assumption that the local pixel motion is to a small number of positions on the coarse grid, then sparse recovery techniques can be used.
- Improve the accuracy of the sub-pixel displacement estimate. A coarse estimate can be taken into account when selecting the position of the aperture for the adaptive filter. This makes it possible for the adaptive filter to have a smaller aperture, which makes the algorithm more efficient and the estimated convolution kernel more accurate.

It is proposed therefore to split the registration algorithm into two parts. The first part performs a forward/backward registration to a pixel-wide precision discrete grid, using a sparse recovery technique. From the forward/backward comparison a coarse (pixel-wide

precision) DM estimate is made. The second part of the algorithm refines the DM estimate further by using an adaptive filter to estimate the motion to sub-pixel accuracy producing a fine DM. A block diagram of the proposed registration algorithm for FLS images is shown in Fig. 3.1.

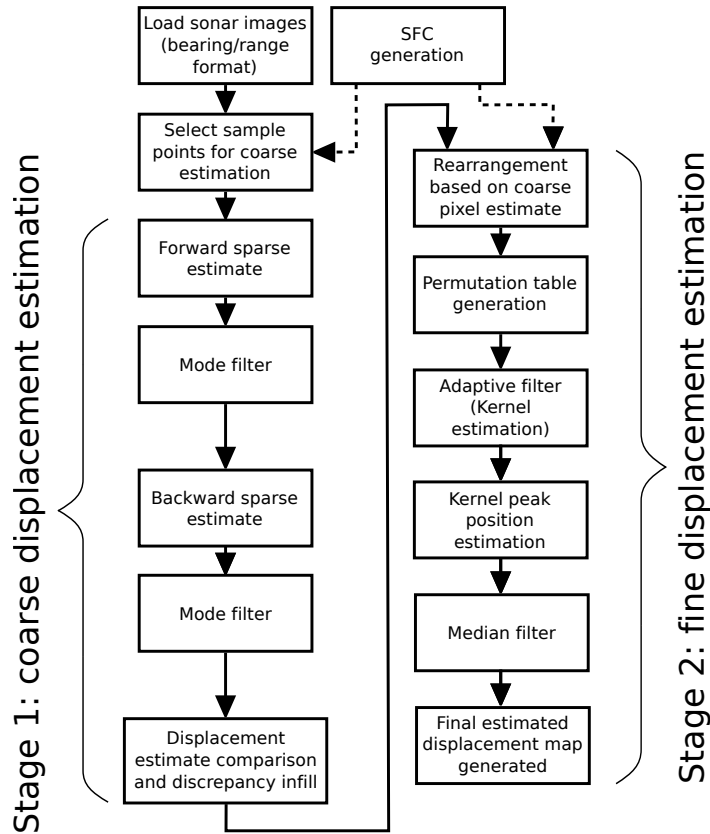


Fig. 3.1 A block diagram showing the proposed registration algorithm for FLS images.

Sections 3.1.1 to 3.1.4 explain the pixel-wide precision DM estimation through the use of sparse recovery techniques. Section 3.1.5 goes on to present the forward/backward comparison and the processing of the resultant DM, to produce a coarse DM estimate. Section 3.2 details the refinement of the coarse DM, where an adaptive filter is used to produce a convolution kernel estimate associated with each sub-pixel displacement. After estimating the motion associated with the convolution kernels, a fine DM is produced. Section 3.3 is an analysis of the complexity of the resultant algorithm. The chapter then ends with some concluding remarks in Section 3.4.

The work in this chapter is presented in the papers: B. Henson and Y. Zakharov, “Attitude-trajectory estimation for forward looking sonar based on acoustic image registration,” *IEEE Journal of Oceanic Engineering*, First submission May 2017, second submission Dec. 2017;

B. Henson and Y. Zakharov, “Local optical-flow estimation for forward looking imaging sonar data,” in *MTS/IEEE OCEANS-Monterey*, 2016, pp. 1–8.

3.1 Coarse DM estimation

The first part of the algorithm estimates a coarse DM. The algorithm uses a sparse recovery technique to find estimated displacement for each individual pixel on a pixel-wide precision grid.

3.1.1 Sparse representation techniques

To perform the pixel width precision displacement estimate, the problem is framed as matching a small block of pixels over a search area, with the expectation of a finite number of possible displacements.

Consider two frames, with a reference image $i_r(\xi)$ and a target image $i_t(\xi)$, where ξ is a pixel position within a frame X : $\xi \in X$. We assume that the relationship between the two frames is described by the convolution

$$i_t(\xi) = \sum_{m=1}^M h(\xi_m, \xi) i_r(\xi - \xi_m) + \varepsilon(\xi), \quad (3.1)$$

where $h(\xi_m, \xi)$ is an unknown kernel at pixel location ξ , ξ_m is a displacement and $\varepsilon(\xi)$ a measurement noise. The displacements ξ_m are limited to a support X_r , $\xi_m \in X_r$, e.g. X_r is a rectangular window around the origin. The support size is defined by dynamics of the sonar platform and sonar frame rate. In this section, the displacements ξ_m are assumed to be discrete to the pixel positions, while in Section 3.2 we will consider the sub-pixel case. This is a simplified signal model that, for example, does not describe the occluded and revealed pixels. At this stage, we assume that these effects are encompassed in the measurement noise. Note that the number M in (3.1) can be high. In our numerical example, the support X_r is a square window of size 41×41 so that $M = 41^2 = 1681$. The size of M is dictated by the maximum estimated pixel motion between frames. In this example the maximum motion was estimated by visually inspecting the sonar data and a value of ± 20 pixels was selected.

Ideally, with a pixel-width movement, the kernel $h(\xi_m, \xi)$ would contain a single non-zero coefficient. However, generally the movements are fractions of a pixel width, which require several discrete pixel-wide movements to approximate accurately. Besides, several objects can move independently within the support X_r and so several (L) coefficients $h(\xi_m, \xi)$ will be non-zero. However, typically $L \ll M$; in our numerical example, we use $L = 3 \ll M = 1681$.

At this stage, the task is to identify the kernel $h(\xi_m, \xi)$ at a pixel position ξ . For the identification, the target frame in the vicinity X_t of pixel ξ is treated as a measurement vector (vect) $\boldsymbol{\gamma}$:

$$\boldsymbol{\gamma} = \text{vect}\{i_t(\xi - \xi_m), \xi_m \in X_t\}. \quad (3.2)$$

The size (cardinality) $N = |X_t|$ of the set X_t should be related to the search size M and the number of non-zeros L . The number of measurements N should be sufficiently high to allow the kernel identification at low SNRs. However, if the kernel is fast varying over the frame, N should be reduced. For our numerical example, we choose X_t as a 13×13 square aperture, so that $N = 13^2 = 169$. This is large enough to be comparative with the identifiability criteria for random measurement selection, where $N = \mathcal{O}(L \log(M))$ [89], or the more conservative criterion of $N \geq \mathcal{O}(L^2 \log(M/L))$ [90]. We use the reference frame in the vicinity of pixel ξ to generate an $N \times M$ dictionary matrix \mathbf{A} with columns

$$\mathbf{A}^{(m)} = \text{vect}\{i_r(\xi - \xi_m), \xi \in X_t, \xi_m \in X_r\}. \quad (3.3)$$

The columns of \mathbf{A} are patches of images from the reference frame in the vicinity of the point of interest ξ . Therefore if $\boldsymbol{\gamma}$ were a perfect match with a column of \mathbf{A} then the motion would be an integer pixel width in vertical and horizontal. However, in practice this is not the case because the motion is unlikely to be a whole pixel width, also, the patch may contain more than motion, for instance at a object boundary. These scenarios are represented in the coefficients, \mathbf{h} , associated with each column of \mathbf{A} . Then we can write

$$\mathbf{A}\mathbf{h} = \boldsymbol{\gamma}, \quad (3.4)$$

where $\mathbf{h} = \text{vect}\{h(\xi_m, \xi), \xi_m \in X_r\}$ is an $M \times 1$ vector associated with modelled displacements.

Since the linear system (3.4) has a sparse solution, the first L coefficients of \mathbf{h} are estimated using the Orthogonal Matching Pursuit (OMP) [91, 89]. A summary of OMP is shown in Algorithm 1.

Taking each step in more detail. At initialisation the running signal estimate, \mathbf{a}_0 , is set to zero, this will be updated at each iteration. The initialisation for the residual, \mathbf{r}_0 , is set to be the patch from the target image that is to be matched. Within the algorithm the residual is used to select candidate motions (columns of \mathbf{A}) to be represented in the final coefficient vector \mathbf{f} . Line 1: L coefficients are to be found therefore the algorithm needs to iterate L times; this is the stopping criterion. Line 2: \mathcal{NCC} is the normalised cross correlation (NCC) as discussed further in Section 3.1.2 and set out in Eq. 3.10. The matrix \mathbf{G} is the result of

Algorithm 1 Orthogonal Matching Pursuit (OMP)

Input: \mathbf{A} : combined sense and representation matrix, $\boldsymbol{\gamma}$: the measurement vector, L : the number of sparse coefficients to be recovered.

Initialise: $\mathbf{a}_0 = 0, \mathbf{r}_0 = i_t(\xi - \xi_m), \xi_m \in X_t$

- 1: **for** $l = 1$ to L **do**
- 2: $\mathbf{G} \leftarrow \mathcal{NCC}\{\mathbf{r}_{l-1}, (i_r(\xi - \xi_m), \xi \in X_t, \xi_m \in X_r)\}$
 $\{\mathbf{r}_{l-1} \text{ is made into square aperture}\}$
- 3: $\Lambda_l \leftarrow \Lambda_{l-1} \cup \text{supp}(\mathbf{H}_1(\text{vec}(\mathbf{G})_{\Lambda_{l-1}^c}))$
 $\{\text{add largest residue index to support}\}$
- 4: $\mathbf{R} \leftarrow \mathbf{A}_{\Lambda_l}^T \mathbf{A}_{\Lambda_l}$
- 5: $\mathbf{f} \leftarrow \mathbf{A}_{\Lambda_l}^T \boldsymbol{\gamma}$
- 6: $\mathbf{a}_l|_{\Lambda_l} \leftarrow (\mathbf{R} + \boldsymbol{\Gamma}_l)^\dagger \mathbf{f}, \mathbf{a}_l|_{\Lambda_l^c} \leftarrow 0$ {update signal estimate}
- 7: $\mathbf{r}_l \leftarrow \boldsymbol{\gamma} - \mathbf{A} \mathbf{a}_l$ {update residual vector}
- 8: **end for**

the NCC between the residual for this iteration, \mathbf{r}_{l-1} , and the larger aperture to be search in the reference frame, $(i_r(\xi - \xi_m), \xi \in X_t, \xi_m \in X_r)$. Line 3: The index corresponding to the peak position is then added to the accumulated support vector Λ_l ; where $\mathbf{H}_\beta(\boldsymbol{\alpha})$ is an operator that sets to zero all but the β largest magnitude entries of $\boldsymbol{\alpha}$. More specifically, $\Lambda_l \leftarrow \Lambda_{l-1} \cup \text{supp}(\mathbf{H}_1(\text{vec}(\mathbf{G})_{\Lambda_{l-1}^c}))$, the single largest coefficient position is added to the support from the results of the NCC not including the positions of coefficients selected in previous iterations. Lines 4, 5 and 6 show the creation of a new signal estimate based only on the selected columns of \mathbf{A} . The algorithm used in this system has the addition of regularisation; where $\boldsymbol{\Gamma}_l$ is a diagonal matrix of small regularisation values of size l , in this case of the order of 10^{-6} . Line 7 is the calculation of a new residual vector that is to be used in the next iteration of the algorithm.

After L iterations (3 in this numerical example) the position of the largest coefficient of \mathbf{a} is taken to be the best position estimate, thus indicating the displacement.

In the estimate $\hat{h}(\xi_m, \xi)$ only the displacement $d(\xi)$ corresponding to the coefficient with the highest magnitude is kept for our further processing:

$$d(\xi) = \arg \max_{\xi_m \in X_r} |\hat{h}(\xi_m, \xi)|; \quad (3.5)$$

3.1.2 Pixel block matching techniques

The OMP algorithm relies on selecting a good candidate displacement at each iteration, therefore the quality of the pixel block match is important to obtaining a good result. This section discusses some block matching methods and why NCC was used.

Cross Correlation

A similarity measure that simply looks at the correlation:

$$\rho_{cc}(\xi, \xi_m) = \sum_{n=1}^N i_r(\xi - \xi_n) i_t(\xi - \xi_n - \xi_m) \quad (3.6)$$

where ξ is the pixel location within the frame X : $\xi \in X$. The displacements ξ_m are limited to a support X_r , $\xi_m \in X_r$, e.g. X_r is a rectangular window around the origin of size M . The displacements ξ_n are limited to a support X_t , $\xi_n \in X_t$, e.g. X_t is a rectangular window around the origin of size N representing the pixel block to be matched, where $N < M$. The best correlation over the the support X_r is taken to be the displacement for ξ . However, simply using the correlation can give poor results if the energy for the block area varies over the image. That is, a displacement that leads to a perfect match for the pixel block area might give a lower cross-correlation value than a result for a brighter part of the image [43]. One of the prominent problems with the registration of FLS images is the large variation in the intensity due to the directional nature of the single point illumination (insonification). Pre-whitening of the signal before the correlation is one way to improve the robustness to this illumination change.

Phase Correlation

One common approach to image matching is phase correlation. The image is transformed to the frequency domain, where the magnitude of the frequency components are set to unity. Correlation then takes place using the phase components only, thus making the method insensitive to intensity changes [14]. Although in practice this method is successful, it does have the disadvantage of equally weighting all frequency components [43]. This would have the effect of raising noise levels for low amplitude frequency components.

A shift in the spatial domain is equivalent to a phase shift in the frequency domain. Therefore if

$$i_t(\xi - \xi_n) = i_r(\xi - \xi_n - \xi_m), \quad (3.7)$$

where ξ is the pixel location within the frame X : $\xi \in X$. The displacements ξ_m are limited to a support X_r , $\xi_m \in X_r$, e.g. X_r is a rectangular window around the origin of size M . The displacements ξ_n are limited to a support X_t , $\xi_n \in X_t$, e.g. X_t is a rectangular window around the origin of size N representing the pixel block to be matched, where $N < M$. Then

$$I_t(\psi) = I_r(\psi) e^{-j(\psi + \xi_m)}, \quad (3.8)$$

where $I_t(\boldsymbol{\psi})$ is the 2D Fourier transform of the target image X_r , but where $X_r \neq X_t$ the image pixel is set to zero. $I_r(\boldsymbol{\psi})$ is the 2D Fourier transform of the reference image limited to the support X_r . $\boldsymbol{\psi}$ is the frequency component within the limited support X_f : $\boldsymbol{\psi} \in X_f$. The normalising cross-power spectrum gives the phase correlation matrix $S_{\text{ph}}(\boldsymbol{\psi})$:

$$e^{-j(\boldsymbol{\psi} + \boldsymbol{\xi}_m)} = \frac{I_t(\boldsymbol{\psi})I_r^*(\boldsymbol{\psi})}{|I_t(\boldsymbol{\psi})I_r^*(\boldsymbol{\psi})|} = S_{\text{ph}}(\boldsymbol{\psi}), \quad (3.9)$$

where $()^*$ denotes the complex conjugate. Finding the peak of the inverse 2D Fourier transform of S_{ph} gives an estimate of $\boldsymbol{\xi}_m$

Normalised Cross Correlation

The Normalised Cross Correlation (NCC) is where the target image aperture is normalised together with the aperture around a search point in the reference image [43]. The NCC is defined as follows:

$$\rho_{\text{ncc}}(\boldsymbol{\xi}, \boldsymbol{\xi}_m) = \frac{\sum_{n=1}^N [i_r(\boldsymbol{\xi} - \boldsymbol{\xi}_n) - \bar{i}_r(\boldsymbol{\xi} - \boldsymbol{\xi}_n)][i_t(\boldsymbol{\xi} - \boldsymbol{\xi}_n - \boldsymbol{\xi}_m) - \bar{i}_t(\boldsymbol{\xi} - \boldsymbol{\xi}_n - \boldsymbol{\xi}_m)]}{\sqrt{\sum_{n=1}^N [i_r(\boldsymbol{\xi} - \boldsymbol{\xi}_n) - \bar{i}_r(\boldsymbol{\xi} - \boldsymbol{\xi}_n)]^2 \sum_{n=1}^N [i_t(\boldsymbol{\xi} - \boldsymbol{\xi}_n - \boldsymbol{\xi}_m) - \bar{i}_t(\boldsymbol{\xi} - \boldsymbol{\xi}_n - \boldsymbol{\xi}_m)]^2}} \quad (3.10)$$

where $\bar{i}_r(\boldsymbol{\xi} - \boldsymbol{\xi}_n)$ is the mean of the support X_t , $\boldsymbol{\xi}_n \in X_t$ in the reference frame. $\bar{i}_t(\boldsymbol{\xi} - \boldsymbol{\xi}_n - \boldsymbol{\xi}_m)$ is the mean of the support X_t in the target frame, at the displacement $\boldsymbol{\xi}_m$. The displacements $\boldsymbol{\xi}_m$ are limited to a support X_r , $\boldsymbol{\xi}_m \in X_r$, e.g. X_r is a rectangular window around the origin of size M . The displacements $\boldsymbol{\xi}_n$ are limited to a support X_t , $\boldsymbol{\xi}_n \in X_t$, e.g. X_t is a rectangular window around the origin of size N representing the pixel block to be matched, where $N < M$.

The NCC is used as the block matching method in the algorithm due to the good performance with the high contrast images.

3.1.3 Selection of measurement (sample) points

Ideally, the coarse displacement estimation would be made for each pixel position in the frame. However, this would make the problem too complex in practice. Besides, with a measurement window X_t of size N , the number of estimates in a frame can be reduced by a factor comparable to N . In order to ensure a uniform density of sample points over the frame, a selection from a uniform distribution over the frame could be made. However, for a sonar image, portions of the image that give high intensity returns are likely to provide most accurate displacement estimates. Therefore, in our algorithm the reference image intensity is used as a probability density function (PDF) for generating the sample points $\boldsymbol{\xi} \in X_{\text{sample}}$

[92, 84]. An example distribution of sample points X_{sample} can be seen in plot 3 of Fig 3.8, where the PDF is defined from the reference frame (plot 1). In our numerical example, generating the sample points stops when $|X_{\text{sample}}| = 2891 \approx 1/17|X|$.

To demonstrate the improved efficiency of the sampling scheme a comparison has been made between a uniform distribution of sampling points and the intensity based PDF. Comparing the position of the sample points to the map of displacement estimates used for the platform motion estimate. A proportion of sample points coinciding with reliable displacement estimates can be found (Q). Using the notation from Section 4.2, X_B is the reliable displacement estimates, thus,

$$Q = \frac{|X_{\text{sample}} \cap X_B|}{|X_{\text{sample}}|}. \quad (3.11)$$

The mean Q value for the uniform and intensity based PDF was taken over the first 50 frames of the dam inspection dataset (see Section 5.2.2). For the uniform distribution the mean Q value is 0.3291, and for the intensity based PDF sampling the mean Q is 0.4110. It can be seen that the intensity based PDF sampling has over 8% more sampling points contributing to the final result.

3.1.4 Mode filter

From the displacement estimates $d(\xi) \in X_{\text{sample}}$, we wish to interpolate displacement values for all pixels in the frame, whilst eliminating outliers. Moreover, we wish to preserve the pixel precision estimates. To that end we can consider the individual motions as a nominal data type, and apply a square aperture mode filter to propagate the displacement values. More formally, the operation is as follows. An interpolated DM is given by

$$\hat{d}(\xi) = \arg \max_k g(k), \quad \xi \in X, \quad (3.12)$$

where $g(k)$ are elements of a vector \mathbf{g} representing a histogram (hist) of displacements measured in the vicinity X_{mode} of pixel ξ :

$$\mathbf{g} = \text{hist}\{d(\xi - \xi_m), \xi_m \in X_{\text{mode}}, (\xi - \xi_m) \in X_{\text{sample}}\}. \quad (3.13)$$

In our numerical example, X_{mode} is a square window of size 13×13 . Fig. 3.2 illustrates the interpolation using the Mode filter.

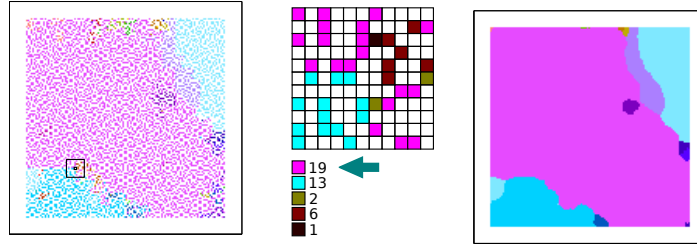


Fig. 3.2 An illustration of the interpolation using the Mode filter

3.1.5 Forward-backward registration

To estimate the movement between two frames, the registration can be performed not only from the reference to the target frame (forward), but also from the target to the reference frame (backward). A comparison between these two estimated displacements allows validation of the quality of the registration because the two displacements should complement, in that one should be the reverse of the other [25, 45]. In addition, the comparison can be used to remove estimates where pixels are revealed or occluded and therefore have no complement. To make the comparison the DM estimate in the forward direction $\hat{d}_f(\xi)$ is compared with the DM estimate in the backward direction $\hat{d}_b(\xi)$ as follows.

A selection of displacement estimates for the forward to backward comparison are $\hat{d}_f(\xi), \xi \in X_{fb}$ where X_{fb} represents the pixels where

$$|\hat{d}_b(\xi - \hat{d}_f(\xi)) + \hat{d}_f(\xi)| < d_t, \quad (3.14)$$

where d_t is the threshold vector magnitude (in this example $d_t = 2$).

Similarly, a selection of displacement estimates are made from the backward to forward comparison are $\hat{d}_f(\xi), \xi \in X_{bf}$ where X_{bf} represents the pixels where

$$|\hat{d}_f(\xi - \hat{d}_b(\xi)) + \hat{d}_b(\xi)| < d_t, \quad (3.15)$$

The final selection of displacement estimates are,

$$\hat{d}(\xi) = \hat{d}_f(\xi), \quad \xi \in X_{fb} \text{ or } \xi \in X_{bf}. \quad (3.16)$$

The comparison is illustrated in Fig. 3.3, for every pixel position $\xi \in X$ is made by taking the forward displacement estimate and examining the backward estimate at the location $\xi - \hat{d}_f(\xi)$. If the magnitude of the sum of these two displacements is less than a threshold ($d_t = 2$ pixels in the example) then the forward estimate is retained. This comparison is

also made by considering the backward and the forward estimate, again for a successful comparison the forward estimate is retained.

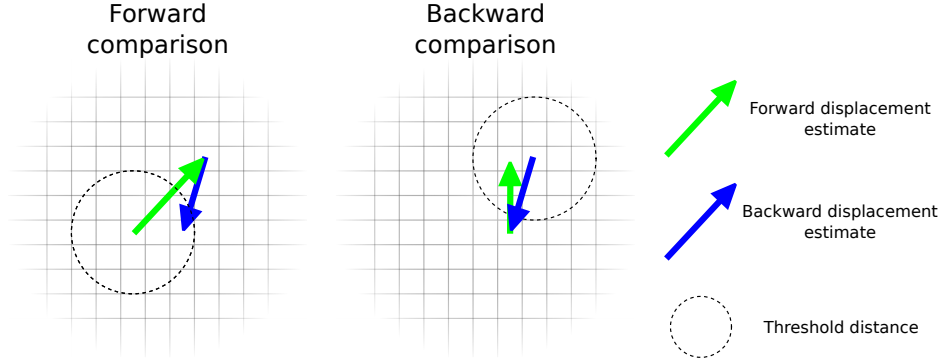


Fig. 3.3 A block diagram showing the forward and backward estimates and the sum of the vectors. The displacements where the difference is within a certain magnitude are retained.

Example forward and backward DMs can be seen in plots 4 and 5 of Fig. 3.8. The accepted displacements from the forward and backward comparison are shown in plot 6 of Fig. 3.8. The pixel locations where the displacement estimate is not retained are interpolated, taking the value of the nearest (in terms of pixels) accepted estimate. This interpolation allows a full frame DM to be generated and allows the adaptive filter to smooth the results between the accepted estimates from the forward/backward comparison. An example of the interpolated DM $\tilde{d}(\xi)$ is shown in plot 7 of Fig. 3.8.

3.2 Fine DM estimation

3.2.1 Re-centring and reordering

At this point in the proposed algorithm, there is a coarse displacement estimate $\tilde{d}(\xi)$ associated with each pixel location in the reference frame. Here an adaptive filter is used to identify the convolution kernel $h(\xi_m, \xi)$, related to the sonar sensor motion, to a sub-pixel precision.

The adaptive filter works most effectively if the change in the convolution kernel from one adaptive iteration to the next is a slow evolution. With this aim, the order that the pixels are presented to the filter is chosen to try to group similar movements. One way to achieve this is to alter the path of the SFC based on extracted features from the image such as in [93], however, as already discussed this is more difficult for FLS images. Therefore, we propose and use the following ordering algorithm based on the coarse DM estimate to produce a permutation table. Firstly, each displacement $\xi_m \in X_r$ represents a bin in the permutation table, i.e. M bins in total. The bins are ordered to provide the slowest evolution between

consecutive bins. The algorithm scans the DM $\tilde{d}(\xi)$ along a pseudo-Hilbert space filling curve (SFC) pixel-by-pixel [84]. The use of the SFC itself ensures a slow evolution of the kernel. For every pixel, the pixel position is added into the bin associated with the value $\tilde{d}(\xi)$. After the scan, every bin in the permutation table contains positions of all pixels with the same displacement, thus providing the slowest evolution of the convolution kernel within the bin. The position within a bin is ordered according to appearance along the SFC, as illustrated in Fig. 3.4. In total, I positions, I being the number of pixels in a frame, are recorded in all bins, thus the table describes an $I \times I$ permutation matrix. Thus, when reading from the permutation table from the first to the last bin, and within a bin from the first to the last pixel, the convolution kernel to be estimated would have a slow evolution.

Plot 8 of Fig. 3.8 shows an example of the rearranged pixels; where the pixel colour represents where in the order it is, blue is processed first and red last.

3.2.2 Adaptive filtering and centre estimation

The input to the adaptive filter is a sequence of square pixel apertures representing the regressor vectors from the reference frame, in the order determined by the permutation table. The size M_{fine} of the filter kernel $h_{\text{fine}}(\xi_m, \xi)$ to be estimated, and consequently the size of the input vector (*regressor*) should be kept as small as possible to improve the identification accuracy. To guarantee this small size, the *desired signal* of the adaptive filter, which is taken from the target image $i_t(\xi)$, is translated according to the coarse estimate $\tilde{d}(\xi)$, i.e. the desired signal is $i_t(\xi - \tilde{d}(\xi))$, this is illustrated in Fig. 3.5. Thus, the fine estimation is performed only in a part of area X_r , in the vicinity of the pixel position $\xi - \tilde{d}(\xi)$, therefore allowing a reduction in the filter aperture.

