Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingeniería y Sistemas de
Telecomunicación

Contributions to robotic underwater perception and mapping based navigation

Doctoral Thesis

Xin Yuan

Master in Ingeniería de Sistemas y Servicios para la Sociedad de la
Información

2018

Universidad Politécnica de Madrid

POLITÉCNICA

Escuela Técnica Superior de Ingeniería y Sistemas de

Telecomunicación

Departamento de Ingeniería Telemática y Electrónica

Contributions to robotic underwater perception and mapping based navigation

Doctoral Thesis

Xin Yuan

Master in Ingeniería de Sistemas y Servicios para la Sociedad de la
Información

Supervisors:

Prof. Ph.D. José-Fernán Martínez Ortega

2018

Doctorando en Ingeniería de Sistemas y Servicios para la Sociedad de la Información

Tesis Doctoral		
Título	Contributions to robotic underwater perception and mapping based navigation	
Autor	Xin Yuan	
Tutor	Dr. José-Fernán Martínez Ortega	VºBº
Ponente		
Tribunal		
Presidente		
Secretario		
Vocal		
Vocal		
Vocal		
Suplente		
Suplente		
Fecha de lectura	E.T.S.I. y Sistemas de Telecomunicación (UPM)	
Calificación		

El Presidente

La secretaria

Los vocales

To my beloved family.

Acknowledgements

This thesis would not have been possible without the supports from many people. Besides the knowledge I have acquired in the past three years, it has been a lot of fun to work with my colleagues in the “Next-Generation Networks and Services” research group at Universidad Politécnica de Madrid.

First and foremost, I would like to express my sincere gratitude to my supervisor Prof. José-Fernán Martínez Ortega for his extraordinary and selfless supervision and support during my whole Ph.D. life. Prof. José-Fernán encouraged me during difficult times, and inspired me when writing scientific papers and working on research projects. Most importantly, he provided me the perfect balance of guidance and freedom during the past years. Prof. José-Fernán is more than professors and tutors acting as my friend as well. I would also like to appreciate him for spending countless time to read this thesis. I have to admit that this thesis cannot be accomplished without his contributions and help. Without his supports, I would not be where I am today.

I am also deeply grateful to Prof. Martina Eckert, Prof. Juana Sendra Pons and Mohammed Al-Rawi, who contributed to ideas, insights and important feedback. It was my pleasure to discuss with them, and learn lots of techniques from them as well.

Moreover, I want to appreciate the supports from my research colleagues. They are more knowledgeable and skilled in programming than me. Their help and care for my research means a lot for me. Also everyone is so kind-hearted and friendly, working in such perfect atmosphere is a joy.

Finally, no one deserves my deepest thanks more than my family, for their enormous love and support to me. Special thanks also are due to my beloved parents, Shuping Yuan and Guiling Qin, they always support me either financially or emotionally. Although now we are in different countries and have six hours’ time difference, they always comfort me whenever I feel desperate and make me to firmly believe that I am never alone because they are always there caring for me. I would never have done this without you.

Abstract

In recent years, one of the fundamental challenges of the research on robotics is to obtain robust and efficient mechanisms for modelling increasingly complex environments, using mobile robots for their exploration. Nowadays, underwater vehicles are increasingly being used in complicated and rigid environments like oceans, harbors or at dams, such as using the on-board Side Scan Sonars (SSSs) of the Autonomous Underwater Vehicles (AUVs) to image the seabed. And underwater robotic mapping has been an active research area in robotic research community. Localization and mapping are the fundamental abilities for underwater robots to carry out exploration and searching tasks autonomously. Autonomous localization and mapping requires a vehicle to start at an unknown location in an unknown environment and then to incrementally build a map of landmarks present in this area while simultaneously using this map to compute absolute vehicle position. The problem of solving both map building and robot localization is addressed by Simultaneous Localization and Mapping (SLAM).

The main focus of this thesis is on extracting features with Sound Navigation And Ranging (SONAR) sensing for further robotic underwater landmark-based SLAM navigation. First and foremost, a detailed overview of currently used and popular solutions for the underwater SLAM problem is presented and characteristics like accuracy, robustness, computational complexity, etc. are compared. Besides, different types of commonly used map representations are compared regarding their suitability for a priori map localization, in particular with regard to large-scale underwater SLAM based navigation, which are computational requirements, reliability, robustness, etc. In our case, consider the sparse spatial distribution of the marine features, thus the landmark map is chosen to represent the underwater region to be explored.

According to the characteristics of sonar images, we propose an improved Otsu Threshold Segmentation Method (TSM) for fast and accurate detecting underwater objects of various feature shapes, including a shipwreck, a branch and a plastic mannequin in this thesis. For all the SSS and Forward Looking Sonar (FLS) images of different resolutions and qualities presented in this thesis, simulation results show that the computational time of our improved Otsu TSM is much lower than that of the maximum entropy TSM, which achieves the highest segmentation precision than other three classic TSMs including the traditional Otsu method, the local TSM and the iterative TSM. Experimental results justify that the improved Otsu TSM maintain more complete information and details of the objects of interests after segmentation, also the effect of noise, the holes and gaps in the foreground objects are greatly reduced.

Furthermore, the MIXed exponential Regression Analysis (MIRA) method is presented for handling the echo decay in sonar images. Our MIRA model is compared with the Dark Channel Prior (DCP) model, which is an adaption to well-known fog removal technique to sonar imaging, in terms of one ping, local similarity and global quality analysis between a landmark and its 180° - rotated counterpart. Simulation results prove that the proposed MIRA approach has superior normalization performances. In addition, a detailed state of the art underwater feature matching methods are summarized and compared, including the classic Harris, Speeded Up Robust Features (SURF), Binary Robust Invariant Scalable Keypoints (BRISK), Scale Invariant Feature Transform (ASIFT) feature detector. Consider their suitability for subsea sonar map fusion, in terms of computational requirements, reliability, accuracy, etc. We propose to employ the underwater wireless sensor network (UWSN) into the sonar map fusion task, and present UWSN-based Delaunay Triangulation (UWSN-DT) algorithm for enhancing the performances of sonar map fusion accuracy with low computational costs. Experimental results justify that the UWSN-DT approach works efficiently and robustly, especially for the subsea environments where distinguishable feature points are few and difficult to be detected.

Most importantly, as a result of the segmentations, the centroids of the main extracted regions are computed to represent point landmarks which can be used for navigation, e.g., with the help of our newly proposed Augmented Extended Kalman Filter (AEKF)-based robotic underwater SLAM algorithm, which stores the robot pose and the map landmarks in a single system state vector, and estimates the state parameters by using a iterative, estimation-update process, which besides a prediction, an update stage (as well as in the classic Extended Kalman Filter), includes a newly proposed augmentation stage. Several MATLAB simulated experiments are carried out for both our proposed AEKF-SLAM based navigation algorithm and classic FastSLAM 2.0 for dense loop mapping and line mapping, where environmental landmarks include not only the calculated centroids of the shipwreck, branch and plastic mannequin, but also those centroids of certain parts of the background detected by the proposed improved Otsu TSM in Section 3. Simulation results prove that for both dense loop mapping and line mapping experiments, our AEKF-SLAM based robotic navigation approach has the best performances of localization and mapping accuracy with relatively low computational costs. The AEKF achieves more precise and robust estimations of the robot pose and the landmark positions on account of the landmark positions detected by our improved Otsu TSM, than those derived by the maximum entropy TSM. In all, our presented AEKF-based robotic underwater SLAM algorithm achieves reliable detection of cycles in the map and consistent map update on loop closure.

Keywords: Simultaneous Localization and Mapping (SLAM); Underwater Object Detection; Sonar Image Processing, Augmented Extended Kalman Filter (AEKF); FastSLAM 2.0; Speeded Up Robust Features (SURF); Computational Complexity

Resumen

En los últimos años, uno de los desafíos fundamentales de la investigación en robótica es obtener mecanismos robustos y eficientes para modelar entornos cada vez más complejos, utilizando robots móviles para su exploración. Hoy en día, los vehículos submarinos se usan cada vez más en entornos complicados y rígidos, como océanos, puertos o represas, como el uso de los sonar de barrido lateral (SSS) de los vehículos submarinos autónomos (AUV) para crear imágenes del lecho marino. Y el mapeo robótico subacuático ha sido un área de investigación activa en la comunidad de investigación robótica. La localización y el mapeo son las habilidades fundamentales para que los robots subacuáticos realicen tareas de exploración y búsqueda de forma autónoma. La localización y el mapeo autónomos requieren que un vehículo comience en un lugar desconocido en un entorno desconocido y luego construya incrementalmente un mapa de los puntos de referencia presentes en esta área al mismo tiempo que utiliza este mapa para calcular la posición absoluta del vehículo. El problema de resolver tanto la creación de mapas como la localización de robots se aborda mediante localización y asignación simultáneas (SLAM).

El objetivo principal de esta tesis es extraer las características con la detección de sonido y rango (SONAR) de detección para la navegación SLAM basada en hitos subacuático robótica adicional. Primero, se presenta una descripción detallada de las soluciones actualmente utilizadas y populares para el problema de SLAM submarino y se comparan características tales como precisión, robustez, complejidad computacional, etc. Además, se comparan diferentes tipos de representaciones de mapas de uso común con respecto a su idoneidad para el mapeo a priori, en particular con respecto a la navegación basada en SLAM subacuático a gran escala, que son requisitos de cómputo, fiabilidad, solidez, etc. En nuestro caso, tenga en cuenta la distribución espacial dispersa de las características marinas, por lo que el mapa de referencia se elige para representar la región submarina que se explorará.

De acuerdo con las características de las imágenes del sonar, proponemos un Método de Segmentación de Umbral Otsu (TSM) mejorado para la detección rápida y precisa de objetos submarinos de varias formas características, incluyendo un naufragio, una rama y un maniquí de plástico en esta tesis. Para todas las imágenes SSS y Forward Looking Sonar (FLS) de diferentes resoluciones y cualidades presentadas en esta tesis, los resultados de la simulación muestran que el tiempo de cálculo de nuestro TSM de Otsu mejorado es mucho menor que la TSM de entropía máxima, lo que permite precisión de segmentación que otros tres TSM clásicos, incluido el método tradicional de Otsu, el TSM local y el TSM iterativo. Los

resultados experimentales justifican que la TSM mejorada de Otsu mantenga la información más completa y los detalles de los objetos de interés después de la segmentación, también el efecto del ruido, los agujeros y las brechas en los objetos en primer plano se reducen en gran medida.

Además, se presenta el método de análisis de regresión exponencial mezclado (MIRA) para manejar el decaimiento del eco en las imágenes del sonar. Nuestro modelo MIRA se compara con el modelo Dark Channel Prior (DCP), que es una adaptación a la bien conocida técnica de eliminación de niebla para imágenes sonar, en términos de un ping, similitud local y análisis de calidad global entre un hito y su 180° - contraparte girada. Los resultados de la simulación demuestran que el enfoque MIRA propuesto tiene mejores resultados de normalización. Además, se resumen y comparan los métodos avanzados de comparación de características subacuáticas, que incluyen el clásico Harris, Funciones rápidas y rápidas (SURF), Puntos clave escalables invariables binarios (BRISK), Transformación de característica invariable de escala (SIFT) y detector de característica de transformación de característica invariable de escala afin (ASIFT). Considere su idoneidad para la fusión de mapas de sonar submarinos, en términos de requisitos computacionales, confiabilidad, precisión, etc. Proponemos emplear la red de sensores inalámbricos subacuáticos (UWSN) en la tarea de fusión de mapas de sonar y presentar Triangulación de Delaunay basada en UWSN (UWSN). DT) Algoritmo para mejorar el rendimiento de la precisión de fusión del mapa de sonar con bajos costos computacionales. Los resultados experimentales justifican que el enfoque UWSN-DT funcione de manera eficiente y sólida, especialmente para los entornos submarinos donde los puntos característicos distinguibles son pocos y difíciles de detectar.

Lo que es más importante, como resultado de las segmentaciones, los centroides de las principales regiones extraídas se calculan para representar puntos de referencia que pueden usarse para la navegación, por ejemplo, con la ayuda de nuestro recién propuesto filtro de Kalman ampliado aumentado (AEKF) basado en robótica submarina Algoritmo SLAM, que almacena la posición del robot y los puntos de referencia del mapa en un único vector de estado del sistema, y estima los parámetros de estado mediante un proceso iterativo de actualización de estimación que, además de una predicción, una etapa de actualización (así como en el clásico Extended Kalman Filter), incluye una nueva etapa de aumento propuesta. Se realizan varios experimentos simulados de MATLAB para nuestro algoritmo de navegación basado en AEKF-SLAM propuesto y FastSLAM 2.0 clásico para mapeo de mallas densas y mapeo de líneas, donde los hitos ambientales incluyen no solo los centroides calculados del naufragio, la rama y el maniquí de plástico, sino también centroides de ciertas partes del fondo detectadas por la propuesta Otsu TSM mejorada en la Sección 3. Los resultados de simulación demuestran que

tanto para el mapeo de bucle denso como para el mapeo de líneas, nuestro enfoque de navegación robótica basada en AEKF-SLAM tiene los mejores resultados de localización y precisión de mapeo costos computacionales relativamente bajos. El AEKF logra estimaciones más precisas y robustas de la postura del robot y las posiciones históricas a causa de las posiciones históricas detectadas por nuestro TSM de Otsu mejorado, que aquellas derivadas por la TSM de máxima entropía. En total, nuestro algoritmo de SLAM robótico subacuático basado en AEKF logra una detección confiable de ciclos en el mapa y una actualización de mapas consistente en el cierre de bucle.

Palabras clave: Localización y Mapeo Simultáneos (SLAM); Detección de Objetos Bajo el Agua; Procesamiento de Imágenes de Sonar, Filtro de Kalman Ampliado Aumentado (AEKF); FastSLAM 2.0; Funciones Robustas y Aceleradas (SURF); Complejidad Computacional.

Table of Contents

Acknowledgements.....	V
Abstract	VI
Resumen	IX
Table of Contents	XII
List of Figures	XV
List of Tables.....	XIX
Glossary	XXI
1 Introduction.....	1
1.1. Introduction	2
1.2. Motivation.....	4
1.2.1. Navigation	5
1.2.2. Systematic exploration	5
1.2.3. Communication	6
1.3. Problem statement and state of the art	6
1.3.1. AUV navigation and localization techniques	6
1.3.2. Robotic mapping	10
1.3.3. Solutions to the SLAM problem	13
1.3.4. Challenges in underwater SLAM	19
1.4. Research contributions and thesis structure.....	21
2 Navigational map representations.....	25
2.1. Navigational maps and their applications to underwater SLAM	26
2.2. The occupancy grid maps	27
2.3. The topological maps	29
2.4. The landmark maps	30
2.5. Summary.....	32
3 Underwater feature detection	33

3.1. Sensor selection	34
3.2. Sonar image characteristics	35
3.2.1. Side scan sonar images.....	37
3.3. The proposed improved Otsu TSM	38
3.3.1. The power-law transformation.....	41
3.4. TSM results for sonar images.....	41
3.4.1. TSM results for high-resolution SSS image.....	42
3.4.2. TSM results for low-resolution SSS image.....	46
3.4.3. TSM results for FLS image.....	50
3.5. Summary.....	54
 4 Underwater sonar map fusion	55
4.1. Intensity normalization of SSS imagery.....	56
4.1.1. The tested SSS image data	57
4.1.2. The proposed MIXed exponential Regression Analysis (MIRA) method.....	60
4.1.3. Dark Channel Prior (DCP) method	63
4.1.4. Local similarity analysis	63
4.1.5. Global quality analysis.....	66
4.2. Underwater feature matching approaches	67
4.2.1. SURF, SIFT and ASIFT keypoint based underwater SSS image matching	69
4.2.2. Simulation results of SURF keypoint matching.....	71
4.2.3. Simulation comparison of SURF, SIFT and ASIFT keypoint matching performances	74
4.3. Wireless Sensor Network (WSN) based underwater sonar image fusion	76
4.3.1. Underwater Wireless Sensor Network (UWSN).....	77
4.3.2. Simulation results of the communication between underwater nodes and AUV	81
4.3.3. Building the Delaunay triangulated mesh on the detected nodes	82
4.4. Summary.....	85
 5 The posterior-estimated AEKF-SLAM system.....	86
5.1. Vehicle model.....	87
5.2. Feature model	87

5.3. Observation model.....	88
5.4. Simultaneous Localization and Mapping (SLAM)	89
5.4.1. The SLAM process	90
5.4.2. Loop closing.....	91
5.5. Augmented Extended Kalman Filter (AEKF)	93
5.5.1. Time update.....	94
5.5.2. Measurement update	94
5.6. The estimation process of the AEKF-SLAM	95
5.6.1. Vehicle, map and augmented state vectors	98
5.6.2. Prediction stage	100
5.6.3. Update stage	101
5.6.4. Augmentation Stage	103
5.7. Summary.....	104
 6 The AEKF-SLAM and the FastSLAM simulated experiments ...	106
6.1. The AEKF-SLAM and FastSLAM 2.0 based underwater robotic navigation simulations.....	107
6.1.1. Dense loop map.....	107
6.1.2. Line map.....	110
6.2. AEKF-SLAM dense loop map using landmarks detected by the improved Otsu TSM.....	112
6.3. SWARMs seabed mapping use case	116
6.4. Summary.....	119
 7 Conclusions and future work.....	121
7.1. Conclusions	122
7.2. Future work	125
 References	127
List of Author's Publications and Projects.....	137

List of Figures

1.1	Outline of underwater navigation classifications. The approaches are often combined to provide enhanced performances.....	8
1.2	The problem of robotic localization and mapping.....	11
1.3	Bayes net for robotic localization.....	12
1.4	Bayes net for robotic mapping.....	12
2.1	The occupancy grid map of the San José Tech Museum, in [18].....	28
2.2	The topological map.....	29
2.3	The landmark map.....	31
3.1	Sonar images with different colormaps.....	36
3.2	(a) High-resolution SSS images recorded with the DE340D SSS at Stockholm sea; Low-resolution SSS image generated by the 3500 Klein SSS (ECA Group company [82]).....	38
3.3	The procedure of the improved Otsu TSM.....	39
3.4	The plots of the power-law equation with different r values.....	41
3.5	(a) Traditional Otsu TSM, $T_h = 0.3216$; (b) Local TSM, $T_h = 0.1628$; (c) Iterative TSM, $T_h = 0.4238$; (d) Maximum entropy TSM, $T_h = 0.6627$	42
3.6	(a) Canny edge detection after applying the traditional Otsu method, $bw = \text{edge}(b, 'canny', 0.33)$, $N_{30} = 752 > 300$; (b) Improved Otsu TSM, $T=0.3216$, $T^* = 0.6784$; (c) Result of the improved Otsu TSM after morphological operations marking the centroids of the obtained regions; (d) Result of the maximum entropy TSM after the same morphological operations marking the centroids of the acquired areas.....	44
3.7	(a) Traditional Otsu TSM, $T_h = 0.1137$; (b) Local TSM, $T_h = 0.0941$; (c) Iterative TSM, $T_h = 0.2609$; (d) Maximum entropy TSM, $T_h = 0.3176$	46
3.8	(a) Canny edge detection after applying the traditional Otsu method, $bw = \text{edge}(b, 'canny', 0.1255)$, $N_{15} = 419 > 100$; (b) Improved Otsu TSM, $T=0.1137$, $T^* = 0.3529$; (c) Result of the improved Otsu TSM after morphological operations marking the centroids of the obtained regions; (d) Result of the maximum entropy TSM after the same morphological operations marking the centroids of the acquired areas.....	47
3.9	The original FLS image comes from [89], and there is a plastic mannequin in the down center.....	50
3.10	(a) Traditional Otsu TSM, $T_h = 0.1176$; (b) Local TSM, $T_h = 0.0941$; (c) Iterative TSM, $T_h = 0.2990$; (d) Maximum entropy TSM, $T_h = 0.4118$	51

3.11	(a) Canny edge detection after applying the traditional Otsu method, $bw = \text{edge}(b, \text{canny}', 0.13)$, $N_{40} = 1341 > 600$; (b) Improved Otsu TSM, $T=0.1176$, $T^* = 0.5412$; (c) Result of the improved Otsu TSM after morphological operations marking the centroids of the obtained regions; (d) Result of the maximum entropy TSM after the same morphological operations marking the centroids of the acquired areas.....	51
4.1	The scanning paradigm used to record the seabed images in Lake Vättern, Sweden. Overlapping regions are marked with a similar canvas pattern. Colours have no meaning here but visual clarity. The arrow exemplifies the movement of the vehicle carrying the imaging sonar.....	58
4.2	One original SSS image (WE3) of the seabed displayed without further processing. The vertical axis shows the total number of pings, and the horizontal axis shows the number of samples (1000 in the left, and 1000 in the right), although -50 to 50 in the area covered in meters, $\log(x)$ has been used to compress the received acoustic signal x	58
4.3	One ping showing some form of exponential decay in the received acoustic signal, where $\log(x)$ has been used to compress the original values.....	58
4.4	Extraction of corresponding pairs from, the original image (left), image intensity normalized with the DCP method (middle), and image intensity normalized with MIRA method (right). We see how the inhomogeneous echo decay produces an irregular illumination field in regions containing natural seabed landmarks, left part. This artefact seems to be decreased after applying the proposed methods, although the DCP method introduces a considerable amount of artificial contrast in the extracted regions. The shown above are corresponding pairs of distinguished landmarks. (a) Two landmark pairs $\{(a_1, a_2), (b_1, b_2)\}$ in original images; (b) Two landmark pairs $\{(\alpha_1, \alpha_2), (\beta_1, \beta_2)\}$ in DCP; (c) Two landmark paris $\{(a_1, a_2), (b_1, b_2)\}$ in MIRA.....	61
4.5	Four of the 32 pairs contained in the benchmark set. Distinctive landmarks from the original SSS images, same landmarks after illumination correction by the DCP method and by the MIRA method are shown. It is obvious that some SSS images are difficult to recognize even for the human observer. For illumination purpose, the brightness and contrast of the images shown in this figure have been set to 35%, and the picture boarder of each image was set to black. (a) Non-normalized SSS images; (b) SSS images normalized by the DCP; (c) SSS images normalized by the MIRA.....	62
4.6	Using image WE3. (a) Ping at index 500 used to estimate the MIRA model, where two models are estimated one for the left scan/ping and another for the right scan/ping; (b) Ping 500 after normalizing it with	

	MIRA; (c) Another ping from the same image at index 4000; (d) After normalizing ping 4000 with the MIRA model estimated from ping 500; (e) Ping 500 normalized with the DCP method; (f) Ping 4000 normalized with the DCP.....	64
4.7	SSIM scores measuring the similarity of the two methods proposed in this chapter and compared to the non-normalized images. Images normalized with the MIRA method which is depicted as the dark thick line, outperforms the both the DCP method and the non-normalized images.....	65
4.8	AUV round trip scans with overlapping.....	71
4.9	180° rotation difference, 51 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 242 SURF keypoints detected.....	72
4.10	135° rotation difference, 27 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 255 SURF keypoints detected.....	72
4.11	90° rotation difference, 50 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 239 SURF keypoints detected.....	73
4.12	45° rotation difference, 24 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 267 SURF keypoints detected.....	73
4.13	20° rotation difference, 24 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 259 SURF keypoints detected.....	73
4.14	180° rotation difference, 74 SIFT matching pairs, 4.720715 s. The left SSS image scale 1, 502 SIFT keypoints detected; the right SSS image scale 1.2, 2366 SIFT keypoints detected.....	75
4.15	180° rotation difference, 331 ASIFT matching pairs, 4.720715 s. The left SSS image scale 1, 502 SIFT keypoints detected; the right SSS image scale 1.2, 2366 SIFT keypoints detected.....	75
4.16	(a) The diagram illustrating the assumed geometry of the AUV embedded SSS in seabed navigation; (b) The communication range and scan range of AUV; (c) The geometry relationship between the communication range and scan range of AUV.....	78
4.17	Simulation results of the communication between the nodes underwater nodes and AUV.....	82
4.18	Building the Delaunay triangulation on the detected nodes within the overlapped region between two SSS images (a) and (b).....	83
4.19	Constructing the Delaunay triangulated mesh on the detected nodes in	

	partial SSS image pairs where contains landmark pairs a_1 and a_2	84
5.1	A robot measuring relative observations to environmental landmarks.....	90
5.2	The flowchart of the SLAM process.....	90
5.3	The SLAM graphical model, redrawn after [123].....	91
5.4	(a) Before closing the loop; (b) After closing the loop.....	92
5.5	The AEKF estimator.....	95
5.6	The flow chart of SLAM procedure based on an AEKF.....	95
5.7	The architecture of AEKF-SLAM based robotic navigation system, as in [9].....	98
5.8	The robot motion model.....	100
5.9	The robot observation model.....	102
6.1	The robot is observing the landmarks A and B in the AEKF-SLAM dense loop map; (b) The robot is getting measurements A and B in the FastSLAM 2.0 dense loop map.....	108
6.2	(a) Partial magnification of the AEKF-SLAM line map; (b) Partial magnification of the FastSLAM 2.0 line map.....	110
6.3	(a) The robot is observing the centroids of certain parts of the body before loop closure; (b) The final AEKF-SLAM loop map where the landmarks are detected by the improved Otsu TSM.....	113
6.4	(a) The robot is observing the centroids of certain parts of the body before loop closure; (b) The final AEKF-SLAM loop map where the landmarks are detected by the maximum entropy TSM.....	115
6.5	The simulated SWARMs vehicles.....	118
6.6	Link all the actors for landmark localization and seabed mapping in the SWARMs project.....	119

List of Tables

1.1	List of pros and cons of filtering approaches applied to the SLAM framework.....	14
1.2	The summary of state of the art underwater SLAM approaches.....	18
3.1	Common instrumentations used in underwater SLAM.....	35
3.2	The confusion matrix of real and detected ship centroids using the improved Otsu TSM.....	45
3.3	The confusion matrix of real and detected ship centroids using the maximum entropy TSM.....	45
3.4	Computational costs of different segmentation methods on Figure 3.2 (a)...	46
3.5	The confusion matrix of real and detected branch centroids using the improved Otsu TSM.....	48
3.6	The confusion matrix of real and detected branch centroids using the maximum entropy TSM.....	48
3.7	Computational costs of different segmentation methods on Figure 3.2 (a)...	49
3.8	The confusion matrix of real and detected body centroids using the improved Otsu TSM.....	52
3.9	The confusion matrix of real and detected body centroids using the maximum entropy TSM.....	52
3.10	Computational costs of different segmentation methods on Figure 3.9.....	53
4.1	Mean and Standard Deviation of SSIM scores on the entire set of 19 landmark pairs in this study.....	65
4.2	Results of the SIQEM metric on each of the five SSSs in the used dataset, the lower SIQEM value the better image quality.....	66
4.3	The computation time and the number of matched SURF feature pairs with different scales and rotation angles.....	74
4.4	The number of detected SURF, SIFT, ASIFT keypoint matching pairs and their computational complexity.....	76
4.5	The number of detected SURF, SIFT, ASIFT keypoint matching pairs and their computational complexity.....	81
5.1	The AEKF operations for achieving underwater SLAM.....	97
6.1	The comparisons of the computational time and the estimated landmark A, B positions in the dense loop map derived by the AEKF-SLAM and FastSLAM 2.0.....	109
6.2	The comparisons of the computational time and the estimated landmark A, B, C, D positions in the line map derived by the AEKF-SLAM and FastSLAM 2.0.....	112
		XIX

6.3	The landmark point positions of the ship, branch and body estimated by the AEKF and the true ones detected by the improved Otsu TSM.....	114
6.4	The landmark point positions of the ship, branch and body estimated by the AEKF and the true ones detected by the maximum entropy TSM.....	116
6.5	The parameters of the underwater vehicles (AUVs and ROVs) employed in the SWARMs environmental sensing mission.....	117

Glossary

SSS: Side Scan Sonar

AUV: Autonomous Underwater Vehicle

SLAM: Simultaneous Localization and Mapping

SONAR: Sound Navigation And Ranging

TSM: Threshold Segmentation Method

FLS: Forward Looking Sonar

MIRA: Mixed exponential Regression Analysis

DCP: Dark Channel Prior

SURF: Speeded Up Robust Features

BRISK: Binary Robust Invariant Scalable Keypoints

SIFT: Scale Invariant Feature Transform

ASIFT: Affine Scale Invariant Feature Transform

UWSN: Underwater Wireless Sensor Network

UWSN-DT: Underwater Wireless Sensor Network-based Delaunay Triangulation

AEKF: Augmented Extended Kalman Filter

EKF: Extended Kalman Filter

CML: Concurrent Mapping and Localization

GPS: Global Positioning System

DGPS: Differential Global Positioning System

PF: Particle Filter

EM: Expectation Maximization

DVL: Doppler Velocity Log

TDMA: Time-Division Multiple-Access

LBL: Long Base Line

SBL: Short Base Line

USBL: Ultra Short Base Line

ToF: Time of Flight

INS: Inertial Navigation System

DEM: Discrete Elevation Map

IMU: Inertial Measurement Unit

KF: Kalman Filter

PF: Particle Filter

CEKF: Compressed Extended Kalman Filter

IF: Information Filter

EIF: Extended Information Filter

MMSE: Minimum Mean-Square Error

GRV: Gaussian Random Variable

UKF: Unscented Kalman Filter

UT: Unscented Transform

MLE: Maximum Likelihood Estimate

SMC: Sequential Monte-Carlo

USAR: Urban Search and Rescue

RBPF: Rao-Blackwellized Particle Filter

MSIS: Mechanical Scanned Imaging Sonar

MES: Multibeam Echo Sounder

FPR: False Positive Rate

PPV: Positive Predictive Value

WE: WestEast

EW: EastWest

SSIM: Structural Similarity Index Metric

SIQEM: Sonar Image Quality Assessment Metric

DoG: Difference of Gaussians

RANSAC: Random Sample Consensus

SM: Stochastic Mapping

RSSI: Received Signal Strength Indication

RMS: Root Mean Square

IRS: Inertial Reference System

ROVs: Remote Operating Vehicles

USVs: Unmanned Surface Vehicles

1

Introduction

1.1. Introduction

More than 70% of the Earth's surface is covered by water. Oceans and seas host an incredibly rich biodiversity almost 90% of all the known species, influence short and long-term climate and have a high impact not only on the economy, but on the life and evolution of human society as a whole. Exploring this large body of water is a matter of the utmost importance, not only because it represents a vast source of natural resources, but also because its study may help us to understand how this complex ecosystem works. In the near future, the ocean will supply a substantial part of human and industrial needs: the oil and gas industry will move into deeper waters, the renewable energy will be harvested from the sea, as well as many other innovative practices will become common. Furthermore, minerals such as cobalt, nickel, copper, rare earths, silver, and gold will be mined from the seafloor. One major challenge in this context is to develop systems and solutions which could guarantee a sustainable development of the maritime activities so that this fragile habitat is protected for future generations. Therefore, the subsea environment is very interesting for exploration, but also one of the most challenging environments, since many activities such as installation and monitoring of oil pipelines are performed by underwater robots. However, the limited range of light with different spectrum than daylight, the presence of marine currents, uniformity of the underwater environment, water pressure and turbidity are some of the problems faced in robotic underwater applications for localization and mapping.

Autonomous Underwater Vehicles (AUVs) have been widely used in oceanographic studies and for military purposes for several years. To date, they are increasingly employed to explore rigid and complicated environments like oceans, harbors or at dams, such as using the on-board Side Scan Sonars (SSSs) of the AUVs to image the seabed. There exists a cyclic nature problem in the robotic mapping and localization: while operating in an unknown environment, in order to build a map of the uncharted territory, the fully autonomous robot needs to know its location, but therefore it requires a map. This chicken and egg problem is often referred to as Simultaneous Localization and Mapping (SLAM) or Concurrent Mapping and Localization (CML). Robotic underwater SLAM has a wide breadth of applications, including coral reef inspection and monitoring [1,2], inspection of archaeological sites [3], geological surveying [4], and profiling of northern icebergs [5,134,135].

Nowadays, the robotic research community has attached great importance to the underwater robotic mapping and navigation problem. The most crucial and the most essential requirement for an autonomous robot is having a precise and detailed map. The robot creates a spatial representation of its working environment from a sequence of on-board navigation sensor

measurements as it surveys that area. This is generally regarded as one of the most important problems in the pursuit of building truly autonomous robots. Accurate estimation of the robot pose is simple, if it is possible to incorporate the Global Positioning System (GPS) or Differential Global Positioning System (DGPS) sensors into the robot. But GPS is unavailable for indoor, underground and underwater robotic applications. In underwater environments, it is also not possible to use lasers or cameras due to poor lighting conditions. And underwater nature imposes a series of challenges and problems, which are not faced by terrestrial or indoor applications. However, with the development of the imaging sonar technology, it is possible to obtain the underwater information by sonar images. After filtering, smoothing, segmenting the sonar image, and then extracting features and recognizing objects, eventually the underwater landmark map will be generated.

In the research area of computer vision, landmark extraction is the preliminary step of many vision tasks such as object recognition, localization, tracking and mapping. Manufactured environments are typically composed of easy to recognize geometric features (planes, lines, corners, etc.), but in subsea environments usually no such particular distinguishable features can be found. Considering the restrictions of underwater perception and localization sensors, which are affected by accumulated uncertainty over long-term manipulations, the implementation of robotic underwater SLAM-based navigation is still a tough research topic. Moreover, acoustic sensors for underwater mapping usually provide noisy and distorted images or low-resolution ranging, while video sensors could return highly detailed images but are always limited by water turbidity and poor illumination conditions [6,7].

There are several types of sensors that can be used to obtain distance measurements, such as laser sensors, infrared sensors, digital cameras, and Sound Navigation And Ranging (SONAR) sensors. In our previous works [8–13], we focused on determining underwater geometric features with sonar sensing. Sonar sensors have been widely used in autonomous mobile vehicles, since sound propagates larger distances than electromagnetic waves, sonar imaging outperforms optical imaging in underwater. Especially the side-scan-sonars (SSSs) are increasingly applied in industry and academic research activities to survey the seafloor [14]. High-resolution SSSs create a 2D photorealistic image of the surveyed seafloor. As long as the sonar scans are incorporated through mosaicking, we could recognize the detected underwater objects easily and these feature interpretation offers valuable insights into the study of the seabed topography [15]. The SSSs create higher qualitative sonar images than the Forward Looking Sonars (FLSs), since SSSs scan the seafloor at a small grazing angle. Generally, the SSS and the FLS could generate large-scale maps on the area of the seabed, which are typically processed for obstacle detection and feature extraction [16], such as in [136], the authors use an

imaging FLS for autonomous underwater ship hull inspection. The sonar images with high resolutions are of crucial importance in decision-making to investigate a detected target or not. Several approaches tackle the issues related to object detection and classification, image segmentation and registration, and map fusion.

The research of estimation approaches for the robotic underwater SLAM-based navigation problem has caught utmost attention by the research community. Among the solutions presented in the literature to solve the underwater SLAM problem, the Bayesian filters-based estimation methods have been the most successful ones over time [17–19]. While most of the successful mapping and navigation techniques are probabilistic [20–22], there is a wide range of ways in which a probabilistic approach can be implemented. At present, there are several types of probabilistic algorithms that have been the focus of research, and the most commonly used ones are Extend Kalman Filter (EKF), Particle filter (PF) and Expectation Maximization (EM). It is difficult to judge which one is the best. Techniques vary in their complexity, assumptions about the environment, such as the area of coverage and the number of features, also sensor usage, computational requirements, flexibility across robot platforms, scalability, and many other characteristics. These three algorithms are mathematical derivations of the recursive Bayes rule. The main reason for this probabilistic techniques popularity is the fact that robot mapping is characterized by uncertainty and sensor noise, and probabilistic algorithms tackle the problem by explicitly modeling different sources of noise and their effects on the measurements. The Stochastic SLAM is subject to three dominating defects: high computational load and storage costs, fragile and even wrong data association, and inconsistent update of non-linearity. Keeping relationships between the robot poses and all correlated landmark positions in the whole system state covariance matrix, leads to high computational requirements and space consumption problem. Although some approaches have been proposed to deal with the underwater environment, the robotic SLAM applications for it still remain unsolved.

1.2. Motivation

Robotic mapping is concerned with developing techniques that enable a mobile robot to construct and maintain a model of its exploration environment based on spatial information measured over time, so that robots can know and understand the physical environment where specific operations are to take place and also be able to adapt to any change occurred in this dynamic environment. Building a map is a key requirement for a robot to operate in an unknown region where prior information of the environment is unavailable. Most approaches to robotic mapping are incremental in the sense that a new observation is used to adjust the current spatial model, leading to a new model. The observation is then discarded. The adjustment of the

model when new information about the environment becomes available can be regarded as a two-step process: First and foremost, in the localization step, corresponding features contained in the new information and in the current model are identified (data association) and the robot position within the model is updated based on the detected correspondences (position update). Second, in the map merging stage, the spatial model information is complemented and updated, based on the results derived by the localization step. The present spatial model serves as the input for the data association and map merging steps and is modified by the map merging stage.

The problem of acquiring and maintaining a long-term spatial model of an environment cannot be studied in isolation and has to be considered in the context of all the operations that are to be supported by the map. Although the concrete set of operations clearly depends on the specific area of application, we believe that it is reasonable to consider the following three crucial tasks as future mobile robot fundamental applications: (1) navigation, (2) systematic exploration and (3) communication on the area.

1.2.1. Navigation

In [23], Montello defines navigation as a “coordinated and goal-directed movement of one’s self or one’s body through the environment.” The purpose of navigation is to move oneself in order to reach a particular known location. In 1990, Levitt and Lawton listed three fundamental questions that needed to be answered so as to achieve successful navigation: “Where am I?”, “Where are other places relative to me?”, and “How do I reach other places from here?” [24]. Therefore, it follows that a spatial representation needs to contain information that supports the ability to localize oneself with respect to environmental features, information on the positions of relevant places, and information that allows us to come up with solutions to reach these places. From the perspective of operations and spatial competencies, three main subtasks can be distinguished under the term navigation: localization, path planning, and locomotion. And only the first two items require a global spatial model of the environment. Locomotion consists of sensor-based movement through the immediate surroundings while avoiding obstacles. Wayfinding comprises all the higher-level cognitive processes involved in navigation, such as localization and path planning. While locomotion can be achieved reactively, wayfinding incorporates those abilities that rely on the spatial information stored in the memory.

1.2.2. Systematic exploration

Exploration sometimes is seen as a part of navigation, namely when an agent is supposed to find a certain goal position in an unfamiliar environment. This task requires an efficient systematic exploration approach. The most important aspect of exploration is to keep track of

covered parts and parts of the environment that still need to be explored in order to obtain complete coverage. Besides, it is often desirable to explore as efficiently as possible in terms of time or travel distance. In this case, information that supports the decision of which part to explore next needs to be provided by the map representation as well.

1.2.3. Communication

Communication with other agents on the area takes many forms. First and foremost, communication is categorized into the direct and indirect one. For the first circumstance, information is directly passed between different agents, typically in verbal form, possibly supported by gestures. In contrast, indirect communication makes use of external representations. Examples are sketches, written descriptions and all kinds of maps, such as topographical maps, you-are-here maps, or floor plans. The external representation is usually either graphical or verbal.

As for the second distinction, one could look at the purpose of the communicated information. Here, we could differentiate communication with the goal of conveying information about the environment from communication with the goal of providing navigation information like route instructions which are supposed to guide you to a particular place.

Finally, communicated spatial information can be position-related or irrelevant, depending on whether it is adapted to the receiving agent's current position or not. In order to communicate with agents that have completely different sensor and motion capabilities, a suitable level of representation needs to be established on which spatial information could be exchanged. To facilitate this, an agent's spatial representation needs to bridge from low-level representations tuned for its own technical abilities to more abstract levels of representation that other types of agents can utilize [25].

1.3. Problem statement and state of the art

1.3.1. AUV navigation and localization techniques

The development of Autonomous Underwater Vehicles (AUVs) began in the early of 1970s. As a new type of underwater equipment, the AUV has been widely used for activities such as oceanographic surveys, bathymetric data collection in marine and riverine environments, offshore oil exploration and demining. In many AUV applications, navigation is a critical issue. Some means of navigation, such as GPS and radio navigation, can only be applied effectively when on the surface [26]. Alternatively, the position of the AUV can be calculated based on

Doppler Velocity Log (DVL) by the method of dead-reckoning, but the growth of positional errors related with the method makes its use impractical for long-term navigation [27]. Although there is highly precise underwater navigation equipment such as inertial navigation, the cost has precluded its widespread use in AUVs until now.

AUV navigation and localization in underwater environments is particularly challenging due to the rapid attenuation of higher frequency signals and the unstructured nature of the undersea environment. Above water, most autonomous systems rely on radio or spread-spectrum communications and global positioning. However, underwater communications are low bandwidth and unreliable, and there is no access to a GPS. Such radio signals propagate only a short distance and acoustic sensors-based communications show better performances. Acoustic communications still suffer from many shortcomings as follows:

- (1) Small bandwidth, which means that communicating nodes have had to use Time-Division Multiple-Access (TDMA) techniques to share information;
- (2) Low data rate, which generally constrains the amount of data that can be transmitted;
- (3) High latency since the speed of sound in water is only 1500 m/s , which is much slower compared with the velocity of light;
- (4) Variable sound speed due to fluctuating water temperature and salinity;
- (5) Multiple transmissions due to the presence of an upper (free surface) and lower (sea bottom) boundary coupled with highly variable sound speed;
- (6) Unreliability, resulting in the need for a communication system designed to handle frequent data loss in transmissions.

Notwithstanding these primary challenges, research in AUV navigation and localization has exploded in the last ten years. Past approaches to solve the AUV localization problem have employed expensive inertial sensors, used installed beacons in the area of interests, or required periodic surfacing of the AUV. Although these solutions are useful, their performance is fundamentally limited. The field is in the midst of a paradigm shift from conventional technologies, such as Long Base Line (LBS), Short Base Line (SBL) and Ultra Short Base Line (USBL), which require pre-deployed and localized equipment, toward dynamic multi-agent system approaches, which allow for rapid deployment and flexibility with minimal infrastructure. Moreover, SLAM techniques, which are developed for above ground robotic applications are being increasingly employed to underwater systems. Advances in underwater

communications and the application of SLAM technology to the underwater field have yielded new possibilities. The result is that bounded error and accurate navigation for AUVs is becoming possible with less cost and overhead.

AUV navigation and localization techniques can be categorized into three main branches: Inertial Navigation, Acoustic Navigation and Geophysical Navigation, see Figure 1.1. The type of chosen navigation system is highly dependent on the type of operation or mission, and that in many cases different systems are incorporated to achieve improved performances. The most important considerations are the size of the area of interests and the desired localization precision. Until recently, the state-of-the-art approach has been combining acoustic and inertial navigation systems [137].

Figure 1.1 Outline of underwater navigation classifications. These approaches are often combined to provide enhanced performances

- (1) Inertial/Dead Reckoning: Inertial navigation is a self-contained navigation technique with a very good short-term accuracy, and it uses accelerometers and gyroscopes for enhancing the accuracy of the current system state. Nevertheless, all of those methods in this sort have unbounded growing position error.
- (2) Acoustic transponders and modems: Techniques in this class are based on measuring the Time of Flight (ToF) of signals from acoustic beacons or modems to perform navigation.

They could provide navigational aids, usually in form of range and bearing signals, but have a low update rate.

- (3) Geophysical: Techniques that use external environmental information as references for navigation. This must be done with sensors and a series of processing that are capable of detecting, identifying and classifying some environmental features.

Sonar sensors are based on acoustic signals, however, navigation with imaging or bathymetric sonars are determined by detection, identification and classification of features in the environment. Therefore, navigation that is sonar based falls into the acoustic and geophysical categories. A distinction is made between sonar and other acoustic-based navigation schemes, which depend on externally generated acoustic signals emitted from beacons or other vehicles. In particular, the development of acoustic communications through the use of underwater modems has led to the development of new algorithms. What's more, advancement in SLAM research has been applied to the underwater domain in a number of new ways.

SLAM has been extensively studied in a land environment. Many of the systems use laser scanners, which provide precise 2D and 3D depth information [28]. However, in underwater environments, it is not possible to use lasers nor cameras due to poor lighting conditions. With the development of imaging sonar techniques, it is likely to obtain the undersea information by sonar images. They represent the environment similar to nature, and also contain an energy distribution map of the imaging space. Nevertheless, since the water mediums in the underwater sound channel and the seabed and sea level boundaries have complex and changeable characteristics, and an acoustic wave has the ability of penetration, the sonar image has low resolution, strong disturbance, and the obscure boundaries of objects. This not only increases the difficulties of interpreting the information in the images directly, it also frequently leads to more inaccurate measurements.

As far as the data association problem is concerned, it is well known that it is completely dependent on the ability to find proper features, which in underwater scenarios is a considerably critical task, mainly due to the enormous amount of noise, the dynamism of the subsea environment and the lack of significant features. Obviously, if this landmark detection is not robust enough there won't be a possibility for the SLAM to work well. For this reason, the use of vision to define landmarks seems to open a wide range of different possibilities. In addition, a vision system is considerably less expensive than other sensorial systems. The main drawback lay in its high computational cost on extracting and matching features. Further problems that need to be addressed from the limitations of using computer vision in SLAM are the dynamism of the robot environment. For instance, the changes of location of other agents in the

environment cause a big challenge, because they introduce inconsistent sensor measurements and since there are almost no algorithms that can deal with mapping in dynamic environments. Until now, the implementation of SLAM in real underwater environment can be considered as an unsolved and challenging research area.

1.3.2. Robotic mapping

Mapping underwater surfaces using an AUV is typically a multi-step process. First, the AUV flies a pattern over the surface collecting multi-beam sonar data as well as navigation data from an Inertial Navigation System (INS). Second, an accurate navigation history of the AUV is constructed offline. Finally, the sonar measurements are projected from these location estimations to form a point cloud, which can then be processed to produce a final map in a variety of formats, like Discrete Elevation Map (DEM) or octrees [29]. In [136], the author presents a complete system for building bathymetric maps of the seafloor incorporating visual texture, as well as 3D maps of rotating ice floes using an AUV. The authors in [138] propose a method of localization of an AUV using visual measurement of underwater structures and artificial landmarks.

The research about robotic mapping has a long history. In the 1980s and early 1990s, the mapping workloads were widely divided into metric and topological approaches. Metric maps capture the geometric properties of the environment, whereas topological maps represent the connectivity between different landmarks. An early representative of the metric maps was Elfes and Moravec's Occupancy grid mapping algorithm [30,31], which describes maps by fine-grained grids that model the occupied and vacant space of the environment. This approach has been used in plenty of robotic systems, such as [18,32,33]. An alternative metric mapping algorithm was proposed by Chatila and Laumond [34], using a set of polyhedra to represent the geometry in the area. Topological maps such as the work by Mataric [35] and Kuipers [36], describe environments as a series of significant places that are connected with each other via arcs. Arcs are usually annotated with information on how to navigate from one place to another. However, the difference between metric and topological maps has always been fuzzy, since actually all working topological approaches are based on geometric information. In practice, metric maps are finer grained than topological ones.

Since the 1990s, the field of robot mapping has been dominated by probabilistic techniques. A series of papers by Smith, Self, and Cheeseman [37,38] introduced a powerful statistical framework for simultaneously solving the mapping problem and the induced problem of localizing the robot relative to its growing map. There exists a cyclic nature problem in the robotic mapping and localization: while operating in an unknown environment, in order to

generate a map of the uncharted territory, the fully autonomous robot needs to know its location, but to know its position, it requires a map. This chicken and egg problem of robotic field is often referred to as Simultaneous Localization and Mapping (SLAM) or Concurrent Mapping and Localization (CML) [39,40], where the robot creates an environmental map, while concurrently localizing itself within the map. This cyclic nature of robotic mapping problem is illustrated in Figure 1.2. The robot has to maintain two representations at the same time: one is an environment observation model or map, the other is a localization model or position estimate. That means to execute them iteratively, with the output of one being the input of the other [41]. It must be able to run two perceptual procedures, namely map-building and self-localization [6,42].

Figure 1.2 The problem of robotic localization and mapping

One family of probabilistic approaches use Kalman filters to estimate the map and the robot pose [21,43]. The achieved maps usually describe the location of landmarks or particular features in the area, although recent extensions exist that represent environments by large numbers of raw range measurements. An alternative approach is based on Dempster's expectation maximization algorithm [44,45]. These approaches specifically address the correspondence problem in mapping, which is the problem of determining whether sensor measurements at different time correspond to the same physical entity in the real world. Another probabilistic method seeks to identify objects in the environment, which may correspond to ceilings, walls, doors that might be open or closed, of furniture and other moving objects. Many of these techniques have counterparts in the computer vision and photogrammetry literature, while a connection is still somewhat underexploited.

To map an environment, current systems use probabilistic methods and landmarks. They regard SLAM as an estimation problem. During robot localization, the state of the robot x_t for instance, the robot pose has to be estimated from all previous sensor measurements $z_1, z_2, \dots, z_{t-1}$ and previous robot controls $u_1, u_2, \dots, u_{t-1}$. Robot localization is often resolved using the Markov assumption, such as the world is assumed to be static, noise to be independent

and no approximation errors during modelling to be made. Using these Markov assumptions, the estimation problem for localization simplifies to the Bayes net illustrated in Figure 1.3.

Figure 1.3 Bayes net for robotic localization

Mapping extends the estimation problem. In addition to the robot state, the map m has to be estimated, as given in Figure 1.4. Usually, the map consists of n landmarks. In this case, the positions of these landmarks have to be estimated to perform mapping.

Figure 1.4 Bayes net for robotic mapping

If a robot moves through a known environment, that means a map is given to the robot, the uncertainties of the robot poses can be bounded. The observation of landmarks allows the precise estimation of the robot poses depending on the uncertainties of the landmark positions and observation. However, if the robot moves through an unknown environment, the uncertainties of the robot poses can be arbitrarily large as odometry errors sum up. Tracked landmarks provide a basis for reducing the uncertainty of the robot poses. For many sensors, such a tracking outperforms pure odometry [46]. However, this does not change the principle of error accumulation. This circumstance makes the SLAM problem different from other

estimation problems. And thereby, closed loops play an important role in known SLAM algorithms. If a robot can detect a position where it has been before and could correctly match landmarks, then the accumulated error will be bounded.

1.3.3. Solutions to the SLAM problem

Since radio waves cannot penetrate water very far, an AUV loses its Global Positioning System (GPS) signal as soon as it is diving. Therefore, a standard solution for AUVs to navigate below the water superficies is through dead reckoning. However, navigation can be improved by employing an acoustic underwater positioning system. For underwater robotics, the Inertial Measurement Unit (IMU) and the Doppler Velocity Log (DVL) sensors are most commonly applied for measuring the navigation data on AUVs. While acoustic sensors, such as Side Scan Sonars (SSSs) or imaging sonars are used to detect the environmental information. Currently, the fusion of data measurements derived from various perception and on-board navigation sensors plays an important role in robotic underwater mapping researches. In general, we employ acoustic sensors to build large-scale maps of the exploring areas, and optical sensors to return much higher quality images of the targets of interest. To a large extent, the navigation time of the AUV greatly influences the expense of subsea mapping, thus any enhancement in the characteristics of sonar images is of vital concern to the marine and offshore research community.

The research of estimation approaches for the robotic SLAM-based navigation has caught utmost attention by the research community. Among the solutions presented in the literature to solve the underwater SLAM problem, the Bayesian filters-based estimation methods have been the most successful ones over time [18,19,47]. Initial works such as [38,48] built a statistical basis for illustrating relationships between landmarks and manipulating geometric uncertainty. At the same time, [34,49] considered employing the Kalman Filter (KF)-based algorithms for visual robotic navigation. After that, [37] demonstrated that when a mobile robot explores an unknown region observing relative measurements of landmarks, the estimates of these landmark positions are all necessarily correlated with each other, due to the common uncertainty in the estimated robot pose. The SLAM-based robotic navigation problem can be categorized into various solutions, relying on the number of features, the area of coverage, computational complexity, flexibility, reliability, etc. [50]. A general classification of the state of the art recursive Bayesian filters-based estimation strategies, including Kalman Filter (KF), Extended Kalman Filter (EKF), Expectation Maximization (EM), Particle Filter (PF), Compressed Extended Kalman Filter (CEKF) and Information Filter (IF) in underwater applications is given, contrasting the advantages and disadvantages summarized in Table 1.1. The Stochastic SLAM is subject to three dominating defects: high computational load and storage costs, fragile and

even wrong data association, and inconsistent update of non-linearity. Keeping relationships between the robot poses and all correlated landmark positions in the whole system state covariance matrix, leads to high computational requirements and space consumption problem. For n state parameters, the n^2 covariance elements need to be updated with each new observation. Generally, the number of vehicle parameters is insignificant compared to the number of feature parameters, and so the computation and storage are specified as being $O(n^2)$ where n is the number of map features. Although some approaches have been proposed to deal with the underwater environment, the SLAM applications for it still remain unsolved.

Table 1.1 List of pros and cons of filtering approaches applied to the SLAM framework

Pros	Cons
Kalman Filter and Extended KF (KF/EKF)	
1. High convergence; 2. Handle uncertainty.	1. Gaussian assumption; 2. Slow in high-D maps.
Compressed Extended KF (CEKF)	
1. Reduced uncertainty; 2. Reduction of memory usage; 3. Handle large areas; 4. Increase map consistency.	1. Require robust features; 2. Data acquisition problem; 3. Require multiple map merging.
Information Filters (IF)	
1. Stable and simple; 2. Accurate; 3. Fast for high-D maps.	1. Data association problem; 2. Recover a state estimates; 3. Computationally expensive in high-D.
Particle Filters (PF)	
1. Handle nonlinearities; 2. Handle non-Gaussian noise.	1. Growth in complexity
Expectation Maximization (EM)	
1. Optimal to map building; 2. Solve data association.	1. Inefficient, growing cost; 2. Unstable for large scenarios; 3. Only successful in map building.