The size of the filter is therefore set with reference to the discrepancy allowed in the sum of the coarse forward-backward DM comparison, and in our example is 2 pixels width. To encompass a variation of ± 2 pixels width, an aperture of 7×7 pixels is used in our example. This translation of the *desired signal* is illustrated in Fig. 3.6. The signal model in (3.1) is now refined to

$$i_t(\xi - \tilde{d}(\xi)) = \sum_{m=1}^{M_{\text{fine}}} h_{\text{fine}}(\xi_m, \xi) i_r(\xi - \xi_m) + \varepsilon(\xi - \tilde{d}(\xi)). \quad (3.17)$$

For each pixel position ξ , the adaptive filter produces an estimate of the convolution kernel h_{fine} , by minimizing the least squares error [94, 95, 80]. The adaptive filter is applied to the whole frame and a convolution kernel of 7×7 coefficients is produced for each pixel

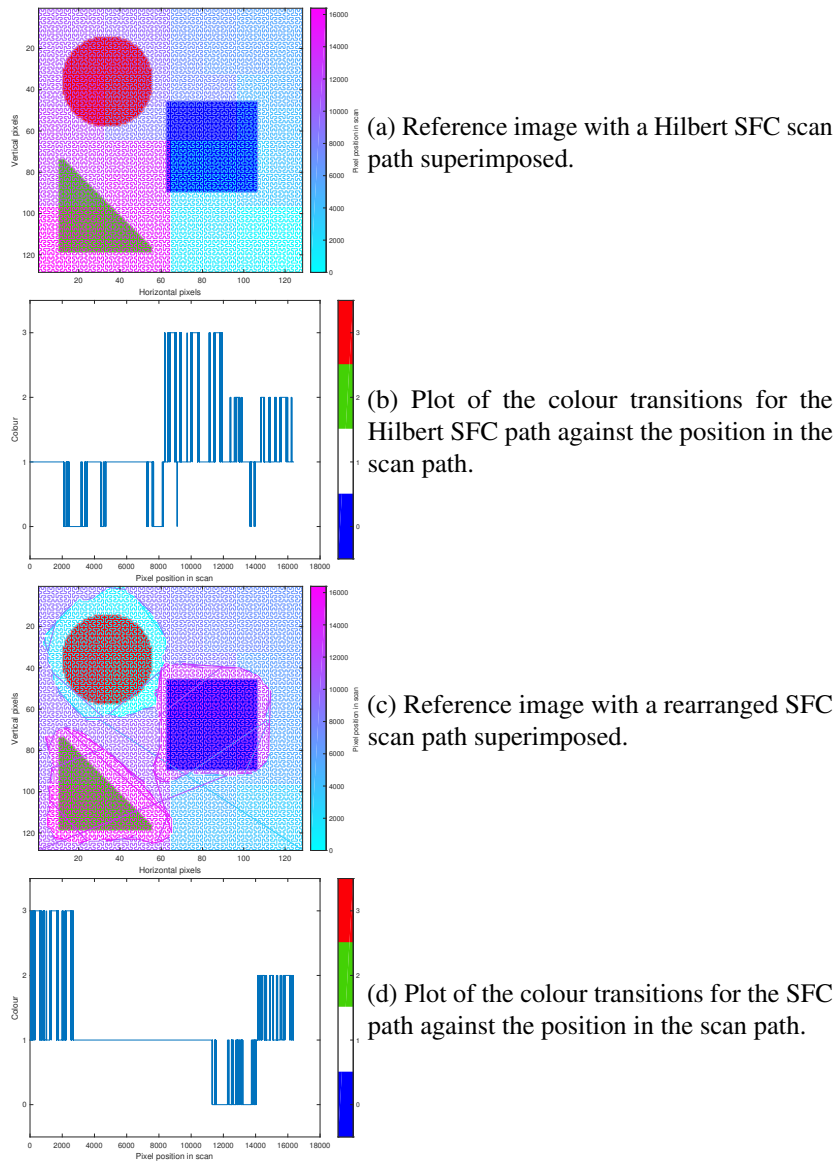


Fig. 3.4 An illustration (with the same image as used in Fig. 2.9) of how the SFC might be rearranged based on the results of the coarse DM estimate. In this example all three shapes would have different movements. The number of motion transitions can be seen in graph where the colour indicates each object (motion) boundary.

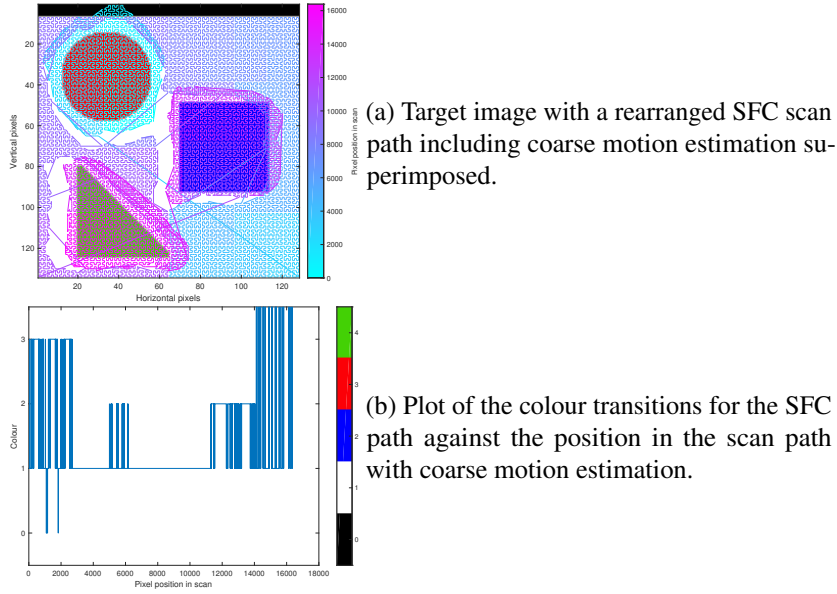


Fig. 3.5 An illustration (with the same image as used in Fig. 2.9) of how the SFC might be rearranged based on the results of the coarse DM estimate. In this example all three shapes would have different movements.

location. The adaptive filter algorithm used is an exponentially weighted recursive least squares (ERLS) [80] with a forgetting factor $\lambda = 0.98$.

3.2.3 Convolution kernel peak position estimation

Assuming that in reality there is a continuous 2-D convolution kernel whose samples at the pixel positions have been estimated by the adaptive filter as a matrix h_{fine} , we can identify the pixel displacement using interpolation. In our example, we use parabolic interpolation leading to the two-sided technique for estimation of the peak position as described in [96]. First the pixel-wide estimate is made by identifying the position of the largest entry,

$$[\hat{i}, \hat{j}] = \arg \max_{\substack{i \in 2 \dots p-1, \\ j \in 2 \dots p-1}} h_{\text{fine}}^{i,j} \quad (3.18)$$

where $h_{\text{fine}}^{i,j}$ is the i^{th} row and j^{th} column of the estimated $p \times p$ convolution kernel. Note a border of a single pixel is excluded from the search. The sub-pixel element is then identified using a two-sided parabolic technique:

$$d_{\text{sub}}^{(1)} = \frac{h_{\text{fine}}^{\hat{i}, \hat{j}+1} - h_{\text{fine}}^{\hat{i}, \hat{j}-1}}{h_{\text{fine}}^{\hat{i}, \hat{j}+1} + |h_{\text{fine}}^{\hat{i}, \hat{j}+1} - h_{\text{fine}}^{\hat{i}, \hat{j}-1}|}, \quad (3.19)$$

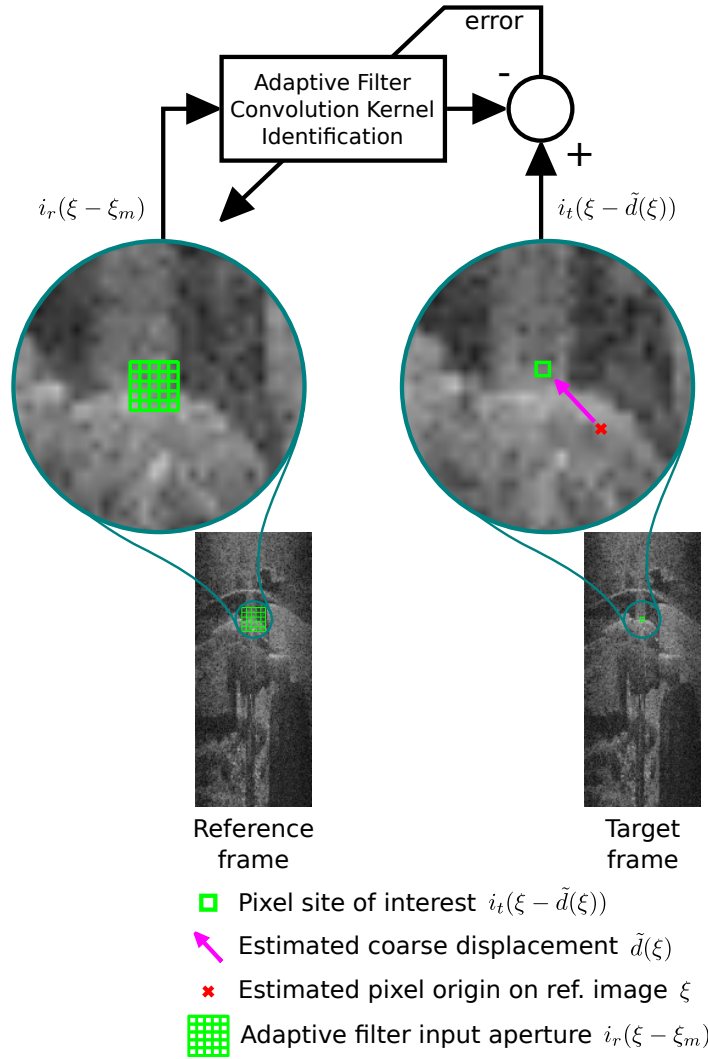


Fig. 3.6 An illustration of a single target pixel and the reference image aperture from which a convolution kernel is estimated.

$$d_{\text{sub}}^{(2)} = \frac{h_{\text{fine}}^{\hat{i}+1, \hat{j}} - h_{\text{fine}}^{\hat{i}-1, \hat{j}}}{h_{\text{fine}}^{\hat{i}+1, \hat{j}} + |h_{\text{fine}}^{\hat{i}+1, \hat{j}} - h_{\text{fine}}^{\hat{i}-1, \hat{j}}|} \quad (3.20)$$

Equations 3.19 and 3.20 estimate the peak of the convolution kernel to sub-pixel precision in the horizontal and vertical directions respectively. The position of this peak is then taken to be the end point of the pixel displacement. An initial fine DM is then created from the sum of the coarse and fine pixel displacement estimates.

Finally, the fine DM $\bar{\mathbf{d}}(\xi)$ is obtained by applying a median filter to all pixels in the initial fine DM. This is to remove isolated extreme results whilst preserving the boundaries between

larger regions of motion. In our numerical example, the aperture of the median filter is a square of size 13×13 .

An example of the output of the adaptive filter and the median filtered fine DM are shown in plots 9, and 10, respectively, of Fig. 3.8.

3.3 Complexity

The precise complexity of the algorithm depends ultimately on the image from the sonar and the size of the expected displacement. These factors dictate the size of the subset of pixels to be processed and the search space for individual registrations respectively. However,

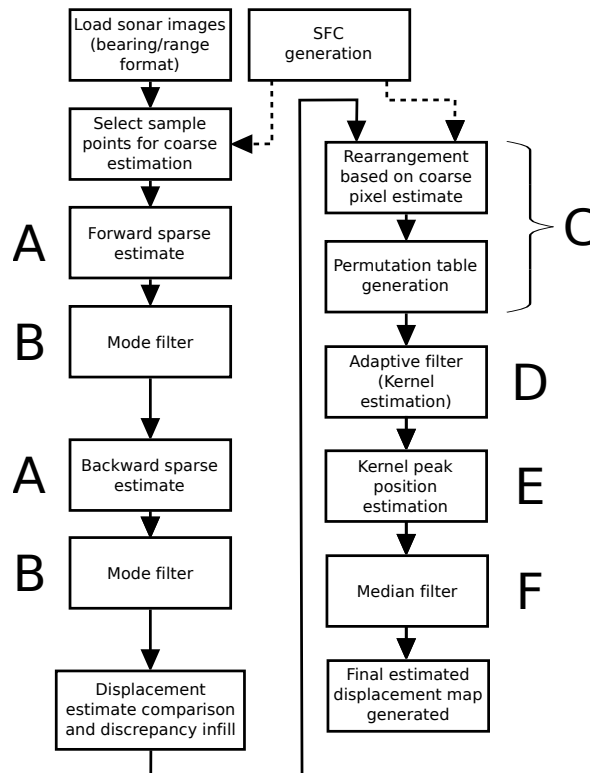


Fig. 3.7 A block diagram showing the proposed registration algorithm.

the order of the complexity can be calculated on a per pixel basis, given two parameters: firstly, an estimate of the maximum expected movement and hence the number of considered displacements (M in (3.1)), this would be from the dynamics of the system. Secondly, the size of the measurement vector needs to be specified (the cardinality of N , $N = |X_t|$ with reference to (3.2)), this represents a compromise between the locality of the registration and therefore how fine the resolution of the displacement map is, against obtaining more

smoothed results that can be more reliable. The complexity can be broken down as per the system diagram as shown in Fig. 3.7.

The coarse displacement estimation (labelled ‘A’ in Fig. 3.7) may be described with reference to Algorithm 1. Where the measurement vector is from a square aperture $N = a \times a$, and the number of considered displacement is from a square aperture $M = b \times b$. A fast implementation of the NCC is set out in [43]. Where the denominator is calculated using a running sum tables and the numerator of (3.10) is calculated in the frequency domain. For a fast implementation of the DFT, both a and b are padded with zeros to the next integer power of 2; that is, $a_{p2} = 2^{\lceil \log_2 a \rceil}$ and $b_{p2} = 2^{\lceil \log_2 b \rceil}$. The number of sparse coefficients searched for is small, in this implementation $L = 3$ [97]. The complexity of the coarse displacement estimation algorithm is summarised in Table 3.1.

Table 3.1 Complexity associated with the coarse displacement estimate

Line	Algorithm	\times	$+$	$/$	$\sqrt{\quad}$
2	$\mathbf{G} \leftarrow \mathcal{NCC}\{\mathbf{r}_{l-1}, i_r\}$	$12b_{p2}^2 \log_2 b_{p2} + 4b^2$	$18b_{p2}^2 \log_2 b_{p2} + 9b^2$	b^2	b^2
5	$\mathbf{R} \leftarrow \mathbf{A}_{\Lambda_l}^T \mathbf{A}_{\Lambda_l}$	$b_{p2}^2 \alpha^2$	$b_{p2}^2 \alpha^2$	-	-
6	$\mathbf{f} \leftarrow \mathbf{A}_{\Lambda_l}^T \boldsymbol{\gamma}$	$b_{p2}^2 \alpha$	$b_{p2}^2 \alpha$	-	-
7	$\mathbf{a}_l _{\Lambda_l} \leftarrow (\mathbf{R} + \boldsymbol{\Gamma})^\dagger \mathbf{f}$	$\sum_{l=1}^L l^4$	$\sum_{l=1}^L l^4$	-	-
8	$\mathbf{r}_l \leftarrow \boldsymbol{\gamma} - \mathbf{A} \mathbf{a}_l$	$b_{p2}^2 \alpha$	$2b_{p2}^2 \alpha$	-	-
Where $\alpha = \sum_{l=1}^L l$					

The mode filter is used to interpolate the coarse displacement estimates over the frame (labelled ‘B’ in Fig. 3.7). If the size (cardinality) $N_{\text{sample}} = |X_{\text{sample}}|$ of the set X_{sample} , then there are N_{sample}^2 additions used in binning the classifications. Then to determine the bin with majority, N_{sample}^2 comparisons are required. Therefore, approximately $2N_{\text{sample}}^2$ additions are required per pixel.

For the fine displacement estimation (labelled ‘D’ in Fig. 3.7), the complexity is presented with reference to an efficient implementation of an RLS filter such as exponential RLS using a Dichotomous Coordinate Decent (ERLS-DCD) described in [95]. A summary of the complexity using this algorithm is shown in Table 3.2.

The values for P_m and P_a are the number of multiplication and additions required to solve the normal equations respectively.

Table 3.2 Complexity associated with the fine displacement estimate [95]

Line	Equation	×	+
1	$\mathbf{R}(q) = \lambda \mathbf{R}(q-1) + \mathbf{i}_r(q)\mathbf{i}_r(q)^T$	$2M_{\text{fine}}^2$	M_{fine}^2
2	$y(q) = \mathbf{i}_r(q)\hat{\mathbf{h}}(q-1)$	M_{fine}	$M_{\text{fine}} - 1$
3	$e(q) = i_t(q) - y(q)$	-	1
4	$\mathbf{b}_0(q) = \lambda \mathbf{r}(q-1) + e(q)\mathbf{i}_r(q)$	$2M_{\text{fine}}$	M_{fine}
5	$\mathbf{R}(q)\Delta\mathbf{h}(q) = \mathbf{b}_0(q)$	P_m	P_a
6	$\hat{\mathbf{h}}(q) = \hat{\mathbf{h}}(q-1) + \Delta\hat{\mathbf{h}}(q)$	-	M_{fine}
Total: $(2M_{\text{fine}}^2 + 3M_{\text{fine}} + P_m)$ multiplies, $(M_{\text{fine}}^2 + 3M_{\text{fine}} + P_a)$ additions.			

In [95] a fast ERLS algorithm is described entitled ERLS-DCD. The ERLS-DCD algorithm reduces the complexity by optimising what needs to be calculated at each iteration and using a efficient Dichotomous Coordinate Decent (DCD) for solving the normal equations. Using a transversal structure and a value for λ that will allow the multiplication to be implemented as a binary shift, the complexity for the ERLS-DCD algorithm becomes $(2M_{\text{fine}} + pM_{\text{fine}}) + P_m$ multiplications and $(5M_{\text{fine}} + pM_{\text{fine}}) + P_a$ addition. These values are modified from the values in [95] because of the square ($p \times p$) aperture, and therefore at each pixel step p pixel are updated. Using the DCD algorithm to solve the normal equations removes the need for the multiplication giving $P_m = 0$. For additions $P_a \leq M_{\text{fine}}(2M_u + M_b - 1) + M_u$, where M_u is the number of line search iterations and M_b is the fixed bit width of the representation. For $M_u \gg M_b$, P_a can be approximated to the upper bound $2M_{\text{fine}}M_u$ [95]. Table 3.3 give a direct comparison with a classical RLS

Table 3.3 Complexity comparison for classical and advanced implementation of RLS algorithm [95]

Algorithm	×	+	÷
ERLS-DCD	$2M_{\text{fine}} + pM_{\text{fine}}$	$2M_{\text{fine}}M_u + 5M_{\text{fine}} + pM_{\text{fine}}$	-
RLS	$M_{\text{fine}}^2 + 5M_{\text{fine}} + 1$	$M_{\text{fine}}^2 + 3$	1

implementation.

The convolution kernel peak position estimation (labelled ‘E’ in Fig. 3.7), requires 6 additions and 2 divisions per pixel.

After the adaptive filter the estimate is no longer a coarse classification therefore a median filter is used to remove erroneous results from the final estimate (labelled ‘F’ in Fig. 3.7). For a square filter of size S^2 a histogram is constructed of the classification. A sorting algorithm such as Quicksort [98] can give a result in $\mathcal{O}(S^2 \log_2 S^2)$ operations on average.

This section details the complexity analysis for each section of the registration algorithm. The coarse displacement estimate is the most computationally intensive section, however, as the computation for each sample pixel is independent, then the computation can be performed in parallel. This would suit current computer hardware architectures. Next, the coarse displacement estimates are combined and the processing for the adaptive filter becomes a single thread. However, there is an efficient implementation for the RLS style adaptive filter making it a good candidate algorithm for the sub-pixel displacement estimation. For a particular dataset, the size of the measurement vector is fixed, and the maximum motion in the scene is estimated prior to the registration. This means that there are only two sections of the algorithm that may vary in the complexity. The first is the number of times that the adaptive filter is reset due to the pixel re-ordering. The second is the sort time for the final median filtering. The small number of areas with variation is useful when estimating hardware requirements and execution time.

3.4 Conclusion and discussion

This chapter details a proposed registration algorithm that builds upon the conclusions of Chapter 2. That is, that the adaptive filter performs best when the number of transitions over motion boundaries is minimised. Also, that the sparse promoting penalty was effective at discriminating motion at the boundaries.

The proposed algorithm is partitioned into two parts: There is a coarse estimator that produces an DM to a pixel-wide accuracy. This estimate is then further refined by using an adaptive filter to produce a fine DM estimate. Using an initial coarse estimator has the following benefits for the proposed algorithm:

- The areas of similar motion can be grouped together - the aim is to improve the conditions for the evolution of the adaptive filter.
- The estimation of the motion to a discrete grid allows the easy comparison for forward/backward DMs.
- The aperture for the adaptive filter can be reduced when it is repositioned to take account of the coarse displacement estimate.

From the analysis of the complexity it can be seen that there is an efficient implementation for many parts of the proposed algorithm. In addition, most of the algorithm is suited to a parallel architecture.

The result, a DM for the whole image, is a starting point for further processing. The most simple application is to create a ‘corrected’ image by interpolating pixels from the new position indicated by the DM, thus reversing the motion. A single image can be created by blending multiple ‘corrected’ images together on a pixel by pixel basis, thus improving the SNR and enhancing the detail. If the motion of a pixel is known (estimated) to a sub-pixel accuracy then a super resolution image can be produced. Another application is to use the DM to estimate the motion of the sonar sensor as proposed in Chapter 4. This information can be used for navigation, assuming that the sonar is attached to an underwater platform, as well as building a mosaic from the individual images, this is demonstrated in Chapter 5.

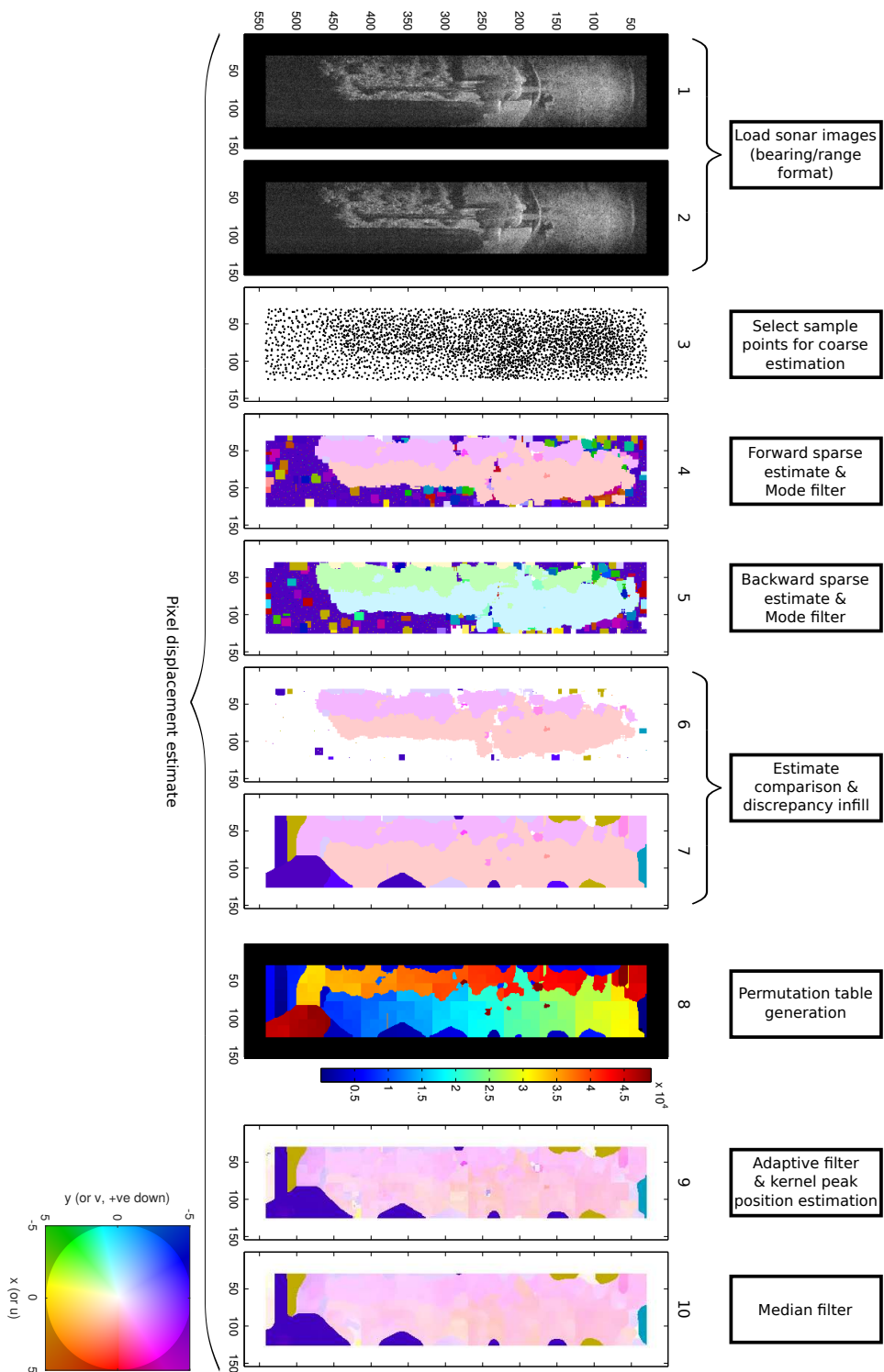


Fig. 3.8 Example outputs at selected stages of the proposed inter-frame DM estimation algorithm.

Chapter 4

Sensor motion estimation

The aim of this chapter is to use the estimated inter-frame DMs to deduce the motion of the sonar sensor. There are two aspects to this: firstly, an estimation needs to be made for the inter-frame motion. Secondly, the motion estimates for all of the frames need to be combined into an attitude-trajectory estimate for the sonar sensor. Once obtained, the estimated attitude-trajectory can be used for navigation or the construction of a single mosaic image which combines all of the frames.

Section 4.1 discusses other work on motion estimation and outlines the proposed method. Section 4.2 describes some preprocessing performed on the estimated DM. Sections 4.3 and 4.4 describe the sonar motion model used for the proposed motion estimation algorithm and how it is used to produce a motion estimate for the sonar sensor. Section 4.5 explains the proposed method for compiling a sequence of sensor motion estimates into an attitude-trajectory. Finally, Section 4.6 closes the chapter with some conclusions.

The work in this chapter is presented in the papers: B. Henson and Y. Zakharov, “Attitude-trajectory estimation for forward looking sonar based on acoustic image registration,” *IEEE Journal of Oceanic Engineering*, First submission May 2017, second submission Dec. 2017; B. Henson and Y. Zakharov, “Local optical-flow estimation for forward looking imaging sonar data,” in *MTS/IEEE OCEANS-Monterey*, 2016, pp. 1–8.

4.1 Introduction

The work in Chapter 3 details a proposed algorithm to estimate the motion between FLS frames. The work in this chapter aims to interpret this motion and create from it an estimate for the attitude and position of the sensor. In order to do this a model of the environment needs to be created. With reference to Fig 4.1, there are six possible degrees of freedom (DOF) of the sonar sensor, which are translations in axis x, y, z and rotations around each axis.

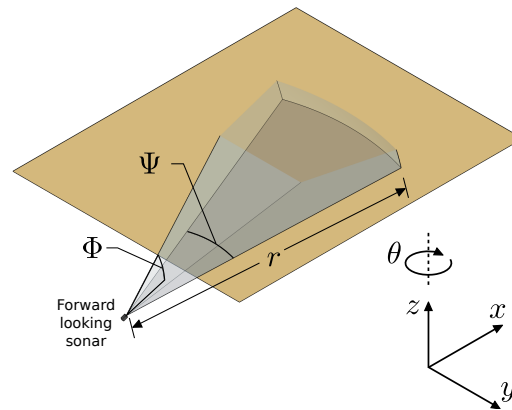


Fig. 4.1 An illustration of a typical field of view of a forward looking sonar. The sonar looks upon the surface being inspected, for instance the underside of a ship's hull.