(1) KF and its variations

a. KF

Kalman filters are Bayes filters that represent posteriors using Gaussians, i.e. unimodal multivariate distributions that can be represented compactly by a small number of parameters. KF SLAM relies on the assumption that the state transition and the measurement functions are linear with added Gaussian noise, and the initial posteriors are also Gaussian. There are two main variations of KF solutions the state of the art SLAM problem: the Extended Kalman Filter (EKF) and its related Information Filter (IF) or Extended Information Filter (EIF). The advantage of KF and its variants is that they provide optimal Minimum Mean-Square Error (MMSE) estimates of the system state, which is composed of the robot pose and landmark

positions as a whole and its covariance matrix seems to converge strongly. But, the Gaussian noise assumption restricts the adaptability of the KF for data association and number of landmarks.

b. EKF

KF performs properly in the linear cases, and is accepted as an efficient method for analytically propagating a Gaussian Random Variable (GRV) through a linear dynamic system. Most frequently the EKF is applied to solve the SLAM problem. EKF accommodates the nonlinearities from the real world by approximating the robot motion model using linear functions. The filter integrates all measurements of landmarks in a covariance matrix of landmark positions and robot poses. EKF uses a landmark-based map and operates recursively in two stages: Prediction and Update steps. In the prediction stage, the control signal u_k and the robot motion model are utilized to estimate the robot pose. Then, in the later update stage, to update the landmark locations and refine the estimation of the robot pose, the new observation z_k from a perception sensor is applied.

However, the KF based SLAM algorithms have two limitations. The first is that the computational complexity increases when newly detected landmarks in the environment are added to the system state vector. The second is the requirement of matching the observations with landmarks in the state vector. This disadvantage is critical since wrong data association may lead to divergence of the filter. The KF based SLAM algorithms create computational complexity and the number of landmarks stored in the state vector is limited. If n implies the number of landmarks in the environment, then each update step of EKF requires $O(n^2)$ and the total computational cost of EKF is $O(n^3)$ [51]. In order to use EKF in outdoor applications, where many objects exist, some improvements were proposed to reduce the computational complexity and memory usage. Due to the quadratic computational complexity of the EKF, some alternatives such as Compressed EKF (CEKF), Sparse Weight Kalman Filter [52] and Map Joining SLAM [53] were proposed. They all work on local areas of the stochastic map, and they are essentially constant most of the time and aim to decrease the total processing load.

Due to the quadratic nature of the EKF's covariance matrix is a key limitation of the EKF solution to the SLAM problem. Many researchers have proposed extensions to the EKF-SLAM algorithms that gain remarkable scalability by decomposing the map into the sub-maps, for which covariances are maintained separately [28,54,55]. Other researchers have developed hybrid SLAM techniques that combine EKF-style SLAM techniques with volumetric map representation, see [56,57]. Finally, researchers have developed data association techniques for SLAM [58] based on advanced statistical techniques, such as Dempster's Expectation

Maximization (EM) algorithm [28].

c. CEKF

CEKF differs from EKF since it does not perform a full SLAM update when working in a local area and this reduces the computational requirement of the SLAM algorithm. CEKF divides the state vector into two parts: Active and Passive. Only the active local area part is updated while a robot is performing SLAM. The full update is performed when the robot moves out of the local area. The CEKF algorithm significantly reduces the computational requirement with only $O(2n^2)$ computation complexity and it does not introduce any penalties in the accuracy of the results compared with EKF.

d. UKF

Another variation of the KF called Unscented Kalman Filter (UKF) addresses the approximation issues of the EKF and the linearity assumptions of the KF. It is based on the Unscented Transform (UT) to deterministically sample Sigma points, which are a set of representative points from the distribution of current estimates. These points are propagated through the nonlinear models, from which the new mean and covariance are recovered. UKF reduces estimation errors but it is computationally more costly than EKF. UKF differs from EKF and CEKF as it does not require deriving the Jacobian matrices. The computational complexity of the UKF is $O(n^3)$.

e. IF

The IF is implemented by propagating the inverse of the state error covariance matrix. There are several advantages of the IF over the KF. First of all, the data is filtered by simply summing the information matrices and vectors, providing more accurate estimates. Moreover, IF is more stable than KF. Finally, EKF is relatively slow when estimating high dimensional maps, since every single vehicle measurement generally affects all Gaussian parameters, and the updates require prohibitive times when dealing with environments with many landmarks.

However, IF have some limitations, a primary disadvantage is the need to recover a state estimate in the update step when applied to nonlinear systems. This step requires the inversion of the information matrix. Further matrix inversions are required for the prediction step of the IF. For high dimensional state spaces the need to compute all these inversions is generally believed to make the IF computationally poorer than the KF. In fact, this is one of the reasons why the EKF has been more popular than the EIF [18].

(2) Expectation Maximization methods (EM)

Expectation Maximization (EM) estimation is a statistical algorithm that was developed in the context of Maximum Likelihood Estimate (MLE), which offers an optimal solution for map building, but not for localization. The EM algorithm is able to build a map when the robot pose is known, for instance, by means of expectation. EM iterates two steps: an expectation stage called E-step, where the posterior over robot poses is calculated for a given map, and maximization stage called M-step, in which the most likely map is calculated given these pose expectations. The final result is a series of maps with increasing accuracy. The main advantage of EM relative to KF is that it tackles the data association problem extraordinarily well. This is possible due to the fact that it repeatedly localizes the robot with respect to the present map in the E-step, generating various hypotheses as to where the robot might have been. In the latter M-step, these possible correspondences are translated into features in the map, which then get reinforced in the next E-step or gradually disappear.

However, the need to process the same data several times to obtain the most likely map makes it inefficient, not incremental and not suitable for real-time applications [59]. Even using discrete approximations, when estimating the robot pose, the computational cost grows exponentially with the size of the map, and the error is not bounded, and therefore the generating map becomes unstable after long cycles. These problems could be avoided if the data association was known, that is the same, if the E-step was simplified or eliminated. For this reason, EM usually is incorporated with Particle Filter (PF), which represents the posteriors by a set of particle samples that imply a guess of the pose where the robot might be. For instance, some practical applications only use M-step in EM to construct the map, while the localization is performed by different means, for example using PF-based localizer to estimate poses from odometer readings.

(3) Particle Filter based method (PF)

The Particle Filter, also called the Sequential Monte-Carlo (SMC) method, is a recursive Bayesian filter implemented in Monte Carlo simulations. It executes SMC estimation by a set of random point clusters, known as particles to represent the Bayesian posterior. In contrast to parametric filters, such as KF, PF represents the distribution by a series of samples drawn from it, which makes it capable of handling highly nonlinear sensors and non-Gaussian noise. However, this ability causes a growth in computational complexity on the state dimension as new landmarks are detected, becoming not suitable for real-time applications. For this reason, PF has only been successfully applied to localization, such as determining position and orientation of the robot, but not to map-building. Therefore, could not find research works only applying PF for the whole SLAM framework, but there exist few literatures that deal with the

SLAM problem using a combination of PF with other techniques, for instance, the FastSLAM and the FastSLAM2.0 [60].

FastSLAM takes advantage of an important characteristic of the SLAM problem, which is that landmark estimates are conditionally independent given the robot path [60]. The FastSLAM algorithm decomposes the SLAM problem into a robot localization problem, and a collection of landmark estimation problems, which are conditioned on the robot pose estimate. A key characteristic of FastSLAM is that each particle makes its own local data association. By contrast, EKF techniques must commit to a single data association hypothesis for the entire filter. In addition, FastSLAM uses a particle filter to sample over robot paths, which requires less memory usage and computational time than a standard EKF or KF.

Table 1.2 resumes the current popular underwater SLAM methods. Many successful applications exist in the literature employing an EKF to solve nonlinear estimation problems, such as robotic localization, object tracking and it remains to be a popular choice for the solution to the robotic SLAM problems [61].

Table 1.2 The summary of state of the art underwater SLAM approaches.

Method [Reference]	Research Group	Underwater Vehicle	Sensor	Underwater Map	Filter
Daniel [22]	SRV ¹	EcoMapper	Side Scan Sonar	Point Features	EKF
He [23]	SISE ²	C-Ranger	Forward Looking Sonar	Point Features	PF
Burguera [24]	SRV ³	Ictineu	Imaging Sonar	Vehicle Poses	EKF
Aulinas [25]	ViCoRob	SPARUS	Imaging Sonar	Point Features	EKF
Mallios [26]	ViCoRob	Ictineu	Imaging Sonar	Vehicle Poses	EKF
Barkby [27]	CAS	Sirus	Multibeam	Bathymetry	PF
Ribas [28]	ViCoRob ⁴	Ictineu	Imaging Sonar	Line Features	EKF
Fairfield [29]	WHOT	MBAUV	Sonar Beams	Evidence Grid	PF
Roman [30]	WHOI ⁵	JASON	Multibeam	Bathymetry	EKF
Fairfield [31]	CMU ⁶	DEPTHX	Sonar Beams	Evidence Grid	PF
Williams [32]	CAS ⁷	Oberon	Camera + Sonar	Point Features	EKF
Tena-Ruiz [33]	OSL ⁸	REMUS	Side Scan Sonar	Point Features	EKF
Williams [34]	ACFR ⁹	Oberon	Imaging Sonar	Point Features	EKF

¹ SRV: Systems Robotics & Vision, Universitat de les Illes Balears, Spain; ² SISE: School of Information Science and Engineering, Ocean University of China, China; ³ SPV: Systems, Robotics and Vision Group, Islas Baleares, Spain; ⁴ ViCoRoB: Computer Vision and Robotics group, Girona, Spain; ⁵ WHOI: Woods Hole Oceanographic Institution, Woods Hole, MA, US; ⁶ CMU: Carnegie Mellon University, Pittsburgh, PA, US; ⁷ CAS: Centre of Excellence for Autonomous Systems, Sydney, Australia; ⁸ OSL: Ocean Systems Laboratory, Edinburgh, UK; ⁹ ACFR: Australian Center for Field Robotics, Sydney, Australia.

However, the quadratic computational complexity of the EKF makes it difficult to apply in underwater scenarios. Unscented Kalman Filter (UKF) [62] is a more reliable estimator than EKF when the system model is highly nonlinear, since it approximates the probability density function instead of the nonlinear function itself, but it does not make any improvement in the high

computational load of the EKF. The Hector SLAM algorithm is based on a Gauss-Newton approach and relies on scan matching alone to estimate the pose. It is only suitable for few limited unstructured environments, such as the Urban Search and Rescue (USAR) scenarios [63]. Another state of the art solution is GMapping, based on Rao-Blackwellized Particle Filter (RBPF), which uses odometry for pose estimation [64]. Hector SLAM and GMapping are two laser scanner based SLAM algorithms, and they create an occupancy grid map from laser scans. However, laser range finders are imprecise working in subsea due to light attenuation. Besides, occupancy grid SLAM can only work in indoor environments over a limited period and it cannot deal with an appropriate uncertainty model, so it would diverge in the long-term navigation required under water. Therefore, both Hector SLAM and GMapping are not adequate in our considered case of underwater environments.

In 2002, Montemerlo et al. proposed the classic FastSLAM algorithm [65], which utilizes the particle filters and handles the computational complexity considerably well compared to EKF and UKF. In FastSLAM version 1.0, only the control signal is adopted to sample the new robot pose for each particle according to the robotic motion model, however FastSLAM version 2.0 not only utilizes the robotic control signal but also the sensor measurements together to sample the new robot pose. Therefore, FastSLAM 2.0 requires fewer particles and is more robust and accurate than FastSLAM 1.0, besides it will be particularly efficient if an accurate sensor is employed. Although the classic FastSLAM 2.0 algorithm is not a new approach, it has been widely used in many robotic applications, such as robot path planning. In this work, a new solution, the Augmented Extended Kalman Filter-based Simultaneous Localization and Mapping (AEKF-SLAM) algorithm is presented, and the performances in terms of accuracy and computational complexity of robotic localization and landmarks mapping are compared. Also the computational cost of the FastSLAM 2.0 is compared to that of the proposed AEKF based robotic underwater SLAM algorithm.

1.3.4. Challenges in underwater SLAM

The autonomy of robotic underwater vehicles is dependent on the ability to perform long-term and long-range missions without need of human intervention. While current state-of-the-art underwater navigation techniques are able to provide sufficient levels of precision in positioning, they require the use of support vessels or acoustic beacons. This can pose limitations on the size of the survey area, but also on the whole cost of the operations [157].

The SLAM problem is generally regarded as one of the most important problems in the pursuit of building truly autonomous mobile robots. Despite significant progress in this area, it still poses great challenges. Two of the major challenges of the SLAM problem are (1) defining

fixed features from raw sensor data and (2) establishing measurements to feature correspondence, such as the data association problem [66]. Both of these two tasks can be nontrivial, especially in an unstructured underwater environment. In man-made areas, typically composed of planes, lines, corners and points features can be defined easily; but, complex subsea environments pose a pretty tough task for feature extraction and matching.

At present, robust methods exist for mapping environments that are static, structured, and of limited size. These methods deal with issues related to computational complexity, data association and representation of the environment [6,42]. The SLAM problem is to acquire a spatial model of the robot environment while localizing itself relative to this model. In SLAM there is bi-directional dependency between map accuracy and pose uncertainty. Sensor limitations, measurement noise, data dimensionality, data association, dynamic environments, obstacles, and exploring unknown areas are the main factors affecting the solution to SLAM problem [67]. The implementation of SLAM in real underwater environments still can be considered as an unsolved and challenging research topic area due to the limitations of subsea localization and perception sensors and error accumulation over long-term operations. Furthermore, acoustic sensors for mapping often provide noisy and distorted images or low-resolution ranging, while video images provide highly detailed images but are always limited due to turbidity and lighting [7]. Nowadays, large-scale dynamic environments still present various challenging problems for the robotic underwater SLAM-based navigation applications due to the following reasons:

- (1) Sensors. Generally, the data observed by the sonar sensors has limited accuracy, usually with a high angular uncertainty of 22.5% to 30% [68], particularly in environments with low-light, strong ocean currents and turbid waters. Also, the sensors usually only allow limited depth for operation, making the applications costly for deep immersions. The estimated noise ends up causing significant impact on the tasks of localization and mapping, often leading to a non-convergent system. The modeling of sensor noise is an important and difficult task, especially for large-scale environments since the noise may be different in certain situations, thus requires a new calibration to obtain a better estimation of the system. The sensors usually have a limited depth for operation, making applications costly for high depths.
- (2) Feature extraction. In the underwater SLAM context, as many as possible distinguishable features must be observed repeatedly, in order to decrease the uncertainty caused by significant vehicle shift. Here, distinct landmarks can simplify the data association process of fusing new observations to corresponding map features, already stored in the map. In man-made environments, typically composed of planes, lines, corners and points, features

can be identified easily; but there are no similar objects or particular features and located easily distinguishable in complex subsea environments. Multiple sources of perturbations in sonar measurements can be listed such as hydrostatic effects of waves or currents, inhomogeneous pressure distribution, vessel traffic and the propellers of its own vehicle, marine animals, suspense material in the water column, seafloor reflection and surface sonar altitude corrections. Besides the noise in the sonar images, one needs to manage reflections and poor resolutions of the derived acoustic imageries for feature extraction. What's more, the dynamics of underwater natural resources is another factor that makes the recognition of previously visited locations difficult or even impossible.

- (3) Absolute location. Since the underwater environment does not allow using Global Positioning System (GPS), alternative solutions such as the use of triangulation systems Long Base Line (LBL), Short Base Line (SBL), and Ultra Short Base Line (USBL) have been proposed. When operating within a net of seafloor deployed baseline transponders, this is known as LBL navigation. When a surface reference such as a support ship is available, SBL or USBL positioning will be used to calculate where the subsea vehicle is relative to the known GPS position of the surface craft by means of acoustic range and bearing measurements. However, these systems require great effort for installation, a special demand of logistics, and high cost. Besides, the above solutions limit the working area of robotic operations.
- (4) Computational complexity: The computational requirements of SLAM applications are closely related with the size of the exploring environment and the methods used for feature extraction, tracking, data association, and filtering. The uncertainty calculations of the robot and landmark positions and their correlations will become larger as the number of map elements increases.

1.4. Research contributions and thesis structure

This dissertation makes a set of contributions that can be summarized as follows:

- (1) Existing underwater navigation approaches, including inertial navigation, acoustic transponders and modems and geophysical techniques are described and carefully compared. Also, a detailed overview of currently used and popular solutions for the robotic underwater SLAM problem is presented and compared including the Kalman filter (KF) and its variations, Expectation Maximization (EM) methods and Particle filter (PF) based FastSLAM. Besides, a survey of several most commonly used and popular map

representations for a priori map localization problem is made, particularly with regard to large-scale underwater SLAM, including computational complexity, storage, convergence, robustness, reliability, etc.

- (2) An improved Otsu Threshold Segmentation Method (TSM) is proposed as a real-time, fast and accurate segmentation solution for underwater feature detection, when compared with several classic and popular used segmentation approaches including the traditional Otsu TSM, the local TSM, the iterative TSM and the maximum entropy TSM. In combination with the contour detection algorithm, the foreground objects of interest, in our case, a shipwreck, a branch and a plastic mannequin are separated more accurately, in sonar images of various resolutions and qualities, with low computational costs. As a result of the segmentations, the centroids of the main extracted regions are computed to represent point landmarks which can be used for further robotic navigation, e.g. with the help of the proposed Augmented Extended Kalman Filter-based Simultaneous Localization and Mapping (AEKF-SLAM) algorithm.
- (3) An optimal solution, the MIXed exponential Regression Analysis (MIRA) method is proposed to normalize inhomogeneous echo decay in acoustic shadows of sonar images. The MIRA model has superior normalization performances compared to the Dark Channel Prior (DCP) approach, which is an efficient fog removal technique to sonar imaging, both from one ping, local similarity and global quality analysis.
- (4) A detailed state-of-the-art underwater feature matching approaches are summarized and compared, including the classic Harris, Speeded Up Robust Features (SURF), Binary Robust Invariant Scalable Keypoints (BRISK), Scale Invariant Feature Transform (SIFT) and Affine Scale Invariant Feature Transform (ASIFT) feature detector. Consider their suitability for subsea sonar map fusion, in terms of computational requirements, reliability, accuracy, etc.
- (5) The Underwater Wireless Sensor Network (UWSN) is newly introduced into the sonar map fusion task, and with the help of the Delaunay triangulation technique, the Underwater Wireless Sensor Network-based Delaunay Triangulation (UWSN-DT) algorithm is proposed for improving the performances of sonar map fusion accuracy with low processing burden, especially for the subsea environments where only exist few distinguishable feature points. To the best of our knowledge, this is the first algorithm which adopts the UWSN into the application of sonar map fusion.
- (6) The mathematical framework of the posterior-estimated “Robotic AEKF-SLAM based navigation” algorithm, which introduces the vehicle model, feature model and observation model, is proposed. In order to generate the best estimation of the system states, the AEKF-SLAM algorithm stores the robot pose and the map landmarks in a single system state

vector, and estimates the state parameters via a recursive, iterative, three-stage process. Hereby, the prediction and update state (which exist as well in the conventional EKF) are complemented by a newly proposed augmentation stage.

- (7) For both performed dense loop mapping and line mapping navigation experiments, the presented AEKF-SLAM based robotic navigation approach has the best performances of self-localization and landmark mapping accuracy with much lower computational costs compared with the classic and popular FastSLAM 2.0 algorithm. Besides, the AEKF achieves more robust and accurate estimations of the robot pose and the landmark positions due to the landmark positions detected by our improved Otsu TSM, than those identified by the maximum entropy TSM, which achieves the highest segmentation precision than other three classic TSMs compared in this thesis. Altogether, the presented AEKF-SLAM method achieves reliably map revisiting and consistent map upgrading on loop closure.

Before ending this chapter, here we introduce the rest of this thesis, which consists of six sections. The outline of them is organized as follows:

- (1) Chapter 2 introduces the existing work and the state-of-the-art related to the topic of this dissertation, and discusses the models and techniques which are employed in this thesis.
- (2) Chapter 3 proposes an efficient and fast threshold segmentation method (TSM), the improved Otsu TSM for real-time and accurate underwater feature detection.
- (3) Chapter 4 presents an optimal solution, the MIRA method for handling the echo decay in sonar images. Besides, a detailed state-of-the-art underwater feature matching methods are summarized and compared. Moreover, the UWSN-DT algorithm is proposed for improving the precision of sonar map fusion with low computational costs, especially when underwater feature points are few, fuzzy and difficult to be detected and identified.
- (4) Chapter 5 presents the mathematical framework of the proposed “AEKF-based robotic underwater SLAM” algorithm. With the assistance of the presented AEKF approach, the underwater robot achieves a more precise and robust self-localization and mapping of the surrounding landmarks with low processing burden.
- (5) Chapter 6 performs several MATLAB simulated experiments for both our proposed AEKF-SLAM based navigation algorithm and the classic FastSLAM 2.0 for dense loop mapping and line mapping, where environmental landmarks consist of not only the calculated centroids of the shipwreck, branch and plastic mannequin, but also those centroids of certain parts of the background detected by our proposed improved Otsu TSM in Chapter 3. As a part of the European Smart and Networking Underwater Robots in Cooperation Meshes (SWARMs) project, several use cases are also introduced, especially the No. 5 Seabed Mapping, which our presented work is mostly concerned with.

- (6) Finally, Chapter 7 provides a summary of the major contributions of this thesis. Besides, remaining challenges are pointed out, and some future work recommendations for this research topic are discussed as well.

2

Navigation map representations

Learning maps is one of the most fundamental and important problems in mobile robotics, since successful robotic systems depend on geological maps that demonstrate the surrounding environment of the robot for a path planning and to determine a position with respect to the environment. Maps are necessary for robotic path planning and self-positioning with respect to the environment, in order to prevent collisions. The basic functionalities of the three most commonly used and popular map representations are summarized in this part and a survey of their suitability to a priori map localization problem is made, such as computational requirement and storage, convergence, robustness, reliability, etc.

2.1. Navigational maps and their applications to underwater SLAM

In order to produce an accurate map, SLAM is needed to estimate the position of the mapping device. SLAM algorithms with multiple autonomous robots have received considerable attention by the research community in recent years. There are three well-known map representations in SLAM, i.e., occupancy grid maps, topological maps, and landmark maps. For each of these maps, a basic functional description is given and a survey is made of their suitability to a priori map localization, in terms of computational complexity, reliability, etc. An estimation of suitability for SLAM follows, with attention to criteria considered essential for the development of a tractable and consistent SLAM algorithm. These criteria are illustrated as follows, note the first two criteria are only valid for metric maps. Considering the sparse spatial distribution of the marine features, and the far distances between distinct features, landmark maps are most suitable to represent subaquatic areas.

- (1) Representation of uncertainty. Since mobile robot sensors cannot measure positions in the environment with total accuracy, a degree of uncertainty should be considered in the map representations. Meanwhile, the vehicle position is acquired from this map, so the pose estimate is also uncertain. This thesis stipulates that both, environmental map and robot pose estimates, require a proper quantitative uncertainty, and that uncertainty model is able to accurately reflect the error between the estimated and actual system state.
- (2) Monotonic convergence. The primary purpose of an uncertainty measurement is to ensure map convergence. A convergent map is given if the estimated environmental geometry is equal to true physical geometry when new observation information is incorporated. In other words, the map uncertainty decreases monotonically. Without this uncertainty model, a stationary object with estimated location (x_1, y_1) may drift to some arbitrarily distant place

- (x_2, y_2) , with subsequent map updates. Therefore, explicit uncertainty is needed to assess map accuracy and it constrains the influence of subsequent observation information.
- (3) Data association. The map representations should permit reliable correspondence between the measurements derived from the robot sensors and the information gained from the stored map. First of all, the search for observation-to-map data association should be efficient enough for real-time operation. Secondly, the data association must be robust to partial views and large-scale searching areas, since an observation is composed of a combination of the currently mapped region, unexplored area and dynamic features. Due to the size of the exploring area is determined by the robot pose uncertainty, an accurate uncertainty model could enhance both efficiency and robustness, by stipulating a minimal detecting region.
 - (4) Loop closure. When a mobile robot explores a working environment by navigating in a large loop which is much bigger than its sensing range, then return and recognition of an already mapped area is called the “Loop Closure Problem”. Other denominations are “Cycle Detection Problem” or “Map Revisitation Problem”. Loop closure mainly considers two previous criterions: one is the data association, which is distinct from local association as a result of the much larger robot pose uncertainty, and hence the related search space. Exploring efficiency is one important aspect, but it is more crucial to robustly decide whether an association is correct or an artifact of environmental similarity. The other issue, after associating sensor measurements correctly, is the convergence problem, where long-term accumulated error in the map loop must be compensated properly during the map update stage by propagating the error-offset back through the map cycle.
 - (5) Computation and storage. The map has to store sufficient information to enable data association and convergence. The computation and the storage required to update the map with newly observed measurements, must be scaled reasonably with the detected environmental area.

2.2. The occupancy grid maps

The 2D occupancy grid maps, which are also known as evidence grid maps were introduced by Moravec [69]. In the occupancy grid maps, the environment is represented in a discrete grid composed of rectangular cells with the same shape and size. Each cell is assigned a value representing the probability of occupancy. A probability of 1 means that the cell is definitely occupied and the robot cannot pass through it. If it is 0, the cell is definitely vacant, and the robot can traverse it. An occupancy probability of 0.5 declares an unknown state of occupancy. The larger the cell value is, the more the cell tends to be occupied; the smaller its value is, the

more the cell tends to be free. Usually in the occupancy grid maps, black cells imply occupied areas, white cells stand for empty regions and the gray ones mean unexplored spaces. The occupancy grid map of the San José Tech Museum is shown in Figure 2.1.

Figure 2.1 The occupancy grid map of the San José Tech Museum, in [18]

The occupancy grid SLAM interleaves the localization and map update stages by first registering the short-term local map with global map (localization), and then updating the perceived occupancy of the global map grid cells (map building). In dynamic indoor areas, the occupancy grid map works robustly over a limited exploring time. The occupancy grid maps are adequate for local navigation and obstacle avoidance purposes [70], and thereby this kind of map is a popular approach to represent the environment of a mobile robot, given known poses. The occupancy grid representation has some merits: above all, it can represent general unstructured environments. In addition, it can model the map uncertainty. Moreover, an occupancy grid map is similar to maps used by humans, which makes the method suitable for an easy handling.

Nevertheless, as a SLAM map representation, the occupancy grids do not process an appropriate uncertainty model, and so will tend to diverge in the longer-term operations. The main drawback of the occupancy grid maps is that they do not scale well to large environments, since they can only represent the uncertainty model from the local robotic view, but not at a global level which is essential for map convergence. Also, memory size is a limiting factor, when the goal is to global map a large region. The high memory consumption for larger environment resulting from the fact that the required space depends on the size of the environment and not on its complexity, and which also implies strongly increasing computational costs. To adequately capture the details in more complex underwater environments, a high resolution of cell distribution is required, which is wasted in less complex

areas. Techniques like quad trees or octrees [29] have been presented to deal with the space consumption issue, but they also lead to increased processing burden.

2.3. The topological maps

Both the occupancy grids and the feature maps are metric maps where location is defined as a set of coordinates in Cartesian space. However, the topological maps do not rely on metric measurements and instead represent the exploring regions in terms of connecting paths led by the feature points, as shown in Figure 2.2. Topological maps consist of vertices and edges. A vertex represents a landmark, such as a particular location in the environment, and an edge indicates the traversability between two connected nodes. Thus, navigation between two non-adjacent places is determined by a sequence of transitions between intermediate location nodes, and so standard graph shortest path algorithm can be used (e.g., A to D requires travelling through the sequence $A \rightarrow B \rightarrow C \rightarrow D$). The concept works on the assumptions that distinctive places are locally distinguishable from the surrounding area, and the procedural information is sufficient to enable the robot to travel within recognizing distance of a specified place [65]. Topological SLAM [36] operates by performing exploration of the environment guided by a set of path following criteria, and recoding place descriptions at appropriate locations.

Figure 2.2 The topological map

Topological maps are attractive for their efficient and compact representation form, and their logical organization for tasks like fast path planning and robotic navigation. The departure from metric representation makes pose uncertainty estimation irrelevant, and instead, qualitative measures are used like “follow the path from A to B” or “at B”. The ability to utilize standard graph algorithms for high-level planning operations, such as finding the shortest path between non-adjacent nodes, is of particular advantage.

Nevertheless, without some form of metric position measurement, the significant weakness of topological maps is that they cannot ensure reliable navigation between distinct places, and subsequent place recognition. Although the relationship between different points is maintained, distance and direction are subject to change and variation. The employment of purely qualitative trajectory information, such as wall following [36] to travel between distinguishable landmarks, is suitable for static structured environments, but may guide the robot to an improper vicinity of the right place in more complex and dynamic environments. Essentially, unless the environment possesses at least one globally unique sequence of landmark positions, loop closure must always be ambiguous. This is the key drawback in the topological map paradigm, since environmental similarity may eventually generate a consistently similar sequence of places and result in wrong data association. The solution to this problem is to introduce some form of metric information, which would enable the estimation of pose uncertainty between landmarks. By bounding the cycle exploring space, place sequences only need to be locally unique.

In our underwater scenario, in undersea environments, where position identification is quite complicated, there is more risk of fail to localize [71]. Due to the lack explicit physical measurements in the seabed, the topological maps are not adequate for robotic underwater SLAM-based navigations.

2.4. The landmark maps

Landmark maps, also called as the feature maps, use geometric primitives such as corners, points, and lines to describe the working environment, as shown in Figure 2.3. The introduction of new geometric primitives increases the complexity of the map. The features can be artificial landmarks [65]; natural landmarks, such as outdoor trees [72], indoor line segments; and they can be of abstract form, detected by certain algorithms [73,74]. Localization is performed by extracting features from sensor measurements and associating them to other features that are already stored in the map. Then, the differences between the predicted feature locations and the detected positions are used to calculate the robot pose by the estimation filters. In this way, localization is very like a multiple target tracking problem [75], but unlike normal target tracking, here the targets are static and the observer is in motion.

Figure 2.3 The landmark map

Localization using a landmark map is a parameter estimation problem to determine the vehicle pose (x, y, φ) , given the map feature information and a set of feature observations. Assuming the measurements are correctly associated to the appropriate map features, the vehicle pose can be tracked using standard estimation techniques, the EKF, which is the most common method applied to this problem. Recursive EKF pose estimation has the advantages of efficient data fusion from multiple sensor measurements and the ability to incorporate explicit sensor uncertainty models. Besides, the size of storing a landmark map is very small comparing to an occupancy grid map or a 3D map, and it has high flexibility in map adjustment. In a 2D landmark-based map, feature positions are stored in the Cartesian coordinate system. A covariance matrix associated with the map is used to represent the uncertainties of both landmark positions and robot poses [76].

Besides the advantages, landmark maps have few limitations. First of all, the underwater landmarks have to be extracted from noisy sensor measurements, so it is required to identify features in the noisy observations. Secondly, a correct data association is essential to build consistent landmark-based maps [76]. Incorrect data association will reduce the accuracy of a landmark map and even leads the filter to diverge. A further problem concerning feature maps is that they are only suitable for environments where the observed objects can be reasonably depicted by basic geometric feature models, but the introduction of new geometric primitives would increase the map complexity.

Landmark map-based SLAM comprises the dual task of adding newly observed features to the map using the robot pose as a reference, while applying existing map features to reckon the robot pose iteratively. Therefore, the uncertainty of sensor measurements results in uncertain estimates of both the robot pose and the map feature positions, and these uncertainties are dependent. Correlated uncertainty is of importance to feature-based SLAM since it inevitably

couples the individual landmarks to each other and the robot to the map. Attempts to estimate the robot pose and map features independently have been shown to produce inconsistent uncertainty estimates.

In summary, the landmark maps are the most suitable ones to demonstrate the robotic underwater SLAM-based navigation, where stable landmarks are observable and tractable landmarks are present. The stationary points, which are the least complex features, are taken into account in this paper to describe the landmarks within a local subsea region. This simplification not only decreases challenges with feature recognition and interpretation, but also increases the focus on the presented AEKF-SLAM algorithm itself. Its computation is tractable and accumulated state uncertainty does not exceed conservative limits. So as to achieve robust and convergent applications in larger marine environments with continuously moving objects, modifications to the conventional stochastic SLAM approaches are required.

2.5. Summary

Learning maps is a crucial task for autonomous mobile robots, since successful robotic systems rely on maps for localization, path planning, activity planning, etc. This chapter summarizes and compares several classic and popular map representations in robotic underwater navigation applications. A survey of their suitability for large-scale reliable underwater SLAM problem has been discussed, taking into account computational complexity, storage, convergence, robustness, reliability, etc. And thereby this evaluation leads to the selection of metric landmark-based maps as a best choice for most situations. For our case, consider the sparse spatial distribution of the marine features, landmark maps are the most suitable one to describe the underwater environment.

Occupancy grid maps can model the map uncertainty and they are excellent for local navigation and obstacle avoidance purposes, but they do not scale well to large environments. As for topological maps, they are excellent for fast path planning and global navigation. However, in unstructured environments like underwater where location recognition is more complex, there is more risk to fail the localization. Considering the sparse spatial distribution of the marine features, the distances between them is too far, thus landmark maps have been chosen to describe the undersea environment.

3 Underwater feature detection

Exploring the seabed is a vital operation that has great importance for industrial, military, and environmental applications. One essential and distinct part of exploring the seas is imaging the seabed. Here, two main aims are usually considered for underwater imaging: seabed mapping and object recognition also called feature identification with special interest in objects lying on the seabed. An automatic detection and localization of underwater objects is of great importance in AUV navigation and mapping applications. Object detection aims at separating a foreground object from the background and generating a binary image for every frame, which is of critical importance for the sonar image processing since the result will directly affect the accuracy of the following feature extraction and object localization [77]. Thresholding is an important technique for image segmentation, and many researchers have paid a lot of attention to the methods of how to select reasonable thresholds in sonar images [78]. Feature extraction is an important aspect of SLAM, in which a mobile robot with a known kinematic model, starting at an unknown location, moving through the exploring environment where contains multiple features to incrementally generate a consist map. Geometric features, such as points, lines, circles and corners are determined as a part of the SLAM process, since these features can be used as landmarks. Generally, SLAM consists of the following parts including motion sensing, environment sensing, robot pose estimation, features extraction and data association. The main focus of this work is on first accurate and fast extracting features from underwater sonar images of different types, and then using them as landmarks for an AEEK-based robotic underwater SLAM navigation.

3.1. Sensor selection

Considering the nature of the underwater environment, sonars have been widely used for detecting objects on the ocean bottom since sound waves propagate long distances in the aquatic environment. When the acoustic signals encounter an object, part of the energy will be reflected and can be detected by the sensor, another part of the energy will be diffracted and keeps propagating. Echo strength is determined not only by the distance the acoustic signal travels, but also by the material, shape and size of the object.

There are many robotic SLAM applications in underwater environments. The application defines the selection of the perception sensors and the on-board navigation sensors as illustrated in the following Table 3.1. The SLAM problem can be stated from different approaches depending on the number of landmarks, area of coverage, computational requirements, flexibility, etc [50]. In this work, we use a depth sensor based on pressure for navigation and Mechanical Scanned Imaging Sonars (MSIS) as perception sensors to get horizontal positions of

features of interest, therefore by combining with the vertical positioning data obtained through pressure sensor, a 3D undersea representation map could be generated.

Table 3.1 Common instrumentations used in underwater SLAM

On-Board Navigation Sensors	Description	Perception Sensors	Description
Global Position System (GPS)	Used as an initial reference position.	Multibeam Echo Sounder (MES)	Array of transducers that emits and receives a fan-shaped beam to the seabed.
Depth Sensor	Vertical positioning through pressure sensors.	Mechanical Scanned Imaging Sonars (MSIS)	Mechanically rotated transducer that scans a horizontal 2D area.
Inertial Measurement Unit (IMU)- Accelerometer and Gyroscope	3-axis linear acceleration and angular rates.	Forward Looking Sonar (FLS)	Gives an acoustic image of what is in front of the vehicle.
Magnetometer	Sense magnetic field to estimate roll, pitch and roll angles.	Side Scan Sonar (SSS)	Provides downward looking acoustic image.
Doppler Velocity Logs (DVL)	Sense water velocity relative to the vehicle.	Video	Provides high-resolution images easy to interpret but limited in ranges.

3.2. Sonar image characteristics

Due to the particular optical properties of light in water and the presence of suspended particles, sonar images are very noisy, the lighting is not uniform, the colors are muted and the contrast is pretty low. Acoustic images include three parts of information, two parts of which come from objects and unknown objects and the third one comes from echo noise. The acoustic image is formed by the echo intensity of reflection from the 3D underwater environment. Compared with optical images, the acoustic images have some characteristics listed as follows. (1) The correlation in the acoustic image is strong and the resolution of image is very low. (2) The signal to noise ratio is lower than that of an optical image. (3) Different regions may be generated for the same object in the acoustic images.

In recent years, the resolution of sonar imagery has improved significantly, such that it can be used in a much better way for further processing and analyzed with advanced digital image processing techniques. Noise filtering, radiometric corrections, contrast enhancement, deblurring through constrained iterative deconvolution, and feature extraction are usually employed to correct or to alleviate flaws in the recorded data [79]. The first step of underwater

object detection is to segment the foreground features from the background. Segmentation is the process of assigning a label to every pixel in the image such that pixels with the same label share homogeneous characteristics, like color, intensity, or texture, and thereby different entities visible in the sonar imagery could be separated. A sonar image is made up of a matrix of pixels having a gray level typically on a scale from 0 to 255. The gray levels of pixels associated with foreground objects are essentially different from those belonging to the background. Normally, in typical sonar imagery, the object is composed of two parts: the highlighted areas (echo) and the shadow regions. The echo information is caused by the reflection of the emitted acoustic wave on the object while the shadow zones correspond to the areas lack of acoustic reverberation behind the object. Based on this characteristic, the threshold segmentation methods (TSM) can be used to detect the foreground object in the sonar image. In general, the Otsu method is one of the most successful adaptive methods for image thresholding [78].

The colors used for representing the sonar image are referred to as the colormap. Figure 3.1 shows the same sonar image with different colormaps. Each colormap would be appropriate in certain condition, and the most commonly used ones are Copper's and Jet's colormaps (both rightmost images). In this paper, Copper's colormap is chosen to display the sonar images since we do not need much more details of the features to achieve the capability of detection and localization, and we want to spot objects as soon as possible.

Figure 3.1 Sonar images with different colormaps

Typically, the SLAM implementations require a set of sparse features that are sufficiently distinguishable to allow correct association of observations with corresponding map features. Underwater environments may cause large uncertainty on the position of a platform without inertial sensors. Thus, it is desirable to have a sufficient number of features to ensure that multiple observations are continuously available. There are two classes of features used in the

underwater SLAM implementations. A large number of transient features are tracked to provide accurate short-term motion estimation, and a smaller set of stable features are maintained as landmarks.

We have chosen an AEKF-based implementation of the stochastic map to perform robotic SLAM navigation. The algorithm relies on a MSIS for extracting features [80]. Although these mechanically actuated devices usually have a low scanning rate, they are quite popular due to their lower cost. When working with MSIS, it is common to assume that the robot remains static or moves slowly enough in order to neglect the induced acoustic image distortion.

3.2.1. Side scan sonar images

Under optimal conditions, side-scan sonars (SSSs) can generate an almost photorealistic, two-dimensional picture of the seabed. Once several swatches are joined via mosaicing, geological and sedimentological features could be easily recognized and their interpretation would provide a valuable qualitative insight into the topography of the seabed [12]. Due to the low grazing angle of the SSS beam over the seabed, SSSs provide far higher quality images than Forward Looking Sonars (FLSs), such that the feature extraction and the data association processes will behave better. The accuracy of SLAM using SSS is more dependent on the distribution of landmarks. In general, the SSS and the multibeam FLS provide large-scale maps of the seafloor that are typically processed for detecting obstacles and extracting features of interest on the areas of the seafloor [16].

By transmitting and receiving sound via an underwater sonar system, the seafloor terrain and texture information can be extracted with relevant data from the acoustic reflection image of the seabed. High image resolutions are very important for determining if an underwater target is something worth investigating. Two SSS images are used in this work (see Figure 3.2). Figure 3.2 (a) is of high resolution, whereas Figure 3.2 (b) is a low-resolution SSS image. Actually, the DeepEye340D SSS (Deep VisionAB company, Linköping, Sweden), the 3500 Klein SSS (Klein Marine System, Inc., Salam, USA), and the Blue View P900-90 2D FLS (Teledyne BlueView, Bothell, USA), all are employed in our “Smart and Networking Underwater Robots in Cooperation Meshes”-SWARMs European project [133], that is the reason why we perform feature detection on these sonar images. DE340D SSS is a product of the Deep VisionAB company [81], working at the frequency of 340 kHz with an optimized resolution of 1.5 cm. The beam reaches from 15 to 200 m and the maximum operation depth is 100 m. The size of the converted Figure 3.2 (a) is $800 \times 800 = 640,000$ pixels. Due to the impact of sediment and fishes, many small bright spots appear in sonar images. The area size of these background spots is usually smaller than 30 pixels, and their gray level is similar to that of certain areas of

foreground object. When using the traditional TSM to separate the foreground object, in this case a shipwreck, it can be found that most of these spots are still retained in the segmentation results (see Figure 3.5). To solve this problem, an improved Otsu method is proposed that constrains the search range of the ideal segmentation threshold to separate the foreground object inside the sonar imagery.

Figure 3.2 (a) High-resolution SSS image recorded with the DE340D SSS at Stockholm sea; (b) Low-resolution SSS image generated by the 3500 Klein SSS (ECA Group company [82])

It is typical for SSS images to show a vertical white line appearing in the center, indicating the path of the sonar. A dark vertical band on both sides represents the sonar return from the water-column area. Notice that this band has no equal width, and is curved in some parts. The distance from the white line to the end of the dark area is equivalent to the depth to the sea bottom below the sonar device. The seabed is imaged on either side, corresponding to port-side and starboard-side views. Bright areas indicate ascending ground, while dark areas correspond to descending regions or shadows produced by objects, vegetation or rocks. The length of a shadow can be used to calculate the height of an object.

3.3. The proposed improved Otsu TSM

When using the traditional threshold segmentation method (TSM) to segment the foreground object, in this case a shipwreck, most of the background spots are still retained in the segmentation results (see Figure 3.5). To solve this problem, an improved Otsu TSM is presented that constrains the search range of the ideal segmentation threshold to extract the foreground object inside the image. Since the area size of the background spots shown in Figure 3.2 (a), is usually no bigger than 30 pixels, the parameter N_{30} has been defined as the number of contours to be found with an area size smaller than 30 pixels. The procedure of the improved

Otsu approach is illustrated in Figure 3.3. At first, the traditional Otsu method [83] is used to calculate the initial segmentation threshold T . Then, the Moore Neighbor contour detection algorithm [84,85] is employed to compute N_{30} . If $N_{30} > 300$, ($64000/30 \times 300 = 71.1:1$), it means that there are still many small bright spots remaining in the segmentation result, and the threshold needs to be improved. The final segmentation threshold T^* can be calculated as explained further on. If $N_{30} \leq 300$, the final segmentation threshold T^* is set as the initial segmentation threshold T , and segmentation is end. Notice that both values, N_{30} and 300 should be changed depending on the characteristics of the used sonar images.

Figure 3.3 The procedure of the improved Otsu TSM

In the gray level range of one plus the initial segmentation threshold T calculated by the traditional Otsu method to the gray level of 255, denoted as $[T + 1, \dots, 255]$, the number of pixels at gray level i is denoted by n_i , and the total number of pixels is calculated by:

$$N = \sum_{i=T+1}^{255} n_i. \quad (3.1)$$

The gray level histogram is normalized and regarded as a probability distribution:

$$p_i = \frac{n_i}{N}, p_i \geq 0, \sum_{i=T+1}^{255} p_i = 1. \quad (3.2)$$

Supposing that the pixels are dichotomized into two categories C_0 and C_1 by a threshold T^* . The set C_0 implies the background pixels with a gray level of $[T + 1, \dots, T^*]$, and C_1 means those pixels of the foreground object with a gray level of $[T^* + 1, \dots, 255]$. The probabilities of gray level distributions for the two classes are the following: ω_0 is the probability of the background and ω_1 is the probability of the object:

$$\omega_0 = P_r(C_0) = \sum_{i=T+1}^{T^*} p_i, \quad \omega_1 = P_r(C_1) = \sum_{i=T^*+1}^{255} p_i = 1 - \omega_0. \quad (3.3)$$

The means of the two categories C_0 and C_1 are:

$$u_0 = \sum_{i=T+1}^{T^*} i p_i / \omega_0, \quad u_1 = \sum_{i=T^*+1}^{255} i p_i / \omega_1. \quad (3.4)$$

The total mean of gray levels is denoted by:

$$u_M = \omega_0 u_0 + \omega_1 u_1. \quad (3.5)$$

The two class variances are given by:

$$\sigma_0^2 = \sum_{i=T+1}^{T^*} (i - u_0)^2 p_i / \omega_0, \quad \sigma_1^2 = \sum_{i=T^*+1}^{255} (i - u_1)^2 p_i / \omega_1. \quad (3.6)$$

The within-class variance is:

$$\sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2. \quad (3.7)$$

The between-class variance is:

$$\sigma_B^2 = \omega_0 (u_0 - u_M)^2 + \omega_1 (u_1 - u_M)^2 = \omega_0 \omega_1 (u_0 - u_1)^2. \quad (3.8)$$

The total variance of the gray levels is:

$$\sigma_M^2 = \sigma_W^2 + \sigma_B^2. \quad (3.9)$$

The final threshold T^* is chosen by maximizing the between-class variance, which is equivalent to minimizing the within-class variance, since the total variance, which is the sum of the within-class variance σ_W^2 and the between-class variance σ_B^2 , is constant for different partitions:

$$\therefore T^* = \arg \left\{ \max_{T+1 \leq i \leq 255} \{\sigma_B^2(i)\} \right\} = \arg \left\{ \min_{T+1 \leq i \leq 255} \{\sigma_W^2(i)\} \right\}. \quad (3.10)$$

3.3.1. The power-law transformation

Nevertheless, the Otsu method does have its limitations in that the correct image segmentation cannot be obtained when the gray level of objects is closely approximate to that of the background or the proportion of objects is low. To solve this problem, the gray level for objects should be increased and the one for the background should be decreased for enhancing the contrast and highlighting before using the segmentation method. For this purpose, the power-law transformation has been employed, which has the following basic form:

$$y = cx^r, \quad (3.11)$$

where x and y are the input and output gray levels, c and r are positive constants. With the help of this power-law transformation, the gray level of each pixel can be easily changed. The plots of this equation with various r values are displayed in Figure 3.4. Different values of c and r affect the segmentation result directly, many experiments have been done and it has been figured out that the best thresholding result is achieved when the value of c is 0.1 and the value of r equals 1.415.

Figure 3.4 The plots of the power-law equation with different r values

3.4. TSM results for sonar images

For comparison, the SSS images in Figure 3.2 (a) and Figure 3.2 (b) have been segmented with the traditional Otsu method, the local TSM, the iterative TSM, the maximum entropy TSM and our method, respectively. The local TSM adapts the threshold value on every pixel to the local image characteristics, and a different threshold is selected for each pixel in the image. As for the iterative TSM, it compares the threshold value of each pixel with the average of the

maximum and the minimum thresholds in the image. The maximum entropy TSM employs the entropy of the foreground and background regions, the cross-entropy between the original and binarized image, etc. [86].

3.4.1. TSM results for high-resolution SSS image

The thresholding segmentation results of the above four classic segmentation methods and the improved Otsu TSM on Figure 3.2 (a) are shown in Figure 3.5 and Figure 3.6, respectively. The output after the TSM operation is a binary image which indicates the object with a gray level of 255 (white pixels in 8-bit images) and the background with a gray level of 0 (black pixels in 8-bit images). Compared with other conventional TSM, it is obvious that the improved Otsu TSM could reduce the influence of noise and also the small bright spots in sonar images, since most of them have been divided into the background. In addition, the segmentation results of the presented method are more precise.

(a)

(b)

(c)

(d)

Figure 3.5 (a) Traditional Otsu TSM, $T_h = 0.3216$; (b) Local TSM, $T_h = 0.1628$; (c) Iterative TSM, $T_h = 0.4238$; (d) Maximum entropy TSM, $T_h = 0.6627$

Edges exist between objects in contact, between objects and background between different fields. As a crucial characteristic of an image, an edge indicates the position of its outline. Consequently, edge detection is the first necessary step in detecting objects and its result directly domains the ability of system recognition and classification [Sensors 1: 46]. In this paper, the Canny edge detector is used as a feature extraction tool. The main stages of the Canny edge detection algorithm (Algorithm 1) [87] are described as follows, and the result is illustrated in Figure 3.6 (a).

Algorithm 1: Canny edge detection

1. Smooth the image with a Gaussian filter, $h = fspecial('gaussian', [3\ 3], 0.5)$;
 2. Calculate the gradient's amplitude and orientation with the finite-difference for the first partial derivative;
 3. Non-Maxima Suppression;
 4. Detect and link the edge with double threshold method, $y = edge(b, 'canny', 0.33)$, the high threshold for Figure 3.2 (a) is 0.33, and the 0.4 times high threshold is used for the low threshold.
-

The initial segmentation threshold T computed by the traditional Otsu method is 0.3216, and the parameter N_{30} returned from the above Canny edge algorithm equals to 752, which is bigger than 300. Therefore, our improved Otsu TSM has been applied, and the segmentation result is shown in Figure 3.6 (b), with final threshold T^* of 0.6784. In order to detect the centroids of each segmented region, we need to do the following morphological operations (Algorithm 2) with Figure 3.6 (b).

Algorithm 2: Morphological operations for detecting feature centroids

1. Remove all connected components that have fewer than 30 pixels in Figure 3.6 (b);
 2. Bridge previously unconnected pixels;
 3. Perform dilation using the structuring element ones (3) with the size of a 3×3 square;
 4. Fill the holes in the image;
 5. Compute the area size, the centroid and the bounding box of different contiguous regions;
 6. Concatenate structure array which contains all centroids into a single matrix.
-

Figure 3.6 (a) Canny edge detection after applying the traditional Otsu method, $bw = \text{edge}(b, 'canny', 0.33)$, $N_{30} = 752 > 300$; (b) Improved Otsu TSM, $T=0.3216$, $T^* = 0.6784$; (c) Result of the improved Otsu TSM after morphological operations marking the centroids of the obtained regions; (d) Result of the maximum entropy TSM after the same morphological operations marking the centroids of the acquired areas

In Figure 3.6 (c), the red stars ' * ' mark the centroids for each contiguous region or connected component in this image. We set the top left corner as the coordinate origin, and the horizontally to the right direction as the x -axis, and the vertically downward direction as the y -axis. The centroid coordinates of all connected regions within the foreground shipwreck are (535, 603), (542.3, 653.9), which will be used as point landmarks in the further test of AEKF-based robotic SLAM loop mapping. So the central centroid of this ship is (538.3, 628.5), which is calculated as the average of the above two centroid positions. The same morphological operations for marking the feature centroids is performed on the segmentation result of the maximum entropy TSM, shown in Figure 3.6 (d). The confusion matrix of the real ship centroids and the ones detected by our proposed improved Otsu TSM is shown in the following Table 3.2.

Table 3.2 The confusion matrix of real and detected ship centroids using the improved Otsu TSM

		Detected	
		Ship Centroids	Non-Ship Centroids
Real	Ship Centroids	2	0
	Non-Ship Centroids	4	21

Thus, the False Positive Rate (FPR) of ship centroids detected by the improved Otsu TSM is:

$$FPR = \frac{FP}{FP+TN} = \frac{4}{4+21} = 0.16. \quad (3.12)$$

Among several classic segmentation methods compared above, the maximum entropy TSM achieves the best segmentation performances. To compare it here with the proposed method, the following Table 3.3 shows also the confusion matrix of real centroids and the ones detected by the maximum entropy TSM.

Table 3.3 The confusion matrix of real and detected ship centroids using the maximum entropy TSM

		Detected	
		Ship Centroids	Non-Ship Centroids
Real	Ship Centroids	2	0
	Non-Ship Centroids	8	20

In this case, the FPR of ship centroids detected by the maximum entropy TSM is:

$$FPR = \frac{FP}{FP+TN} = \frac{8}{8+20} = 0.29, \quad (3.13)$$

which is higher than the FPR of the proposed improved Otsu TSM. As a further performance indicator, the detection precision, also called Positive Predictive Value (PPV), has been calculated for both segmentation methods. For the improved Otsu TSM it is:

$$PPV = \frac{TP}{TP+FP} = \frac{2}{2+4} = 0.33, \quad (3.14)$$

while the maximum entropy TSM only leads to a lower value:

$$PPV = \frac{TP}{TP+FP} = \frac{2}{2+8} = 0.2. \quad (3.15)$$

Indeed, both precision rates are not really high because the proportion of the foreground ship feature is small, and the centroids in some parts of the background, where their gray levels are similar to that of the ship, are also detected. So, we consider even more important that the improved Otsu TSM shows a better performance and therefore seems to be more robust. The

computational cost of the improved Otsu TSM has also been compared with the above four conventional segmentation approaches on Figure 3.2 (a) is shown in Table 3.4.

Table 3.4 Computational costs of different segmentation methods on Figure 3.2 (a)

Segmentation Method	Computational Time [s]
Traditional Otsu TSM	0.178226
Local TSM	0.913942
Iterative TSM	0.289513
Maximum entropy TSM	1.562499
Improved Otsu TSM	0.868372

In general, the improved Otsu TSM achieves a more accurate and fast segmentation on the SSS image shown in Figure 3.2 (a). Although the computational time of the improved Otsu TSM is 4.9 times higher than that of the classic Otsu method, it is only half of that of the maximum entropy TSM, which achieves the highest segmentation precision among the four classic segmentation methods compared above.

3.4.2. TSM results for low-resolution SSS image

To further compare the performance of the proposed segmentation algorithm with lower-resolution images, the same process has been realized over the image shown in Figure 3.2 (b). The size of the image is $417 \times 228 = 95,076$ pixels. Since the background spots, whose gray level are similar to those of some parts of the object (a branch) in the foreground, usually have an area size not bigger than 15 pixels, the parameter N_{15} has been defined as the number of contours to be found within an area size smaller than 15 pixels. If $N_{15} > 100$, $(^{95076}/_{15} \times 100 = 63.4:1)$, this assigned threshold of 63.4 is lower than that of 71.1 for Figure 3.2 (a), since the proportion of background spots in this low-resolution SSS image is higher than that in Figure 3.2 (a).