It is possible for the sensor to move in all of the six DOF, however in practice the sensor is attached to a platform that is moving over a surface of interest, such as the seabed [10, 11]. Such an arrangement is illustrated in Fig. 4.1. Therefore, it is assumed that there is a zero roll angle (rotation around x), and a fixed pitch angle (rotation around y) [14, 10, 55]. It is also assumed that the altitude z is maintained, this may be with additional sensors as described in [10]. This leaves translations in x and y and a yaw angle (rotation around z). These assumptions were used in [14] where, in the registration step, the whole image was used to estimate a single motion for the image, as opposed to generation of a DM for individual pixels. This motion estimate was then used directly for the sensor motion, with appropriate scaling to move from pixels in the image to metres in a real world coordinate system. These assumptions may be good enough where the surface being inspected, whether it be a ship's hull or the seabed, is relatively flat and objects are small relative to the altitude. However, for more complicated scenes where there are prominent features a more sophisticated model would be required. Both parallax and 2-D projection changes create different, but related, motion for different parts of the image. To address this more complicated models have been developed in [20] where 3-D features were estimated through interpretation of the FLS image content. This 3-D height map was then fit to the measured motion giving an estimate of the position of the sensor in space. With an accurately modelled height map of the seabed this method would give a better registration, however, estimation from the image content is a difficult problem. For the work in this thesis the simple model is used with the assumption that the surface being inspected is flat. However, a more general model could be used in conjunction with the displacement map, this makes the proposed method more versatile than that described in [14]. Also because it is the motion of the individual pixels that is estimated more complex models such as those estimated in [20] could also be used.

An outline of the proposed attitude-trajectory algorithm is shown in Fig. 4.2. Based on the DM $\bar{d}(\xi)$ produced by the fine displacement estimator, an inter-frame motion vector α is estimated. Firstly, the preprocessing of $\bar{d}(\xi)$ is described, this selects an area X_B of the frame that contains most reliable information relevant to the dominant motion of the sensor. Then the motion model is introduced and the relationship between the motion parameters (α) and the DM is established. Then formulation of the motion estimation as a least squares (LS) minimization problem is described. This LS problem can be solved using standard optimization techniques, such as the interior-point method [99], which would however require a high computational load. A significantly more efficient method for real-time processing is firstly to transform the DM into a small vector of auxiliary statistics and to apply a dichotomous coordinate descent (DCD) search to match the statistics with a set of precomputed modeled ‘statistics’. These statistics are compared with a set of ‘statistics’ that are precomputed from a set of modelled sensor motions. The motion that generates the most similar DM statistics is taken to be the motion for that frame pair. The estimated inter-frame motions are then combined with a smoothing spline to produce the final attitude-trajectory for the sensor. Finally, a mosaic is generated by plotting all of the individual frames onto a single global grid.

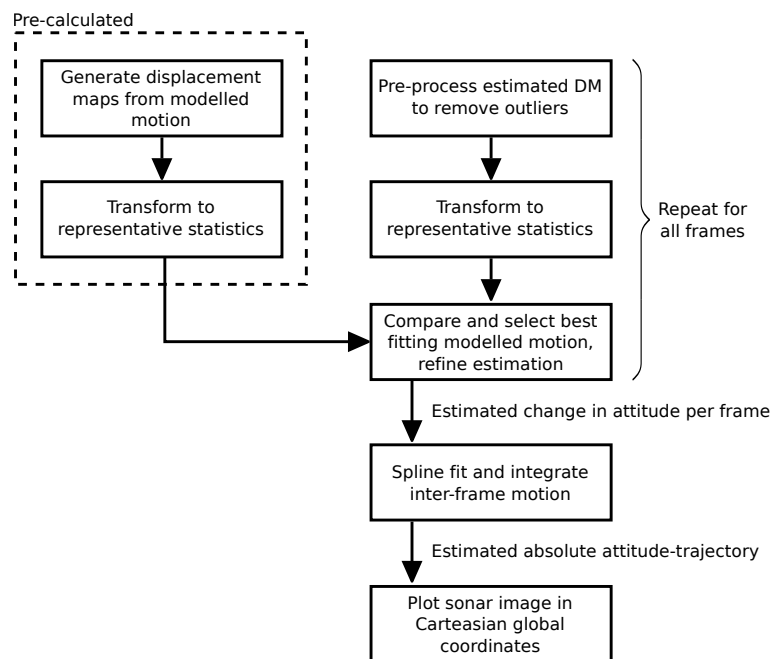


Fig. 4.2 A block diagram on the proposed algorithm for the attitude-trajectory estimation.

4.2 Preprocessing of the fine DM

The purpose of the preprocessing is to remove unreliable parts of the DM to reduce their influence on the final estimate. This is done in two steps. The first step is a thresholding of the reference frame to remove low intensity image parts with weak sonar returns. The threshold is based on computing a histogram of intensity in the reference frame and choosing a predefined quantile, for example 25%. The second step is based on processing the DM only at pixels kept after the first step. In this processing, a histogram for magnitudes of the displacements is computed, to which Tukey's rule [100] is applied to identify outliers. More specifically, if Q_1 and Q_3 are lower and upper quartile, respectively, then an outlier is a displacement outside the range: $[Q_1 - \gamma(Q_3 - Q_1), Q_3 + \gamma(Q_3 - Q_1)]$ for a nonnegative constant γ , typically $\gamma = 1.5$ [100]. The outlier pixels are also removed from further processing, thus finally identifying a reliable set X_B on the reference frame. An example X_B is shown (in white) in plot 11 of Fig. 4.5, this is based on the DM shown in plot 10.

4.3 Motion model

There are six possible degrees of motion of the sonar sensor, which are translations in axis x, y, z and rotations around each axis. We assume that the altitude z is predefined and is constant for the duration of the experiment. Also, the assumption is made that the platform roll (rotation around axis x) and pitch (rotation around axis y) are negligible (see Fig.4.1 for an illustration of the coordinate system). Therefore for estimation we consider only translations Δ_x and Δ_y in x and y and rotation Δ_θ around z : $\boldsymbol{\alpha} = [\Delta_x, \Delta_y, \Delta_\theta]$. These assumptions are specific for our numerical example. However, a more general case can be similarly considered. The projection onto the seabed is illustrated in Fig. 4.3. The displacement is described by a model

$$d_{\text{model}}(\xi, \boldsymbol{\alpha}) = [\xi^{(\psi)} - \xi_t(\boldsymbol{\alpha})^{(\psi)}, \xi^{(r)} - \xi_t(\boldsymbol{\alpha})^{(r)}], \quad (4.1)$$

where $\xi = [\xi^{(\psi)}, \xi^{(r)}]$ is a pixel position (beam angle and range, respectively) in the reference frame, $\xi \in X_B$, and $\xi_t(\boldsymbol{\alpha}) = [\xi_t(\boldsymbol{\alpha})^{(\psi)}, \xi_t(\boldsymbol{\alpha})^{(r)}]$ is the new pixel position (beam angle and range, respectively) after the modelled motion. The pixel position is transformed from polar coordinates in the reference frame to Cartesian coordinates $[u_x, u_y]$ on the seabed:

$$\begin{aligned} u_x &= \cos\left(\delta_\psi \xi^{(\psi)}\right) \sqrt{[\delta_r \xi^{(r)}]^2 - z^2}, \\ u_y &= \sin\left(\delta_\psi \xi^{(\psi)}\right) \sqrt{[\delta_r \xi^{(r)}]^2 - z^2}, \end{aligned} \quad (4.2)$$

where δ_r and δ_ψ are range and angle resolutions defining a single pixel and z is the predefined sensor altitude. The motion $\boldsymbol{\alpha}$ is transformed into a new position on the seabed:

$$\begin{aligned} v_x &= u_x \cos(\Delta\theta) + u_y \sin(\Delta\theta) - \Delta_x, \\ v_y &= -u_x \sin(\Delta\theta) + u_y \cos(\Delta\theta) - \Delta_y. \end{aligned} \quad (4.3)$$

Finally, the polar coordinates in the target frame for the new position on the seabed are given by

$$\begin{aligned} \xi_t^{(\psi)} &= \frac{1}{\delta_\psi} \left(\arctan \left(\frac{v_y}{v_x} \right) \right), \\ \xi_t^{(r)} &= \frac{1}{\delta_r} \left(\sqrt{v_x^2 + v_y^2 + z^2} \right). \end{aligned} \quad (4.4)$$

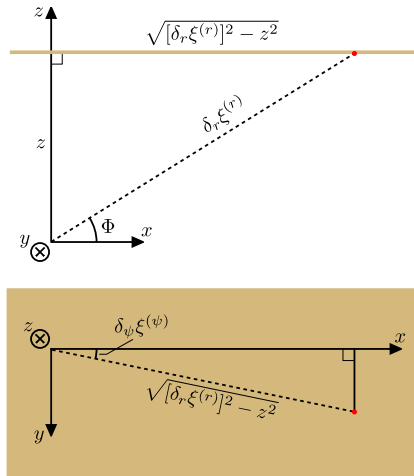


Fig. 4.3 Position of the point $[u_x, u_y]$ on the seabed.

4.4 Estimation of the motion vector

The motion can be estimated by solving the LS optimization problem:

$$\hat{\boldsymbol{\alpha}}_{LS} = \arg \min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}), \quad (4.5)$$

where the LS cost function is given by

$$J(\boldsymbol{\alpha}) = \sum_{\xi \in X_B} \|\bar{d}(\xi) - d_{\text{model}}(\xi)\|^2, \quad (4.6)$$

and $d_{\text{model}}(\xi)$ depends upon $\boldsymbol{\alpha}$ as described in Section 4.3. Obtaining the solution to (4.5) has high complexity because, for an iterative LS solver, a model DM must be regenerated multiple times using the complicated non-linear transforms (4.2) to (4.4) for every pixel $\xi \in X_B$. To reduce the complexity of the inter-frame motion estimator, the fine DM $\bar{d}(\xi)$, $\xi \in X_B$, is represented by a small number of statistics (12 in our example) as follows. Four histograms are computed from the DM:

$$\mathbf{G}_{\pm}^{(j)} = \text{hist}\{\bar{d}^{(j)}(\xi), \xi \in X_B^{\pm}\}, j = 1, 2, \quad (4.7)$$

where X_B^+ and X_B^- denote parts of X_B for positive and negative beam angles and $\bar{d}^{(j)}(\xi)$ denotes the j^{th} coordinate of vector position $\bar{d}(\xi)$. For each of the histograms, three quartiles are computed, thus twelve parameters in total; we denote all of these quartiles as a vector \mathbf{s} .

The vector \mathbf{s} is then compared with vectors $\mathbf{s}_{\text{model}}(\boldsymbol{\alpha})$ computed on a grid of motions $\boldsymbol{\alpha} \in T$ to find the best match, where T is a grid of sensor motions bounded by the dynamics of the platform. Specifically, the modelled DMs for all possible sensor motions $\boldsymbol{\alpha} \in T$ are pre-computed, and for each of them we store the twelve ‘statistics’ as was described above. The match is based on the minimization of the Euclidean distance

$$\hat{\boldsymbol{\alpha}}_s = \arg \min_{\boldsymbol{\alpha} \in T} \|\mathbf{s} - \mathbf{s}_{\text{model}}(\boldsymbol{\alpha})\|^2. \quad (4.8)$$

The solution of this minimization problem can be found using an exhaustive search over the grid. Our comparison of run time in Matlab found that the LS optimization in (4.5) was about 5 times more complex than the grid search for the minimisation of 4.8. In our numerical example, the cardinality of T is $|T| = 97 \times 97 \times 97 \approx 9 \times 10^5$ which results in approximately 11×10^6 addition and multiplication operations to find $\hat{\boldsymbol{\alpha}}_s$. This is still high for real-time processing. To further reduce the complexity, we propose to use the DCD search [95] on the grid T . With the DCD search, the complexity is $12 \log_2 |T| \approx 250$ multiplications and additions which is significantly lower than that of the other two techniques; about 4×10^4 times lower than using the exhaustive search. The last two techniques require a storage space for the $12 \times |T|$ statistics, which is a payment for faster computation. However, this is comparable to the storage for a single frame and therefore is not a significant overhead.

The result of the processing described above when applied to all frames in a sequence of P frames is a sequence of motion vectors $\{\boldsymbol{\alpha}_k\}_{k=1}^{P-1}$. For further processing we also retain the values $l_k = J_{LS}(\boldsymbol{\alpha}_k)$ or $l_k = J_s(\boldsymbol{\alpha}_k)$, depending on the estimation technique used. These will be transformed into a set of weights $\{w_k\}_{k=1}^{P-1}$ that characterize the accuracy of the inter-frame

motion estimation. In our example, we use the weights

$$w_k = \max \left\{ 0, 1 - \frac{\beta l_k}{\max_n l_n} \right\}, \quad (4.9)$$

where β is a positive constant set to $\beta = 0.9$. An example of the inter-frame registration weighting is shown in Fig. 4.4.

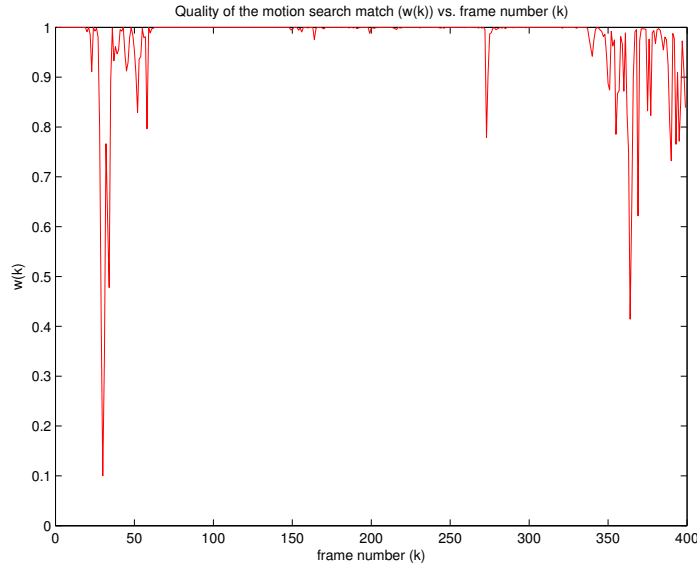


Fig. 4.4 An example of the inter-frame registration weighting (first 400 frames of the dam dataset - see section 5.2.2).

Fig. 3.8 showed an example registration pipeline to generate an inter-frame DM. Fig. 4.5 shows the match of a DM obtained from the motion model. Fig. 4.5 plot 10 shows the DM as estimated from the inter-frame registration $\bar{d}(\xi)$, plot 11 shows the displacement estimates selected for further processing X_B (selected estimates are shown in white). Plot 12 shows the DM from a modelled motion that best fits the estimated DM $\bar{d}(\xi)$, $\xi \in X_B$, the selection was made using the minimisation shown in (4.8) using a DCD search. Plot 13 shows a DM that is the difference between the estimated DM and the matched modelled DM. It can be seen that most discrepancies are less than a pixel width in magnitude.

4.5 Attitude-trajectory estimation

The aim is now to produce an estimate of the sonar sensor attitude and position at any time t within an experiment. More specifically, the incremental movements

$\alpha_k = [\Delta_x(t_k), \Delta_y(t_k), \Delta_\theta(t_k)]^T$, where $t_k = kT_{\text{frame}}$ need to be processed to produce positions

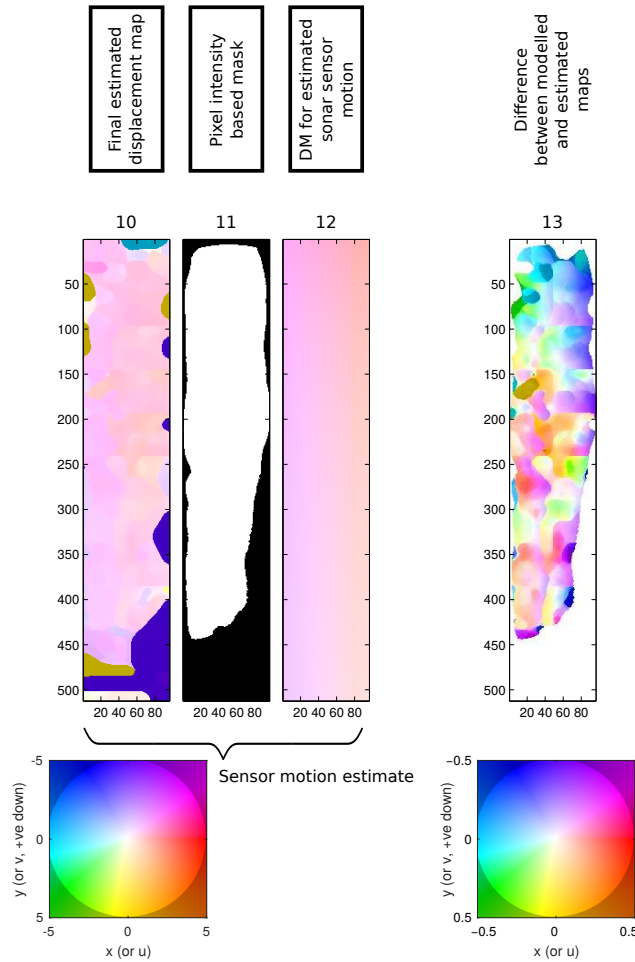


Fig. 4.5 An example match for the estimated DM and a modelled motion DM.

and attitudes $\mathbf{p}(t) = [x(t), y(t), \theta(t)]$ that are in a coordinate system fixed on the seabed. In order to create an attitude-trajectory the sequence of estimated inter-frame motions needs to be combined and converted into a position in 3-D space. It is desirable to smooth the resultant attitude-trajectory taking into account the relative quality of the motion estimates. If additional data for the trajectory and attitude were available then a popular method for data fusion is the Kalman filter [101] or Extended Kalman filter [102]. The Kalman filter is a method by which not only additional sensor data may be incorporated into the estimation, such as inertial navigation measurements or acoustic beacon localisation, but also a model of the motion dynamics of the sensor platform [31]. In this work only the relative inter-frame motion estimation is available and therefore regularised splines provide an efficient way to construct the estimated attitude-trajectory.

We represent $\theta(t)$ as a smoothed spline found as a trade off between an error in the fit to the data points and smoothness of the spline. One efficient way is to use P-splines [103].

4.5.1 P-splines

The spline is given by

$$\theta(t) = \sum_{m=1}^{N_b} c_m B(t - (m-1)\tau), \quad (4.10)$$

where $B(t)$ is a B-spline [104], N_b the number of basis functions, c_m are basis expansion coefficients, and in our example $\tau = T_{\text{frame}}/4$. The basis functions used in this work are cubic B-splines. The angular velocity $\theta'(t)$ is then given by

$$\theta'(t) = \sum_{m=1}^{N_b} c_m b(t - (m-1)\tau), \quad (4.11)$$

where $b(t) = B'(t)$. For finding the spline coefficients c_m , a weighted LS optimization problem with a penalty is formulated as follows. For P-splines, the smoothness penalty is efficiently calculated based on the difference in the values of the spline coefficients themselves [103, 105]. Therefore, the cost function takes the form

$$S = \sum_{k=1}^{P-1} w_k \left[\frac{1}{T_{\text{frame}}} \Delta\theta(t_k) - \theta'(t_k) \right]^2 + \mu \sum_{m=n+1}^{N_b} (\Delta^n c_m)^2, \quad (4.12)$$

where $\mu > 0$ is a regularization parameter, Δ^n is the n^{th} difference operator. In our example, we use $n = 2$, i.e. $\Delta^2 c_m = c_m - 2c_{m-1} + c_{m-2}$. From [103], minimization of S over the vector of expansion coefficients $\mathbf{c} = [c_1, \dots, c_{N_b}]^T$ is equivalent to solving the system of equations

$$(\mathbf{B}^T \mathbf{W} \mathbf{B} + \mu \mathbf{D}^T \mathbf{D}) \mathbf{c} = \mathbf{B}^T \mathbf{W} \Delta\theta, \quad (4.13)$$

where \mathbf{D} is the difference operator in matrix form and \mathbf{B} is a matrix with the basis functions $b(t_k - (m-1)\tau)$ as its $P-1$ length columns, \mathbf{W} is a diagonal matrix where the diagonal elements are the weights w_k , which are found from the quality of the modelled DM fit (see (4.9) and Fig. 4.4), and $\Delta\theta = [\Delta\theta(t_1), \dots, \Delta\theta(t_{P-1})]^T$.

4.5.2 Attitude-trajectory generation

Having obtained the spline $\theta(t)$ we are able now to rotate the incremental motions $\Delta_x(t_k)$, $\Delta_y(t_k)$ onto the seabed coordinate system,

$$\begin{aligned} \Delta_{xx}(t_k) &= \Delta_x(t_k) \cos[\theta(t_k)] + \Delta_y(t_k) \sin[\theta(t_k)], \\ \Delta_{yy}(t_k) &= -\Delta_x(t_k) \sin[\theta(t_k)] + \Delta_y(t_k) \cos[\theta(t_k)], \end{aligned} \quad (4.14)$$

where $\Delta_{xx}(t_k)$ and $\Delta_{yy}(t_k)$ are the incremental movements aligned to x and y axis of the seabed coordinates, respectively. The same spline smoothing procedure as described above is then applied to increments $\Delta_{xx}(t_k)$ and $\Delta_{yy}(t_k)$ to find the smoothed splines $x(t)$, and $y(t)$, respectively.

The dynamics of the system can be incorporated into the smoothing regularization, for instance the regularizing parameter μ can be increased until the attitude-trajectory conforms to a maximum acceleration which is known in advance from the expected behaviour of the system.

Fig. 4.6 shows two examples of attitude-trajectory for the 3 DOF for the dam inspection dataset (see Section 5.2.2). Each attitude-trajectory uses a different value for μ . The effect of this smoothing parameter can be seen on the maximum acceleration for each DOF. The value can therefore be adjusted to take account of the maximum expected acceleration for the platform. In Fig. 4.6a, $\mu = 10^{-5}$, giving maximum accelerations in x of 0.030 m/s^2 , y of 0.11 m/s^2 , and rotation around z of 0.015 rad/s^2 . In Fig. 4.6b, $\mu = 10^2$, giving maximum accelerations in x of 0.00096 m/s^2 , y of 0.0020 m/s^2 , and rotation around z of 0.00022 rad/s^2 . Fig. 4.7 show these smoothing parameters applied to the dam inspection dataset.

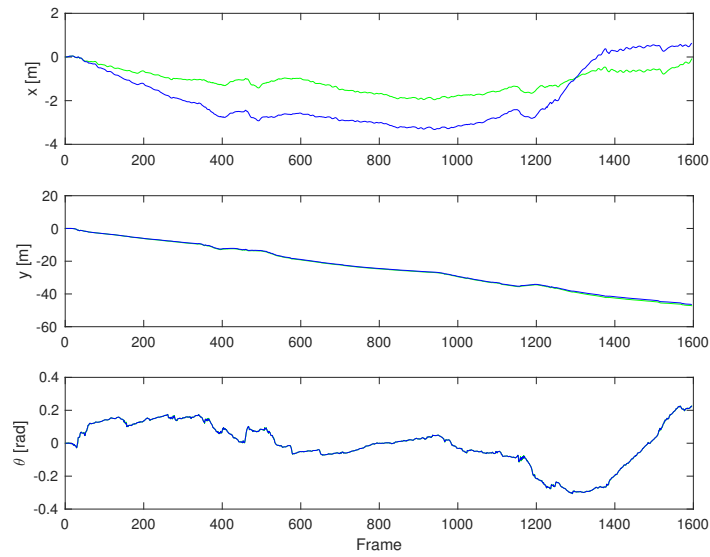
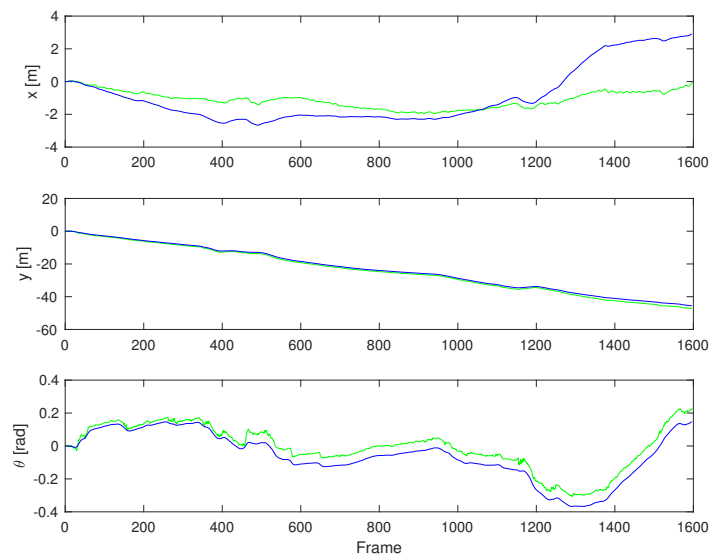
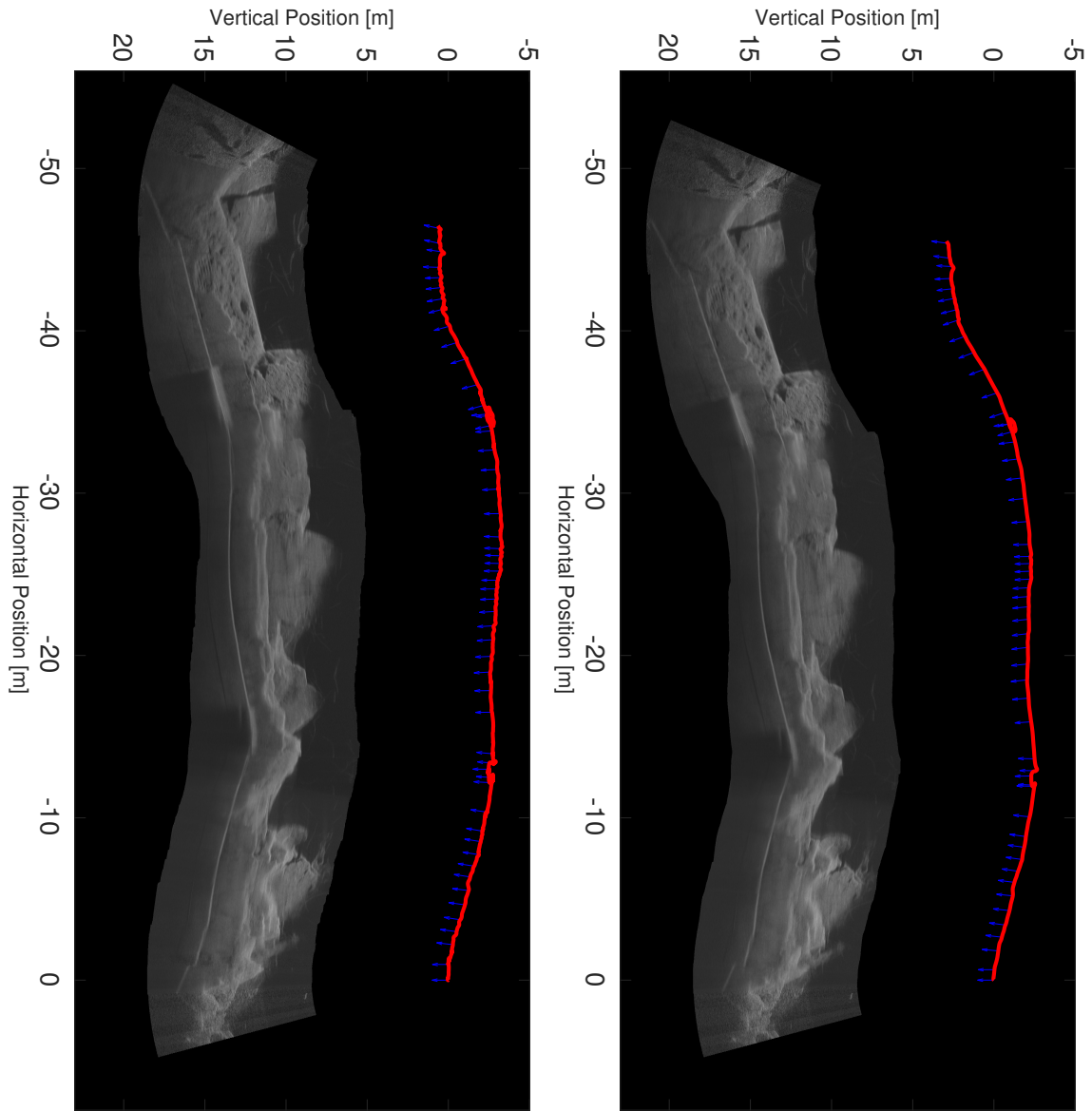
(a) $\mu = 10^{-5}$.(b) $\mu = 10^2$.

Fig. 4.6 Two examples of attitude-trajectory for the 3 DOF, unsmoothed is plotted in green, smoothed in blue (dam inspection dataset - Section 5.2.2).



(a) A mosaic showing a track along a dam wall where the motion is minimally smoothed.

(b) A mosaic showing a track along a dam wall where the motion is more heavily smoothed.

Fig. 4.7 The result for the dam wall attitude-trajectory estimation with the smoothed parameters $\mu = 10^{-5}$ and $\mu = 10^2$. The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

4.5.3 Mosaic generation

The mosaic is generated by projecting the FLS data onto a flat seabed in a reference frame relative to the sonar, then transforming the positions with respect to a global reference according to the estimated attitude-trajectory for the absolute position of the sonar.

The pixel position for the sonar frame is transformed from polar coordinates into Cartesian coordinates $[u_x, u_y]$ on the seabed, where the position of the sonar sensor is the local origin:

$$\begin{aligned} u_x &= \cos\left(\delta_\psi \xi^{(\psi)}\right) \sqrt{[\delta_r \xi^{(r)}]^2 - z^2}, \\ u_y &= \sin\left(\delta_\psi \xi^{(\psi)}\right) \sqrt{[\delta_r \xi^{(r)}]^2 - z^2}, \end{aligned} \quad (4.15)$$

where δ_r and δ_ψ are range and angle resolutions defining a single pixel and z is the predefined sensor altitude. Using the absolute attitude-trajectory (relative to a global reference) the image data can be transformed to a new position on a global relative to the origin for the mosaic.

$$\begin{aligned} m_x &= u_x \cos(\theta(t_k)) + m_y \sin(\theta(t_k)) - x(t_k), \\ m_y &= -u_x \sin(\theta(t_k)) + m_y \cos(\theta(t_k)) - y(t_k). \end{aligned} \quad (4.16)$$

A single frame aligned to the mosaic is then created by bi-linearly interpolating the sonar data onto a high resolution pixel grid in the coordinate system for the mosaic. Outside the field of view of the sonar the image is set to zero. In addition a binary image is created '1' in the sonar field of view '0' outside. The single frame is then blended with the mosaic, averaging all of the single frame values at each pixel location in the mosaic. To reduce the memory requirements, for each mosaic pixel location a running average and a total number of contributing frames are recorded. Hence when a new frame is added the binary image created is added to the contributing frames map and the mosaic regenerated with new averages. The running average, M_k , is found as follows [106],

$$M_k = M_{k-1} + \frac{x_k - M_{k-1}}{k}, \quad (4.17)$$

where k is the number of frames contributing to the mosaic pixel and the iterative algorithm is initialised $M_0 = x_0$. Fig. 4.8 shows an example of a single frame being blended with a mosaic. Fig. 4.8a is a single frame moved and interpolated onto the mosaic grid, Fig. 4.8b the map of the number of contributing frames for each pixel, and 4.8c the mosaic with the example frame blended into it.

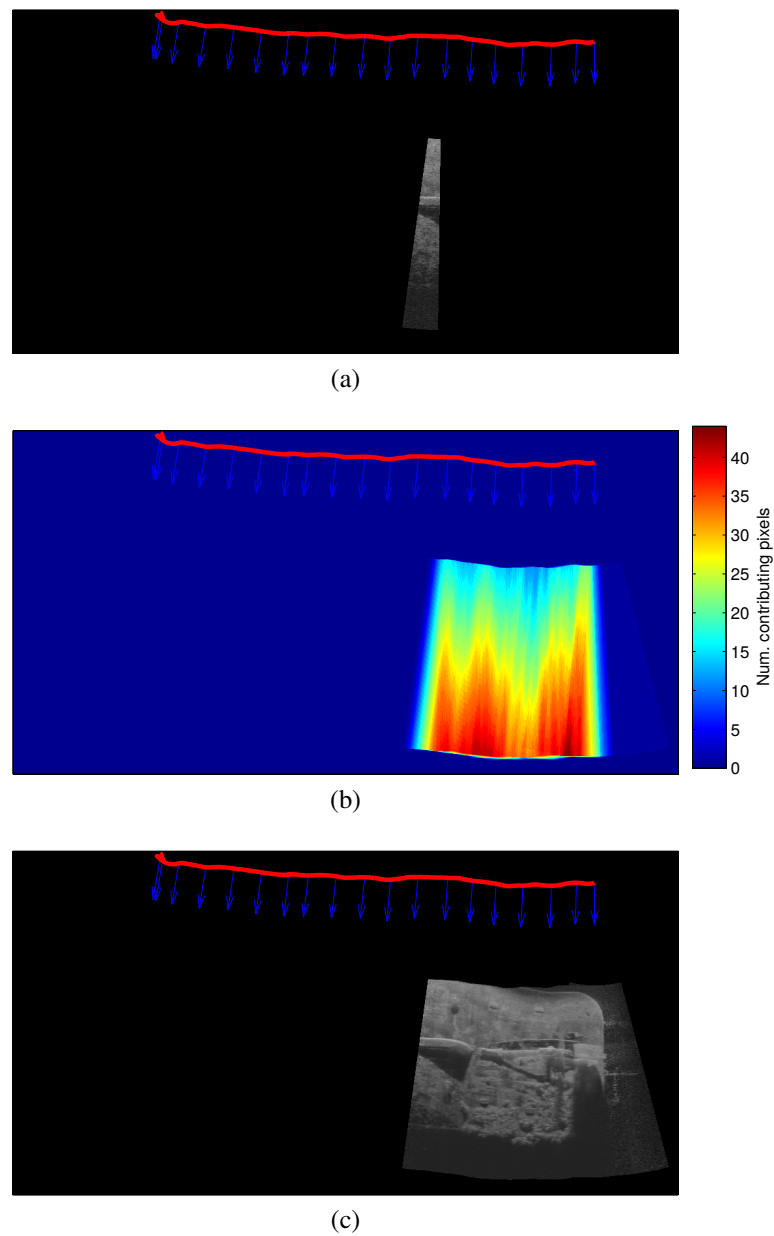


Fig. 4.8 (a) is a single frame moved and interpolated onto the mosaic grid. (b) the map of the number of contributing frames for each pixel. (c) the mosaic with the example frame blended into it.

4.6 Conclusion and discussion

This chapter describes the proposed sensor attitude-trajectory estimator. The estimator uses the DM generated from the proposed registration algorithm described in Chapter 3. After preprocessing the estimated inter-frame DM it is converted into a descriptor using order statistics. This descriptor is then compared with a library of descriptors that come from a set of DMs that are generated from models of various sensor motions. From a sequence of estimated DMs the inter-frame motion of the sonar sensor is estimated. A regularised spline is then employed to compile the sequence of motions into an attitude-trajectory. The use of a full frame DM allows more pixels to contribute to the sonar sensor motion estimation. Also, the DM is flexible in the range of motion models it can represent.

As mentioned, the attitude-trajectory algorithm begins with the pre-processing for the estimated DM. For a practical system a reliable way of automatically setting this level would need to be developed. The use of a descriptor to describe the DMs, reduced the complexity of the search and reduced the time it took to make an estimate. A problem with this method was that if after the pre-processing and the rejection of outliers, the coverage over the DM frame was poor then a good match was difficult to obtain. Further work in this area would be to develop a more robust DM descriptor. The regularised spline is an effective method for integrating the sensor motion estimates into an attitude-trajectory. This integration was able to take into account the quality of the DM match and incorporate other constraints such as the dynamics of the underwater platform. The use of B-splines meant that there is a small local support allowing local estimation of the odometry. In further work this local P-spline estimate could be used to extrapolate an estimate where the inter-frame DM motion match is poor. Also, extrapolated results could be fed back into the next inter-frame DM estimation to reduce the search space for the coarse pixel displacement. The sonar sensor motion considered in this chapter used a basic geometric model that used broad assumptions about the environment, such as the scene being observed is a flat plane and the propagation through the water column is free from distortions. A more sophisticated model of the acoustic propagation would give a more accurate modelled DM for a motion, and so a more accurate match for the estimated inter-frame DM. Modelling of underwater acoustic propagation is discussed further in Appendix B. The results of the estimation of the attitude-trajectory for three in field dataset are presented in Chapter 5.

Chapter 5

Experiment results

The aim of this chapter is to apply the proposed algorithms to a variety of different dataset types. The goal is to validate the results and observe the universality of the proposed algorithm. Section 5.1 shows the registration of optical images from the Middlebury dataset [77]. For each of the images from this dataset there is an accompanying ground truth DM, allowing a quantitative comparison to be made with the estimate from the proposed algorithm. Section 5.2 shows the results of the applying the proposed registration algorithm followed by the interpretation of the resultant DMs using the proposed motion estimation algorithm. Three FLS datasets were used: the first in a high quality short range view of a ship's hull. This dataset contained multiple passes over the same scene, allowing the consistency of the registration to be compared. In addition, this dataset contained clear linear features that can also be used to assess the quality of the registration. The second dataset used was a single pass along a dam wall. The dam is seen at longer range and the dataset shows interference from fast motions and fish obscuring the receiver. The final dataset is of a seabed scene featuring a pipeline and pile. This dataset differs from the first two as it was recorded using a different type of FLS, which exhibits a poorer SNR in the received image. These three FLS images also show that the proposed algorithms are applicable to different image types. In Section 5.3 the results are discussed and some conclusion drawn.

5.1 Middlebury dataset

The DM estimation part of the proposed algorithm was applied to the same selection of images from the Middlebury dataset [77] that were assessed in Section 2.3.5. The results can be seen in Fig. 5.1. The two main parameters for the registration algorithm, the number of possible displacements accounted for (M , see (3.1)) and the size of the measurement vector (N , see (3.2)), were the same for all of the experiments - $M = 121^2 = 14641$ and

$N = 13^2 = 169$. For the coarse estimates the sampling the images reduced the computation by 17 times, that is, from Section 3.1.3, $|X_{\text{sample}}| = 1/17|X|$. However, the part of the algorithm that selects the sample distribution based on the intensity in the image (Section 3.1.3) is disabled for optical images. The mode and median filters both had an aperture of 13×13 . For the adaptive filter section, the forgetting factor $\lambda = 0.998$. Figure 5.1 shows the colour scale for the adaptive filter registration results.

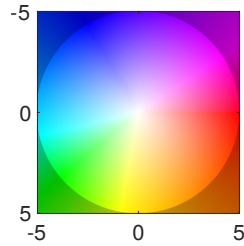
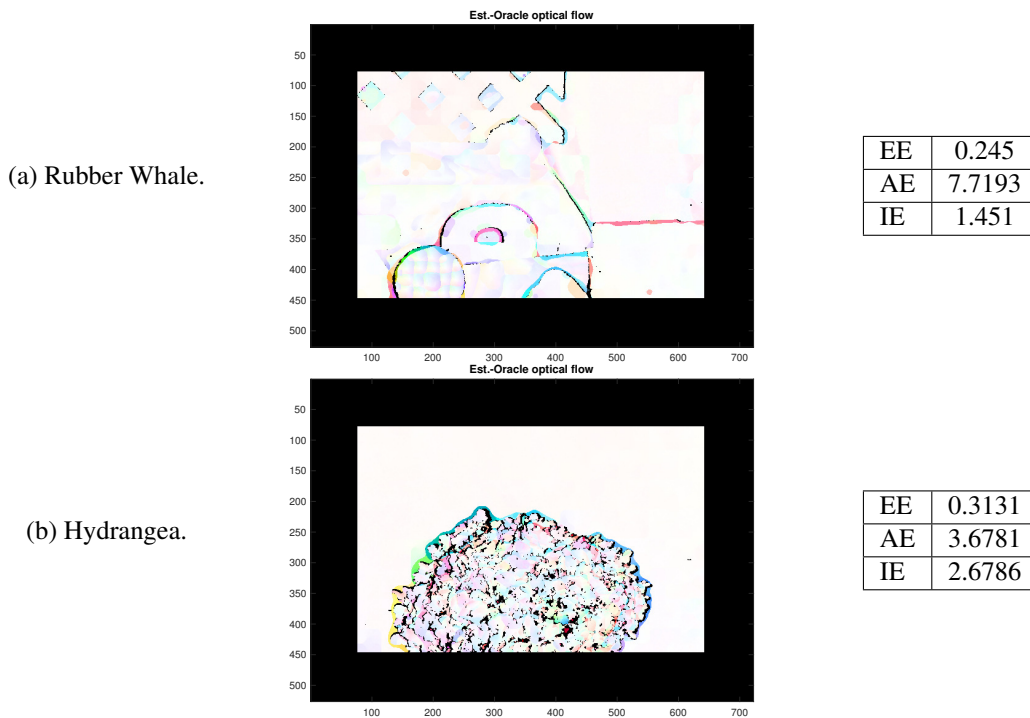


Fig. 5.1 Colour wheel key for the results of the coarse-fine registration.



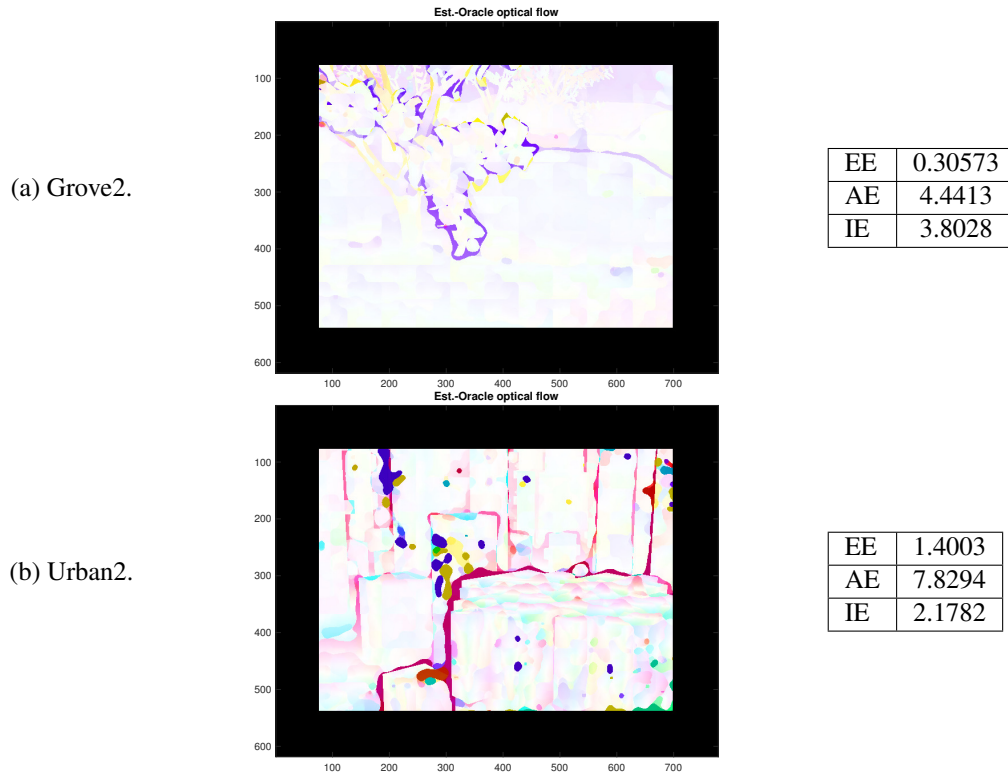


Fig. 5.1 The results from four experiments using selected images from the Middlebury dataset [77].

Comparing the results from those shown in Section 2.3.5, it can be seen from the images that the results are improved. The segmentation using the coarse estimate has allowed the adaptive filter to more accurately measure the motion within each segment (which often is the motion of an individual object). This is because the adaptive filter is able to evolve over a larger area of similar pixel displacements. The segmentation reduces the incidences where adaptive filter propagates a displacement estimate across a motion boundary. This leads to better definition of the motion boundaries in the DM. With the exception of the Urban2 image, the errors are less extreme for the final proposed algorithm. This improvement is due to the forward/backward registration comparison removing erroneous results from the DM. All of the numerical measures for the registration accuracy have improved, although this would not be a fair comparison for the Urban2 image because there is a large section where the registration using the initial algorithm fails completely.

Table 5.1 shows the results for the Hydrangea image pair with added noise. It can be seen that the registration remains at a high quality until test three where the error begins to increase. This improvement is due to the better rejection of poor pixel displacement estimates using the forward/backward registration comparison.

Table 5.1 A comparison of the RLS, RLS-Sparse filter and the final sonar registration algorithm with different amounts of added noise. The Hydrangea image was uses with a single scan path and 0 iterations.

Test image	SFC, RLS	SFC, RLS-sparse	Sonar reg. algorithm
EE			
No added noise	0.8871	0.6666	0.3131
Noise test 1	2.6782	1.5303	0.3467
Noise test 2	5.1588	4.1251	0.7578
Noise test 3	6.3069	6.1393	2.7183
AE			
No added noise	10.1126	8.3509	3.6781
Noise test 1	29.0911	15.8455	4.0269
Noise test 2	58.8554	43.1149	7.9653
Noise test 3	71.9067	64.7966	27.8156
IE			
No added noise	5.7291	4.6241	2.6786
Noise test 1	10.9635	9.4866	7.1455
Noise test 2	15.7257	14.7276	12.0161
Noise test 3	20.0736	19.8731	17.0731

5.2 Acoustic imaging datasets

The proposed registration algorithm was applied to three FLS datasets, and the resultant sequences of DMs were then used to estimate attitude-trajectories for each of the sonar sensors. Due to there not being a ground truth for the field data, a mosaic was produced for each dataset to asses the quality of the DM and attitude-trajectory estimation. Data from three different FLSes were considered, two with an acoustic lens design, the DIDSON 300 and ARIS 3000, both manufactured by Sound Metrics [48]. The other FLS is Gemini 720i produced by Tritech [49], this sonar uses beamforming applied to a hydrophone array. The sonar sensor trajectories are estimated from consecutive frames with all of the frames in the dataset considered. The mosaics were created without any additional data such as the inertial navigation data or a model for the motion dynamics of the platform. The attitude-trajectory can be generated continually as the frames arrive from the sonar. The attitude-trajectory approximation can be made using the estimated inter-frame motion from all of the frames or a sliding window can be used to calculate the local position of the sonar sensor. From the running attitude-trajectory estimate loop closure can be searched for where the path is close to crossing itself. A potential overlap can then calculated and the two candidate frames registered. If the registration is successful then the additional closed loop constraint can be added to the fit for the P-spline. This constraint would take the form of a motion estimate between two non-consecutive frames.

5.2.1 Ship's hull example dataset

The example dataset is from the inspection of a ship's hull. The data was recorded using a Bluefin Robotics Hovering Autonomous Underwater Vehicle (HAUV) [107] equipped with a DIDSON 300 FLS. The data was captured in a manner described in [11], where the trajectory of the surveying robot was a zig-zag path over the underside of the ship's hull as illustrated in Fig. 5.3. The coordinate system for the FLS viewing the ship's hull is shown in Fig. 5.4. The data was captured in a manner described in [11], where the trajectory of the surveying robot was a zig-zag path over the underside of the ship's hull as illustrated in Fig. 5.2. From a visual inspection of the data an approximate trajectory for the sonar sensor is shown in Fig. 5.3. Track 4 and track 5 were used for the evaluation resulting in the mosaics shown in Fig. 5.6 and Fig. 5.7, respectively. For this dataset, the number of possible displacements

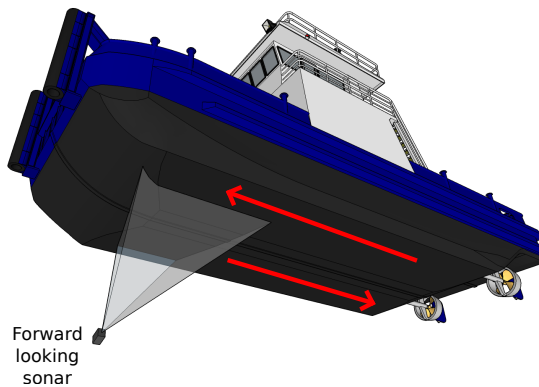


Fig. 5.2 An example application for a FLS is the inspection of the underside of a ship. An inspection robot with an FLS would follow multiple tracks along the ship's hull (illustrated here as red arrows) to ensure that whole surface is inspected.

accounted for $M = 41^2 = 1681$ and the size of the measurement vector $N = 13^2 = 169$. For the coarse estimates the sampling of the images reduced the computation by 17 times, that is, from Section 3.1.3, $|X_{\text{sample}}| = 1/17|X|$. The mode and median filters both had an aperture of 13×13 . For the adaptive filter section, the forgetting factor $\lambda = 0.998$.

Two mosaics have been produced, each from a separate track along the ship's hull. Fig. 5.6 shows a mosaic of 560 frames along the ship's hull, the robot moving predominantly in a positive y direction. Fig. 5.7 shows a mosaic of 550 frames with the robot moving predominantly in a negative y direction. For comparison, a single frame from the dataset is shown in Fig. 5.5. Using the sensor's estimated attitude-trajectory, the pixel location from each frame is projected onto the seabed with reference to a fixed Cartesian coordinate system. The pixel values for the seabed mosaic image are then interpolated from the frame pixels. In order to build the mosaic the FLS frames need to be blended together, ideally there needs to

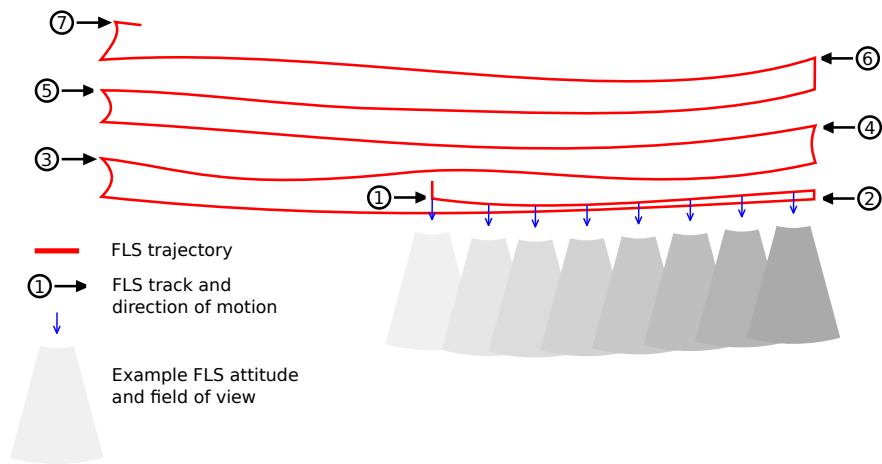


Fig. 5.3 Zig-zag path of the inspection robot. The FLS path is shown as a red line and the attitude as a blue arrow. The FLS footprints through time are shown in grey scale.

be enough frames to improve the SNR and enhance the image, however, not so many that small discrepancies either from errors in the attitude-trajectory estimation or from a change in viewpoint distort the image [108]. To that end for this dataset, only the centre 24 beams are plotted, apart from the first and last frames where all of the beams are plotted. Where the frames overlap the pixel intensities are averaged over the number of frames to produce the final mosaic. The resolution of the ship’s hull mosaic is 512 pixels per metre. It can be seen that using the estimated attitude-trajectory for the sonar sensor a coherent mosaic can be produced with an improved SNR over the image. The linear features, which are cooling pipes on the ship’s hull, are mostly straight before the hull changes with the rear section reduced before propeller. The small details, such as the sacrificial anodes, show good registration with the detail preserved.

The two mosaics produced are consistent in that all of the features are approximately the same size. These observations give confidence as to the accuracy of the attitude-trajectory estimation.

Fig. 5.8 illustrates the performance of the registration algorithm around the region of shadow caused by the keel. Fig. 5.8a, Fig. 5.8b and Fig. 5.8c show the coarse forward, backward and complementing forward/backward registrations, respectively. The poor result from the shadow are rejected in the forward/backward comparison. Fig. 5.8d shows the reference image and Fig. 5.8e shows the reference image with the positions of the forward/backward registrations overlaid. The unreliable results in the shadow region having been rejected are not part of the platform motion estimation.

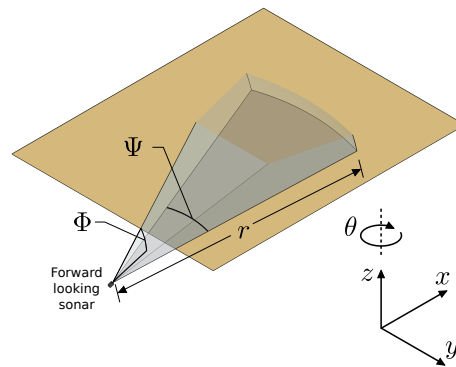


Fig. 5.4 The coordinate system for the ship's hull dataset.

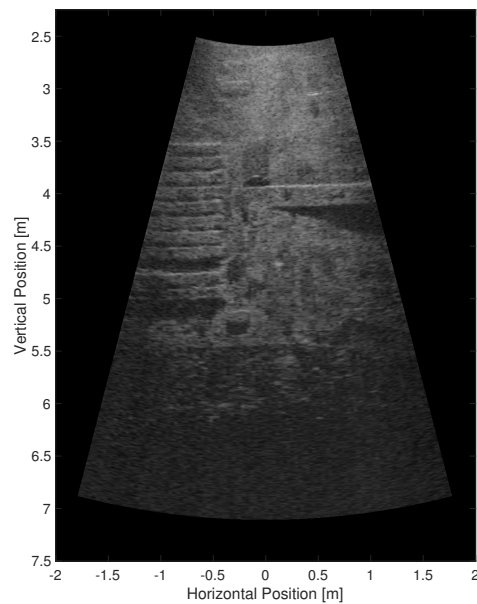


Fig. 5.5 A single frame from the ship's hull dataset, shown in Cartesian coordinates.