(a)

(b)

Figure 3.7 (a) Traditional Otsu TSM, $T_h = 0.1137$; (b) Local TSM, $T_h = 0.0941$; (c) Iterative TSM, $T_h = 0.2609$; (d) Maximum entropy TSM, $T_h = 0.3176$

Also, the lower resolution of the image in Figure 3.2 (b) and the rough texture of the seabed leads to many small bright spots in the segmentation result, and the threshold needs to be improved. Figure 3.7 shows the results of several classic segmentation methods and Figure 3.8 (b) our improved Otsu TSM. The initial segmentation threshold T calculated by the traditional Otsu method is 0.1137. In Figure 3.8 (a), the parameter N_{15} computed by the Canny contour detection algorithm is 419, which is bigger than 100. As a result, the proposed improved Otsu TSM has been applied, and the segmentation result is shown in Figure 3.8 (b), with the final threshold T^* of 0.3529. The morphological operations for marking the centroids of every segmented region within the branch are similar to that of the ship. Only in step 1, the parameter is set to 15 to remove all connected components that have fewer than 15 pixels. The red stars ' $*$ ', shown in Figure 3.8 (c) and Figure 3.8 (d), imply the centroids for every contiguous region or connected component in the segmentation results of our improved Otsu TSM and the maximum entropy TSM, separately. The centroid coordinate of the branch detected by our method is (187.3, 115.6), which will be used as a landmark point in the further simulation test of an AEKF-based SLAM loop mapping. The confusion matrices of the real centroids and the ones detected by the improved Otsu TSM on the one hand and the maximum entropy TSM on the other hand are shown in the following Tables 3.5 and Table 3.6, separately.

Figure 3.8 (a) Canny edge detection after applying the traditional Otsu method, $bw = \text{edge}$ (b, 'canny', 0.1255), $N_{15} = 419 > 100$; (b) Improved Otsu TSM, $T=0.1137$, $T^* = 0.3529$; (c) Result of the improved Otsu TSM after morphological operations marking the centroids of the obtained regions; (d) Result of the maximum entropy TSM after the same morphological operations marking the centroids of the acquired areas

Table 3.5 The confusion matrix of real and detected branch centroids using the improved Otsu TSM

		Detected	
		Branch Centroids	Non-Branch Centroids
Real	Branch Centroids	1	0
	Non-Branch Centroids	1	13

Therefore, the FPR of branch centroids detected by the improved Otsu TSM is:

$$FPR = \frac{FP}{FP+TN} = \frac{1}{1+13} = 0.07. \quad (3.16)$$

And the precision of branch centroids detected by the improved Otsu TSM is:

$$PPV = \frac{TP}{TP+FP} = \frac{1}{1+1} = 0.5. \quad (3.17)$$

Table 3.6 The confusion matrix of real and detected branch centroids using the maximum entropy TSM

		Detected	
		Branch Centroids	Non-Branch Centroids
Real	Branch Centroids	1	0
	Non-Branch Centroids	7	11

As a result, the FPR of branch centroids detected by the maximum entropy TSM is:

$$FPR = \frac{FP}{FP+TN} = \frac{7}{7+11} = 0.39, \quad (3.18)$$

which is 5.5 times bigger than the FPR of the proposed improved Otsu TSM. The precision of branch centroids detected by the maximum entropy TSM is:

$$PPV = \frac{TP}{TP+FP} = \frac{1}{1+8} = 0.11, \quad (3.19)$$

which is much lower than the precision of the improved Otsu TSM. As in the previous image, both precision values are low, since the ratio of the foreground branch feature is small, and the centroids of certain parts of the background, where their gray levels are also close to that of the branch are also detected, but the difference between precision values is even bigger, 4.5 times higher for the improved Otsu TSM in this image, 1.65 times higher in the formerly presently high-resolution image. This proves that the proposed improved Otsu TSM is very valuable, above all for standard sonar images. The computational cost of the proposed improved Otsu TSM is compared with the above four classic segmentation approaches on Figure 3.2 (b) and is shown in Table 3.7.

Table 3.7 Computational costs of different segmentation methods on Figure 3.2 (b)

Segmentation Method	Computational Time [s]
Traditional Otsu TSM	0.120458
Local TSM	0.261021
Iterative TSM	0.227290
Maximum entropy TSM	0.378283
Improved Otsu TSM	0.241164

In conclusion, the testing of different SSS images with different resolutions shows that the proposed improved Otsu TSM keeps a good tradeoff between segmentation precision and computational cost. The computational time of the improved Otsu TSM on this low-resolution SSS image, shown in Figure 3.2 (b), is only two times higher than that of the traditional Otsu TSM. As for the former high-resolution SSS image, shown in Figure 3.2 (a), it is nearly five times higher. Therefore, the improved Otsu TSM performs even better for lower-resolution images both in segmentation precision and processing time and seems to be more robust. In comparison, the configuration-conjunct TSM proposed in [88], which is only suitable for detecting simple linear objects with neat and obvious edges, needs a computing time of 0.371 s on a low-resolution sonar image of size $140 \times 228 = 31,920$ pixels. Executed on that same image, our improved Otsu TSM only spends 0.117 s segmenting the object (a pipe), this means that the configuration-conjunct TSM consumes three times more processor time than our improved Otsu TSM. As a result, the improved Otsu TSM provides a much faster segmentation than other state of the art approaches for detecting underwater objects of different shapes.

3.4.3. TSM results for FLS image

Usually, a preliminary mission of AUV is data collection, generally accomplished by means of SSS or multibeam echosounder, another key issue is to ensure the safety of the AUV. For the purpose of avoiding obstacles, the AUV could be equipped with a Forward Looking Sonar (FLS) to sense the working environment at a certain distance in the direction of navigation. The FLS platform emits a short acoustic pulse in forward direction on a horizontal sector of around 120° , and on a vertical sector from 15° to 20° . The original FLS imagery used here (see Figure 3.9) is provided by Desistek Robotik Elektronik Yazilim company (Ankara, Turkey) [89]. It is recorded with the Blue View P900-90 2D Imaging Sonar, which has a 90° horizontal field of view, and works at a frequency of 900 kHz. Its update rates are up to 15 Hz, the scanning range is 100 m, and its resolution is 2.51 cm.

Figure 3.9 The original FLS image comes from [89], and there is a plastic mannequin in the down center

The size of the converted FLS imagery is $1920 \times 932 = 1,789,440$ pixels, the area size of the background spots is usually fewer than 40 pixels, with gray levels similar to that of certain areas of the foreground objects. In this case, N_{40} is defined as the number of contours which area size is smaller than 40 pixels, and it is computed by the Canny contour detection algorithm. If $N_{40} > 600$, $(1789440/40 \times 600 = 74.6:1)$, this assigned threshold 74.6 is higher than that of 71.1 for the former high-resolution SSS image of the ship, that is because the black background proportion in this presented FLS image is higher than that in that SSS image. This means that there are many small bright spots still left in the segmentation result.

The initial FLS image in Figure 3.9 has been segmented with the traditional Otsu method, the local TSM, the iterative TSM, the maximum entropy TSM and our method, respectively. The simulation results of the segmentations are shown in Figure 3.10 and Figure 3.11. The initial segmentation threshold T obtained via the traditional Otsu method is 0.1176. In Figure 3.11 (a), the parameter N_{40} calculated from the above Canny edge detection algorithm is 1314, which is

bigger than 600. Thus our improved Otsu TSM has been applied, and the segmentation result is shown in Figure 3.11 (b), with the final threshold T^* of 0.5412. The morphological operations for computing the centroids of every segmented region within the body are similar to that of the ship. Only in step 1, the parameter is set to 40, in order to remove all connected components that have fewer than 40 pixels. Besides, in step 3, it applies dilation two times.

Figure 3.10 (a) Traditional Otsu TSM, $T_h = 0.1176$; (b) Local TSM, $T_h = 0.0941$; (c) Iterative TSM, $T_h = 0.2990$; (d) Maximum entropy TSM, $T_h = 0.4118$

Figure 3.11 (a) Canny edge detection after applying the traditional Otsu method, $bw = \text{edge}(b, 'canny', 0.13)$, $N_{40} = 1341 > 600$; (b) Improved Otsu TSM, $T=0.1176$, $T^* = 0.5412$; (c) Result of the improved Otsu TSM after morphological operations marking the centroids of the obtained regions; (d) Result of the maximum entropy TSM after the same morphological operations marking the centroids of the acquired areas

The red stars ' $*$ ', shown in Figure 3.11 (c), stand for the centroids for each contiguous region or connected component in this image. The centroid coordinates of every connected region within the foreground object are (949.8, 662.9) , (966.9, 660.6) , (1021.2, 615.7) , (1024.1, 703) , (1065.5, 811.3) , which will be used as landmark points in the further simulation test of an AEKF-based robotic SLAM loop mapping. So the center centroid of the person is (1005.5, 690.7), which is calculated as the average of the above five centroids. The same morphological operations for marking the feature centroids is employed on the segmentation result of the maximum entropy TSM, shown in Figure 3.11 (d). The confusion matrices of the real body centroids and the ones detected by the improved Otsu TSM on the one hand and the maximum entropy TSM on the other hand are shown in the following Table 3.8 and Table 3.9, respectively.

Table 3.8 The confusion matrix of real and detected body centroids using the improved Otsu TSM

		Detected	
		Branch Centroids	Non-Branch Centroids
Real	Branch Centroids	5	0
	Non-Branch Centroids	11	26

As for the former SSS images, we calculate the FPR and PPV indicators:

$$FPR = \frac{FP}{FP+TN} = \frac{11}{11+26} = 0.297, \quad (3.20)$$

$$PPV = \frac{TP}{TP+FP} = \frac{5}{5+11} = 0.3125. \quad (3.21)$$

Table 3.9 The confusion matrix of real and detected body centroids using the maximum entropy TSM

		Detected	
		Branch Centroids	Non-Branch Centroids
Real	Branch Centroids	2	3
	Non-Branch Centroids	44	40

In this case, the FPR and the precision value are:

$$FPR = \frac{FP}{FP+TN} = \frac{44}{44+40} = 0.524, \quad (3.22)$$

$$PPV = \frac{TP}{TP+FP} = \frac{2}{2+44} = 0.043. \quad (3.23)$$

As can be seen, also for the FLS images, similar performance is achieved with the improved Otsu TSM. The FPR value is 1.8 times lower and the precision is seven times higher, even better than for the SSS images, which means that, lower the image quality, the better the detection performance for important features. Besides, three real body centroids are not detected at all by the maximum entropy TSM. In general, in all SSS and FLS images presented in this work, the proposed improved Otsu TSM has much lower FPR and much higher precision rate on the detected feature centroids than those values of the maximum entropy TSM.

Finally, again the computational cost of the proposed improved Otsu TSM is also compared with above four conventional segmentation methods on Figure 3.9 and is shown in Table 3.10. It is higher than the traditional Otsu TSM, the local TSM and the iterative TSM, but nearly one third of the maximum entropy TSM, while achieving much better detection rates.

Table 3.10 Computational costs of different segmentation methods on Figure 3.9

Segmentation Method	Computational Time [s]
Traditional Otsu TSM	0.244472
Local TSM	0.941853
Iterative TSM	0.428126
Maximum entropy TSM	3.903889
Improved Otsu TSM	1.452562

In general, the improved Otsu TSM constrains the search range of the ideal segmentation threshold, and combined with the contour detection algorithm, the foreground object of interest, a ship, a branch and a body have been separated more accurately, in sonar images of very different resolutions and qualities, with a low computational time. Compared with the maximum entropy TSM, which has the highest segmentation accuracy among the four conventional segmentation approaches compared above, our improved Otsu TSM just needs half of the processing time for segmenting the ship in the high-resolution SSS image. Regarding the branch in the low-resolution SSS image, our method consumes two thirds of the time, and for the body in the presented FLS image, it only spends one third of the computational time used by the maximum entropy TSM. As a consequence, the improved Otsu TSM achieves precise threshold segmentation performances for underwater feature detection at a fast computational speed. Since the computational time of the improved Otsu TSM is very short, it achieves real-time results which can be used afterwards for underwater SLAM. The centroids that have been

calculated for the different objects will be used as landmark points in the following AEKF-SLAM loop map simulation. Moreover, compared to the traditional Otsu method, the presented approach could retain more complete information and details of objects after segmentation, also the holes and gaps in objects are reduced.

3.5. Summary

No matter for low-resolution and even for high-resolution underwater sonar images, which are recorded by the Side Scan Sonar (SSS) or Forward Looking Sonar (FLS), most of the spots are still remained when using the traditional Threshold Segmentation Methods (TSMs) to separate foreground objects of the interest from background, including the traditional Otsu method, the local TSM, the iterative TSM and even the maximum entropy TSM, which has the highest segmentation accuracy among these four conventional approaches compared in this section. To solve this problem, an improved Otsu TSM has been presented by constraining the search range of the ideal segmentation threshold. And combined with the contour detection algorithm, the foreground objects of interest, a shipwreck, a branch and a body have been separated more precisely, in sonar images of various resolutions and qualities, with low computational complexity. It achieves real-time results which can be used afterwards for underwater SLAM based navigation applications. The centroids that have been calculated for the different objects will be used as landmark points in the following AEKF-SLAM loop map simulation experiments in Chapter 6.

Moreover, the proposed improved Otsu TSM algorithm could significantly reduce the effect of noise, and also the small bright spots in sonar images since most of them have been joined with the background, also the segmentation results are more accurate. What's more, compared to the traditional Otsu method, our presented approach could retain more complete information and details of objects after segmentation, also the holes and gaps in objects are greatly reduced. In all, the improved Otsu TSM algorithm achieves precise threshold segmentation performances for underwater feature detection at a fast computational speed.

4 Underwater sonar map fusion

Computer vision is a division of artificial intelligence, which impersonates human intelligence to extract information from images [90]. Image matching has been an active topic of research in computer science for the last two decades. The task of finding keypoint correspondences between two measured images of the same scene or object is an important part of computer vision applications, including image registration, camera calibration, object recognition, image retrieval, etc.

Sonar imaging is currently the exemplary choice applied in underwater imaging. However, since sound signals are absorbed by water, an image acquired by sonar will have gradient illumination; thus, underwater maps will be difficult to process. In this chapter, we will investigate this phenomenon with the objective to propose methods to normalize the sonar images with regard to illumination. To overcome the weakness found in previous approaches, in this section, first we propose to use Mixed exponential Regression Analysis (MIRA) estimated from each image that requires normalization. Two SSSs are used to capture the seabed in Lake Vättern in Sweden in two opposite directions WestEast (WE) and EastWest (EW); hence, the task is extremely difficult due to differences in the acoustic shadows. Using the structural similarity index, similarity analyses between corresponding regions extracted from the sonar images will be performed. Simulation results showed that MIRA has superior normalization performances.

Moreover, with the help of high-resolution underwater features in SSS images, an efficient feature detector and descriptor, Speeded Up Robust Feature (SURF) is employed to seabed sonar image fusion task. In this section, the underwater wireless sensor network-based Delaunay Triangulation (UWSN-DT) algorithm is proposed for improving the performances of sonar map fusion accuracy with low computational complexity, in which we regard the wireless nodes as underwater feature points. Since nodes could provide sufficiently useful information for the underwater map fusion, such as the location. In the following simulated experiments, it shows that our presented UWSN-DT algorithm works efficiently and robustly, especially for the subsea environments where there are few distinguishable feature points.

4.1. Intensity normalization of SSS imagery

Due to the high absorption of electromagnetic waves by water, sonar is the prime imaging device used underwater. Nevertheless, because of the imaging geometry and the way sonar imaging works, acquiring one image of some area of the seabed may not depict the seabed topology correctly. One of the major problems related to underwater sonar imaging is that the acoustic shadow depends on the direction of the sonar head relative to the seabed objects of

interests [91]. Maps fusion is an extremely complex task that requires sophisticated planning and prudent analysis; where different maps of the seabed may be acquired by different sonars, or using the same sonar but scanning is performed at different directions, and the objective would be fusing all these maps into one entity.

Contrary to the simple description aforementioned, the task is exceptionally challenging due to the harsh underwater environment that imposes problems in image acquisition by sonars. One major problem is the echo decay due to the high absorption rate of sound by water [91]. These factors affect the received echo and thus its strength will depend on the distance the sound has travelled. It is known that the echo decay follow the inverse square law for spherical sound [91,92]. The consequences of the echo decay on the image quality is a gradient of intensity values ranging from brighter values (seabed points close to the sonar head) that are gradually reduced to the outer borders of the image (seabed points far from the sonar head) [93]. Few models have been proposed to handle the echo decay, for example [92,93]. The models are based on some hypothesis adapted or modified from the inverse square law may be perturbed due to the environmental factors and the used imaging sonar.

4.1.1. The tested SSS image data

Two SSSs have been simultaneously used to capture the seabed in Lake Vättern in Sweden, an area that is relatively constant in depth. The scanning missions have been performed in two different directions, WestEast (WE) and EastWest (EW), thus a difference of 180° between the two overlapping regions. To test MIRA method and an adaption of known fog removal technique to sonar imaging, known as Dark Channel Prior (DCP) method, a benchmark has been manually extracted from the original non-normalized images in the form of pairs containing templated from the overlapping regions in WE and EW. The images have been recorded with a Deep Eye 680D SSS (chirp sonar, 680 kHz) This sonar is also available in an AUV mounted version that works with the same quality. Two sonars have been used, a left SSS and a right SSS. Most of the area is recorded in two directions WE and EW, with an overlapping region between WE and EW. The way of the acquisition has been performed is illustrated in Figure 4.1. The purpose of this dataset is to provide data for image processing in order to prepare maps fusion methods and a toolbox that will be used in the demonstrations and the use cases. Combing the left and right sonar images, each sonar image has more than 5000 pings with each ping having 2000 samples. The acquired data have three WE images and two EW images; for illustration WE3 is shown in Figure 4.2, and the outline of the amplitude of one single ping is depicted in Figure 4.3.

Figure 4.1 The scanning paradigm used to record the seabed images in Lake Vättern, Sweden. Overlapping regions are marked with a similar canvas pattern. Colours have no meaning here but visual clarity. The arrow exemplifies the movement of the vehicle carrying the imaging sonar

Fig. 4.2 One original SSS image (WE3) of the seabed displayed without further processing. The vertical axis shows the total number of pings, and the horizontal axis shows the number of samples (1000 in the left, and 1000 in the right), although -50 to 50 in the area covered in meters, $\log(x)$ has been used to compress the received acoustic signal x

Figure 4.3 One ping showing some form of exponential decay in the received acoustic signal, where $\log(x)$ has been used to compress the original values

Preprocessing is the very first necessary step in any computer vision and image processing task. There are various sources of noise and artefacts in sonar images that need to be removed using specialized filters and methods. One serious artefact that often occurs in sonar imaging is the existence of different intensities in the form of gradient. In sonar imaging, there is a decay in the received echoes as a function of range, due to sound absorption in seawater. This phenomenon is expressed as a non-linear form $f(\mathcal{A}, r)$, where $\mathcal{A}$ is a vector of parameters representing the absorption coefficient depending on sonar frequency, water temperature, depth, etc. And r is the range between the sonar and the object. There is also spherical spreading, expressed as $40 \log(r)$ in the same log scale. In addition, the seabed backscatter strength varies strongly with grazing angle, hence range and typical variations are in excess of 30 dB. Because of all this total dynamic range requirements of sonars exceed 100 dB. This dynamic range is much too large to be pictured in a single image, thus all sonar manufacturers should implement a proprietary dynamic range compression with the objective of reducing the dynamic range to around 40 dB. The goal is not only to make the image “nice to look at”, i.e. subjectively, but also to enable using computer vision and image analysis techniques on those correctly normalized images. In all the images used in this section, $\log(x)$ was used to compress the received acoustic signal x .

With regard to the dataset obtained in this chapter, the images show clearly this deficiency in relation to the range that the acoustic signals have travelled. To demonstrate this, the image in Figure 4.2 illustrates that the central region, along the vertical axis, which represents the water column where the left and right signals are combined to form the image, has a brighter appearance (higher intensity values). In the same time, the ping at both the left and right borders appears darker. To illustrate this phenomenon further, one ping is shown in Figure 4.3 and it is obvious that the relation between the echo strength and the distance is non-linear, in fact the relation appears to have some form of an exponential model. Therefore, the intensity drift with respect to the closest and the farthest points in the sonar image has some form of exponential decay. One way to normalize the intensities is by using regression analysis.

To test the similarity of the overlapping regions between WE and EW, sub-images of size 400×400 pixels have been extracted from each of them, and a couple of sub-images representing the same region is called a pair. To that end, 19 pairs have been extracted from the non-normalized images and 13 pairs have been extracted from the images normalized by the MIRA method. Afterwards, a template matching method has been implemented to find the

corresponding pair in the normalized images (using the 19 pairs) and the non-normalized images (using the 13 pairs).

Similarly, template matching has been implemented to extract 32 pairs from the images normalized by DCP. Due to the difficulty of extracting these pairs from the sonar images, human observer error is highly probable, therefore we tossed few pairs from the benchmark after extraction. To give an example, the extraction of few pairs from overlapping regions is illustrated in Figure 4.4, and four pairs extracted from different overlapping regions, along with the effect of intensity normalization, are illustrated in Figure 4.5.

Although imaging was performed at roughly the same distance between the sonars and the seabed, image scaling effects were obvious while manually extracting the overlapping regions, not to mention the possible error due to the manual extraction of each pair from the overlapping images.

4.1.2. The proposed MIXed exponential Regression Analysis (MIRA) method

One way to resolve the illumination change due to the decay in the acoustic signal is by normalizing every pixel in the image according to an estimated weight value. Using exponential regression analysis, the weights can be estimated from the intensity characteristics of the image by treating each ping separately, in our case these are the intensity values of one row in the image, i.e., the received echo of one ping. The simplest exponential model that one can assume is as follows:

$$f(z) = ae^{bz}, \quad (4.1)$$

where z is the index value at one sample in the ping, a and b are the parameters that should be estimated and f is the output of the fitting model that should be compared to the observed intensity $I(z)$ (via exponential regression analysis). Then, the normalized image should be estimated by using the following formula (4.2):

$$\hat{f}(z) = \frac{I(z)}{f(z)}. \quad (4.2)$$

For the entire ping, however, we may rewrite formula (4.1) as follows:

$$f(z) = ae^{bz}, \quad (4.3)$$

where now $\mathbf{z}$ is the ping of either the left or the right SSS sonar, a and b are the weights that can be estimated from all the samples in the ping via regression analysis. While this model might seem sound and might work well, our preliminary experiments showed that it did not fit well to the observed intensity values. As an alternative, we find out that using MIXed exponential Regression Analysis (MIRA) has superior performances, which can be written as follows:

$$f(\mathbf{z}) = ae^{b\mathbf{z}} + ce^{d\mathbf{z}}, \quad (4.4)$$

where a, b, c, d are now the four weights representing the echo decay for each ping signal, and $\mathbf{z}$ is the spatial location (or index) of each sample within the ping. We normalize each ping using the formula (4.5):

$$\hat{f}(\mathbf{z}_i) = \frac{I(\mathbf{z}_i)}{f(\mathbf{z}_k)}, \quad (4.5)$$

where k is the index of one ping from the image that will be used to find the parameters of MIRA, which could be arbitrary selected or by using a more dedicated selection criteria, $i = 1, \dots, n$, and n is the total number of pings in the image.

Estimating this model for each ping is not only time consuming, but may lead to overfitting. Furthermore, the exponential regression model should be estimated from each of the left and the right images using at least one ping, respectively. The selection of the ping that can be used in the exponential regression analysis is also a problem, since it will be used to normalize all the pings in the image. Nonetheless, our analysis showed that choosing any ping from the image will do, and this case can be used safely if there are not enormous depth changes among the different pings. In this work, we opted to use an arbitrary ping index, at $k = 500$. Additional work, however, might be performed in the future to probably enhance the method further.

(a) Two landmark pairs $\{(a_1, a_2), (b_1, b_2)\}$ in original images; (b) Two landmark pairs $\{(\alpha_1, \alpha_2), (\beta_1, \beta_2)\}$ in DCP; (c) Two landmark pairs $\{(a_1, a_2), (\beta_1, \beta_2)\}$ in MIRA.

Figure 4.4 Extraction of corresponding pairs from, the original image (left), image intensity normalized with the DCP method (middle), and image intensity normalized with MIRA method (right). We see how the inhomogeneous echo decay produces an irregular illumination field in regions containing natural seabed landmarks, left part. This artefact seems to be decreased after applying the proposed methods, although the DCP method introduces a considerable amount of artificial contrast in the extracted regions. The shown above are corresponding pairs of distinguished landmarks

Figure 4.5 Four of the 32 pairs contained in the benchmark set. Distinctive landmarks from the original SSS images, same landmarks after illumination correction by the DCP method and by the MIRA method are shown. It is obvious that some SSS images are difficult to recognize even for the human observer. For illumination purpose, the brightness and contrast of the images shown in this figure have been set to 35%, and the picture boarder of each image was set to black

For illumination purpose, one ping, at $k = 500$, regression analysis and signal normalization are depicted in Figure 4.6, where the left MIRA model coefficients (with 95% confidence bounds) are, $a = 229.4$ (210.5, 248.3), $b = -0.005883$ (-0.006692, -0.005074), $c = 81.63$

(57.32, 105.9), $d = -0.001001$ (-0.001335, -0.0006677). Right MIRA model coefficients (with 95% confidence bounds) are, $a = 275.2$ (270.6, 279.9), $b = -0.00262$ (-0.002793, -0.002448), $c = 0.7158$ (-1.999, 3.431), $d = 0.002802$ (-0.0009682, 0.006572).

4.1.3. Dark Channel Prior (DCP) method

The Dark Channel method [94] was initially designed to perform image dehazing. The task of image dehazing consists of removing fog from outdoor degraded images [95,96]. The Dark Channel method attempts to solve the following physical model of atmospheric degradation:

$$I(x) = J(x)t(x) + A(1 - t(x)), \quad (4.6)$$

where $I(x)$ is the observed intensity, $J(x)$ is the scene radiance, which corresponds to the non-degraded image, A is a constant additive term usually called airlight, and $t(x)$ is a scalar quantity called transmission, that describes the amount of light that is not scattered nor absorbed, and reaches the observer. According to the Beer-Lambert law, transmission is exponentially decreasing with respect to the distance, and can be modelled as $t(x) = \exp^{-\beta(x)}$, where $d(x)$ denotes the distance and β is the attenuation coefficient of the medium the signal traverses.

To invert the above model and retrieve $J(x)$, one needs to impose some constraints on the problem. The Dark Channel method imposes that no large bright areas without contrast should be present in the retrieved radiance, see [94] for more technical details. In this sense, although in principle image dehazing and echo decay compensation seem to be unrelated problems, both of them share a common feature: their goal is to produce contrast and shrink intensities in too bright areas of the image, while keeping intensity unaltered in dark areas. As shadows contain useful information in sonar scans, and the Dark Channel method will leave those areas unmodified, it seems to be a reasonable solution to test in order to solve the echo decay mitigation problem.

4.1.4. Local similarity analysis

In order to assess the performances of the two above proposed methods, MIRA and DCP approaches, we analyze the similarity of the landmarks extracted from different scans of the same area after correction of the echo decay.

At the first stage, the well-known Structural Similarity Index Metric (SSIM) is employed to evaluate the similarity between a landmark and its 180°-rotated counterpart, see Figure 4.4. The SSIM measures the similarity of two images in terms of three independent variables: luminance,

contrast and structure. We define the mean and variance of a signal x (in this case containing several pixels) of size N as:

$$\mu_x = \frac{1}{N} \sum_{i=1}^N x_i, \quad (4.7)$$

$$\sigma_x = \frac{1}{N-1} \left(\sum_{i=1}^N (x_i - \mu_x)^2 \right)^{\frac{1}{2}}, \quad (4.8)$$

and the covariance between two signals x and y as:

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^N (x_i - \mu_x)(y_i - \mu_y). \quad (4.9)$$

Finally, we define a comparison function as follows:

$$L(a, b) = \frac{2ab}{a^2 + b^2}. \quad (4.10)$$

Then, we have that the SSIM index is the product of a luminance comparison, a contrast comparison, and a structure comparison:

$$SSIM(x, y) = L(\mu_x, \mu_y) L(\sigma_x, \sigma_y) s(x, y), \quad (4.11)$$

where $s(x, y) = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$.