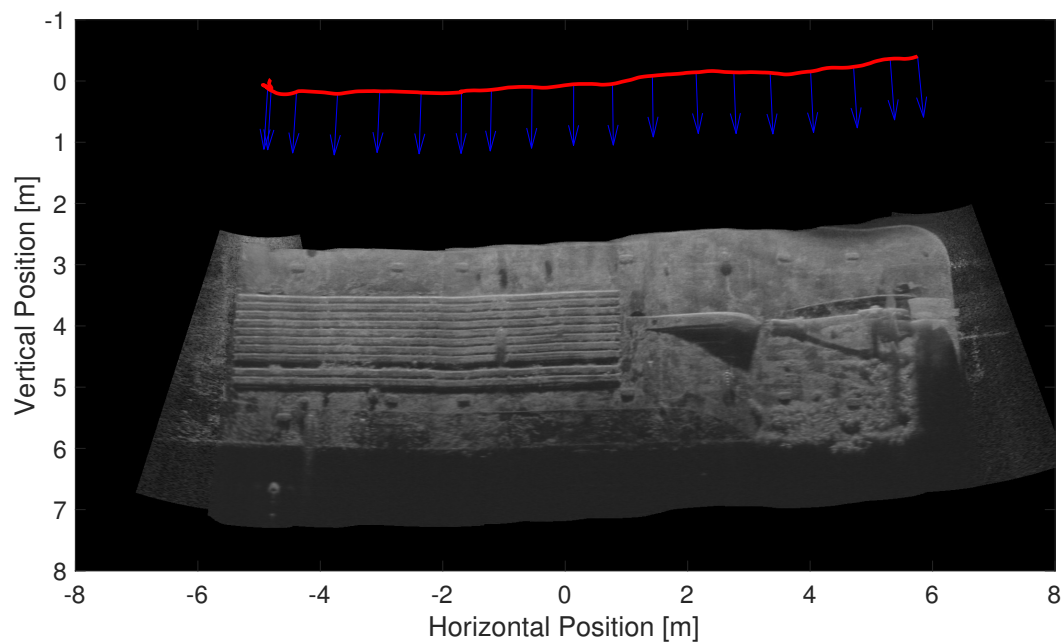


Fig. 5.6 A mosaic of 560 frames showing a track along the ship's hull (track motion is in positive y direction). The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

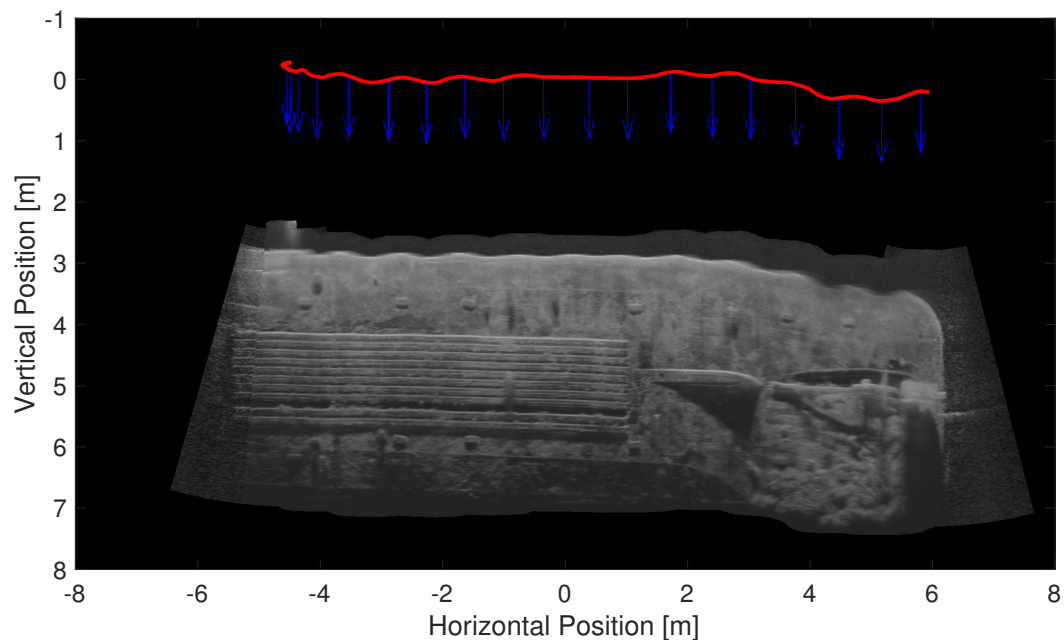


Fig. 5.7 A mosaic of 550 frames showing a second track along the ship's hull (track motion is in negative y direction). The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

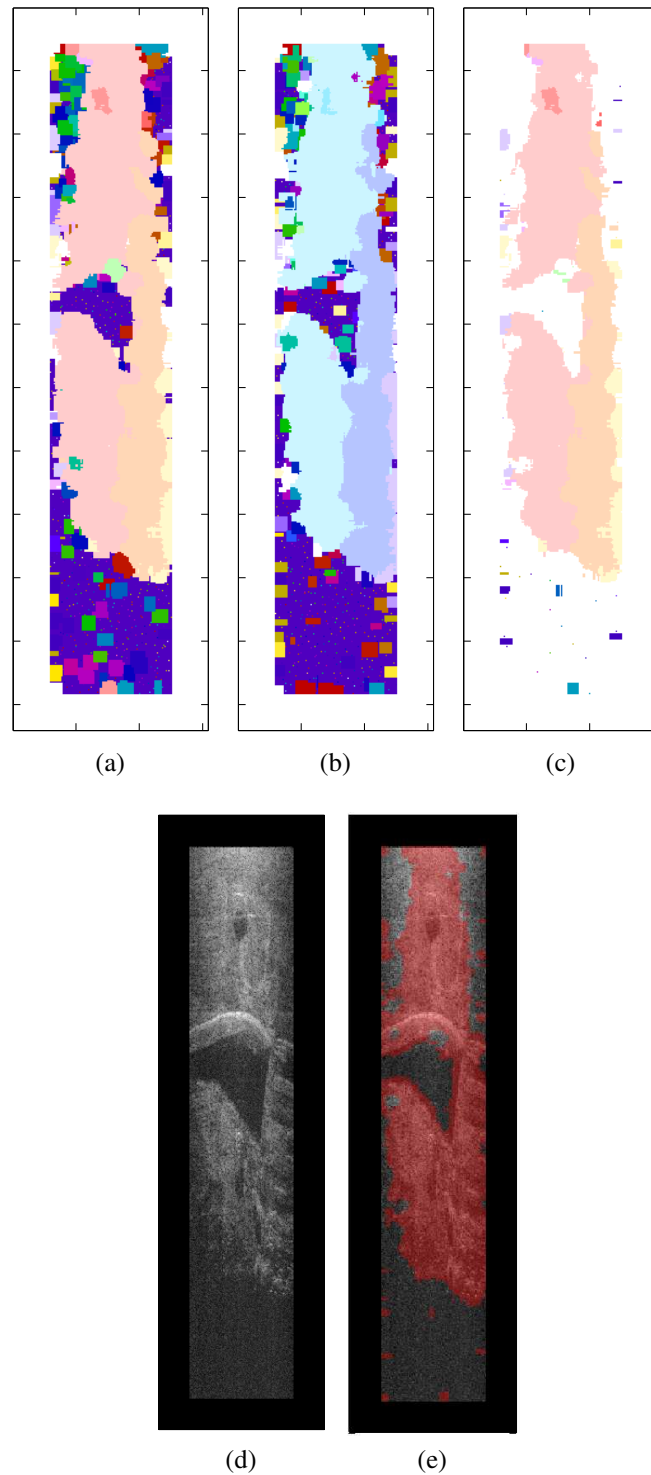


Fig. 5.8 (a), (b) and (c) show the coarse forward, backward and complementing forward/backward registrations, respectively. (d) shows the reference image and (e) shows the reference image with the positions of the forward/backward registrations overlaid. It can be seen that the unreliable results in the shadow region is have been rejected and therefore are not part of the platform motion estimation.

5.2.2 Dam inspection example dataset

The example dataset is from the inspection of a dam wall, [32, 33]. Fig. 5.13a shows a mosaic of 1596 frames (ARIS 3000 sonar [48]) from a single track along the dam wall moving predominantly in the negative y direction. The mosaic is an aerial view of the dam wall. Fig. 5.10 shows a photograph of the dam wall after the dam had been drained. The approximate position of the camera relative to the mosaic is shown in Fig. 5.13b.

The coordinate system for the FLS viewing the dam is shown in Fig. 5.9 and an example single image is shown in Fig. 5.11.

For this dataset, the number of possible displacements accounted for $M = 61^2 = 3721$ and the size of the measurement vector $N = 13^2 = 169$. For the coarse estimates the sampling of the images reduced the computation by 17 times, that is, from Section 3.1.3, $|X_{\text{sample}}| = 1/17|X|$. The mode and median filters both had an aperture of 13×13 . For the adaptive filter section, the forgetting factor $\lambda = 0.998$.

Again, only the centre 24 beams are plotted, apart from the first and last frames where all of the beams are plotted. Where the frames overlap the pixel intensities are averaged over the number of frames to produce the final mosaic. The resolution of the dam wall mosaic is 512 pixels per metre.

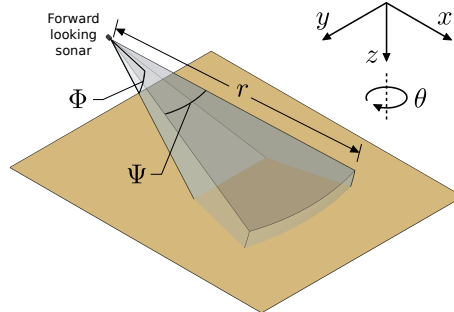


Fig. 5.9 The coordinate system for the dam inspection dataset.



Fig. 5.10 A photograph showing the dam wall after the dam had been drained, copied from [33]. Exposed re-enforcing bar can be seen in the foreground.

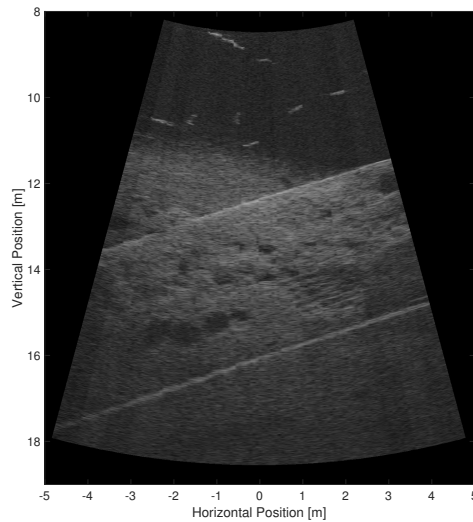


Fig. 5.11 A single frame from the dam inspection dataset, shown in Cartesian coordinates. Exposed re-enforcing bars are shown faintly on the right of the image. Also, a number of fish can be seen at the top of the image, these cast an acoustic shadow over the scene.

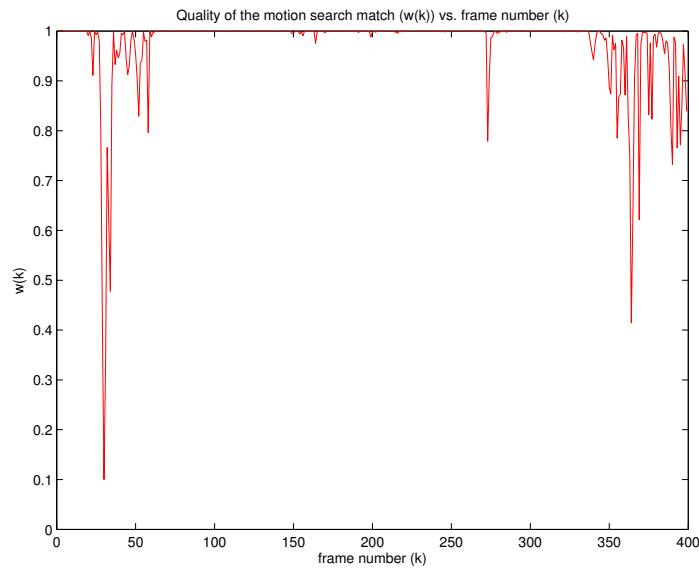
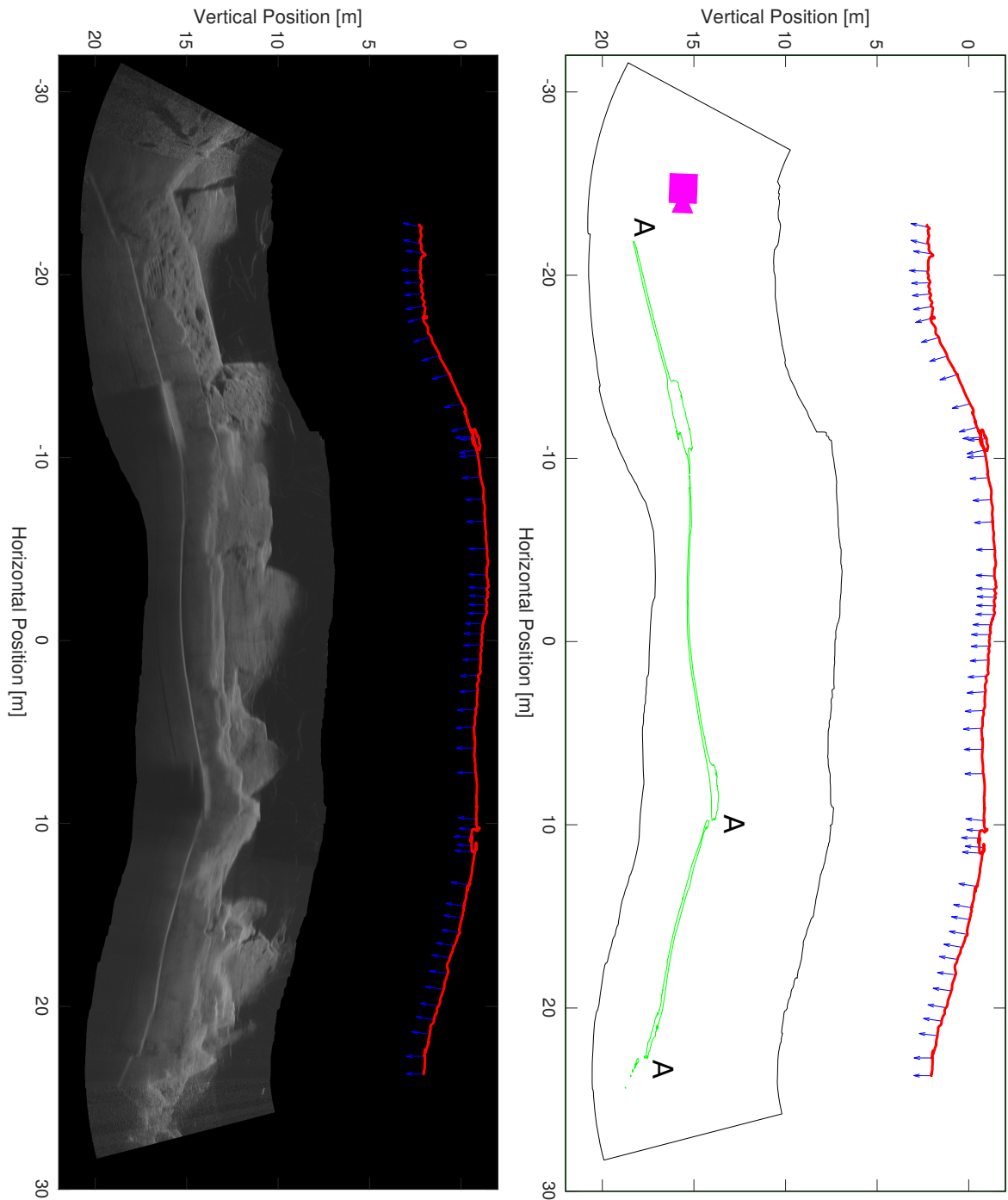


Fig. 5.12 Inter-frame registration weighting for the dam inspection dataset.

It can be seen that using the estimated attitude-trajectory for the sonar sensor a coherent mosaic can be produced, with enhanced details in sections of the image showing good registration. The bright line at the bottom of the mosaic is the dam wall and this should be straight. This linear feature highlights problems with the registration, the sharp discontinuities (marked with ‘A’ in Fig. 5.13b) are where the quality of the sonar frames is poor. This can be seen in Fig. 5.12 which shows the quality of each sensor motion estimate (w_k , see (4.9)). This poor quality image may be due to rapid motion of the sonar sensor causing the image to break up, or a fish in front of the receiver array blocking the view. Another problem shown by the straight feature of the wall is the curve in the centre of the mosaic. There is a high density of frames in this section of the sequence and the curve shows the build up of errors in the attitude-trajectory estimation.

The result from this dataset shows that there are improvements that can be made to the attitude-trajectory estimation in the rejection of poor results. The result also shows that details can be enhanced when the registration is good. Also, the proposed algorithm is robust enough to remove the effects of shadows from fish except in the more extreme examples.



(a) A mosaic of 1596 frames showing a track along a dam wall (track motion is predominantly in negative y direction).

(b) Showing the outline of the mosaic (black) and the position of the dam wall (green). Also, the approx. camera position for the photograph in Fig. 5.10, is shown in magenta.

Fig. 5.13 The result for the dam wall attitude-trajectory estimation. The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

5.2.3 Pipeline and pile dataset

This example dataset is of a motion over the sea floor, the mosaic shows a pipeline and pile. Fig. 5.17a shows a mosaic of 398 frames (Tritech Gemini 720i [49]) from a single track along the sea bed with the inspection robot moving predominantly in the positive x direction.

The coordinate system for the FLS viewing the sea floor is shown in Fig. 5.14 and an example single image is shown in Fig. 5.15.

For this dataset, the number of possible displacements accounted for $M = 121^2 = 14641$, this large search area was needed to accommodate rapid rotations that appear in the dataset. The size of the measurement vector $N = 13^2 = 169$. For the coarse estimates the sampling of the images reduced the computation by 17 times, that is, from Section 3.1.3, $|X_{\text{sample}}| = 1/17|X|$. The mode and median filters both had an aperture of 13×13 . For the adaptive filter section, the forgetting factor $\lambda = 0.998$.

In this mosaic all of 256 beams in each frame are plotted. Where the frames overlap the pixel intensities are averaged over the number of frames to produce the final mosaic. The resolution of the sea floor mosaic is 512 pixels per metre.

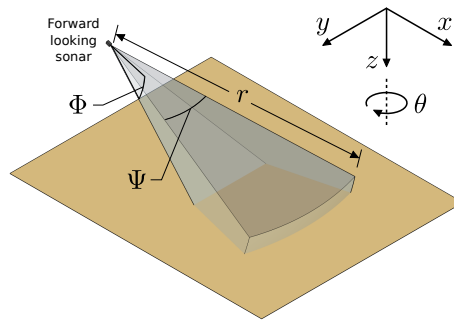


Fig. 5.14 The coordinate system for the pipeline and pile dataset

The estimated attitude-trajectory shows the inspection robot moving forward (in the positive x direction) and rotating back and forth three times. With multiple frames covering the same area a small error in the motion estimation can impact the quality of the final mosaic, blurring the final image. The quality of each sensor motion estimate as represented by w_k is shown in Fig. 5.16. However, it can be seen that the pile has a coherent image (see Fig. 5.17b), and therefore the attitude-trajectory estimation has accumulated a relatively small error for 398 frames. The start point and end point for the estimated attitude-trajectory can be seen more clearly in Fig. 5.18 which shows the 1st and 398th frames blended into a single image. It can be seen that after 398 frames the estimate for the position of the frames in space is close enough that the first and last frames could be registered together.

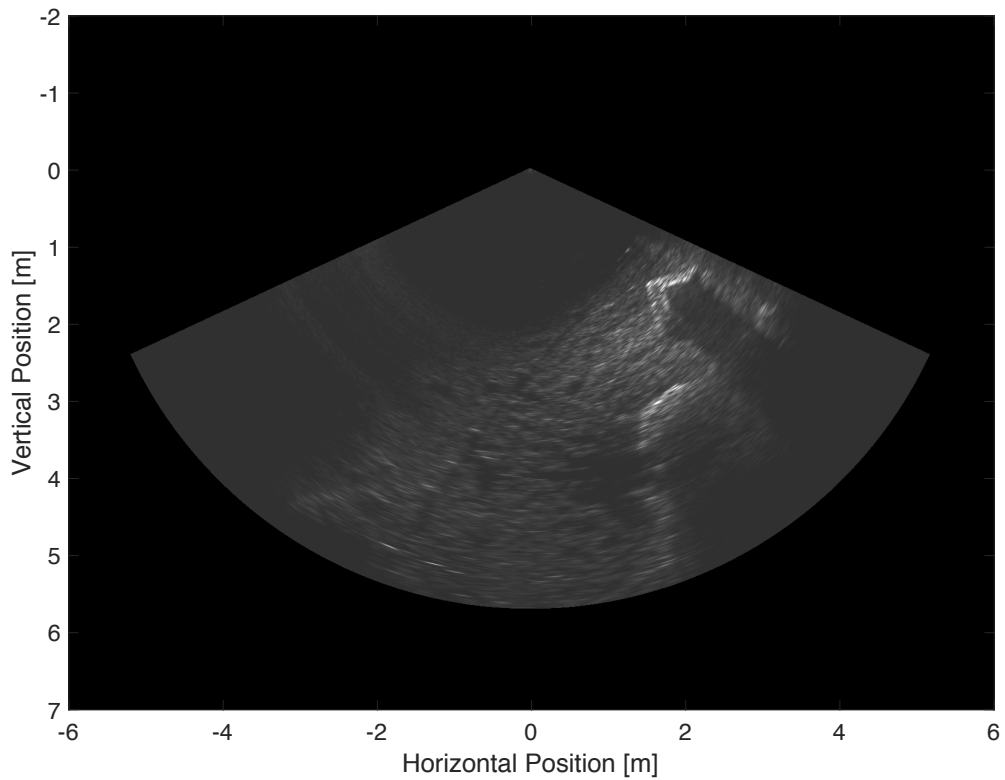
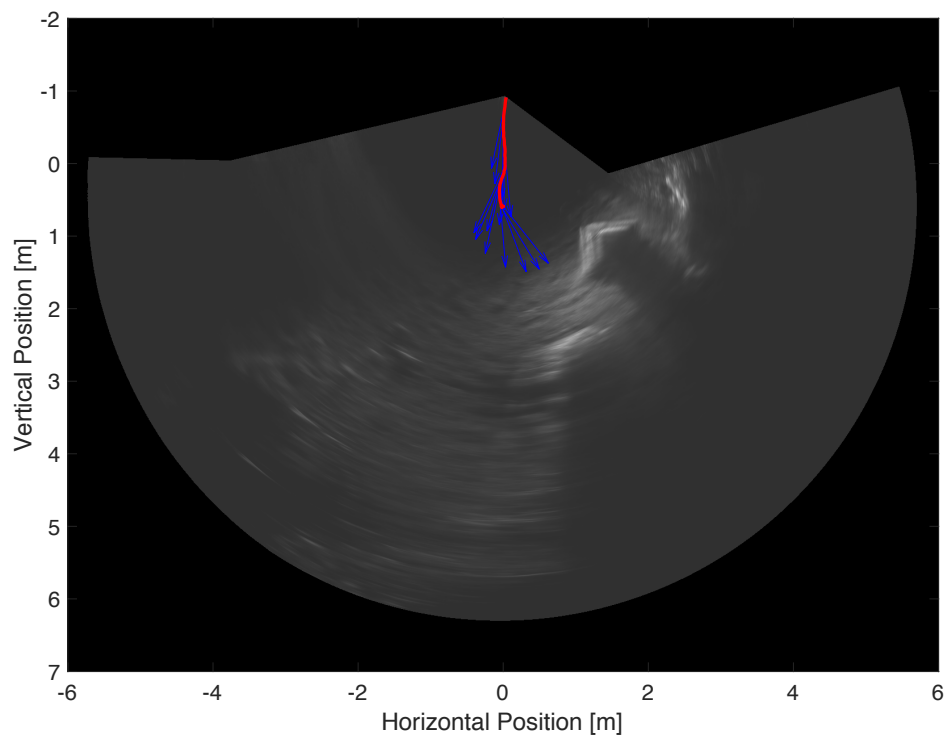


Fig. 5.15 A single frame from the pipeline and pile dataset.

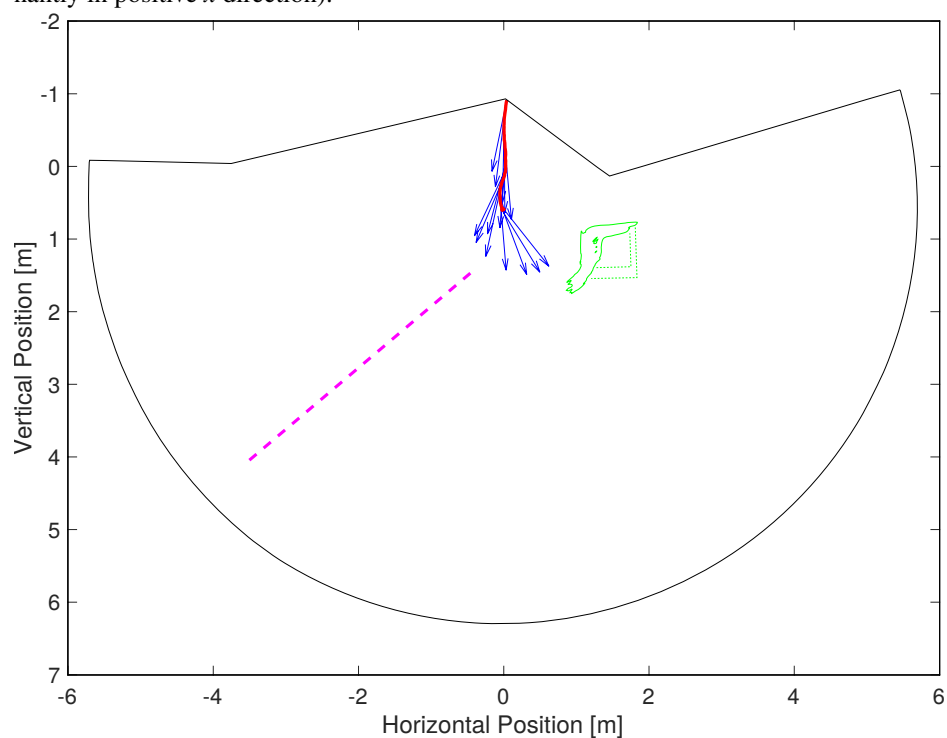


Fig. 5.16 Inter-frame registration weighting for the pipeline and pile dataset.

Thus the proposed algorithm shows reasonable performance even with FLS images with a relatively poor SNR compared to other FLS devices.



(a) A mosaic of 398 frames showing a track along the sea floor (track motion is predominantly in positive x direction).



(b) Showing the outline of the mosaic (black) and the position of the pile (green). Also, the approx. location of a pipeline on the seabed is shown (magenta)

Fig. 5.17 The result for the sea floor attitude-trajectory estimation. The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

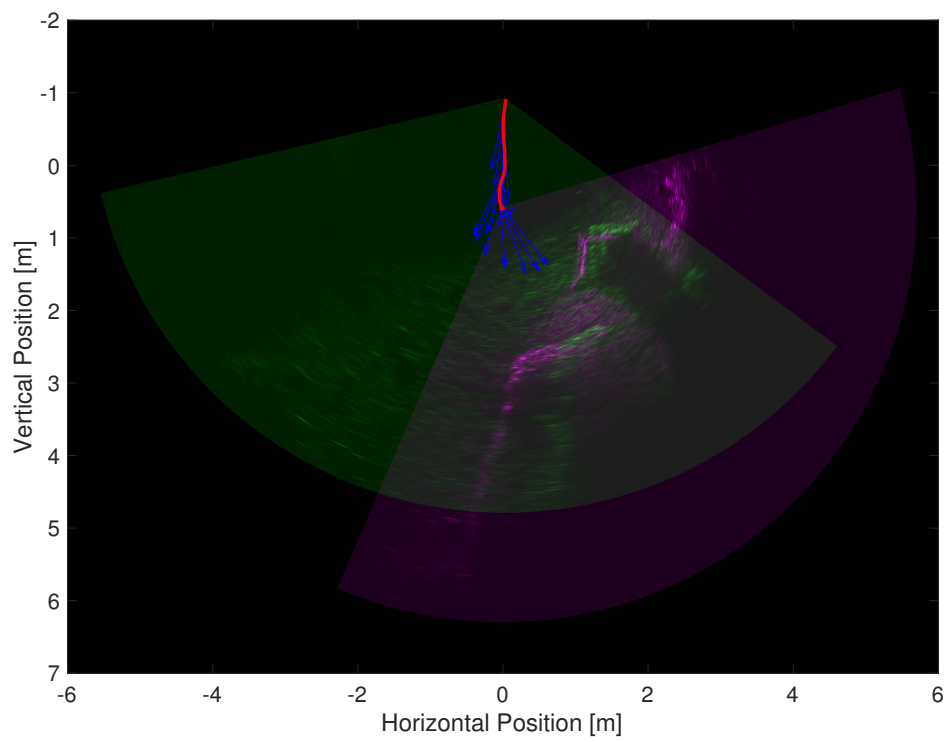


Fig. 5.18 A blended image of the 1st (green) and 398th (magenta) frames.

5.3 Conclusion and discussion

The results for the registration of the selected images from the Middlebury dataset show a large improvement over the results presented in Section 2.3.5. The motion boundaries are better defined, this is particularly notable in the images with more detailed motion. However, it can be seen that in areas where there are revealed or occluded pixels the algorithm has more difficulties. This is particularly apparent where there are large motions showing large errors at the boundaries. These areas are however often identified by the forward/backward registration, and the results could be improved with more sophisticated interpolation taking greater account of the direction of the surrounding DM estimates. Within the motion boundaries, the segmentation from the coarse estimation allowed the adaptive filter to evolve over a smoother motion area thus improving the sub-pixel DM estimation.

From application of the proposed algorithm to FLS datasets it can be seen that both the ship's hull and the dam inspection datasets show good detail in the mosaics. More critically, the scale of the scenes are reasonable and where the registration is good the image is enhanced. Considering the pipeline and pile dataset where the SNR is relatively poor, it can be seen that even though there is drift in the estimated attitude-trajectory over the 398 frames, the overlapping portion of the mosaic is close enough to identify that a registration can be made from the first to the last frame. This would allow for loop closure and the resultant bounding on the errors in the estimated attitude-trajectory [14]. The mosaics show good coherence for the estimated attitude-trajectories when considered in the context of local odometry, where a simple sequence of inter-frame DMs are considered and all of the frames are included.

Chapter 6

Conclusion and further work

6.1 Conclusion

The work in this thesis addresses some of the problems associated with navigation underwater. More specifically, the use of images from a FLS to estimate the motion of the sonar sensor itself. Research has been done into image registration algorithms and what aspects are important for the registration of FLS images. The aim of this research has been to produce a set of inter-frame DMs from a sequence of FLS frames. From each estimated DM a motion for the sonar sensor is estimated. These motions are then integrated into an estimate of the attitude-trajectory for the sonar sensor.

This research has been based on a simple geometric motion model. However, one way to improve the results would be to use a more accurate acoustic propagation model. To work towards this improvement research has been undertaken to develop further the Waymark underwater channel model [109]. The work allowed the processing of the signal to be carried out in the baseband, therefore in turn allowing the processing to be performed at a lower sample rate, after which the signal is restored to the passband. This work is pertinent to the FLS navigation, this is because a more detailed model on the acoustic wave propagation can give a better estimate of the sensor motion based on the DM.

The contribution from the work presented in this thesis is as follows:

1. Inter-frame DM estimator - An algorithm has been proposed that uses an adaptive filter to estimate the inter-frame motion of each pixel in the FLS frame. The proposed algorithm is further improved by the addition of a coarse pixel displacement estimator that employs a sparse recovery technique. This two step process of a coarse and fine estimation allows large motion to be accommodated whilst still estimating the DM for the whole image to a sub-pixel accuracy. Due to the flexibility of this approach the pixel motion estimation can be made in the native polar format of the FLS data, removing

the need for interpolation to a Cartesian coordinate system. Another advantage of generating a full frame DM is that complicated motions for objects within the scene can be represented.

2. Attitude-trajectory estimator for the sonar sensor - An efficient method for the comparison of estimated and modelled DMs is proposed. A fast search and matching technique has been proposed through reducing the DMs dimensions using order statistics, and the application of a dichotomous coordinate descent algorithm. Regularised splines have been exploited to compile estimated inter-frame sensor motion into a smoothed attitude-trajectory for the sensor. Regularised splines also allow additional factors to be incorporated into the motion estimation, such as the quality of the registration and the dynamics of the underwater platform. In addition, the localised nature of the B-spline support means that local odometry is possible.
3. Validation on in field datasets - The proposed algorithm has been validated through the application to in field datasets where mosaics of the scene have been produced. These show that a coherent attitude-trajectory for the sonar sensor can be produced, with appropriate scaling and enhanced detail. They also show that the proposed algorithm is appropriate for a range of image types.
4. Underwater channel model - A baseband Waymark acoustic channel model has been developed that allows greater efficiency in processing of a virtual signal transmission between a moving transmitter/receiver. This faster variant of the channel model was developed to allow processing of a signal in the baseband, facilitating a lower sample rate, thus taking advantage of significant computational savings. This makes the Waymark model a more usable tool for research into underwater signal propagation and signal processing. This is demonstrated in [7] where efficient modelling of the underwater acoustic channels for the transmission of communication signals formed an essential part of the research. The Waymark model can also form the basis for efficiently generating accurate motion models for the navigation algorithm described in this thesis.

6.2 Further research

This sections covers some aspects that would be the direction for future work and is further to the development and improvements that are in the discussion section at the end of each chapter. Further work would initially focus on validation of the registration and attitude-trajectory

estimation using more quantitative methods. Sonar data with a motion ground truth that is accurate enough for a meaningful assessment is unavailable, therefore, a sonar simulation would be a practical means of generating a dataset. Simulators for FLSs have been created with different trade off between accuracy [110] and speed [111], but in each example the motion can be defined to a high accuracy. In terms of the registration algorithm further work would fall into three areas: Firstly, with the integration of an acoustic propagation simulator into the sonar attitude-trajectory estimation. A more detailed acoustic propagation model can produce more accurately modelled DMs. This in turn allows the sonar sensor motion to be estimated with greater accuracy because the DM matching would be more discriminating. As an initial step the channel model will be generated using a known environment, such as a ship's hull or a man-made structure like a dam wall. A detailed simulator would then be used to model the channel at grid points over the scene. The DMs for the motion would then be generated by interpolating between them. This will mean a development on the Waymark model to allow arbitrary points in space to be interpolated. This makes use of the impulse response interpolation section of the algorithm. Another way that a more detailed acoustic simulator can be used, is in the generating a modelled DM at a pre-beamformed stage for the FLS. In the same way that the proposed algorithm processes the FLS frame in polar coordinate format, removing the need to interpolate the image onto a Cartesian grid, processing the motion as it would appear to each element of an array removes the beamforming stage.

The second part of the work would be to improve the odometry from the sonar sensor motion. One method is to introduce a way of detecting that the attitude-trajectory forms a loop. This detection would be performed incrementally using the attitude-trajectory generated from the regularised spline. The extra constraint from a loop closure means that the motion error is bounded [14]. This extra constraint could be incorporated into the regularised spline. Another method to improve attitude-trajectory estimate would be to detect poor quality FLS frames. This is where the image is broken due to rapid movement or something obscuring the sensor array. This would use a local interpolation of the position to give some confidence to the DM result. Likewise, a strategy for staggering the registration would produce two (or more) running position estimates that would help to corroborate the results. Detection and rejection of very poor results is difficult, however by extrapolating from a local regularised spline a basis for comparison can be made.

In addition, this extrapolation of the motion could be used to feedback into the inter-frame DM estimation stage of the proposed algorithm. This would reduce the search area for the coarse pixel displacement estimation, thus reducing the complexity and improving the accuracy.

Finally, further work would also explore further the use of the inter-frame DMs as a means of improving the precision of the rendered mosaic. This could be achieved by using the DM to position the pixels in the mosaic taking account of the individual pixel motions as opposed to the motion of the whole sonar sensor. In addition, the DM could be used to infer an estimate of the object height by observing the motion parallax relative to the expectation of the pixel motion of a flat plane.

An additional consideration for future work, applicable to the whole algorithm, would be an improved implementation which would take advantage of the efficient methods for each part of the algorithm and exploit a parallel processing architecture. This would be with the aim of achieving real-time performance.

References

- [1] B. Henson, J. Li, Y. V. Zakharov, and C. Liu, “Waymark baseband underwater acoustic propagation model,” in *Underwater Communications and Networking (UComms)*. IEEE, 2014, pp. 1–5.
- [2] B. Henson and Y. V. Zakharov, “Adaptive filter based image registration,” in *European Modelling Symposium (EMS)*, Madrid, Spain, Oct., 2015.
- [3] B. Henson and Y. Zakharov, “Local optical-flow estimation for forward looking imaging sonar data,” in *MTS/IEEE OCEANS-Monterey*, 2016, pp. 1–8.
- [4] B. Henson and Y. Zakharov, “Estimating attitude and trajectory of forward looking imaging sonar using inter-frame registration,” in *International Conference and Exhibition on Underwater Acoustics (UACE)*, Skiathos, Greece, 2017.
- [5] B. Henson and Y. Zakharov, “Attitude-trajectory estimation for forward looking sonar based on acoustic image registration,” *IEEE Journal of Oceanic Engineering*, First submission May 2017, second submission Dec. 2017.
- [6] L. Liao, Y. Zakharov, and B. Henson, “Grid waymark baseband underwater acoustic transmission model,” in *International Conference and Exhibition on Underwater Acoustics (UACE)*, Skiathos, Greece, 2017.
- [7] J. Li, Y. V. Zakharov, and B. Henson, “Multibranch autocorrelation method for doppler estimation in underwater acoustic channels,” *IEEE Journal of Oceanic Engineering*, 2017.
- [8] A. Alcocer, P. Oliveira, and A. Pascoal, “Underwater acoustic positioning systems based on buoys with GPS,” in *Proceedings of the Eighth European Conference on Underwater Acoustics*, vol. 8, 2006, pp. 1–8.
- [9] A. Mallios, P. Ridao, D. Ribas, M. Carreras, and R. Camilli, “Toward autonomous exploration in confined underwater environments,” *Journal of Field Robotics*, vol. 7, no. 33, pp. 994–1012, 2016.
- [10] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, “Imaging sonar-aided navigation for autonomous underwater harbor surveillance,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4396–4403.
- [11] J. Vaganay, M. Elkins, S. Willcox, F. Hover, R. Damus, S. Desset, J. Morash, and V. Polidoro, “Ship hull inspection by hull-relative navigation and control,” in *Proceedings of OCEANS 2005 MTS/IEEE*. IEEE, 2005, pp. 761–766.

References

- [12] G. T. Donovan, "Position error correction for an autonomous underwater vehicle inertial navigation system (INS) using a particle filter," *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 431–445, 2012.
- [13] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of intelligent and robotic systems*, vol. 53, no. 3, p. 263, 2008.
- [14] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi, "Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments," *Journal of Field Robotics*, vol. 32, no. 1, pp. 123–151, 2015.
- [15] J. N. Bakambu, C. Langley, G. Pushpanathan, W. J. MacLean, R. Mukherji, and E. Dupuis, "Field trial results of planetary rover visual motion estimation in mars analogue terrain," *Journal of Field Robotics*, vol. 29, no. 3, pp. 413–425, 2012.
- [16] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip *et al.*, "Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in GPS-denied environments," *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.
- [17] K. Richmond and S. M. Rock, "An operational real-time large-scale visual mosaicking and navigation system," in *IEEE OCEANS*, 2006, pp. 1–6.
- [18] A. Elibol, N. Gracias, and R. Garcia, "Augmented state–extended kalman filter combined framework for topology estimation in large-area underwater mapping," *Journal of Field Robotics*, vol. 27, no. 5, pp. 656–674, 2010.
- [19] R. Garcia and N. Gracias, "Detection of interest points in turbid underwater images," in *IEEE OCEANS-Spain*, 2011, pp. 1–9.
- [20] M. D. Aykin and S. Negahdaripour, "On feature matching and image registration for two-dimensional forward-scan sonar imaging," *Journal of Field Robotics*, vol. 30, no. 4, pp. 602–623, 2013.
- [21] S. Davide and F. Friedrich, "Visual Odometry: Part I: The First 30 Years and Fundamentals," *IEEE Robotics & Automation Magazine*, 2011.
- [22] B. Zitova and J. Flusser, "Image registration methods: a survey," *Image and vision computing*, vol. 21, no. 11, pp. 977–1000, 2003.
- [23] J. Le Moigne, N. S. Netanyahu, and R. D. Eastman, *Image registration for remote sensing*. Cambridge University Press, 2011.
- [24] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE Signal Processing Magazine*, vol. 20, no. 3, pp. 21–36, 2003.
- [25] C. Chailloux, J.-M. Le Caillec, D. Gueriot, and B. Zerr, "Intensity-based block matching algorithm for mosaicing sonar images," *IEEE Journal of Oceanic Engineering*, vol. 36, no. 4, pp. 627–645, 2011.

-
- [26] F. Ferreira, V. Djapic, and M. Caccia, "Real-time mosaicing of large scale areas with forward looking sonar," *IFAC-PapersOnLine*, vol. 48, no. 2, pp. 32–37, 2015.
- [27] J. Tian and K.-K. Ma, "A survey on super-resolution imaging," *Signal, Image and Video Processing*, vol. 5, no. 3, pp. 329–342, 2011.
- [28] K. Nasrollahi and T. B. Moeslund, "Super-resolution: a comprehensive survey," *Machine vision and applications*, vol. 25, no. 6, pp. 1423–1468, 2014.
- [29] W. R. Crum, T. Hartkens, and D. Hill, "Non-rigid image registration: theory and practice," *The British journal of radiology*, vol. 77, no. suppl_2, pp. S140–S153, 2004.
- [30] S. Negahdaripour, H. Assalih, Y. Petillot, and L. N. Brisson, "Performance and accuracy in visual motion computation from FS sonar video sequences," in *IEEE OCEANS*, 2010, pp. 1–7.
- [31] Y. Petillot, I. T. Ruiz, and D. M. Lane, "Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 2, pp. 240–251, 2001.
- [32] L. Conti, M. Rodrigues, and B. Hanot, "Hydroelectric power plant inspections," *Hydro International Magazine*, 2016.
- [33] (2017, Accessed 8th, May, 2017) Acquest Subaquatic Geology and Geophysics website. [Online]. Available: <http://www.acquest.com.br>
- [34] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.
- [35] M. B. Porter and H. P. Bucker, "Gaussian beam tracing for computing ocean acoustic fields," *The Journal of the Acoustical Society of America*, vol. 82, no. 4, pp. 1349–1359, 1987.
- [36] D. Fortun, P. Bouthemy, and C. Kervrann, "Optical flow modeling and computation: a survey," *Computer Vision and Image Understanding*, vol. 134, pp. 1–21, 2015.
- [37] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15, no. 50. Manchester, UK, 1988, pp. 10–5244.
- [38] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.
- [39] D. G. Lowe, "Object recognition from local scale-invariant features," in *The proceedings of the seventh IEEE international conference on Computer vision*, vol. 2. Ieee, 1999, pp. 1150–1157.
- [40] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.
- [41] P. Viola and W. M. Wells III, "Alignment by maximization of mutual information," *International journal of computer vision*, vol. 24, no. 2, pp. 137–154, 1997.

References

- [42] J. Vanne, E. Aho, T. D. Hamalainen, and K. Kuusilinna, "A high-performance sum of absolute difference implementation for motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 876–883, 2006.
- [43] J. Lewis, "Fast normalized cross-correlation," in *Vision interface*, vol. 10, no. 1, 1995, pp. 120–123.
- [44] T. Brox and J. Malik, "Large displacement optical flow: descriptor matching in variational motion estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 500–513, 2011.
- [45] C. Ballester, L. Garrido, V. Lazcano, and V. Caselles, "A TV-L1 Optical Flow Method with Occlusion Detection," in *Pattern Recognition: Joint 34th DAGM and 36th OAGM Symposium, Graz, Austria, August 28-31, 2012, Proceedings*, vol. 7476. Springer, 2012, pp. 31–40.
- [46] R. Timofte and L. Van Gool, "Sparse flow: Sparse matching for small to large displacement optical flow," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015, pp. 1100–1106.
- [47] N. Hurtós, S. Nagappa, X. Cufí, Y. Petillot, and J. Salvi, "Evaluation of registration methods on two-dimensional forward-looking sonar imagery," in *2013 MTS/IEEE OCEANS-Bergen*. IEEE, 2013, pp. 1–8.
- [48] (Accessed 2nd, April, 2016) Sound Metrics Corp. website. [Online]. Available: <http://www.soundmetrics.com/>
- [49] (2014, Accessed 22, June, 2014) Tritech International Ltd. website. [Online]. Available: <http://www.tritech.co.uk/>
- [50] (Accessed 1st March, 2017) Sony IMX219PQ. [Online]. Available: http://www.sony-semicon.co.jp/products_en/new_pro/april_2014/imx219_e.html
- [51] F. Schmitt, M. Mignotte, C. Collet, and P. Thourel, "Estimation of noise parameters on sonar images," in *Proc. SPIE Conf. Signal Image Process*, 1996, pp. 1–12.
- [52] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *IEEE Conference on Computer Vision and Pattern Recognition.*, 2009, pp. 2294–2301.
- [53] J. A. Noble and D. Boukerroui, "Ultrasound image segmentation: a survey," *IEEE Transactions on medical imaging*, vol. 25, no. 8, pp. 987–1010, 2006.
- [54] M. J. Ledesma-Carbayo, J. Kybic, M. Desco, A. Santos, M. Suhling, P. Hunziker, and M. Unser, "Spatio-temporal nonrigid registration for ultrasound cardiac motion estimation," *IEEE transactions on medical imaging*, vol. 24, no. 9, pp. 1113–1126, 2005.
- [55] J. Zhang, F. Sohel, H. Bian, M. Bennamoun, and S. An, "Forward-looking sonar image registration using polar transform," in *MTS/IEEE OCEANS Monterey*, 2016, pp. 1–6.

- [56] M. M. dos Santos, P. Ballester, G. B. Zaffari, P. Drews, and S. Botelho, "A topological descriptor of acoustic images for navigation and mapping," in *Robotics Symposium (LARS) and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*. IEEE, 2015, pp. 289–294.
- [57] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [58] V. T. Wang and M. P. Hayes, "Synthetic aperture sonar track registration using sift image correspondences," *IEEE Journal of Oceanic Engineering*, 2017.
- [59] S. Song, J. M. Herrmann, K. Liu, S. Li, and X. Feng, "Forward-looking sonar image mosaicking by feature tracking," in *Robotics and Biomimetics (ROBIO), IEEE International Conference on*, 2016, pp. 1613–1618.
- [60] H. Cho, J. Pyo, and S.-C. Yu, "Drift error reduction based on the sonar image prediction and matching for underwater hovering," *IEEE Sensors Journal*, vol. 16, no. 23, pp. 8566–8577, 2016.
- [61] E. Ferrante and N. Paragios, "Slice-to-volume medical image registration: A survey," *Medical Image Analysis*, vol. 39, pp. 101–123, 2017.
- [62] M. D. Aykin and S. Negahdaripour, "Efficient ray-casting of quadric surfaces for forward-scan sonars," in *MTS/IEEE OCEANS Monterey*, 2016, pp. 1–7.
- [63] B. K. Horn and B. G. Schunck, "'Determining optical flow': a retrospective," *Artificial Intelligence*, vol. 59, no. 1, pp. 81–87, 1993.
- [64] B. K. Horn and B. G. Schunck, "Determining optical flow," in *1981 Technical Symposium East*. International Society for Optics and Photonics, 1981, pp. 319–331.
- [65] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision," in *IJCAI*, vol. 81, 1981, pp. 674–679.
- [66] G. Caner, A. M. Tekalp, G. Sharma, and W. Heinzelman, "Local image registration by adaptive filtering," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3053–3065, 2006.
- [67] D. B. Williams and V. Madisetti, Eds., *Digital Signal Processing Handbook*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 1997.
- [68] P. Elad and A. Feuer, "Recursive optical flow estimation-adaptive filtering approach," in *IEEE Nineteenth Convention of Electrical and Electronics Engineers in Israel*, 1996, pp. 387–390.
- [69] M. Elad and A. Feuer, "Superresolution restoration of an image sequence: adaptive filtering approach," *IEEE Transactions on Image Processing*, vol. 8, no. 3, pp. 387–395, 1999.
- [70] M. R. Lynch and P. Rayner, "A new approach to image registration utilising multi-dimensional LMS adaptive filters," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1988. ICASSP-88.*, 1988, pp. 920–923.

References

- [71] J.-N. Lin, X. Nie, and R. Unbehauen, “Two-dimensional LMS adaptive filter incorporating a local-mean estimator for image processing,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 7, pp. 417–428, 1993.
- [72] J. Brauers, N. Schulte, and T. Aach, “Multispectral filter-wheel cameras: Geometric distortion model and compensation algorithms,” *IEEE Transactions on Image Processing*, vol. 17, no. 12, pp. 2368–2380, 2008.
- [73] D. G. Bailey, “Sub-pixel estimation of local extrema,” in *Proceeding of Image and Vision Computing New Zealand*, 2003, pp. 414–419.
- [74] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, “Patchmatch: A randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.
- [75] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, “The generalized patchmatch correspondence algorithm,” in *Computer Vision—ECCV 2010*. Springer, pp. 29–43.
- [76] J. Choi, H. Kim, T.-H. Oh, and I. S. Kweon, “Balanced optical flow refinement by bidirectional constraint,” in *IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 5477–5481.
- [77] (Accessed 31st August, 2015) A Database and Evaluation Methodology for Optical Flow. [Online]. Available: <http://vision.middlebury.edu/flow/>.
- [78] J. Y. Wang and E. H. Adelson, “Representing moving images with layers,” *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 625–638, 1994.
- [79] S. Negahdaripour, “On 3-D motion estimation from feature tracks in 2-D FS sonar video,” *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 1016–1030, 2013.
- [80] S. Haykin, “Adaptive filter theory,” 2nd. ed., *Prentice-Hall, Englewood Cliffs, NJ*, 1991.
- [81] A. D. Poularikas and Z. M. Ramadan, *Adaptive filtering primer with MATLAB*. CRC Press, 2006.
- [82] B. Hayes, “Crinkly curves,” *AMERICAN SCIENTIST*, vol. 101, no. 3, pp. 178–183, 2013.
- [83] E. H. Moore, “On certain crinkly curves,” *Transactions of the American Mathematical Society*, vol. 1, no. 1, pp. 72–90, 1900.
- [84] C.-C. Wu and Y.-I. Chang, “Approximately even partition algorithm for coding the Hilbert curve of arbitrary-sized image,” *IET Image Processing*, vol. 6, no. 6, pp. 746–755, 2012.
- [85] Y. V. Zakharov and V. H. Nascimento, “Homotopy RLS-DCD adaptive filter,” in *Proceedings of the Tenth International Symposium on Wireless Communication Systems (ISWCS 2013)*, pp. 1–5.

-
- [86] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.
- [87] F. Gustafsson, *Adaptive Filtering and Change Detection*. Wiley Online Library, 2000.
- [88] (2015, Accessed 29, April, 2015) NCAP - National Collection of Aerial Photography. [Online]. Available: <http://ncap.org.uk>.
- [89] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [90] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.
- [91] Y. C. Pati, R. Rezaifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *IEEE Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.
- [92] C. Schretter and H. Niederreiter, "A direct inversion method for non-uniform quasi-random point sequences," *Monte Carlo Methods and Applications*, vol. 19, no. 1, pp. 1–9, 2013.
- [93] R. Dafner, D. Cohen-Or, and Y. Matias, "Context-based space filling curves," in *Computer Graphics Forum*, vol. 19, no. 3. Wiley Online Library, 2000, pp. 209–218.
- [94] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.
- [95] Y. V. Zakharov, G. P. White, and J. Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3150–3161, 2008.
- [96] J. Ren, J. Jiang, and T. Vlachos, "High-accuracy sub-pixel motion estimation from noisy images in Fourier domain," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1379–1384, 2010.
- [97] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of computation*, vol. 19, no. 90, pp. 297–301, 1965.
- [98] C. A. Hoare, "Quicksort," *The Computer Journal*, vol. 5, no. 1, pp. 10–16, 1962.
- [99] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.
- [100] J. W. Tukey, *Exploratory data analysis*. Reading, Mass., 1977.
- [101] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

References

- [102] 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*. IEEE, 2010, pp. 4404–4411.
- [103] P. H. Eilers and B. D. Marx, “Flexible smoothing with B-splines and penalties,” *Statistical science*, pp. 89–102, 1996.
- [104] M. Unser, “Splines: A perfect fit for signal and image processing,” *IEEE Signal processing magazine*, vol. 16, no. 6, pp. 22–38, 1999.
- [105] P. H. Eilers, B. D. Marx, and M. Durbán, “Twenty years of p-splines,” *SORT-Statistics and Operations Research Transactions*, vol. 39, no. 2, pp. 149–186, 2015.
- [106] D. E. Knuth, “The art of computer programming, volume 2: seminumerical algorithms,” 1997.
- [107] (2016, Accessed 13th, December, 2016) Bluefin Robotics Corp. website. [Online]. Available: <http://http://www.bluefinrobotics.com/>
- [108] N. Hurtós, X. Cufi, and J. Salvi, “A novel blending technique for two-dimensional forward-looking sonar mosaicing,” in *Oceans-San Diego, 2013*. IEEE, 2013, pp. 1–7.
- [109] C. Liu, Y. V. Zakharov, and T. Chen, “Doubly Selective Underwater Acoustic Channel Model for a Moving Transmitter/Receiver,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 3, pp. 938–950, 2012.
- [110] D. Guériot and C. Sintès, “Forward looking sonar data simulation through tube tracing,” in *OCEANS 2010 IEEE - Sydney*, May 2010, pp. 1–6.
- [111] R. Cerqueira, T. Trocoli, G. Neves, S. Joyeux, J. Albiez, and L. Oliveira, “A novel gpu-based sonar simulator for real-time applications,” *Computers & Graphics*, vol. 68, pp. 66–76, 2017.
- [112] T. Guerneve and Y. Petillot, “Underwater 3D reconstruction using BlueView imaging sonar,” in *IEEE OCEANS*, 2015, pp. 1–7.
- [113] H. Assalih, S. Negahdaripour, and Y. Petillot, “3-D Motion Estimation in passive navigation by acoustic imaging,” in *IEEE OCEANS*, 2010, pp. 1–6.
- [114] O. Pizarro, R. M. Eustice, and H. Singh, “Large area 3-D reconstructions from underwater optical surveys,” *IEEE Journal of Oceanic Engineering*, vol. 34, no. 2, pp. 150–169, 2009.
- [115] P. Ozog, M. Johnson-Roberson, and R. M. Eustice, “Mapping underwater ship hulls using a model-assisted bundle adjustment framework,” *Robotics and Autonomous Systems*, vol. 87, pp. 329–347, 2017.
- [116] P. Lothe, S. Bourgeois, E. Royer, M. Dhome, and S. Naudet-Collette, “Real-Time Vehicle Global Localisation with a Single Camera in Dense Urban Areas: Exploitation of Coarse 3D City Models,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 863–870.