(a)

(b)

Figure 4.6 Using image WE3. (a) Ping at index 500 used to estimate the MIRA model, where two models are estimated one for the left scan/ping and another for the right scan/ping; (b) Ping 500 after normalizing it with MIRA; (c) Another ping from the same image at index 4000; (d) After normalizing ping 4000 with the MIRA model estimated from ping 500; (e) Ping 500 normalized with the DCP method; (f) Ping 4000 normalized with the DCP

The SSIM value of the 19 different landmark pairs present in the derived sonar dataset has been computed, as depicted in Figure 4.7. The mean and standard deviation of the SSIM score obtained by both MIRA and DCP normalization methods, and also without applying any normalization, are shown in the following Table 4.1.

Figure 4.7 SSIM scores measuring the similarity of the two methods proposed in this chapter and compared to the non-normalized images. Images normalized with the MIRA method which is depicted as the dark thick line, outperforms the both the DCP method and the non-normalized images

Table 4.1 Mean and Standard Deviation of SSIM scores on the entire set of 19 landmark pairs in this study

Non-normalized	DCP	MIRA
0.258 ± 0.057	0.217 ± 0.066	0.369 ± 0.071

It is obvious that the MIRA method performs the best at correlating different landmarks coming from different SSSs, while the DCP method reduces the mean of the SSIM score for most of the available landmark pairs and it even lower than the original non-normalized images.

4.1.5. Global quality analysis

Apart from the above local analysis, considering the entire sonar image instead of sub-regions as in Figure 4.4, a global analysis should also be performed to see the effects of applying the proposed MIRA and DCP approaches. While the above local analyses are useful to assess the feasibility of a subsequent landmark matching stage after correcting the initial seabed scans, a global analysis may be useful for several tasks, such as providing a mechanism to automatically tune the parameters of an echo decay correction algorithm, or evaluating its correctness without the need to manually extract natural landmarks.

To perform such a global analysis, we propose to study the difference of echo values between adjacent regions of the sonar scan. To that end, we divide the sonar image into a collection of $M \times N$ patches covering the whole scanned seabed. Then, for each patch (m, n) we compute its mean intensity, which is denoted by $\mu_{m,n}$. Afterwards, to estimate the quality of an image patch, we compare its mean intensity with the adjacent ones following Weber's perceptual law:

$$C_{m,n} = \frac{\max(\mu_{m,n} - \mu_{m_k, n_k})}{\mu_{m,n}}. \quad (4.12)$$

Eventually, we define the following Sonar Image Quality Assessment Metric (SIQEM):

$$SIQEM(S) = \frac{1}{M} \frac{1}{N} \sum_{m=1}^M \sum_{n=1}^N C_{m,n}. \quad (4.13)$$

The lower SIQEM value an image attains, the better quality it has. A similar image quality metric has been recently proposed in [97] for uneven illumination assessment purposes in dermoscopic images of the skin, but to the best of our knowledge, no similar quality metric has been proposed before to measure echo decay in SSS scans of the seabed.

Regarding the global quality evaluation, the values of the SIQEM metric for the five sonar scans available in our dataset, after processing with both MIRA and DCP enhancement methods and without any processing, are shown in Table 4.2 below.

Table 4.2 Results of the SIQEM metric on each of the five SSSs in the used dataset, the lower SIQEM value the better image quality

Image set from	Non- normalized	DCP	MIRA
-------------------	--------------------	-----	------

WE3	0.115	0.129	0.076
EW2	0.120	0.143	0.087
WE2	0.117	0.142	0.080
EW1	0.109	0.125	0.076
WE1	0.137	0.157	0.107
Mean $\pm$STD	0.12$\pm$ 0.01	0.14$\pm$ 0.01	0.08 $\pm$ 0.01

It can be found in every case that the MIRA method outperforms the DCP method, the latter seems to perform worse than the non-normalized SSS images. These results are in line with the above local analysis and validate the value of the SIQEM metric for globally evaluating the behavior of illumination normalization that has been verified subjectively.

Such preprocessing approaches for handling the echo decay in images acquired by SSSs will lead to improved representation and classification of underwater images and thus will facilitate maps fusing, matching, registration, among other objectives. The manual extraction of the pairs for the overlapping regions in two sonar images acquired via opposite direction missions has been a difficult task, due to the severe differences in the acoustic shadows between the overlapping regions in the sonar images. Therefore, the cases where the similarity scores of few sonar images normalized with the MIRA method have been comparable to the scores of the non-normalized images could be related to the imperfect manual pair extraction. The possible translation, slight-rotation, and scale difference, or even an affine transform between every pair that has been provoked during the movement of the vehicle carrying the two SSSs could be another reason of this occasional low performance. Hence, deploying an affine invariant metric to measure the similarity between every pair could greatly enhance performances. Although the current MIRA model only estimates its parameters from one ping, local and global similarity analysis, indicate its effectiveness in solving the echo decay problem in sonar images.

4.2. Underwater feature matching approaches

Currently, seabed mapping techniques put great emphases on the integration of data recorded from different perception and navigation sensors, generally acoustic sonars are used to create large-scale maps of the environments while optical cameras provide more detailed images for the interested objects. Conventional Global Positioning System (GPS) and Differential GPS (DGPS) receivers are not available in subsea environments. Therefore, when submerged, the SSSs must be localized using dead-reckoning with depth sensors, Doppler Velocity Logs (DVL), Inertial Navigation Systems (INS), or compasses. Since the costs of offshore seabed mapping are largely determined by the expense of the vehicle exploring time, any improvement in the quality of the sonar perceived data, and especially reduction in detection costs are of crucial interest to the marine and offshore community.

The vast majority of the state-of-the-art image matching methods consider two views of the same scene where the viewpoints differ by small offsets in position, orientation and viewing parameters such as focal length. Therefore, under such conditions, the images acquired by two viewpoints have comparative resolutions and hence they encapsulate scene features at approximately the same scale. The search for discrete image correspondences can be divided into three main phases. First and foremost, select interest feature points at distinctive positions in every image, such as corners, blobs, and T-junctions. The interest point detector should reliably detect the same interest points under various viewing conditions. Secondly, the neighborhood of each interest feature point is represented by a feature vector. This descriptor has to be distinctive, at the same time, robust to noise, detection errors, and geometric and photometric deformations. Moreover, match the descriptor vectors between different images.

The conventional image matching approaches extract interesting point-features from each image, match them based on cross-correlation, compute the epipolar geometry through the robust estimation of the fundamental matrix, and establish many other matches once this matrix is known. However, on the following conditions, these classic methods cannot be applied anymore [98]:

- (1) Point-feature extraction and matching are resolution dependent processes;
- (2) The high-resolution image corresponds to a small region of the low-resolution one and thereby the latter contains many other features which do not have any match in the former;
- (3) It may be difficult to estimate the epipolar geometry since there is not enough depth associated with the high-resolution image and its associated small area of the low-resolution image.

The special application of subsea sonar image matching requires high matching precision and satisfying real-time performances of the chosen algorithm. Nevertheless, the different imaging time and condition lead to the variance of image resolution, view angle and gray value feature between the reference image and real-time image, which makes scene matching a tough task [99]. For the sonar maps, where feature points are hard to be detected by the traditional feature detectors, such as Harris corner detector [100], the performances of final map fusion also deteriorate. Therefore, an underwater feature matching method, which is based on the most efficient scale-and rotation-invariant detector and descriptor, Speeded Up Robust Features (SURF) for sonar map fusion is presented, SURF can match two sonar images with different resolutions and geometries of sonar acquisitions, because of distinct viewpoints. Moreover, for improving the accuracy and efficiency of map fusion, especially for our considered case

underwater environments where feature points are few and difficult to be detected, here we propose to employ the Underwater Wireless Sensor Network-based Delaunay Triangulation (UWSN-DT) algorithm for sonar map fusion is proposed.

4.2.1. SURF, SIFT and ASIFT keypoint based underwater SSS image matching

Local features are suitable for feature matching and object recognition as well as for many other image processing applications, since they are robust to occlusion, background clutter and other content changes. Obtaining invariance to different viewing conditions is the key to image matching. The most widely used detector probably is the Harris corner detector, which is based on the eigenvalues of the second moment matrix. But, they are not invariant to scale changes. Recently, many local invariant descriptors have been proposed and made remarkable improvements, of which Scale Invariant Feature Transform (SIFT) [101] descriptor is distinctive, relatively fast, which is crucial for online applications. Besides, it shows excellent performances for affine transformations, scale and illumination differences, rotation, blur, etc [102]. The SIFT descriptor still seems to be the most appealing one for the practical uses, thus the most widely used nowadays, including image registration, object recognition, recognizing panoramas, etc. Nevertheless, the heavy computational burden makes SIFT cannot meet the real-time requirements for underwater feature matching. SIFT can detect and describe a feature, since it is invariant to scale, rotation and translation. The detector used in SIFT feature is the Difference of Gaussians (DoG) detector. Affine-SIFT (ASIFT) [103] is a fully affine-invariant feature matching algorithm. ASIFT keypoint detection uses a predefined number of image tilts and rotation and applies a SIFT algorithm for each image. ASIFT is significantly accurate for detecting interest points by considering Harris-Affine or Hessian-Affine. Although good performances of SIFT or even ASIFT feature descriptors compared to others is remarkable. Nevertheless, by integrating the roughly localized information and the distribution of gradient related features seems to yield good distinctive power while lowering the effects of localization errors in terms of scales and spaces. Adopting relative strengths and orientations of gradients reduces the influences of photometric variances.

The Speeded Up Robust Features (SURF) and the Binary Robust Invariant Scalable Keypoints (BRISK) [104] feature-based techniques represent an evolution of the image processing method based on SIFT and FAST feature, respectively. The SURF feature descriptor relies on local gradient computations while the BRISK feature uses a binary descriptor depending on pairs of local intensity differences, which is then encoded into a binary vector. The SURF detector is a Fast Hessian detector, and it identifies the centers of blob-like structures

by using the DoG, which is an approximation of the Laplacian. The BRISK detector is a corner detector. The feature detection technique based on SURF features is more capable to determine more useful landmarks for matching two acoustic images than the method based on BRISK features. The SURF descriptor is based on similar properties, with a complexity stripped down even further. First of all, we fix a reproducible orientation based on information from a circular region around the interest points. Next, construct a square area aligned to the chosen orientation, and we extract the SURF descriptor from it. Finally, match the SURF descriptors between different images.

It is important to find the key feature points for detecting an object in a sonar image, and in our case we detect SURF features. Our interests are the focuses that are the extremes among its 8 neighbors in the present level and its 2×9 neighbors in the above and beneath level. This is a non-most extreme concealment in a $3 \times 3 \times 3$ neighborhood. For this, a 2×2 symmetric Hessian matrix which contains the second order derivatives is considered. Given a point $X = (x, y)$ in an image I , its Hessian matrix at scale σ is defined as follows:

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{yx}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} = \begin{bmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial y \partial x} & \frac{\partial^2 I}{\partial y^2} \end{bmatrix} \quad (4.14)$$

where $L_{xx}(X, \sigma)$ represents the convolution of the Gaussian second order derivative with the image $I(x, y)$ at point X , and similarly for $L_{yx}(X, \sigma)$ and $L_{yy}(X, \sigma)$. Box filter is chosen to approximate the second order Gaussian partial derivatives. The integral image is used in the Hessian square matrix to obtain the sum of the intensity values, thus the computation of intensity values in the area of interests will be decreased dramatically when an integral image is built. The value at any point (x, y) of the integral image is given as the sum of all the intensity values of the points in the image which are lower than or equal to (x, y) . The integral image has the same size as that of the given image. The convolution of each box filter and the image is denoted by D_{xx} , D_{yy} and D_{xy} , respectively, the final matrix that we get with the determinant values is called the blob response map. For location (x, y, σ) , the blob response $\det(H_{approx})$ is calculated as the following formula (4.15). In general, for a 5×5 matrix with $\sigma = 1.5$: $\sigma = 1.5 * (filter\ size/5)$.

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2. \quad (4.15)$$

After the SURF feature points are detected and described in sonar images, we will implant the coarse-to-fine matching based on the bidirectional nearest neighbour method between these SURF feature points. If more than three key point pairs are detected and matched correctly

within these two sonar images, we regard these two sonar images as matched. The key point match criterion for the bidirectional nearest neighbour approach is defined as follows. Assume that P and Q are two SURF key point sets derived from two sonar images; if the point pair (p_i, q_j) satisfies the following A, B, C three requirements, clarified in the formulas (4.16) to (4.18), then the point pair (p_i, q_j) will be regarded as matched. Here, the threshold $\alpha \leq 1$, and the smaller α , the fewer matched pairs, the higher matching accuracy. Since the outliers are eliminated in the coarse matching procedure, thus greatly decreases the influences of outliers on the successive coarse-to-fine matching and enhances the final matching precision.

$$A: d(p_i, q_j) = \min_{q_l \in Q} d(p_i, q_l) = \min_{p_k \in P} d(p_k, q_j); \quad (4.16)$$

$$B: d(p_i, q_j) \leq \min_{q_l \in Q, l \neq j} d(p_i, q_l) * \alpha; \quad (4.17)$$

$$C: d(p_i, q_j) \leq \min_{p_k \in P, k \neq i} d(p_k, q_j) * \alpha. \quad (4.18)$$

4.2.2. Simulation results of SURF keypoint matching

The tested SSS image in this work is recorded by DeepVision AB company with DeepEye340D SSS at Stockholm marine [81]. The DE340D SSS is a chirp sonar working at the frequency of 340 kHz with optimized resolution of 1.5 cm. The sonar beams reaches from 15 to 200 m and the maximum operation depth is 100 m. Figure 4.8 illustrates the AUV round trip scans for exploring multiple areas. Two sonars are employed, one is a left SSS, and the other is a right SSS. The AUV supervises its surroundings as it navigates in the underwater environment, mostly in WE and EW these two directions, with an overlapping region between each scans.

Figure 4.8 AUV round trip scans with overlapping

We choose one SSS image, in which contains a shipwreck. By blurring this SSS image with a Gaussian low pass filter of size $[11 \ 11]$ with standard deviation 1.5, we derive this large-space low-resolution SSS image, shown on the left in Figure 4.9. We regard the scale of it equals to 1, and various scales between 0.6 and 1.4 of the partial high-resolution SSS image have been set, where also contains the feature, a shipwreck, shown on the right in Figure 4.9. Afterwards, we test whether these two image pairs are matched by detecting their SURF feature pairs. Simulation results are shown from Figure 4.9 to Figure 4.13, where these two tested SSS images have a rotation difference of 180° , 135° , 90° , 45° and 20° , separately; the scale of the partial high-resolution SSS image is 1.2.

Figure 4.9 180° rotation difference, 51 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 242 SURF keypoints detected

Figure 4.10 135° rotation difference, 27 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 255 SURF keypoints detected

Figure 4.11 90° rotation difference, 50 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 239 SURF keypoints detected

Figure 4.12 45° rotation difference, 24 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 267 SURF keypoints detected

Figure 4.13 20° rotation difference, 24 SURF matching pairs. The left SSS image scale 1, 156 SURF keypoints detected; the right SSS image scale 1.2, 259 SURF keypoints detected

The following Table 4.3 clarifies and compares the performances of the SURF keypoint matching within the above mentioned large-scale low-resolution and partial high-resolution SSS image pairs under the condition of various scales and rotation angles.

Table 4.3 The computation time and the number of matched SURF feature pairs with different scales and rotation angles

Rotation Angle Variance Matching Pairs; Time [s] Scale2	20°	45°	90°	135°	180°
0.6	26;0.849	23;0.856	40;0.843	29;0.838	40;0.828
0.8	36;0.876	24;0.849	44;0.842	20;0.857	50;0.845
0.9	33;0.866	25;0.867	45;0.855	28;0.870	47;0.847
1.0	35;0.850	28;0.853	53;0.842	28;0.856	53;0.839
1.1	36;0.879	35;0.888	46;0.859	31;0.859	51;0.862
1.2	31;0.904	24;0.893	50;0.899	27;0.863	51;0.860
1.4	26;0.930	22;0.970	50;0.904	25;0.931	50;0.891

Obviously, the two SSS image pairs match perfectly when they have the same scale. However, usually this is not a common case. By comparing the SURF keypoint matching performances between the above scale differences, we found that these tested SSS images match perfectly with more SURF keypoint matching pairs and low computational complexity when the scale of the partial high-resolution SSS image is 1.2.

4.2.3. Simulation comparison of SURF, SIFT and ASIFT keypoint matching performances

The SIFT and ASIFT keypoint matching results are shown in Figure 4.14 and Figure 4.15, respectively. In Figure 4.14, it costs 4.720715 s to detect 74 SIFT keypoint matching pairs between these two tested SSS images. In Figure 4.15, it identifies 332 ASIFT keypoint matches, but it needs 7.859278 s, this means that the ASIFT consumes nearly 1.7 times more processor time than the SIFT.

Figure 4.14 180° rotation difference, 74 SIFT matching pairs, 4.720715 s. The left SSS image scale 1, 502 SIFT keypoints detected; the right SSS image scale 1.2, 2366 SIFT keypoints detected

Figure 4.15 180° rotation difference, 331 ASIFT matching pairs, 4.720715 s. The left SSS image scale 1, 502 SIFT keypoints detected; the right SSS image scale 1.2, 2366 SIFT keypoints detected

To a great extent, the expense of the seabed mapping is determined by the navigation time of the AUV. Therefore any enhancement in the quality of the derived sonar images, in particular lowering the exploration costs, are of crucial importance to the maritime and offshore research community. The number of detected SURF, SIFT, ASIFT keypoint matching pairs and their computational requirements are compared in Table 4.4.

Table 4.4 The number of detected SURF, SIFT, ASIFT keypoint matching pairs and their computational complexity

Scale 2=1.2	Matching Pairs	Computational Time (s)
SURF	51	0.860375
SIFT	74	4.720715
ASIFT	332	7.859278

Overall, the state of the art underwater feature matching approaches are summarized and compared, including the classic Harris, SURF, BRISK, SIFT and ASIFT feature detector, regarding their suitability for subsea sonar map fusion, which are computational requirements, repeatability, efficiency, robustness, etc. For this considered case, both two SSS images have relatively high resolutions and consist of the same distinctive and clear feature, a shipwreck, we compare and analyze the performances of SURF, SIFT, and ASIFT feature detector; we conclude that although ASIFT approach results in more correct associations, in this scenario we could detect their SURF feature matches efficiently with comparatively low computational burden, which is of crucial concern to the maritime and offshore research community.

However, one serious problem is that two tested SSS images might or totally differ when the sonar scanning is performed at different directions, such as the shadow effects. Due to various sonar image resolutions, view angle and illumination changes in the perceived sonar data, underwater features are blurry and hardly to be detected, for this case, the conventional keypoint matching approaches as mentioned above both fail to detect keypoint pairs.

4.3. Wireless Sensor Network (WSN) based underwater sonar image fusion

For improving the accuracy and efficiency of map fusion, especially for our considered case underwater environments where feature points are few and difficult to be detected, we deploy the wireless sensor network to assist the sonar map fusion; the main idea of the WSN assisting seabed mapping is that we regard the nodes of the WSN as feature points. The nodes in the WSN can provide sufficient information for the seabed mapping, such as the location. Moreover, we combine the underwater wireless sensor network (UWSN) and Delaunay Triangulation technology, and propose the underwater wireless sensor network-based Delaunay Triangulation (UWSN-DT) algorithm to enhance the performances of sonar image fusion. To the best of our knowledge, this is the first algorithm which adopts the underwater wireless sensor network into the application of sonar map fusion.

4.3.1. Underwater Wireless Sensor Network (UWSN)

With the rapid developments of the wireless sensor network, more and more applications adopt the WSN to improve their performance, especially in underwater environments, which is not suitable for people to work for a long time due to the high pressure. In subsea environments, since it is not suitable for people to work for a long time under high pressure, so the WSN is applied to assist the ocean exploring. Many wireless sensor network based underwater applications have been proposed. For instance, in [105], the authors propose a paradigm for tracking objects (like trespassing ships approaching a harbor) and monitoring the activities on the ocean surfaces; in [106], the mobile ad hoc network is used for the underwater plumes detection, inspection, and traction; in [107], the underwater WSN is used in the offshore oil field comprising; in [108], the underwater WSN is used for the submarine detection; in [109], the authors use the underwater WSN and underwater robotics to track the underwater targets; in [110], the authors introduce the WSN into the underwater localization scheme with the assists of the underwater vehicles. Not only in the underwater environment but also in the remote sensing, the WSN is applied widely. In [111], the authors introduce the WSN into the soil moisture measurements over heterogeneous cropland; in [112], the water-level and temperature information of the agriculture is sensed by the WSN for the remote monitoring of the agriculture plantation; in [113], the distributed WSN is used in the remote sensing and control of the irrigation system. However, the wireless sensor network is still not used in the seabed mapping mission.

In the underwater region, where there are no obvious feature points, the WSN is deployed. There are N nodes deployed in the interested area randomly. For instance, the area A is defined as $A \triangleq L \times W$, where L is the length of the concerned area and W is the width of the concerned area. Since the nodes are randomly deployed in the area A , the density of the network can be defined as $\rho \triangleq \frac{N}{A} = \frac{N}{L \times W}$. The nodes are deployed in the seabed randomly. The underwater nodes are numbered and know their own locations through the Ultra Short Base Line (USBL) technology or the underwater GPS. Each node can be denoted as $n(i, (x_i, y_i))$, where i represents the node number, and (x_i, y_i) is the coordinate of node i ; moreover, $i \in \{1, N\}$, in which N represents the total number of nodes deployed in the seabed. The modes of the nodes are idle mode and activate mode. In the idle time, the nodes sleep for saving the energy and there are no AUVs communicate with the nodes. The nodes are activated when the AUVs moves into its communication range. The AUVs, which are equipped with sonar, move along the pre-defined path in the concerned area. They can scan the seabed and communicate with the nodes that are located in their communication area. The communication area and the scan area

of AUVs are different, which can be found in Figure 4.16. The circular area is the communication area of AUV, while the rectangular area is the scan area.

When the AUVs are moving, the nodes which have bi-directional communication links with AUVs will be activated. The bi-directional communication link means that the communication range of nodes and AUVs are all larger than that of the distance between the nodes and the AUVs. For instance, in Figure 4.16, the node s can communicate with AUV, since $R \geq |As|$ and $r_s \geq |As|$, where R is the communication range of AUV, r_s is the communication range of node s , $|As|$ is the distance between AUV and node s . In this scenario, the AUV sends HELLO packet periodically; only the nodes which can receive the HELLO packet will be activated. The nodes which cannot receive the HELLO packet will turn to idle model for saving energy. The activated nodes will send their node number and coordinates to the AUV. When the AUVs receive this information, they will embed the node locations and node numbers into the sonar map; moreover, this information will be used as the feature points of our proposed Underwater Wireless Network-based Delaunay Triangulation (UWSN-DT) algorithm for sonar map fusion. Since the sonar knows the boundary coordinates of its scanning area, and the coordinates of the nodes are known, thus the positions and the node numbers in the seabed can be shown in the sonar map. The network model of this scenario can be found in Figure 4.16. Under this assumption, each sonar channel (port and starboard) insonifies a rectangular area on the sea floor. The length of the rectangle is determined by the slant range of the sonar (the maximum range of the sonar) and the height of the underwater vehicle, assuming a flat seabed.

(a)

Figure 4.16 (a) The diagram illustrating the assumed geometry of the AUV embedded SSS in seabed navigation; (b) The communication range and scan range of AUV; (c) The geometry relationship between the communication range and scan range of AUV

In this scenario, for guaranteeing that the nodes in the scan area can be covered by the communication range of AUV successfully, the minimum communication range should be larger than the maximum scan range of AUV. The minimum communication range can be calculated according to Figure 4.16. The width and the length of the scan area are W and L , respectively. The distance of AUV to the seabed is H , which is shown in Figure 4.16 (c). So the minimum communication range can be calculated as:

$$R = \sqrt{\frac{\sqrt{L^2 + W^2}}{2} + H^2}. \quad (4.19)$$

As we know, the larger communication range, the higher energy consumption; so in this thesis, we choose R as the communication range of AUV. The AUV moves with the predefined path, the nodes locating in the communication range of AUV are activated and send their node number and the coordinates to the AUV. The AUV only stores the information of nodes which are within its scan area.

For the nodes which have been activated, the communication ranges should be assigned carefully for saving energy. When the nodes receive the HELLO message, in which the location of the AUV is included, they will calculate the distances to the AUV. The communication range of the underwater node should be larger than this distance. So for each node in the coverage area of the AUV, the minimum communication range of node s should be $|As|$, and the maximum communication range of node s should be R , since the communication range of AUV is R , i.e., $r_s \in [|As|, R]$. Moreover, the AUV should also decide which nodes located in its coverage area

are also in its scan area. As mentioned above, the width of the scan area is W , and the length of the scan area is L . Moreover, the coordinate of the AUV is (x_A, y_A) . So the boundary of the scan area can be given as:

$$x_c = \{x_A - L/2, x_A + L/2\}, \quad (4.20)$$

$$y_c = \{y_A - W/2, y_A + W/2\}. \quad (4.21)$$

The nodes whose coordinates satisfy the constraints as follows will be regarded as located in the scan area: if the node i is located in the scan area, then $x_i \in x_c$ and $y_i \in y_c$. When the AUV gets all needed information from the nodes, then the sequence number and the nodes position are reappeared in the scan map as the feature points. These feature points can be used in the presented Underwater Wireless Sensor Network based Delaunay Triangulation (UWSN-DT) algorithm for sonar map fusion. The procedure of the node discovery algorithm can be expressed as follows.

Algorithm 3: Node discovery algorithm

Initial:

1. $node_ni_status = \text{"Idle"}$,
2. $R_{AUV} = \sqrt{\frac{L^2 + W^2}{2}} + H^2$,
3. $R_{node} = 0$,
4. $x_{AUV_upper} = x_A + L/2$ and $x_{AUV_lower} = x_A - L/2$,
5. $y_{AUV_upper} = y_A + W/2$ and $y_{AUV_lower} = y_A - W/2$,
6. $node_ni_receive_HELLO = \text{False}$,

Main:

7. AUV send HELLO message periodically, the HELLO message include the coordinates of AUV,
 8. if node ni receives the HELLO message
 9. $node_ni_receive_HELLO = \text{True}$,
 10. end if
 11. if $node_ni_receive_HELLO = \text{True}$
 12. $node_ni_status = \text{"Active"}$,
 13. end if
 14. while $node_ni_status = \text{"Active"}$ do
 15. if $x_A - L/2 < x_{ni} < x_A + L/2$ && $y_A - W/2 < y_{ni} < y_A + W/2$
 16. node ni send its coordinates and $node_ni_number$ to AUV,
 17. else
 18. $node_ni_status = \text{"Idle"}$,
 19. end if
 20. end while
-

4.3.2. Simulation results of the communication between underwater nodes and AUV

In this part, we will evaluate the communication performances between the AUV and underwater nodes. In this simulation, the area size is $3000\text{ m} \times 3000\text{ m}$; the communication range of the AUV is set to 447 m ; for both left and right sonar scans, the range are both of $400\text{ m} \times 400\text{ m}$; the total number of nodes in the seabed is 1000. We set two coordinates for the AUV to show the underwater nodes in the scan range of the AUV, the coordinates of the AUV are: $(500\text{ m}, 500\text{ m})$ and $(600\text{ m}, 600\text{ m})$, which can be found in Table 4.5.