- [117] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, 2009.
- [118] M. B. Porter, "The KRAKEN normal mode program," DTIC Document, Tech. Rep., 1992.
- [119] J. Peterson and M. Porter, "Ray/Beam Tracing for Modeling the Effects of Ocean and Platform Dynamics," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 4, pp. 655–665, 2013.
- [120] M. Siderius and M. B. Porter, "Modeling broadband ocean acoustic transmissions with time-varying sea surfaces," *The Journal of the Acoustical Society of America*, vol. 124, p. 137, 2008.
- [121] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, *Computational ocean acoustics*. Springer, 2011.
- [122] M. Stojanovic, "Underwater acoustic communications: Design considerations on the physical layer," in *Fifth Annual Conference on Wireless on Demand Network Systems and Services (WONS)*. IEEE, 2008, pp. 1–10.
- [123] (2010, Accessed 19, March, 2014) Heat, Light & Sound Research, Inc. website. [Online]. Available: <http://oalib.hlsresearch.com/Modes/AcousticsToolbox/>
- [124] O. C. Rodriguez, "General description of the BELLHOP ray tracing program," *Physics Department Signal Processing Laboratory Faculty of Sciences and the University of the Algarve Tecnologia (Galician), Version*, vol. 1, 2008.
- [125] (2003, Accessed 27, March, 2014) The SWellEx-96 Experiment website. [Online]. Available: <http://www.mpl.ucsd.edu/swellex96/>
- [126] X. Geng and A. Zielinski, "An eigenpath underwater acoustic communication channel model," in *MTS/IEEE Proceedings OCEANS'95. Challenges of Our Changing Global Environment.*, vol. 2, 1995, pp. 1189–1196.
- [127] P. Qarabaqi and M. Stojanovic, "Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 4, pp. 701–717, Oct 2013.
- [128] R. Otnes, P. van Walree, and T. Jensenud, "Validation of replay-based underwater acoustic communication channel simulation," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 4, pp. 689–700, Oct 2013.
- [129] W. Tranter, K. Shanmugan, T. Rappaport, and K. Kosbar, *Principles of communication systems simulation with wireless applications*. Prentice Hall Press, 2003.
- [130] M. B. Porter, "The bellhop manual and user's guide: Preliminary draft," *Heat, Light, and Sound Research, Inc., La Jolla, CA, USA, Tech. Rep*, 2011.
- [131] Y. Zakharov and A. Morozov, "OFDM transmission without guard interval in fast-varying underwater acoustic channels," *IEEE Journal of Oceanic Engineering*, vol. PP, no. 99, pp. 1–15, 2014.

References

- [132] T. Chen, C. Liu, and Y. V. Zakharov, "Source localization using matched-phase matched-field processing with phase descent search," *IEEE Journal of Oceanic Engineering*, vol. 37, no. 2, pp. 261–270, 2012.
- [133] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of biomedical optics*, vol. 15, no. 2, pp. 021 314–021 314, 2010.
- [134] T. Schneider and H. Schmidt, "Model-based adaptive behavior framework for optimal acoustic communication and sensing by marine robots," *IEEE Journal of Oceanic Engineering*, vol. 38, no. 3, pp. 522–533, 2013.

Appendix A

Debris dataset example

A.1 Pitch and altitude estimation

The motion estimator can be further extended by incorporating the pitch and altitude estimation. These estimates are then used to select a subset of motion models providing the best fit to the inter-frame DM. Assume that the sonar is looking at a flat plane and that the vertical beam-width Φ_w is known. Fig. A.1 shows that the illuminated area on the seabed plane is dependent on the altitude and the pitch angle of the sonar sensor. The pitch angle Φ

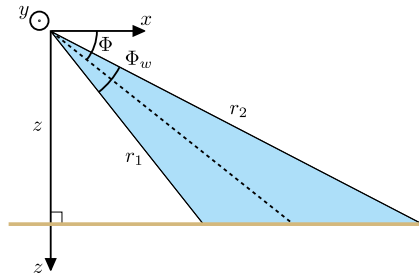


Fig. A.1 The illuminated area from the vertical beam. From this area an estimate of the altitude (z) and pitch angle (Φ) of the sensor is made.

is estimated as

$$\Phi = \frac{\pi}{2} - \arctan \left(\frac{\cos(\Phi_w) - \frac{\xi(r_1)}{\xi(r_2)}}{\sin(\Phi_w)} \right) - \frac{\Phi_w}{2}, \quad (\text{A.1})$$

where $\xi(r_1)$ and $\xi(r_2)$ are the ranges to the first and last points of the illuminated area, respectively, estimated from the sonar image. The altitude z is estimated as

$$z = \xi(r_1) \cos \left(\frac{\pi}{2} - \Phi - \frac{\Phi_w}{2} \right). \quad (\text{A.2})$$

A.2 Debris example dataset

The Debris dataset is now considered where pitch and altitude are changing over a sequence of sonar frames. This example dataset was obtained by a surveying platform moving forward over a debris field (concrete and re-enforcing bars). The sonar rapidly sweeps back and forth in the yaw angle (rotation around z). With reference to Section 4.4, 20 statistics were used to represent the inter-frame motion estimator. For each of the histograms generated using (4.7), five percentiles are computed (5th, 25th, 50th, 75th and 95th), thus twenty parameters in total; denoted as a vector \mathbf{s} .

Fig. A.2 shows the sonar view of the seabed.

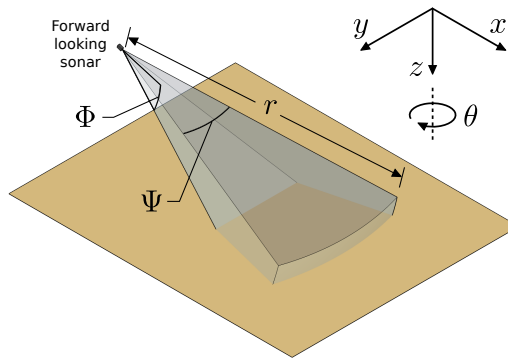


Fig. A.2 Position of the FLS field of view and the seabed. r is the range, Ψ is the field of view, Φ the elevation and θ is the rotation around the z axis.

Fig. A.3 shows a mosaic and attitude-trajectory estimate for 486 frames over the debris field. Fig. A.4 shows the five components of the attitude-trajectory in this scenario. These results demonstrate that the proposed method is also applicable to scenarios with varying pitch and altitude.

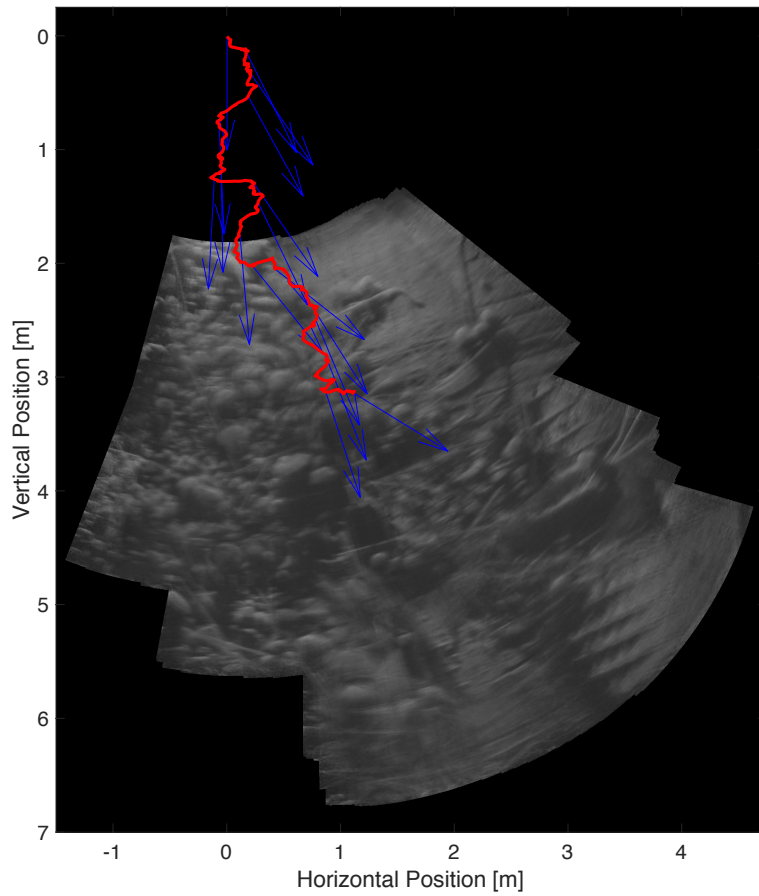


Fig. A.3 A mosaic of 486 frames showing a debris field. The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

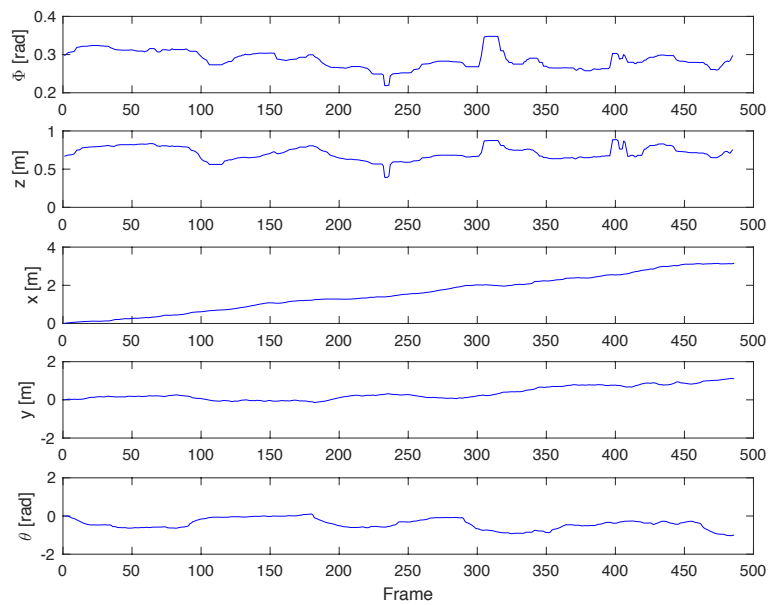


Fig. A.4 The smoothed attitude-trajectory of 486 frames over the debris field.

Appendix B

Underwater Acoustic channel model

In Chapter 4, the generation of a modelled DM was discussed. The motion of the sonar sensor has been modelled with a basic geometric relationship that used broad assumptions about the environment; i.e. the scene being observed is a flat plane and the propagation through the water column is free from distortions. For many scenarios this would be adequate, however, if the topology of the scene is more complex, or the range is longer leading to greater distortion through the water column, then a more detailed model might be needed. The more accurate the model, the more accurate the generated DM will be. A more accurate modelled DM gives better discrimination when being matched to the DM estimate from the FLS inter-frames registration.

In an unknown environment, assessing the topology using the content of the image is a significant problem [112, 113, 20]. The work in [20] made an estimation of the dimensions of 3-D features in the scene from the 2-D FLS image. These estimates were then used to improve the motion estimation of the sonar sensor, by modelling the inter-frame motion of the features in 3-D space. Other work, [114] used optical measurement techniques to build a model, however, this method would be restricted if the visibility is poor [19].

Other environments however, are known in advance, for instance a man-made underwater structure such as a dam wall, or a ship's hull (see Section 5.2). In this environment the topology of the scene is known and can potentially be matched with the features observed. Using this known environment as a reference, navigation accuracy can be improved [115, 116].

The generation of a modelled motion DM can be improved using knowledge about the topology of the scene. Another improvement could come from a more sophisticated model for the propagation of the acoustic wave through the water column. An acoustic signal travelling through a body of water can be subject to a number of distortions, for example, multipath propagation, the Doppler effect and time varying environmental conditions [117].

In order to capture these distortions and incorporate them into a motion model, a physics simulator can be used. Acoustic wave simulators, such as BELLHOP [35] and KRAKEN [118], can take into account environmental conditions to show the propagation distortions to the signal at the receiver. However, an accurate simulation is computationally expensive and so is impractical for each sample period of an long signal transmission.

To reduce the complexity, physics simulations of the environment can be made at snapshot intervals in time (or space), and the results interpolated to build a channel model. One approach for dealing with this problem is the ‘virtual’ signal transmission [119, 120]. This approach was also adopted in the Waymark channel model. In the Waymark model [109], the impulse response for a set of points or waymarks along the transmitter/receiver trajectory are calculated using a physics simulator. The relative delay in the impulse responses between these points is then estimated, allowing the shape of the impulse response and the delays to be interpolated separately, giving an improved result. Also, because the waymarks are snapshots of the environment, changing conditions can be taken into account, for instance, a model of a surface wave can be used.

This model gave good results when compared with a more established model [120], however, all of the processing is performed at a sampling frequency suitable for a passband signal. The processing of the full bandwidth signal may be unnecessary as often the transmitted signal in the baseband is of interest. If this is the case then the signal can be processed in the baseband, this requires the delay associated with each impulse response to be considered when moving from the passband to the baseband and vice-versa [1].

This chapter is organised as follows, in Section B.1 the distortion that an underwater channel is subject to is discussed. Section B.2 describes ray tracing acoustic models. Then in Section B.3, the Waymark model with the baseband processing modification is described [109, 1]. The model is then validated with a comparison with the VirTEX model [120] in Section B.3.4. Finally Section B.5 ends the chapter with some concluding remarks.

The baseband Waymark channel model is presented in the conference paper: B. Henson, J. Li, Y. V. Zakharov, and C. Liu, “Waymark baseband underwater acoustic propagation model,” in *Underwater Communications and Networking (UComms)*. IEEE, 2014, pp. 1–5.

B.1 Signal Distortions

The key distortions that affect a signal transmitted through an underwater medium are multipath and the Doppler effect [117]. Due to the slow speed of sound, when compared to electromagnetic wave propagation, these affects are immediately apparent and would need to be accommodated any UWA system.

Multipath propagation

When the signal takes multiple paths to the receiver the pressure fronts will interfere, that is, the same signal will arrive at the receiver at different times depending on the path delay. The speed of sound varies with the conditions in the water, the primary factors being temperature, salinity and depth (static pressure) [121]. These factors can change on a microscopic scale, and affect the propagation with a wave front being bent locally towards the region with a slower speed of sound; this is described by Snell's Law [121]. The refraction of the wave gives multiple paths that can contribute to the received signal. Fig. B.1 shows example multipaths for an underwater environment, there are reflective paths from the surface and bottom and refractive paths. The speed of sound profile is also shown; it can be seen that there is a minimum, with the area around it acting as a wave-guide for the sound.

The Doppler effect

Due to the slow speed of sound the Doppler effect can create a large distortion for quite modest movement. This movement could be intentional, such as an AUV moving underwater, or it could be unintended such as drift on the Ocean currents. This effect also adds to the complexity of the multipath distortion because the angle of incidence affects the magnitude of the Doppler effect. Therefore, each of the multipath arrivals will have a different Doppler shift associated with it, this being a function of the angle between the receiver/transmitter velocity vector and the incident arrival ray [119]. Fig. B.2 shows some of the sources of the Doppler effect that operate on the signal.

Time Varying Channels

The Ocean is a dynamic environment and so the channel will change continuously. The changes that need to be taken into account will depend to some extent on the time scale of interest. For most communication and sensing applications these time scales will be relatively short, and be with reference to the sample period. These short period changes would be; the motion of the transmitter/receiver, this being particularly relevant if there are complex surroundings, and the motion of any surface wave that has the effect of rapidly changing the multipath environment [122].

B.2 Acoustic Channel Modelling

As computers have become more powerful so the possibility of modelling the complex interactions of underwater acoustics has become more attractive and tractable. There are many

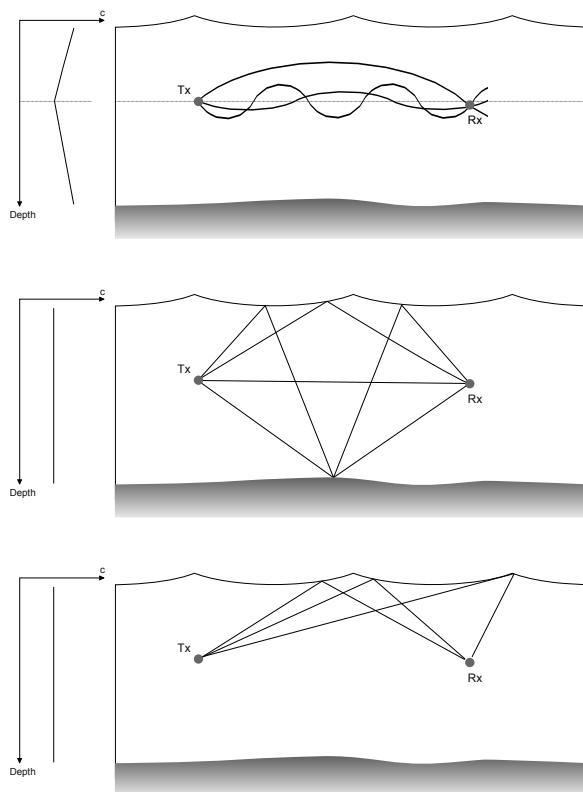


Fig. B.1 An illustration of the multipath effect through refractive (top figure) and reflective (bottom two figures) paths, the speed of sound profile is also shown.

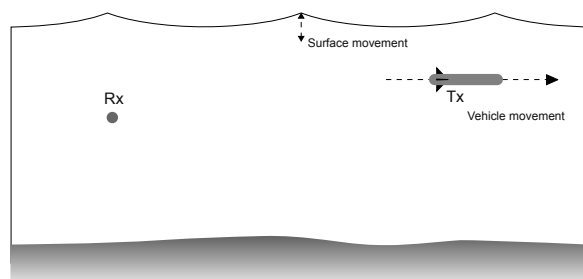


Fig. B.2 Examples of the source of the Doppler effect.

advantages to using a simulated environment in order to develop and evaluate underwater technology. Firstly, the need to conduct expensive and time consuming sea trials is reduced, allowing a wide number of tests to be carried out [109]. Secondly, given the dynamic nature of the Oceans it is difficult if not impossible to control the environment dependent parameters, this makes it hard to compare designs under similar conditions [109].

B.2.1 Modelling Techniques

There are a range of modelling techniques that seek to simulate the physics of an underwater environment and will generate a pressure field for a set of conditions. As with any simulation of the real world there is a trade off between accuracy and speed. There is also a distinction to be made between a single snapshot of a wave propagation in a fixed environment, and a useful simulation for the propagation of a longer signal through a time varying UWA channel.

Two popular physics simulators are BELLHOP [35] and KRAKEN [118] available from Heat, Light & Sound Research, Inc. [123].

Ray Tracing

The BELLHOP program [35] is an example of a ray tracing acoustic model. This program was used as the underlying physics model for the wave propagation, the results of which are an input to the Waymark channel model developed in this chapter.

Ray tracing simulation considers the propagation of the acoustic pressure wave as a series of rays from the source. The rays are launched at known angles from the source and the trajectory is calculated as the solution to the ray equations. The calculation of the amplitude of the acoustic pressure requires the solution of dynamic ray equations [124].

The attraction of using ray tracing methods is that they are better suited for high frequency and range dependent problems [35].

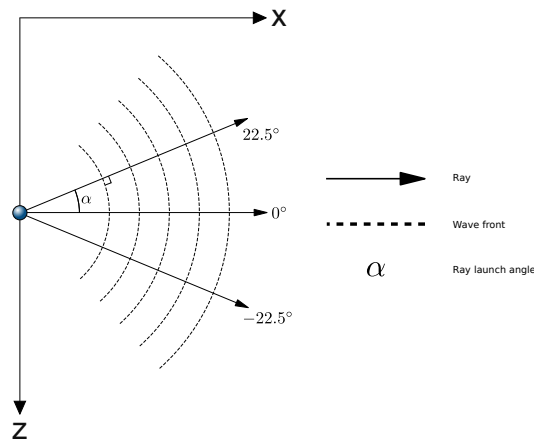


Fig. B.3 A point source with a wave front. The rays that represent it are discrete in the launch angle and perpendicular to the wave front, which is the direction of motion [121].

A ray is considered to be normal to the propagating wave front and therefore denotes the direction of travel. This is illustrated in Fig. B.3. The implementation in terms of a simulation is to discretise the wave front into a number of rays that will be representative of the propagation.

As the speed of sound varies with depth, the ray is subject to refraction through the water, this is consistent with Snell's law:

$$k_1 \cos \theta_1 = k_2 \cos \theta_2 \quad (\text{B.1})$$

where: k_1 and k_2 can be the refractive index of the two media or the speed of sound through them, and θ_1 and θ_2 are the incident and transmission angles respectively. For a continuously varying environment this becomes a function of depth z_1 and z_2 [121]:

$$k(z_1) \cos \theta(z_1) = k(z_2) \cos \theta(z_2) \quad (\text{B.2})$$

With an initial launch angle and the implications of (B.2), the direction on the wave propagation can be calculated.

Using a derivation from the Helmholtz equation [121], the pressure at the point of a disturbance can be related to the wave propagating from it,

$$\nabla^2 p + \frac{\omega^2}{c^2(\mathbf{x})} p = -\delta(\mathbf{x} - \mathbf{x}_0) \quad (\text{B.3})$$

where the coordinates are Cartesian $\mathbf{x} = (x, y, z)$ with \mathbf{x}_0 being the origin, $c(\mathbf{x})$ is the speed of sound at \mathbf{x} and ω is the angular frequency of the source. Details of the derivation of the wave equation are given in [121]. Given the system in cylindrical coordinates the first order ray equations can be written as:

$$\begin{aligned} \frac{dr}{ds} &= c\xi(s), & \frac{d\xi}{ds} &= -\frac{1}{c^2} \frac{\partial c}{\partial r}, \\ \frac{dz}{ds} &= c\zeta(s), & \frac{d\zeta}{ds} &= -\frac{1}{c^2} \frac{\partial c}{\partial z}, \end{aligned} \quad (\text{B.4})$$

where c is the sound speed, s is the arc length along the trajectory, $[r(s), z(s)]$ gives the coordinates of the trajectory point in a radial plane. The tangential vector of the ray may be given by:

$$\mathbf{t}_{ray} = c \left[\frac{dr}{ds}, \frac{dz}{ds} \right] = c[\xi(s), \zeta(s)]. \quad (\text{B.5})$$

In order to numerically solve these equations for a point along the trajectory, the initial conditions need to be specified:

$$\begin{aligned} r = r_0, \quad \xi &= \frac{\cos(\theta_0)}{c(0)}, \\ z = z_0, \quad \zeta &= \frac{\sin(\theta_0)}{c(0)}. \end{aligned} \tag{B.6}$$

The travelling time of the wave $\tau(s)$ is derived from the integral of the instantaneous velocities along the trajectory:

$$\tau(s) = \tau(0) + \int_0^s \frac{1}{c(s')} ds' \tag{B.7}$$

These equations chart the trajectory of the wave front, however, the precise amplitude at each point is dependent on the spread in the ray cross section, and therefore the model used for the shape of the ray.

The cross section of the ray may be calculated using the dynamic ray equations. These define the spread of the ray and hence the amplitude as a function of the arc length [121]. This value is then used to modulate the defined shape of the beam. The cross sectional shape of the beam determines how the area between beams is interpolated. There is the simpler Geometric Beam and the more complex although more accurate Gaussian Beam. The beam shape fills the width between beams as illustrated by Fig. B.4 [121].

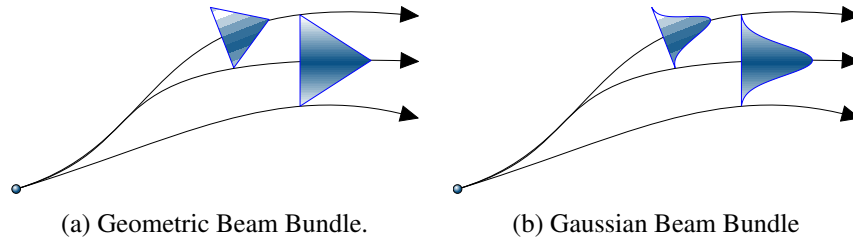


Fig. B.4 Ray tracing beam shapes.

The simulations using BELLHOP in this report use Gaussian beam bundles for the cross section of the ray. This gives more accurate results because there are less of the artefacts associated with geometric beam tracing, such as perfect shadows and infinitely high energies at caustics [35].

Figs. B.5 and B.6 show examples of output from BELLHOP. First the program takes in information about the environment such as the speed of sound and the conditions for the surface and sea floor. In this example the Munk sound speed profile (SSP) was used and is shown in Fig. B.7. Also, there was a generated surface and sea floor topology. The rays

calculated can be seen plotted in Fig. B.5. The transmission loss was also generated, this is shown in Fig. B.6.

Another example shown in Figs. B.8, B.9 and B.10 is for the SWellEx-96 environment [125]. This environment has a SSP that tends to direct the sound downwards to the bottom, and the bottom itself is made up of different sediment layers which each reflect at the boundaries. In addition, the experiment encompasses wide launch angles for the rays. This has the effect of producing a large number of reflected rays which give a mixed pressure field as shown in Figs. B.9 and B.10.

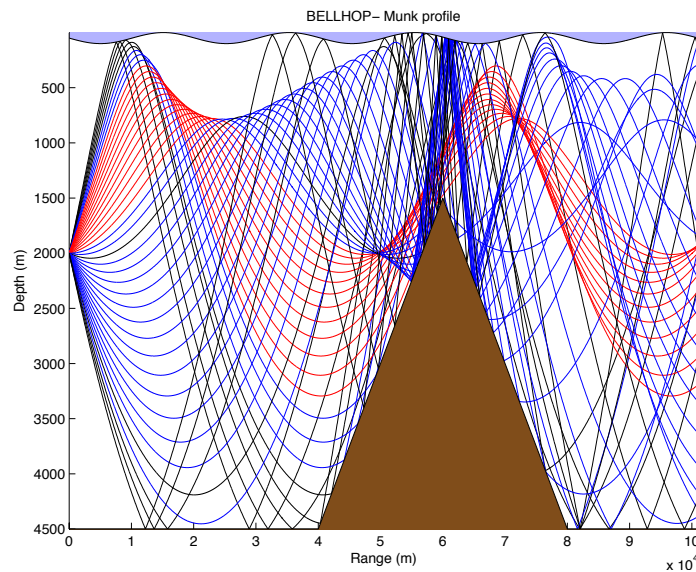


Fig. B.5 An example of a ray diagram for the Munk SSP generated from BELLHOP. The model is able to incorporate a surface and sea floor model.

Time Varying Channels

In many practical applications the time variation in the water, and hence the channel characteristics, is on a time scale that is pertinent to the signal passing through the channel. For example, a communications message could last for tens of seconds which could cover many periods of a surface wave. For a channel simulation to be useful it needs to encompass this variation to be relevant for the period of interest. A simulation of the physics of an environment could be made for each sample period time step of the signal. However, this would be very computationally expensive, and for many signals the sampling period is very small compared to the slower rate of change in the environmental conditions. It is this detailed modelling of the physics and the appropriate approximation for the time scale of interest that make up the full channel model and simulation.

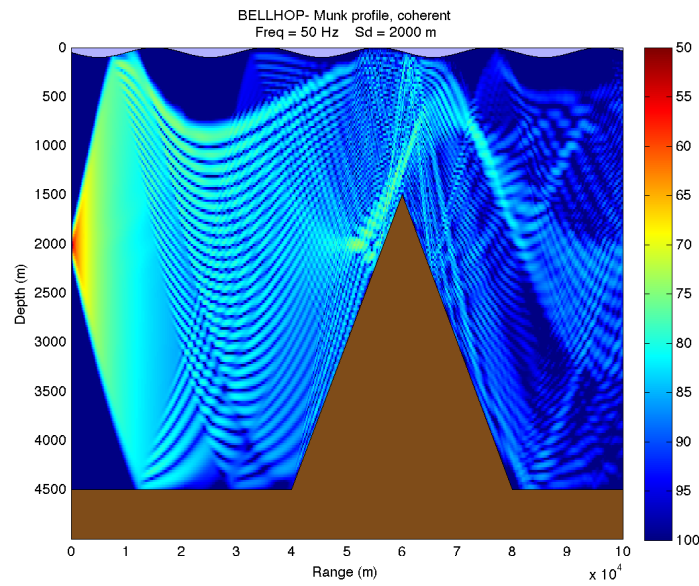


Fig. B.6 An example of a transmission loss diagram for the Munk SSP generated from BELLHOP.

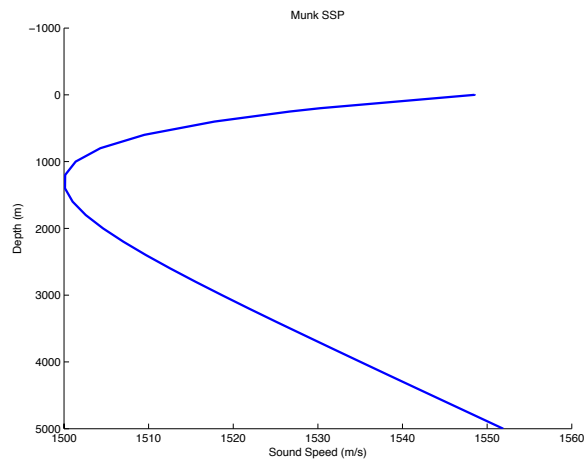


Fig. B.7 The Speed of Sound profile for the Munk environment.

Other models seek to perturb the impulse response in some way in order to generate statistical model variation. There are models that manipulate the eigenpaths adding frequency shifts to represent the Doppler effect, and using statistical models for the amplitudes [126]. Other models represent these effects by adding random local displacements [127]. The simulation of the channel can be avoided altogether by directly replaying a measured time-varying channel response [128].

One example of an existing design of channel model is the ‘virtual’ signal transmission model (VirTEX) [119, 120]. It is this virtual signal transmission method that was used in the

Underwater Acoustic channel model

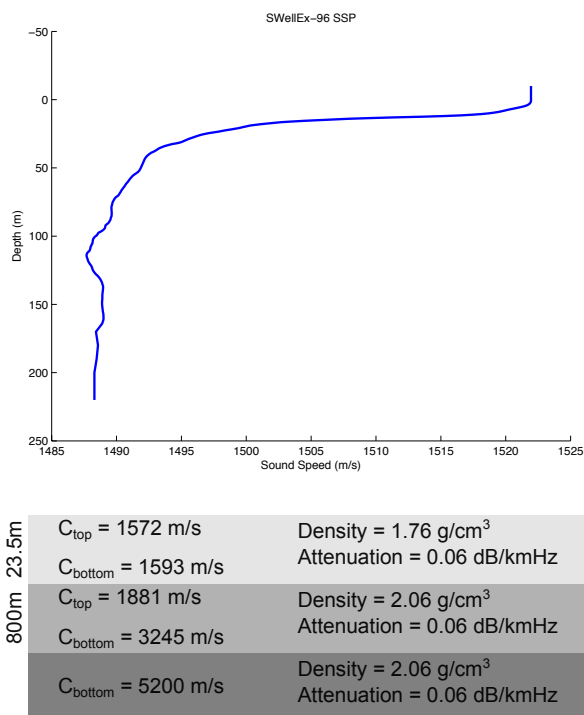


Fig. B.8 The speed of sound profile for the SWellEx-96 environment. Where the attenuation is a linear function of distance and frequency.

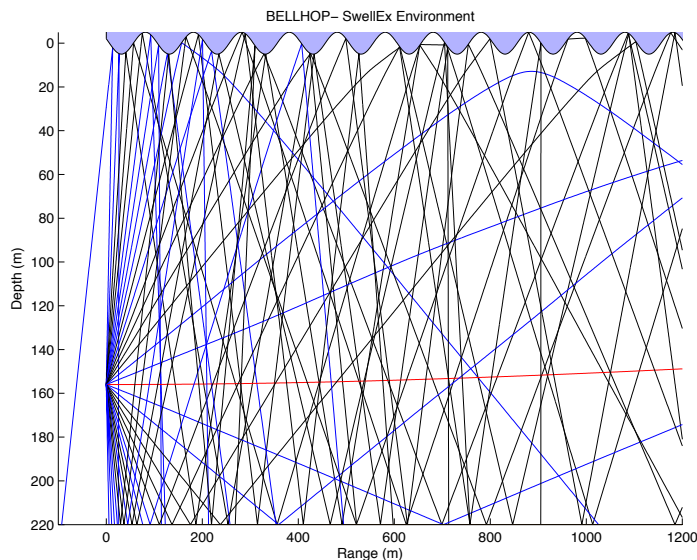


Fig. B.9 An example of a ray diagram for the SWellEx-96 SSP. This is an example of the environment used in the experiments detailed in Section B.3.4. In the experiment a much wider set of launch angles were used.

Waymark model [109], and it is this Waymark model that forms the basis for this section of work.

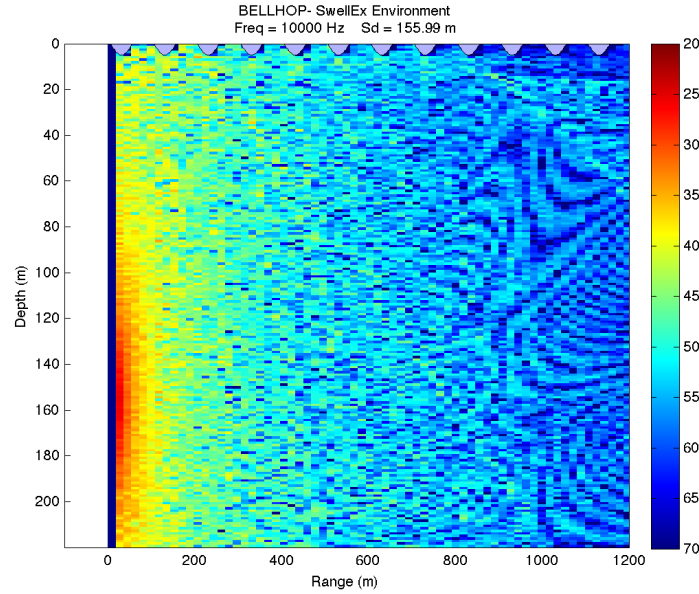


Fig. B.10 An example of a transmission loss diagram for the SWellEx-96 SSP. This is an example of the environment used in the experiments detailed in Section B.3.4.

B.3 Baseband Waymark Channel Model

This section describes the further development of the Waymark underwater channel model [109] to allow processing of a signal as baseband frequencies. This improvement reduced the complexity, meaning that a simulation can be performed in a shorter run time [1].

B.3.1 Underwater Channel Simulation

A noise free received signal in a time-varying linear channel may be described in the general case by [129]:

$$y(t) = \int_{-\infty}^{\infty} h(t, \tau) s(t - \tau) d\tau, \quad t \in [0, T_{sig}] \quad (\text{B.8})$$

where: $h(t, \tau)$ is impulse response of the channel, $s(t)$ the transmitted signal and T_{sig} the signal duration. At time t the baseband channel impulse response may be represented as the sum of baseband multipath components [122] given by:

$$h(t, \tau) = \sum_{p=1}^L c_p(t) \delta(\tau - \tau_p(t)) \quad (\text{B.9})$$

where:

$$c_p = A_p(t) e^{-j2\pi f_c \tau_p(t)} \quad (\text{B.10})$$

c_p is the complex amplitude of path p , $j = \sqrt{-1}$, $\tau_p(t)$ the time-varying delay for path p , $A_p(t)$ the time-varying complex amplitude for path p , f_c is the signal carrier frequency. The delay $\tau_p(t)$, would be affected by the path geometry, which would encompass any movement in the system ultimately representing the Doppler effect.

B.3.2 Previous work

The work in this section is built upon that described in [109]. In this previous research an underwater model was developed that comprised of a number of key stages as follows:

First, the environment over time is described, this includes the trajectory of the transmitter and the receiver relative to the seafloor. Also, the sea surface changing in time can be modelled. The simulation of the transmission is then split up into time steps (multiple signal sample periods), a snapshot of the environment at each time step termed a waymark. The propagation at each of these waymarks is then modelled with a physics simulator, and frequency and impulse responses generated. In [109] both ray tracing (BELLHOP [130]) and normal mode (KRAKEN [118]) physics simulators were used to generate a pressure field, from this the waymark impulse response is calculated. From the waymark impulse responses an impulse response for each signal sample is interpolated. To improve the interpolation, the delay shifts between the waymark impulse responses are estimated. They are then adjusted in time so that they have better alignment. This better alignment allows the interpolation between the waymark impulse responses to be more accurate and the shape of the impulse response and the delays to be interpolated separately. The resultant signal sample impulse responses are then retimed based on interpolated delays.

In addition, the overall common propagation delay of the signal is subtracted from all of the signal delays, thus reducing the length of the impulse response. These operations effectively separate the shape of the impulse response from the delay.

This is an outline of the system as set out in [109], however, the processing of the full bandwidth signal may be unnecessary as often the transmitted signal consists of a baseband signal modulated onto a carrier frequency. If this is the case then the signal can be processed in the baseband, however, this requires the delay associated with each impulse response to be considered when moving from the passband to the baseband and vice-versa. The development of the Waymark model is discussed in the following sections, and outline of the final system is shown in Fig. B.11 [1].

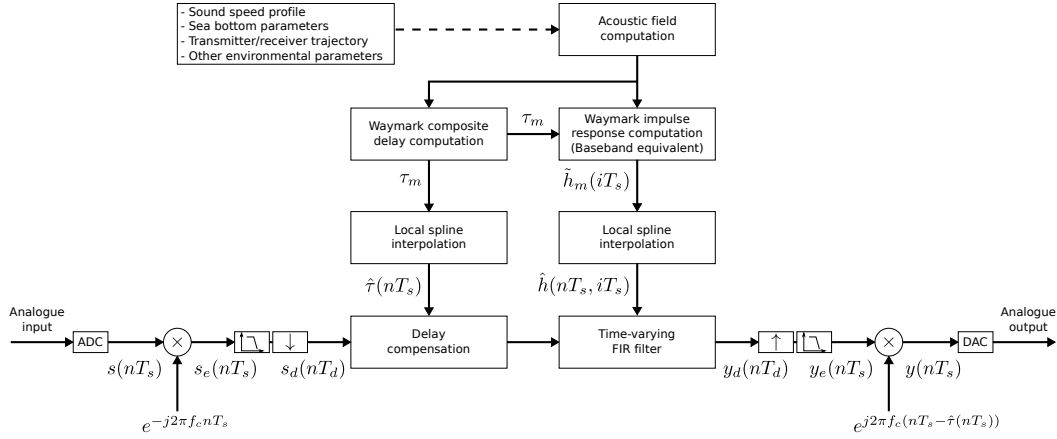


Fig. B.11 A block diagram of the UWA simulator [1] as a development on the system presented in [109]. The link between the delay compensation for the impulse response and the upshifting of the signal can be seen represented with $\hat{\tau}(nT_s)$.

B.3.3 Baseband Waymark Channel Model

The complexity and therefore the runtime of a simulation be obtained by taking into account the bandwidth of the signal that is to be passed through the channel; converting the signal to a baseband representation, and, depending on the signal, perform the processing at a lower sample rate. This is however not as simple with a time varying channel, as any time variation needs to be carried through to the recovery of the passband signal.

From the passband the baseband signal is given by the convolution [1]:

$$y_e(nT_s) = \sum_{i=0}^{I-1} h(nT_s, iT_s) s_e(nT_s - iT_s), \quad n = 0, \dots, N-1 \quad (\text{B.11})$$

where $s_e(nT_s)$ can be approximated by:

$$s_e(nT_s) = \sum_{k=0}^{K-1} s(kT_s) e^{-j2\pi f_c k T_s} r(nT_s - kT_s) \quad (\text{B.12})$$

and

$$r(nT_s) = \text{sinc}(f_0 n T_s) \frac{\cos(\pi f_0 \alpha n T_s)}{1 - (2f_0 \alpha n T_s)^2} \quad (\text{B.13})$$

where: $y_e(nT_s)$ is the baseband output signal, $s_e(nT_s)$ the baseband equivalent signal, T_s is the sample period, I is the length of the impulse response h , $N = T_{\text{sig}}/T_s$, $r(nT_s)$: raised cosine low pass filter impulse response, K : raised cosine filter length, f_0 : upper bound of baseband bandwidth, α : roll-off factor.

The original signal spectrum is shifted to centre around zero and a low pass filter applied. The low pass filter chosen was the raised cosine filter. Once the signal has been moved to be the baseband equivalent, then the sampling period may, with reference to the baseband bandwidth, be increased from T_s to give a longer sample period T_d .

Once passed through the channel the signal may be upsampled and upshifted back to the original sampling frequency and passband. Due to the relatively slow speed of sound the delays are important in the restoration of the signal to the passband. Therefore the upshifting of the signal needs to take into account the delay that was applied to the input signal at each sample point. The upshifted post channel signal may be calculated as follows [1]:

$$y(nT_s) = y_e(nT_s)e^{j2\pi f_c(nT_s - \tau_n)} \quad (\text{B.14})$$

where: $y(nT_s)$ is the output signal, $y_e(nT_s)$ the low frequency equivalent signal, τ_n is the estimated additional delay at each output sample instant.

A diagram of the system is shown in Fig. B.11, this shows the development from the original system in [109]. In this development the waymark impulse response is created in the baseband, in addition the input signal is converted to a downsampled baseband signal and passed through the time varying delay and time varying FIR filter. The splitting of the channel into these two components allows a more accurate interpolation of the channel impulse response between waymarks (for more details see [131]), thus increasing the waymark interval and consequently reducing the computation. However, the time varying delay requires a phase correction when upshifting the signal as shown in Fig. B.11.

The bandlimited channel frequency response at waymark m is generated from multipath arrivals, with their respective excess delays and baseband equivalent complex amplitudes. As described in [109], the common propagation delay is removed from all of the arrivals thus reducing the size of the required impulse response. The frequency response $H_m(f_k)$ at waymark m is computed using BELLHOP. The bandwidth for the channel representation (frequencies f_k) should be selected with reference to the bandwidth of the signal plus any Doppler shift from the environment and movement. The channel impulse response for each waymark is calculated from the inverse DFT of the waymark frequency response. With the delay compensation this gives a set of responses with an alignment based on the cross-correlation of the waymark impulse responses giving a better interpolation between the waymarks.

B.3.4 Comparison of Waymark and VirTEX channel models

A series of tests were performed to compare the Waymark model to the established VirTEX model [120]. The first experiment used a single tone and a flat SSP. The second used an environment with parameters from the SwellEx-96 experiment [125]; these being the measured SSP and the transmitter trajectory.

The VirTEX model [120] uses a regularly spaced grid to describe the volume of water that the signal propagates through. The model interpolation is performed on the amplitude and time of arrivals of the multipath components. An interpolated point between the grid points is the weighted sum of the arrivals at the four surrounding points. So, for instance, if there were two multipath arrivals at each of the surrounding grid points then the interpolated point would comprise of eight multipath arrivals. The delays are adjusted according to the local speed of sound, the geometric distance and incident angle from the interpolated point to the grid point.

Simplified Environment Test

The simulation environment was as follows:

- 10kHz tone ($f_s = 40\text{kHz}$). 100s duration.
- Flat SSP at 1500 m/s.
- Flat bottom at 200m. Speed of sound in sea bed 1600 m/s.
- Flat calm surface or a sinusoid surface (5m amplitude, 8s swell period).
- Receiver and Transmitter depth 100m.
- Range $1000 + v_c t$ meters, ($v_c = 5\text{m/s}$).
- Decimation factor 64, giving $T_d = 1.6\text{ms}$.

Using a 10kHz tone and a velocity of 5m/s a direct path would be expected to have a Doppler shift of around -33.3Hz . The waymark interval was 0.0512s. The spectrum of the channel output from the Waymark and VirTEX models is shown in Fig. B.12. The multipath components on each of the models can be seen to correspond, and the Doppler shift of the main path can be seen to be around -33.3Hz .

Fig. B.13 shows the same experiment but with a sinusoidal surface wave added. It can be seen that the reflections from the surface have been scattered; with the single reflection peak reduced, comprising mostly of the reflection from the bottom, and the double reflection peak reduced almost to background levels. Again, a good correspondence between the Waymark and VirTEX models can be seen.

For both models two minutes of transmission time was simulated, for the Waymark model this required 2354 field calculations.

The signal decimation factor was 64, giving a large saving in the channel impulse response interpolation and convolution calculations (at least linear in the decimation factor) compared to the original Waymark model in [109].

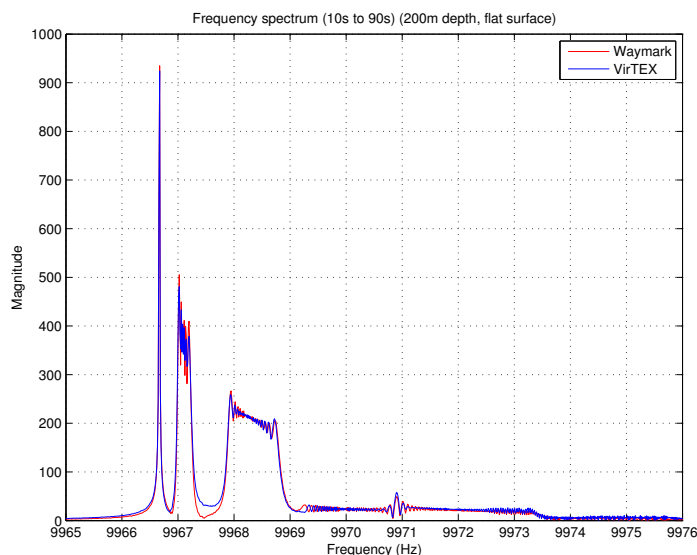


Fig. B.12 The received signal spectrum from the Waymark and VirTEX models with a flat SSP environment and a flat surface model [1].

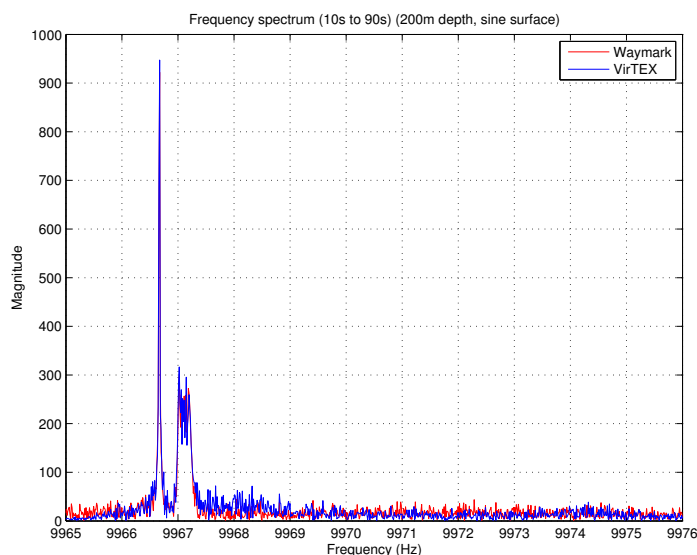


Fig. B.13 The received signal spectrum from the Waymark and VirTEX models with a flat SSP environment and a sinusoidal surface model.

Complex Environment Test

This set of experiments compared the Waymark and VirTEX models with a more complicated and realistic environment. The SSP and the trajectory parameters were taken from the SWellEx-96 experiment [125]. Again a single 10kHz tone was used with a section of the trajectory from ‘Event S5’ [125].

The simulation environment was as follows:

- 10kHz tone ($f_s = 40\text{kHz}$). 100s duration.
- SWellEx environment SSP, as shown in Fig. B.8.
- Flat bottom. Layered sea bed translated into complex reflection coefficients.
- Flat calm surface or a sinusoid surface (5m amplitude, 8s swell period).
- Interpolated trajectory from SWellEx localisation data produced in [132].
- Decimation factor 64, giving $T_d = 1.6\text{ms}$.

In this test the transmitter was moving towards the receiver at 2.5m/s, so a Doppler shift of 16.6Hz was expected for the direct path. The waymark interval was 0.0512s.

The frequency spectrum as output from the Waymark and VirTEX models is shown in Fig. B.14. This is an enlarged section around the tone frequency with the Doppler shift.

It can be seen that the Doppler shift is similar in each case and there is general agreement for the spread around the main peak. Fig. B.15 shows the same experiment but with a sinusoidal surface wave. Some effect can be seen, however it is less pronounced because SSP for the SWellEx experiment tends to bend the sound wave down and so there is less interaction with the surface.

B.3.5 Discussion

This section describes the modification to the Waymark UWA channel model as proposed in [109]. This further work involved developing the channel model and signal representation at the baseband. This however represented a challenge, as there is a time-varying phase shift introduced into the upshifted signal at the channel output. This needs to be synchronized with the time-varying delay which is introduced in the transmitted signal before the baseband time-varying convolution. In addition to this the decimation process needs to be taken into account. This challenge is similar to that in the baseband Doppler effect compensation in UWA modems.

One of the design concepts of the original Waymark model is to use local interpolation, this allows the calculation to be performed in a rolling manner. This has the advantage in that

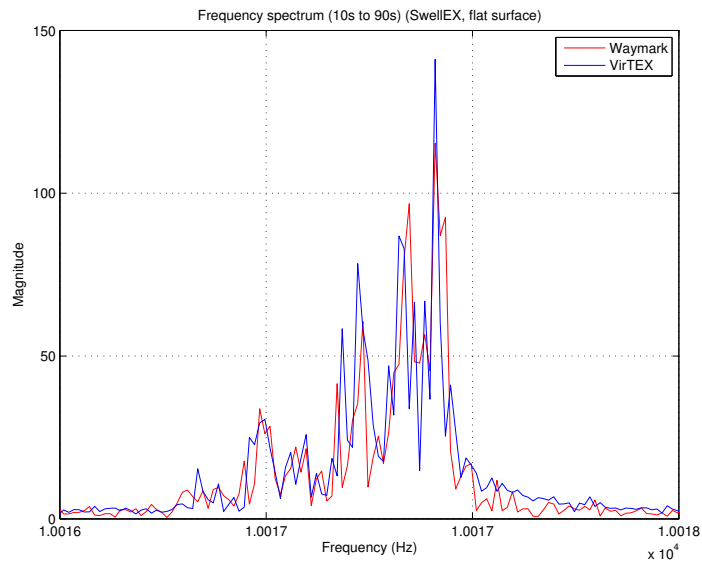


Fig. B.14 The received signal spectrum from the Waymark and VirTEX models with the SWellEx environment and trajectory with a flat surface model [1].

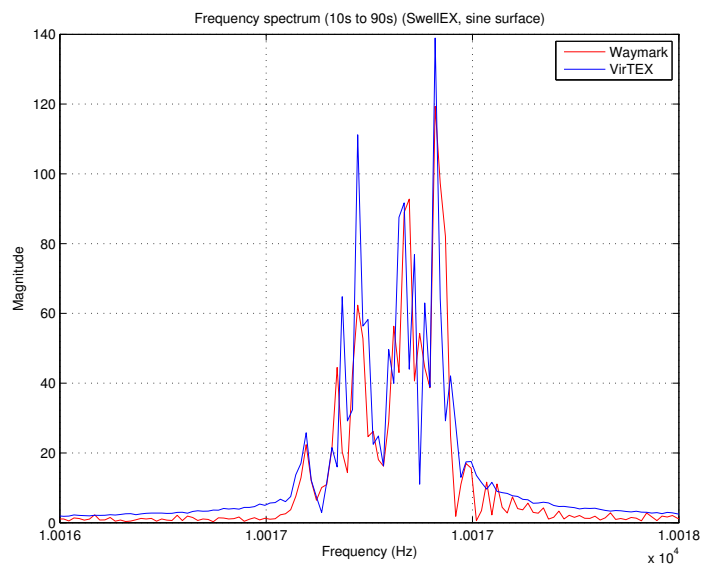


Fig. B.15 The received signal spectrum from the Waymark and VirTEX models with the SWellEx environment and trajectory with a sinusoidal surface model.

fewer computer resources are used at any one time, therefore allowing very long simulations to be performed. This local support approach also allows the problem to be segmented with the potential for parallel processing of the signal.

A set of experiment scenarios have been considered, in which the Waymark and VirTEX models are compared. The results show similarity with a qualitative comparison and the major features such as the Doppler shifts and delays being the same. It is not expected that

the results show perfect agreement, since different interpolation procedures are used in each of the models.

B.4 Channel modelling for imaging

The experiments presented in this chapter have been focused on long range channels, however by using a different propagation simulator different underwater scenarios can be considered. For FLSs, shorter ranges and higher frequencies are more appropriate, with a high level of detail for the representation of the environment. An example simulator is the k-Wave simulator designed for medical ultrasound [133]. Fig. B.16 shows a scene consisting of a flat seabed with two objects (shown in magenta) and a transducer array (shown in yellow). A simulation of this environment would show the interactions of the acoustic wave (sonar ping) with the objects on the sea floor. These interactions would form part of the modelled DM generated from the simulation.

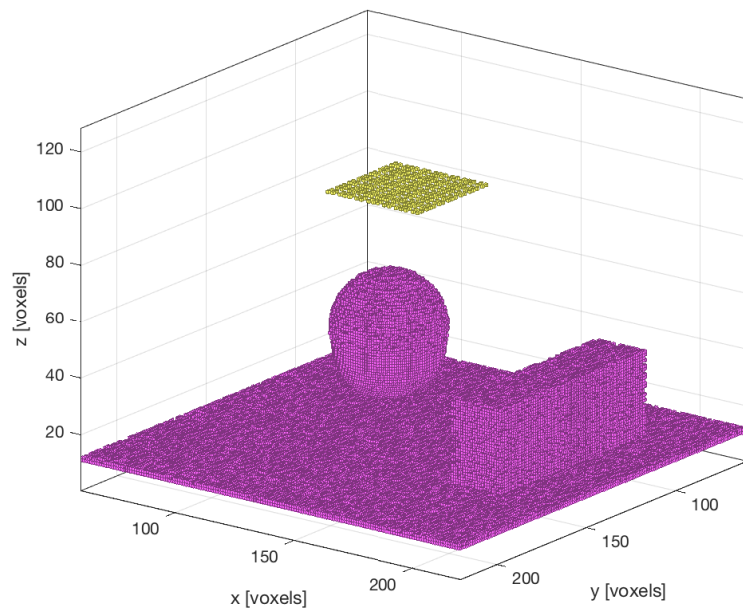


Fig. B.16 An example of a modelled 3-D environment using the k-Wave simulator [133]. The seabed is represented in magenta and the transducer array is shown in yellow. Note the rough surface of the seabed results in diffuse reflections.

B.5 Conclusion and discussion

Simulations and the resultant channel models are useful because they are inexpensive when compared to field trials. They are versatile because different sea conditions can be simulated easily by altering the setup, or the same condition can be simulated many times which is useful for system development. For any channel model there is a trade off between higher accuracy and lower execution time, this trade off being particularly apparent for real-time applications [134]. Therefore, algorithms that make the channel model more efficient are very important. For the Waymark model proposed in [109] greater efficiency was achieved through improvements in the interpolation of the results from the physics simulation. This improved interpolation meant that less simulations were required for a given signal length, thus reducing the complexity of the channel model overall. However, when using the channel model with a signal, all of the processing was performed at the signal sampling frequencies. For a signal where the baseband signal is of interest a significant reduction in the complexity can be achieved by processing the signal at baseband frequencies. The modification to the Waymark model detailed in this chapter successfully allowed the processing of the signal in the baseband frequencies range. The implementation of the proposed modified Waymark model was successfully validated against a more established channel model [120] with the results being alike.

The Waymark channel model shows the effect on a signal that comes from a changing underwater environment. For the work on navigation using FLS it is the motion of the transmitter and receiver that is of interest. If an environment is known, such as a ship's hull then the modelled DM associated with a platform movement could be simulated to better match the results with an estimated inter-frame DM.

In order to achieve this, further work on the Waymark model to give the ability to have uneven spacing of the waymarks in time would be necessary. For longer duration channel models this would allow more difficult areas of the transmitter/receiver trajectory (such as rapid accelerations) to be described by a greater density of waymarks. Conversely, where the the motion was slower there could be larger time gaps between the waymarks, improving the runtime. This change would also allow the model to place the waymarks in space instead of time. An alternative that has been developed in [6] generates a grid of points using the physics simulator. From these points each waymark is interpolated at the appropriate position in the receiver/transmitter trajectory.