Table 4.5 The detected wireless nodes and their coordinates in two SSS scans

AUV Coordinates [m]	(500, 500)	(600, 600)
number	6	9
Node 4	NaN	(990.8, 473.1)
Node 11	(349.5, 331.2)	NaN
Node 14	(600, 607.3)	(600, 607.3)
Node 15	NaN	(703, 765.7)
Node 23	(875.8, 606.1)	(875.8, 606.1)
Node 24	NaN	(750.6, 724.2)
Node 39	(561.6, 468.7)	(561.6, 468.7)
Node 41	NaN	(518.3, 764.7)
Node 42	(471, 663.6)	(471, 663.6)
Node 45	NaN	(920.1, 431.1)
Node 84	(259.7, 347.7)	NaN

In Table 4.5, we can find that when the AUV starts navigating at $(500\text{ m}, 500\text{ m})$, then within the sonar scan range, the following underwater wireless nodes are detected: node 11, node 14, node 23, node 39, node 42, and node 84; when the AUV locates at $(600\text{ m}, 600\text{ m})$, node 4, node 14, node 15, node 23, node 24, node 39, node 41, node 42, and node 45 are in the scan range. As we can find that node 14, node 23, node 39, and node 42 locate within these two scan areas of the AUV. This simulation indicates that when the AUV navigates at $(500\text{ m}, 500\text{ m})$ and $(600\text{ m}, 600\text{ m})$ with an area of $(400\text{ m}, 400\text{ m})$ in both left and right sonar scans, these two SSS images are overlapped. This is illustrated by Figure 4.17 as follows.

Figure 4.17 Simulation results of the communication between the nodes underwater nodes and AUV

4.3.3. Building the Delaunay triangulated mesh on the detected nodes

Given just two sonar images, their interest points, descriptors and no prior information on how these two images may be related. Also, there is no geometric constraint that we can impose on a search for the position transform of $x_i \Leftrightarrow x'_i$. These two sonar images may match at any arbitrary pose, or they are not matched at all. Nevertheless, we still impose epipolar geometry constraint on the detected keypoints. Therefore, we seek to find matched candidates and to use a robust estimation such as Random Sample Consensus (RANSAC) [114] to estimate the true matches. This method works well when the outlier fraction is small. Even though we have an efficient interest point detector and descriptor, we still foresee that there may be many false matches on the following conditions:

- (1) Two sonar images match partially, either from different scales or from distinctive view angle;
- (2) Two sonar images do not match at all;
- (3) There are many other similar objects.

Although the feature, a shipwreck in our derived SSS image is visibly clear and distinctive, it happens that the SSS does not detect this shipwreck when the AUV scans the seabed continuously. Note that there is 180 degrees difference between each SSS image pairs, since the sonar scanning is performed in a round trip mission. In this experiment, the size of the recorded SSS image is $800 \times 748 = 598400$ pixels, in which we randomly deploy 200 underwater nodes. Then, we construct a Delaunay triangulation [115] based on the detected nodes that are located within the derived partial SSS sonar image where almost contains no clear feature, such as in the horizontal range of [44 pixel, 293 pixel], and in the perpendicular area of [1 pixel,

250 pixel], shown in the Figure 4.18 (b). For a set P of discrete points in general position within the two-dimensional Euclidean space, there exists a unique Delaunay triangulation, which is a triangulation $DT(P)$ such that no point in P is inside the circum-hypersphere of any simplex in $DT(P)$. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation.

Figure 4.18 Building the Delaunay triangulation on the detected nodes within the overlapped region between two SSS images (a) and (b)

In Figure 4.18 (b), all the deployed 200 nodes are depicted as green crosses ('+'), and 23 nodes are detected within the above mentioned nearly no clear feature area, drawn as red stars ('*'). Simulation results show that these two SSS images are overlapped in some region, and they are matched with each other.

In order to avoid end up processing large sonar images, we need to extract some regions where contain distinctive landmarks. Therefore, we will perform image matching on our derived sonar image data sets, for instance depicted in above-mentioned Figure 4.4, which is recorded by Deep Vision AB company with DeepEye680D SSS at Lake Vättern in Sweden. The DE680D SSS is a chirp sonar and it works at the frequency of 680kHz with an optimized resolution of 1 cm. The beam reaches from 10 to 200 m and the maximum operation depth is 100 m. Two sonars are employed, one is a left SSS, and the other is a right SSS. The AUV supervises its surroundings as it navigates in the underwater environment, mostly in WE and EW these two directions, with an overlapping region between each scans. Figure 4.4 (a) is the original image, the image intensity normalized with the DCP method and our proposed MIRA method are illustrated in Figure 4.4 (b) and Figure 4.4 (c), respectively. By estimating the parameters of MIRA and DCP model from one ping, local similarity and global quality analysis

described in Section 4.1, indicates the effectiveness of MIRA method in solving the echo decay problem in sonar images compared to the DCP method, which is an adaption to well-known fog removal technique to sonar imaging. Therefore, we will perform the sonar image matching on the image intensity normalized by the MIRA method, shown in Figure 4.4 (c). There are two distinguished landmark pairs a_1 and a_2 , b_1 and b_2 . In this work, we consider the more clear features a_1 and a_2 . To verify the similarity of the overlapping areas between WE and EW SSS images, in our case, we extract sub-image pairs with a size of 400×400 pixels where contains landmarks a_1 and a_2 separately, shown in Figure 4.19.

Figure 4.19 Constructing the Delaunay triangulated mesh on the detected nodes in partial SSS image pairs where contains landmark pairs a_1 and a_2

Note that there is 180° difference between each SSS image pairs, since the sonar scanning is performed in a round trip mission. In the left SSS image, there are 6 SURF keypoints detected, and 6 for the right SSS image. However, due to the fact that there are not enough matching pairs to do epipolar filtering, these two SSS images do not have any SURF keypoint matching pairs. This phenomenon happens the same for both SIFT and ASIFT keypoint detector. We derived 81 SIFT keypoints in the left SSS image, and 79 for the right one; 1803 ASIFT keypoints are detected in the left SSS image, and 4227 for the right one. Although both of them have detected each keypoints, there are none SURF, SIFT or ASIFT keypoint matches between these two SSS images. Nevertheless, we cannot conclude that whether they match with each other or not. Therefore, we deploy 30 wireless nodes into this area, and then we build the Delaunay triangulation, depicted in Figure 4.19. The experimental results show that there exists overlapping region between these two SSS images, and they are matched with each other successfully. Therefore, our presented UWSN-DT algorithm works efficiently when the underwater feature points are few, fuzzy and difficult to be detected.

4.4. Summary

The task of normalizing the sonar images with regard to illumination is challenging due to inhomogeneous echo decay in acoustic shadows. In this chapter, first of all, an optimal solution which employs the MIXed exponential Regression Analysis (MIRA) method has been proposed for handling the echo decay in sonar images. The Structural Similarity Index Metric (SSIM) and the normalized cross-correlation have been used to evaluate the similarity between a landmark and its 180° - rotated counterpart. Our MIRA model has been compared with the Dark Channel Prior (DCP) model, which is an adaption to well-known fog removal technique to sonar imaging, in terms of one ping, local similarity and global quality analysis. The experimental results proved that the proposed MIRA method has superior normalization performances.

Besides, a detailed state of the art underwater feature matching methods have been summarized and compared, including the classic Harris, SURF, BRISK, SIFT and ASIFT feature detector. Consider their suitability for subsea sonar map fusion, in terms of computational requirements, reliability, accuracy, etc.

Moreover, for the high-resolution sonar images, where contain particular and distinguishable features, we can detect their SURF feature matches with comparatively low computational requirements. Nevertheless, due to various sonar image resolutions, view angle and illumination changes in the perceived sonar data, underwater features are blurry, few and difficult to be detected, for this case, the conventional keypoint matching approaches fail to detect keypoint pairs. Therefore, we have introduced the underwater wireless sensor network (UWSN) into the sonar map fusion task, and combined with the Delaunay Triangulation technology, the underwater wireless sensor network-based Delaunay Triangulation (UWSN-DT) algorithm has been proposed for improving the performances of sonar map fusion accuracy with low computational load, in which the wireless nodes are considered as underwater feature points. Since nodes could provide sufficiently useful information for the underwater map fusion, such as the coordinates. In the above-simulated experiments, it shows that our presented UWSN-DT algorithm works efficiently and robustly, especially for the subsea environments where there are few distinguishable feature points. To the best of our knowledge, this is the first algorithm which adopts the underwater wireless sensor network into the application of sonar map fusion.

5 The posterior-estimated AEKF-SLAM system

In this part, the mathematical model of the proposed “Robotic underwater AEKF-SLAM based navigation” algorithm is established. Here the same notation which was employed in [9,10,37, 116] are adopted.

5.1. Vehicle model

The setting for the SLAM problem is that of a robot with a known kinematic model, starting at an unknown position, and moving through the exploring space containing multiple features or landmarks. The terms landmark and feature will be used synonymously. The robot is equipped with sensors to measure the relative location between any detected landmark and the robot itself. The absolute landmark positions are not available. Without previous knowledge, a linear synchronous discrete-time model composed of the evolution of the evolution of the robot poses and the landmark observations is adopted. Although robot motion model and the landmark measurements are usually nonlinear and asynchronous in any real navigation application, the use of linear synchronous models does not affect the validity of the proofs in the SLAM problem other than to require the same linearization assumptions as those normally employed in the development of an EKF [65]. Indeed, the implementation of the SLAM algorithm uses a nonlinear robot models and a nonlinear asynchronous observation models. The state of the system of interest consists of the position and orientation of the robot together with the all landmark locations. The robot state at time k is denoted by $x_v(k)$, and the robot motion is modeled by a conventional linear discrete-time state transition equation:

$$x_v(k+1) = F_v(k)x_v(k) + u_v(k+1) + \omega_v(k+1), \quad (5.1)$$

where $F_v(k)$: State transition matrix; $u_v(k)$: Vector of control inputs; $\omega_v(k)$: Vector of temporally uncorrelated process noise errors; it complied with normal Gaussian distribution, with zero mean and its covariance matrix is denoted as $Q_v(k)$.

5.2. Feature model

The AEKF-SLAM algorithm is based on a landmark map. The class of AEKF-based methods for feature-based SLAM is named Stochastic Mapping (SM). Features are fixed, discrete and identifiable landmarks in the environment. Repeatable observation of features is a mandatory requirement for SLAM. This paper considers the least complicated features, which are stationary point landmarks. This simplification reduces challenges with feature identification and interpretation, increasing the focus on SLAM algorithm itself. More elaborate parametric feature models, such as lines, might also be used, but are not implemented in this thesis.

The position of the i th landmark is denoted as x_{m_i} . Since all landmarks are assumed to be stationary, without loss of generality, the number of landmarks in the environment is arbitrarily set to N . Since the point feature is assumed to be stationary, the state transition equation for it is:

$$x_{m_i}(k+1) = x_{m_i}(k) = x_{m_i} \quad i = 1, 2, \dots, N. \quad (5.2)$$

The matrix of all N landmarks is $X_m = [x_{m_1}^T, \dots, x_{m_N}^T]^T$, where T stands for the transpose and is used both inside and outside the brackets in order to conserve the dimension of space. The augmented state vector is composed of both, the states of the robot and all landmarks locations, and it is expressed as:

$$x_a(k) = [x_v^T(k), x_{m_1}^T, \dots, x_{m_N}^T]^T. \quad (5.3)$$

Consequently, the augmented state transition model for the complete system can be rewritten as:

$$\begin{bmatrix} x_v(k+1) \\ x_{m_1} \\ \vdots \\ x_{m_N} \end{bmatrix} = \begin{bmatrix} F_v(k) & 0 & \cdots & 0 \\ 0 & I_{x_{m_1}} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & I_{x_{m_N}} \end{bmatrix} \begin{bmatrix} x_v(k) \\ x_{m_1} \\ \vdots \\ x_{m_N} \end{bmatrix} + \begin{bmatrix} u_v(k+1) \\ 0_{x_{m_1}} \\ \vdots \\ 0_{x_{m_N}} \end{bmatrix} + \begin{bmatrix} \omega_v(k+1) \\ 0_{x_{m_1}} \\ \vdots \\ 0_{x_{m_N}} \end{bmatrix}, \quad (5.4)$$

or equivalently:

$$x_a(k+1) = F_a(k)x_a(k) + u_a(k+1) + \omega_a(k+1), \quad (5.5)$$

where $I_{x_{m_i}}$ is the $\dim(x_{m_i}) \times \dim(x_{m_i})$ identity matrix and $0_{x_{m_i}}$ is the $\dim(x_{m_i})$ null vector.

Actually, any landmark x_{m_i} is in stochastic motion may be easily adapted to this framework. However, doing so offers little insight into the problem and even the convergence properties may not be held [21].

5.3. Observation model

The robot is equipped with sensors to measure observations of the relative positions of landmarks with respect to the robot. Also, without prejudice, observations are assumed to be linear and synchronous. The observation model for the i th landmark can be denoted in the following form:

$$\begin{aligned}
z_i(k) &= H_i x_a(k) + v_i(k) \\
&= H_{x_{m_i}} x_m - H_v x_v(k) + v_i(k),
\end{aligned} \tag{5.6}$$

where $v_i(k)$ is a vector of temporally uncorrelated observation errors with zero mean and its covariance matrix is denoted as $R_i(k)$. The observation matrix H_i relates the outputs of the sensor $z_i(k)$ to the state vector $x_a(k)$ when detecting the i th landmark. Note that the observation model for the i th landmark has the following form:

$$H_i = [-H_v, 0, \dots, 0, H_{x_{m_i}}, 0, \dots, 0] = [-H_v, H_{m_i}] . \tag{5.7}$$

This structure reflects the fact that the observations between the robot and the landmarks are often in the form of a relative position, or relative range and bearing.

5.4. Simultaneous Localization and Mapping (SLAM)

Simultaneous Localization and Mapping (SLAM), also referred to as Concurrent Mapping and Localization (CML), combines the information provided by the robotic odometers with the Received Signal Strength Indication (RSSI) samples provided by the surrounding objects to build and refine the exploring environmental map, while concurrently localize the robot itself in this map [117–119]. If the robot's path were known with certainty, then mapping would be a straightforward problem. The landmark positions in the robot's environment could be estimated using independent filters. Nevertheless, in the SLAM problem, the robotic trajectory is unknown, thus the uncertainties of the robot poses can be arbitrarily large due to the accumulated odometry errors, which also leads to errors in the robot path correlating errors in the map. Therefore, the state of the robot and the map features must be estimated at the same time. In SLAM, a robot starts at an unknown location and without previous knowledge of the environment to incrementally build a consistent map using its onboard sensors while concurrently localizes itself relative to this same map. To overcome this problem, the robot has to perform observations in order to obtain referential features or landmarks, which are assumed time invariant to create the map and, at the same time, keep track of its relative position and re-observe the features to estimate its position in the map [118,119].

The SLAM problem can be categorized as feature-based, also called landmark-based and view-based, also called pose-based. In a feature-based SLAM, features are measured by the sensors. The state vector is formed by robot pose and landmark positions adding new vectors as new features are found. The structure of landmark-based SLAM is shown in Figure 5.1. In a

view-based SLAM, the whole view of each pose is kept and processed. Visual odometry can then be performed through the comparison between two complete views [43,120,121].

Figure 5.1 A robot measuring relative observations to environmental landmarks

5.4.1. The SLAM process

The general SLAM process is illustrated in Figure 5.2. Each time, after sensor measurements, the robotic local perception map needs to be integrated with the global map in order to update the robot pose and also refine the detected landmark coordinates as well. Selecting the next best view for completing the model is a research issue by itself. The difficulty of the robotic SLAM-based navigation problem is that, in order to obtain a qualified map, an accurate estimation of the vehicle's trajectory is required. Nevertheless, when the unlimited incremental odometry errors are lowered, sensor measurements need to be incorporated into a precise map. SLAM presents a series of tough issues, such as (1) An efficient mapping of large-scale environments; (2) Correct association of sensor measurements; (3) Robust estimation of landmarks map and robot trajectory information.

Figure 5.2 The flowchart of the SLAM process

The SLAM problem is depicted as Bayes network in Figure 5.3 to understand the dependencies in the SLAM problem. The figure shows the changes of robot poses from x_{k-1} to x_k by giving the control signal u_k , the observations z_{k-1} , the pose x_k and the landmarks m_M . In

this representation, the arrows show direct dependencies and it can be found that there is no direct relationship between robot poses and landmarks [122]. Shaded nodes represent data directly observable by the robot, SLAM is that the robot recovers the unobservable variables-landmarks.

Figure 5.3 The SLAM graphical model, redrawn after [123]

5.4.2. Loop closing

Tracked landmarks could provide a basis for reducing the uncertainty of the robot poses. In closed loops, if a robot can detect a position where it has been before and could correctly match landmarks, then the accumulated errors will be bounded and map precision will be increased [124-126]. When it revisits the previously explored environment, it can reflect the changes appeared to the map. The correlation between the robot pose error and map precision is shown graphically in Figure 5.4 (a). The robot is moving along the path specified by the dashed line, observing nearby eight distinguishable features, drawn as dots. The shaded ellipses imply the uncertainties of the robot about its own poses, drawn over time. As a result of the control error, the robot pose becomes more uncertain when the robot moves. The estimations of the landmark positions are shown as white ellipses. One can see that as the robot pose becomes more uncertain, the uncertainty in the estimated locations of newly observed features also increases. In Figure 5.4 (b), the robot completes the loop and revisits a previously observed landmark. Since the position of the first observed landmark is known with high accuracy, the uncertainty of the predicted robot pose will decrease significantly. This newly discovered information about the robot pose increases the certainty such that now past robot poses are also known. Therefore, in turn, also the position uncertainties of the previously perceived landmarks decrease. This resulting effect is that the information spreads to previously observed landmarks, such that gaining information on the robot pose is probably the most important characteristic of the posterior SLAM estimate [127]. In Figure 5.4 (b), it can be seen that the shaded ellipses

obtained before loop closure do not shrink after closure, because they depict a time series of robot pose uncertainties and are not revised afterwards.

Figure 5.4 (a) Before closing the loop; (b) After closing the loop

Being able to re-identify previously detected features are of crucial importance to the cycle detection problem. Notice that, a closed loop can improve map consistency and localization accuracy. Therefore, detecting and selecting proper features is a key issue to solve within the SLAM problem. As a consequence, we present the AEKF-SLAM based robotic navigation algorithm to identify when an observation is corresponding to a new landmark or to an already seen one.

There is no unique best solution to the SLAM problem, the chosen method depends on several factors, such as the desired map resolution, the update time, and the number of features in the map, etc. An idea for a solution to the SLAM problem is to work with both localization model and observation models, which means to execute them iteratively, being the output of one the input of the other [41]. The localization model is obtained by the re-observation of located landmarks, so the robot is localized in reference to those registered landmarks. Given a map, the robot localizes itself in it. For a moving robot the concept can be better described as a motion model in which the mobile robot pose (position and heading) can be estimated based on on-board navigation sensors or on prediction models. Then it is possible to estimate the landmark positions in the subsequent reading. The problem with localization is that sensor noise and model inaccuracies accumulate errors over time which will lead to uncertainties unless corrections are made. Conversely, the observation model builds a map from the perspective of the sensor assuming a known location. The sensors used for perceiving the environment are mounted on the vehicle, which also add their own measurement noises, therefore their position uncertainties will be added to the map.

5.5. Augmented Extended Kalman Filter (AEKF)

Practically an Extended Kalman Filter (EKF) rather than a simple linear Kalman Filter (KF) is employed to generate estimates of the robot pose and landmark positions. Although the EKF suffers from high computational complexity, it has the highest convergence among the current methodologies. It is important to note that if the SLAM problem is linear and Gaussian, then the KF is guaranteed to both, converge and be provably optimal. However, real-world SLAM problems are always nonlinear or even discontinuous as in the presented underwater scenario. It has been successfully and widely employed in large-scale environments, including land, air and even underwater [128]. For the underwater nonlinear discrete SLAM estimation problem, although the linearization errors caused by the standard EKF decrease the localization accuracy, EKF still tends to provide satisfactory results in general. As a consequence, EKF is often treated as the “gold standard” of comparison for online SLAM algorithms due to its conceptual simplicity, high convergence, relatively low computational cost and capacity to handle uncertainty.

EKF is a Bayes filter that represents posteriors with Gaussians that are unimodal distributions. These can be represented compactly by a small number of parameters. At present, research on EKF-SLAM is mainly focused on reducing the linearization errors so that long-term missions can be successfully achieved. We know that the linearization errors become larger when big covariances are involved. Approaches such as hierarchical SLAM or sub-mapping methods build local maps of limited size, which bound the covariances and thus the linearization errors. Then, by incorporating the local maps into a global map or a hierarchy of global maps, EKF-SLAM application in large environments is possible.

The standard solution to manage the nonlinear SLAM system is to first linearize the robotic kinematic model and the landmark observation model by an EKF for generating the system state estimations. It is supposed that the nonlinear discrete system has the following form:

$$\text{State function } f(\cdot): X_k = f(X_{k-1}) + \omega_k, \quad (5.8)$$

$$\text{Observation function } h(\cdot): Z_k = h(X_k) + v_k. \quad (5.9)$$

where ω_k is the procession noise and obeys the standard Gaussian distribution $\omega_k \sim N(0, Q_k)$, Q_k is its covariance matrix; v_k is the observation errors and complies with the standard normal distribution $v_k \sim N(0, R_k)$, R_k is its covariance. As for the AEKF estimator, it mainly consists of five steps, which include state prediction, observation, measurement prediction, matching and estimation [129]. In the prediction stage, the command signal U_k and

the robot motion model are utilized to estimate the robot pose. Then, in the update stage, to update the landmark position and to refine the estimation of the robot pose, the new observation Z_k from an exteroceptive sensor is used. When a landmark is detected for the first time, however, it is added to the system state vector through an initialization process called state augmentation.

5.5.1. Time update

- Predictor step:

$$\hat{X}_k^- = f(\hat{X}_{k-1}^-), \quad (5.10)$$

$$P_k^- = F_k P_{k-1} F_k^T + Q_k, \quad (5.11)$$

where $\hat{X}_k^-$ is the system current state vector, and P_k^- is its covariance matrix. F_k and H_k are the Jacobian matrixes of partial derivatives of $f(\cdot)$ and $h(\cdot)$ with respect to X .

$$F_k = \left. \frac{\partial f(X)}{\partial X} \right|_{X=\hat{X}_{k-1}^-}, \quad H_k = \left. \frac{\partial h(X)}{\partial X} \right|_{X=\hat{X}_{k-1}^-} \quad (5.12)$$

The non-linear functions f and h are linearized by using a Taylor series expansion, where terms of second order and higher are neglected.

5.5.2. Measurement update

- Calculate the Kalman gain K_k :

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}. \quad (5.13)$$

- Corrector step:

First, update the expected value $\hat{X}_k$:

$$\hat{X}_k = \hat{X}_k^- + K_k [Z_k - h(\hat{X}_k^-)]. \quad (5.14)$$

Then, update the error covariance matrix P_k :

$$P_k = P_k^- - K_k H_k P_k^- = (I - K_k H_k) P_k^-. \quad (5.15)$$

Figure 5.5 The AEKF estimator

The simulation of the AEKF estimator is shown in Figure 5.5. Here we suppose that both, the robot control noise and the sensor observation noise, are equal to 1. And we assign the robot's start position at the coordinate $(0, 7m)$ with a velocity of $1 m/s$. The true robot path is depicted as the red line, the green line stands for the robot path estimated by the AEKF. The observations are drawn as black line with black crosses ' + ', which are the surrounding landmarks. The two blue lines mean the $+3$ sigma and -3 sigma around the true robot path. It proves that AEKF could estimate the robot pose and the landmark positions accurately and robustly.

5.6. The estimation process of the AEKF-SLAM

As can be found in [130], the Root Mean Square (RMS) error in the AEKF-SLAM just before loop closure is much smaller than the one in the EKF-SAM. Also, the AEKF-SLAM can compensate the odometric error of the robot more exactly and generate a more precise feature map than the EKF-SLAM. As for the popular FastSLAM algorithm [65], which employs the Rao-Blackwellised method for Particle Filtering (RBPF), is based on an extract factorization of the posterior into a product of conditional landmark distributions and a distribution over robot paths. The FastSLAM behaves much better than the EKF-SLAM at handling the data association issue for the nonlinear system map revisitation. Nevertheless, the biased noises resulting from the unequal wheels misalignment skew the robot's odometric trajectory to one side, but the conventional solutions for the SLAM problem, like FastSLAM or EKF, cannot estimate precisely, since they assume zero mean noise while compensating odometry errors. Particle filters work best in grid constrained areas, such as corridors, and in the representation of occupancy grid maps, not in landmark maps like those considered in this work. Another

challenge in the particle filtering SLAM is to reduce the number of particles while maintaining the estimation accuracy. For indoor environments and aerial structures (such as roads or buildings), particle filters are well suited. As a consequence, here we present the AEKF-SLAM based algorithm to deal with the robotic underwater SLAM problem. The experiments performed later in this work show that the AEKF-SLAM approach builds a more accurate landmark map and also estimates the robot trajectory more precisely than the FastSLAM 2.0.

In any SLAM algorithm, the position and number of environmental landmarks is not known a priori, landmark coordinates must be initialized and inferred from the sensor observations alone. The AEKF-based SLAM algorithm applies the AEKF to online SLAM by using the maximum likelihood data association method for the correspondence test of features. Here, the recursive AEKF-SLAM based robotic navigation algorithm is a recursive and iterative estimation-update process, which besides a prediction stage, an update stage (as in traditional EKF), includes a new augmentation stage additionally. First and foremost, the new underwater robotic pose is estimated by using a motion model, and the environmental feature positions are estimated by using an observation model. Then, the errors between the measurements and the true features are calculated, and the error covariance is used to calculate the Kalman gain W , which is then employed to update the estimated robot pose. Finally, the new features of the environment are merged into the map.

Figure 5.6 The flow chart of SLAM procedure based on an AEKF

The whole procedure of AEKF-SLAM system is illustrated in Figure 5.6. The prediction stage is to reckon the current state of the robot with its on-board navigation sensors. In the update stage, the robot measures the surrounding targets by the imaging sensor and to obtain the relative positions to the robot. Then, the relative positions of detected features are transformed into the robot's coordinates system. The next matching involves a data association process, where features from the predicted and actual observed maps are integrated and their deviations are applied to track the robot path and refine the detected landmark positions. The measurements from the sonar image include three kinds of information: new features, associated features and observations without any direct relationship. After data association, new features will augment the state vector, associated features can improve the precise degree of the state, and offensive features are discarded directly.

Table 5.1 summarizes the necessary procedures of the presented algorithm of the AEKF-SLAM based robotic underwater navigation. As soon as the feature extraction and the data association are in place, the AEKF-SLAM method can be considered as the following three steps. First and foremost, predict the robot current state using the odometry data. Next, update the estimated state from re-observed landmarks. Eventually, add new landmarks into the map. If a feature is perceived for the first time, then it is included to the system state by the proposed augmentation state.

Table 5.1 The AEKF operations for achieving underwater SLAM

Event	SLAM	AEKF
Robot Navigation	Robot Motion	AEKF Prediction
Sensor Detects Known Feature	Map Correction	AEKF Update
Sensor Detects New Feature	Landmark Initialization	State Augmentation
Map Corrupted Feature	Landmark Removal	State Reduction

The architecture of the presented AEKF-SLAM based robotic navigation system is demonstrated in Figure 5.7. Let $\hat{X}_k$ and $\hat{P}_k$ be the estimated system state and its covariance matrix. Then, the filter iteratively refines the state $\hat{X}_k^+$ mean and state covariance $\hat{P}_k^+$ by integrating the predicted state mean $\hat{X}_k^-$ and state covariance $\hat{P}_k^-$ with the new observation Z_k . Q_k and R_k are the covariances of procession noises and observation errors, separately.

Figure 5.7 The architecture of AEKF-SLAM based robotic navigation system, as in [9]

The pseudo code of the presented AEKF-based SLAM algorithm for robotic underwater localization and mapping is summarized in Algorithm 4 as follows. Where z_o stands for the vector of previously observed features; z_n represents the vector of measurements which are unseen and newly detected landmarks.

Algorithm 4: AEKF-SLAM based robotic underwater navigation

1. For $k = 1$ to N
 2. $[X_k^-, P_k^-] = \text{Predict}(X_{k-1}, P_{k-1})$;
 3. $z_k = \text{Get Observations}()$;
 4. $[z_o, z_n] = \text{Data Association}(X_k^-, P_k^-, z_k, R_k)$;
 5. $[X_k^+, P_k^+] = \text{Update Map}(X_k^-, P_k^-, z_o, R_k)$;
 6. $[X_k^+, P_k^+] = \text{Update Map}(X_k^-, P_k^-, z_n, R_k)$;
 7. End for
-

5.6.1. Vehicle, map and augmented state vectors

Varying bathymetric height of the AUV makes it tough to build the underwater map in the same scale and identical resolution. Therefore, we assume that the actual 3D spatial geometry to be perpendicular to the horizontal plane in which the AUV navigates, so that the world can be adequately represented by a simplified 2D model in order to put more attention to the presented AEKF-SLAM based robotic underwater navigation approach. The fundamental formulas for the AEKF-SLAM algorithm are presented as follows. In AEKF-SLAM based robotic navigation system, the system state vector consists of the current robot pose and the positions of all

observed environmental features [131]. That means the state vector grows larger and larger when more and more features are observed. And the robot state is represented by its position and heading angle vectors in Cartesian coordinates, which can be defined as:

$$\hat{X}_v = [\hat{x}_v, \hat{y}_v, \hat{\phi}_v]^T, \quad (5.16)$$

with its covariance matrix P_v :

$$P_v = \begin{bmatrix} \sigma_{x_v x_v}^2 & \sigma_{x_v y_v}^2 & \sigma_{x_v \phi_v}^2 \\ \sigma_{x_v y_v}^2 & \sigma_{y_v y_v}^2 & \sigma_{y_v \phi_v}^2 \\ \sigma_{x_v \phi_v}^2 & \sigma_{y_v \phi_v}^2 & \sigma_{\phi_v \phi_v}^2 \end{bmatrix}. \quad (5.17)$$

The 2D point features observed by the robot to form a map are in the same base coordinate system as the robot. More elaborate parametric feature models, such as lines, might also be used, but not implemented in this thesis. Supposing the position of the n th landmark is represented as $x_{m_n} = (\hat{x}_n, \hat{y}_n)^T$, and the environmental features can be described as:

$$\hat{X}_m = [\hat{x}_1, \hat{y}_1, \dots, \hat{x}_n, \hat{y}_n]^T, \quad (5.18)$$

and its covariance matrix P_m is:

$$P_m = \begin{bmatrix} \sigma_{x_1 x_1}^2 & \sigma_{x_1 y_1}^2 & \dots & \sigma_{x_1 x_n}^2 & \sigma_{x_1 y_n}^2 \\ \sigma_{x_1 y_1}^2 & \sigma_{y_1 y_1}^2 & \dots & \sigma_{y_1 x_n}^2 & \sigma_{y_1 y_n}^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{x_1 x_n}^2 & \sigma_{y_1 x_n}^2 & \dots & \sigma_{x_n x_n}^2 & \sigma_{x_n y_n}^2 \\ \sigma_{x_1 y_n}^2 & \sigma_{y_1 y_n}^2 & \dots & \sigma_{x_n y_n}^2 & \sigma_{y_n y_n}^2 \end{bmatrix}. \quad (5.19)$$

The off-diagonal terms of the covariance matrix P_m are the cross-correlation information between different landmarks. They capture the dependence of each landmark position upon knowledge of the other landmarks in the map. Since the features are assumed to be stationary and their positions do not change over time, the cross-correlations of these features in the map will be enhanced with each re-observation, which makes the map increasing rigid.

The AEKF-SLAM landmark map is represented by an augmented state vector $\hat{X}_a$, which is formed by the concatenation of the present robot state $\hat{X}_v$ and all previously detected landmark locations $\hat{X}_m$ [9.131]. The cross covariance between the robot pose and environmental landmarks is denoted as P_{vm} :

$$\hat{X}_a = \begin{bmatrix} \hat{X}_v \\ \hat{X}_m \end{bmatrix}, P_a = \begin{bmatrix} P_v & P_{vm} \\ P_{vm}^T & P_m \end{bmatrix}. \quad (5.20)$$

Note that the initial conditions of the state estimate are usually given as $\hat{X}_a = \hat{X}_v = 0$ and $P_a = P_v = 0$, meaning that the robot has not perceived any features until now and the base coordinate system is built upon the initial robot pose.

5.6.2. Prediction stage

The SLAM process model specifies that a robot moves relative to its previous pose according to a dead reckoning motion estimate, and the surrounding landmarks remain stationary. The effect of this model on the system state estimate is a change in the $\hat{X}_v$ term of the state vector, and in the P_v and P_{vm} terms of the state covariance matrix, while the $\hat{X}_m$ and P_m portions remain constant. An estimate of the underwater robot pose change $\hat{X}_\delta = [\hat{x}_\delta, \hat{y}_\delta, \hat{\phi}_\delta]^T$ with covariance P_δ (see Figure 5.8) is commonly obtained by the Inertial Navigation System (INS) and a robot kinematic model.

Figure 5.8 The robot motion model

As a result, the estimated system state is $\hat{X}_a^-$ is calculated as:

$$\hat{X}_a^- = f(\hat{X}_a, \hat{X}_\delta) = \begin{bmatrix} g(\hat{X}_v, \hat{X}_\delta) \\ \hat{X}_m \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \end{bmatrix} \begin{bmatrix} \hat{x}_v + \hat{x}_\delta \cos \hat{\phi}_v - \hat{y}_\delta \sin \hat{\phi}_v \\ \hat{y}_v + \hat{x}_\delta \sin \hat{\phi}_v + \hat{y}_\delta \cos \hat{\phi}_v \\ \hat{\phi}_v + \hat{\phi}_\delta \\ \hat{X}_m \end{bmatrix}, \quad (5.21)$$

and its prediction covariance matrix P_a^- is:

$$P_a^- = J P_a J^T + Q P_\delta Q^T, \quad (5.22)$$

where the Jacobian matrices J and Q are given by:

$$J = \left. \frac{\partial f}{\partial X_a} \right|_{(X_a, X_\delta)} = \begin{bmatrix} J_v & 0_{vm} \\ 0_{vm}^T & I_m \end{bmatrix}_{(X_a, X_\delta)}, \quad (5.23)$$

$$Q = \left. \frac{\partial f}{\partial X_\delta} \right|_{(X_a, X_\delta)} = \begin{bmatrix} Q_v \\ 0_{vm}^T \end{bmatrix}_{(X_a, X_\delta)}, \quad (5.24)$$

Here, J_v and Q_v are the Jacobian matrices of partial derivatives of the nonlinear motion model function g with respect to the current robot state X_v and the robot pose change X_δ :

$$J_v = \left. \frac{\partial g}{\partial X_v} \right|_{(X_v, X_\delta)} = \begin{bmatrix} \frac{\partial g_1}{\partial x_v} & \frac{\partial g_1}{\partial y_v} & \frac{\partial g_1}{\partial \varphi_v} \\ \frac{\partial g_2}{\partial x_v} & \frac{\partial g_2}{\partial y_v} & \frac{\partial g_2}{\partial \varphi_v} \\ \frac{\partial g_3}{\partial x_v} & \frac{\partial g_3}{\partial y_v} & \frac{\partial g_3}{\partial \varphi_v} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\hat{x}_\delta \sin \hat{\varphi}_v - \hat{y}_\delta \cos \hat{\varphi}_v \\ 0 & 1 & \hat{x}_\delta \cos \hat{\varphi}_v - \hat{y}_\delta \sin \hat{\varphi}_v \\ 0 & 0 & 1 \end{bmatrix} \quad (5.25)$$

$$Q_v = \left. \frac{\partial g}{\partial X_\delta} \right|_{(X_v, X_\delta)} = \begin{bmatrix} \frac{\partial g_1}{\partial x_\delta} & \frac{\partial g_1}{\partial y_\delta} & \frac{\partial g_1}{\partial \varphi_\delta} \\ \frac{\partial g_2}{\partial x_\delta} & \frac{\partial g_2}{\partial y_\delta} & \frac{\partial g_2}{\partial \varphi_\delta} \\ \frac{\partial g_3}{\partial x_\delta} & \frac{\partial g_3}{\partial y_\delta} & \frac{\partial g_3}{\partial \varphi_\delta} \end{bmatrix} = \begin{bmatrix} \cos \hat{\varphi}_v & -\sin \hat{\varphi}_v & 0 \\ \sin \hat{\varphi}_v & \cos \hat{\varphi}_v & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.26)$$

Since these Jacobians only have influences on the robot portion of the covariance matrix P_v and its cross-correlated covariance P_{vm} , the estimated system covariance matrix P_a^- can be computed and implemented more efficiently as:

$$\therefore P_a^- = \begin{bmatrix} J_v P_v J_v^T + Q_v P_\delta Q_v^T & J_v P_{vm} \\ (J_v P_{vm})^T & P_m \end{bmatrix}. \quad (5.27)$$

5.6.3. Update stage

Suppose that a landmark, which is already stored in the map as an estimate $x_{m_i} = (\hat{x}_i, \hat{y}_i)^T$, is perceived by a range-bearing sonar with the measurement z (see Figure 5.9):

$$z = \begin{bmatrix} r \\ \theta \end{bmatrix}, R = \begin{bmatrix} \sigma_r^2 & \sigma_{r\theta}^2 \\ \sigma_{r\theta}^2 & \sigma_\theta^2 \end{bmatrix}, \quad (5.28)$$

where (r, θ) denotes the distance and the orientation of the observed landmark with respect to the robot coordinate, and the observation covariance matrix is denoted as R .

Figure 5.9 The robot observation model

If we get $i (i > 1)$ observations at a time, thus the measured vector Z and its observation covariance matrix R can be described as:

$$Z = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \end{bmatrix}, R = \begin{bmatrix} R_1 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & R_i \end{bmatrix}. \quad (5.29)$$

Next, the transformation of the derived locations from the global Cartesian coordinate to the local robot side is as follows. Therefore, distinct map landmarks link with each other, and their relationships increase monotonically until their relative locations are known.

$$\hat{z}_i = h_i(\hat{X}_a^-) = \begin{bmatrix} \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2} \\ \arctan\left(\frac{\hat{y}_i - \hat{y}_v}{\hat{x}_i - \hat{x}_v}\right) - \hat{\phi}_v \end{bmatrix}. \quad (5.30)$$

If the measurement z associates with the estimated map feature position $(\hat{x}_i, \hat{y}_i)^T$ correctly, then we update the SLAM results:

$$\hat{X}_a^+ = \hat{X}_a^- + W_i v_i, \quad (5.31)$$

$$P_a^+ = P_a^- - W_i S_i W_i^T. \quad (5.32)$$

The measurements residual v_i , also called as innovation, which is the difference between the actual perceived and predicted measurements, is defined as:

$$v_i = z - h_i(\hat{X}_a^-), \quad (5.33)$$

with its covariance S_i :

$$S_i = H P_a^- H^T + R, \quad (5.34)$$

and the Kalman gain W_i :

$$W_i = P_a^- H^T S_i^{-1}, \quad (5.35)$$

where H represents the Jacobian matrix which linearizes the nonlinear measurements function h around the best estimation of the state $\hat{X}_a^-$. As for H , first $\Delta x, \Delta y, d, H_1, H_2$ are defined in advance as follows:

$$\Delta x = \hat{x}_i - \hat{x}_v, \Delta y = \hat{y}_i - \hat{y}_v, \quad (5.36)$$

$$d = \sqrt{(\hat{x}_i - \hat{x}_v)^2 + (\hat{y}_i - \hat{y}_v)^2}, \quad (5.37)$$

$$H_1 = \left. \frac{\partial h}{\partial x_v} \right|_{\hat{X}_a^-} = \begin{bmatrix} -\frac{\Delta x}{d} & -\frac{\Delta y}{d} & 0 \\ \frac{\Delta y}{d^2} & -\frac{\Delta x}{d^2} & -1 \end{bmatrix}, H_2 = \left. \frac{\partial h}{\partial x_m} \right|_{\hat{X}_a^-} = \begin{bmatrix} \frac{\Delta x}{d} & \frac{\Delta y}{d} \\ -\frac{\Delta y}{d^2} & \frac{\Delta x}{d^2} \end{bmatrix}, \quad (5.38)$$

$$\begin{aligned} \therefore H &= \left. \frac{\partial h}{\partial x_a} \right|_{\hat{X}_a^-} = [H_1 \quad 0_1 \quad H_2 \quad 0_2] \\ &= \begin{bmatrix} -\frac{\Delta x}{d} & -\frac{\Delta y}{d} & 0 & 0 & \dots & 0 & \frac{\Delta x}{d} & \frac{\Delta y}{d} & 0 & \dots & 0 \\ \frac{\Delta y}{d^2} & -\frac{\Delta x}{d^2} & -1 & 0 & \dots & 0 & -\frac{\Delta y}{d^2} & \frac{\Delta x}{d^2} & 0 & \dots & 0 \end{bmatrix}. \end{aligned} \quad (5.39)$$

5.6.4. Augmentation Stage

As the environment is explored, newly observed features need to be included in the stored map. Thus, we come up with an adequate solution for initializing new features. First and foremost, the state vector and covariance matrix are extended by the new observation Z_{new} and its covariance R_{new} , which are perceived relative to the robot coordinate:

$$\hat{X}_{aug} = \begin{bmatrix} \hat{X}_a \\ Z_{new} \end{bmatrix}, P_{aug} = \begin{bmatrix} P_v & P_{vm} & 0 \\ P_{vm}^T & P_m & 0 \\ 0 & 0 & R_{new} \end{bmatrix}. \quad (5.40)$$

Here, a transformation function g_i is employed to convert the polar observation Z_{new} into the global Cartesian coordinate. It consists of the current robot pose state $\hat{X}_v$ and the new sensor measurement Z_{new} :

$$g_i(\hat{X}_v, Z_{new}) = \begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} x_v + r \cos(\theta + \hat{\phi}_v) \\ y_v + r \sin(\theta + \hat{\phi}_v) \end{bmatrix}. \quad (5.41)$$

With the help of a linearized transformation f_i function, the system augmented state is initialized to the correct values. The conversion formula f_i is denoted as follows:

$$\hat{X}_a^+ = f_i(\hat{X}_{aug}) = \begin{bmatrix} \hat{X}_a \\ g_i(\hat{X}_v, Z_{new}) \end{bmatrix}, \quad (5.42)$$

$$P_a^+ = \nabla f_{x_{aug}} P_{aug} \nabla f_{x_{aug}}^T, \quad (5.43)$$

where the sparse Jacobian matrix $\nabla f_{x_{aug}}$ is given by:

$$\nabla f_{x_{aug}} = \frac{\partial f_i}{\partial X_{aug}} \bigg|_{\hat{X}_{aug}} = \begin{bmatrix} I_v & 0 & 0 \\ 0 & I_m & 0 \\ G_{X_v} & 0 & G_{Z_{new}} \end{bmatrix}, \quad (5.44)$$

and the Jacobian matrices G_{X_v} and $G_{Z_{new}}$ are:

$$G_{X_v} = \frac{\partial g_i}{\partial X_v} \bigg|_{(\hat{X}_v, Z_{new})} = \begin{bmatrix} 1 & 0 & -r \sin(\theta + \hat{\phi}_v) \\ 0 & 1 & r \cos(\theta + \hat{\phi}_v) \end{bmatrix}, \quad (5.45)$$

$$G_{Z_{new}} = \frac{\partial g_i}{\partial Z_{new}} \bigg|_{(\hat{X}_v, Z_{new})} = \begin{bmatrix} \cos(\theta + \hat{\phi}_v) & -r \sin(\theta + \hat{\phi}_v) \\ \sin(\theta + \hat{\phi}_v) & r \cos(\theta + \hat{\phi}_v) \end{bmatrix}. \quad (5.46)$$

The matrix multiplication of P_a^+ requires $O(n^3)$ computation complexity where n is the number of features on the map. Due to the sparseness of the Jacobian matrix, a much more efficient transformation can be implemented; it also only affects the block diagonal matrix of the newly observed landmark and off diagonal cross-correlations to the rest of the map.

$$\therefore P_a^+ = \begin{bmatrix} P_v & P_{vm} & P_v G_{X_v}^T \\ P_{vm}^T & P_m & P_{vm}^T G_{X_v}^T \\ G_{X_v} P_v & G_{X_v} P_{vm} & G_{X_v} P_v G_{X_v}^T + G_{Z_{new}} R_{new} G_{Z_{new}}^T \end{bmatrix}. \quad (5.47)$$

5.7. Summary

In this section, the mathematical framework of the proposed ‘‘AEKF-based robotic underwater SLAM’’ algorithm has been established. The main contribution of this chapter is to demonstrate this estimation theoretic solution to the underwater SLAM based navigation problem and to elucidate upon its mathematical structure. The SLAM process consists of generating the best estimation of the system states given the information available to the system. AEKF-SLAM based robotic navigation system stores the robot pose and the map landmarks in a single system state vector, and estimates the state parameters by using a recursive, iterative, three-stage procedure comprising a prediction, an observation (which exists as well as in the conventional EKF) and a newly proposed augmentation step. First and foremost, the prediction phase deals with robot motion model based on the incremental dead reckoning estimates, and it also increases the uncertainty of the robot pose estimate. Then, the update stage occurs with the re-

observation of already stored landmarks, improving the overall state estimate. When a landmark is detected for the first time, however, it is added to the state vector through an initialization process called state augmentation. The AEKF-SLAM based robotic underwater navigation algorithm emphasizes the identification of each sensor perception as a new landmark or a re-visited observed one. It can distinguish new landmarks from those already detected ones. With the assistance of the presented AEKF approach, the underwater robot achieves a more precise and robust self-localization and mapping of the surrounding landmarks.

At present, the research on robotic underwater SLAM applications mainly focuses on reducing the linearization errors so that long-term missions can be achieved successfully. It has been observed that the linearization errors become larger when big covariances are involved. Approaches such as hierarchical SLAM or sub-mapping methods build local maps of limited size, which bound the covariances and thus the linearization errors. Next, by linking the local maps through a global map or a hierarchy of global maps, the AEKF-SLAM based robotic application will be possible in large scenarios, such as in our considered case of underwater environments.

6 The AEKF-SLAM and the FastSLAM simulated experiments

In this section, several MATLAB simulation experiments are carried out for both AEKF based SLAM and FastSLAM 2.0 for dense loop mapping and line mapping, which are executed by a generic autonomous robot observing the environmental landmark points (including not only the calculated centroids of the shipwreck, branch and plastic mannequin, but also those centroids of certain parts of the background detected by the formerly presented improved Otsu TSM in Section 3) with a range-bearing sensor in a 2D area. Here, we assume that an AUV is observing fixed objects in an Inertial Reference System (IRS) in order to enhance the localization performance, since the INS of an AUV suffers from drift [138]. We can change the value of various parameters according to the number of detected landmarks and their positions in the environment, the practical velocity of the AUV, and the maximum perception range of the selected sonar sensors.

6.1. The AEKF-SLAM and FastSLAM 2.0 based underwater robotic navigation simulations

Since FastSLAM 2.0 requires fewer particles and is more robust and accurate than FastSLAM 1.0, the following two MATLAB simulation experiments are carried out for both AEKF based SLAM and FastSLAM 2.0 for dense loop mapping and line mapping, which are executed by a generic autonomous robot measuring the environmental landmark points with a range-bearing sensor in a 2D area.

6.1.1. Dense loop map

The AEKF-SLAM and FastSLAM 2.0 simulation environment for the dense loop map is established as a $200\text{ m} \times 200\text{ m}$ wide area with coordinates ranging from -100 m to 100 m . Indeed, 17 robot waypoints are arranged in form of a circle and 36 landmarks are randomly distributed near the robot trajectory as illustrated in Figure 6.1. We set the robot speed to 3 m/s , its deviation is 0.3 m/s , and its heading angle error is $3\pi/180\text{ rad}$. The range and bearing of the robotic observation variance is 0.1 m/s and $\pi/180\text{ rad}$. The robot observes the surrounding features every 0.1 s with a sampling time of 0.0125 s . This leads to one image each 0.3 m with a maximum perception range of 30 m . As for the FastSLAM 2.0, there are 100 particles to estimate the 36 landmark positions and the robot pose, and we set 75 particles as the minimum number of effective particles before resampling.

Figure 6.1 depicts the 2D dense loop feature map built by our presented AEKF-SLAM algorithm and the conventional FastSLAM 2.0 approach, respectively, where the landmarks are arbitrarily distributed. The sensor scans for the landmark point positions are clearly to be seen,

and few sightings are discarded by the statistical outlier rejection techniques [21], because they are out of the robotic perception distance. The rest denote distinct landmarks and are included into the stored map. The actual landmarks are depicted as blue stars ('*'); green circles ('o') are the robot waypoints, which are used for computing the robot steering angle. The landmark locations estimated by our AEKF-SLAM algorithm are drawn as the red crosses ('+') in Figure 6.1 (a), and the ellipse around every red cross illustrates the uncertainty covariance for the estimated landmark positions. The predicted robot trajectory is shown as the solid black line, led by a cyan triangle. Around this cyan triangle, there is a red ellipse, which denotes the covariance of the posterior AEKF estimate projected into the robot state. The larger it is, the more uncertain about the current robot pose is.

(a)

(b)

Figure 6.1 (a) The robot is observing the landmarks A and B in the AEKF-SLAM dense loop map; (b) The robot is getting measurements A and B in the FastSLAM 2.0 dense loop map

In Figure 6.1 (b), to clearly display the particles in the FastSLAM 2.0 dense loop experiment, they are drawn as the red dots ('.'). The small cyan ellipses represent the covariances of max weight particle for estimating the robot pose and landmark positions. The true positions of the landmarks A and B, in Figure 6.1 dense loop maps, are at $(-55.89\text{ m}, -55.92\text{ m})$ and $(-92.96\text{ m}, -77.73\text{ m})$.

As the coordinate of the first detected landmark point is known with high precision, the uncertainty in the predicted robot state will reduce dramatically when the robot achieves the loop navigation and revisits this landmark. This situation results in the uncertainties of previously observed landmarks are also decreased. It is visible that our proposed AEKF is much more precise and more efficient than the FastSLAM 2.0 for dense loop mapping, since the AEKF estimates both the landmark positions and the robot pose faster and more accurately than the classic FastSLAM 2.0. Table 6.1 illustrates the comparisons of the processing time and the estimated landmark A and B positions in the dense loop map generated by our proposed AEKF-SLAM and FastSLAM 2.0.

Table 6.1 The comparisons of the computational time and the estimated landmark A, B positions in the dense loop map derived by the AEKF-SLAM and FastSLAM 2.0

	Computational Time	Estimated Landmark A	Estimated Landmark B
	[s]	[m]	[m]
AEKF-SLAM	137.937051	$(-56.61, -55.86)$	$(-94.46, -76.53)$
FastSLAM2.0	525.526820	$(-51.38, -61.94)$	$(-87.06, -85.83)$

Therefore, in the dense loop map, the standard deviations between the landmark positions A and B predicted by the AEKF-SLAM and the practical ones are:

$$\sigma_A = 0.723, \sigma_B = 1.921. \quad (6.1)$$

The standard deviations between the landmark positions A and B approximated by the FastSLAM 2.0 and the real ones are:

$$\sigma_A = 7.522, \sigma_B = 10.021. \quad (6.2)$$

which are both 10 times larger than those of our proposed AEKF-SLAM. Besides, the AEKF-SLAM consumes less than one third processing time used by the conventional FastSLAM 2.0

for dense loop mapping. In conclusion, the AEKF-SLAM method has much better performances of localization and mapping accuracy with relatively low computational load.

6.1.2. Line map

The AEKF-SLAM and FastSLAM 2.0 simulation environment for the line map is an area of $1000\text{ m} \times 800\text{ m}$ (from 0 m to 1000 m and from -400 m to 400 m). There are 27 landmarks and 4 robot waypoints. The velocity of the robot is 3 m/s , its variance is 0.3 m/s , the heading orientation error is $3\pi/180\text{ rad}$. The range and bearing of the robotic measurement noise is 0.1 m/s and $\pi/180\text{ rad}$. As before, the sampling time is 0.0125 s , and the robot observes the surrounding features every 0.1 s , obtaining one observation every 0.3 m . The maximum perception distance is again 30 m . In Figure 6.2, the actual robot trajectory is along the x -axis (i.e., $y = 0$), and the true landmark positions are indicated by blue stars ('*'); four green dots ('.') are the robot waypoints which are applied for measuring the robot steering angle. In Figure 14a, the line map which is built by the AEKF-SLAM, the red crosses ('+') are the estimated landmark locations, with red ellipses denoting their uncertainty covariance. The line map also depicts the estimated robot trajectory as the solid black line, guided by the cyan triangle. Around this cyan triangle, we can find a red ellipse, which represents the covariance of the posterior AEKF estimate projected into the robot state. The larger it is, the more uncertain is the current robot pose. As for the line map generated by the FastSLAM 2.0 in Figure 6.2 (b), we set 100 particles for estimating the 27 landmark positions and the robot pose, and 75 particles as the minimum number of effective particles before resampling. These particles are depicted as the red dots ('.') for a purpose of clear visualization. The small cyan ellipses represent the covariances of max weight particle for estimating the robot pose and landmark positions. In Figure 6.2, the true positions of the landmarks A , B , C and D are at $(886.8\text{ m}, -19.04\text{ m})$, $(905.7\text{ m}, 10.68\text{ m})$, $(965.1\text{ m}, 16.08\text{ m})$ and $(989.4\text{ m}, -19.04\text{ m})$.

Figure 6.2 (a) Partial magnification of the AEKF-SLAM line map; (b) Partial magnification of the FastSLAM 2.0 line map

In Figure 6.2 (a), it can be found that with the appearance of strange discontinuities in the estimated robot path, the inconsistency of the AEKF-SLAM algorithm becomes visibly evident. The jagged robot trajectory results from the inconsistency of variations in Jacobian linearization, given a large heading uncertainty. Over several successive observation updates, the estimated robotic pose shows dramatic jumps, which tend to be disproportionately large compared to the real measurement deviation (0.3 m/s). A related symptom also appears in the estimations of landmark positions. Again, the size of the landmark updates tends to be much larger than the actual observation noise (0.1 m/s). However, rather than exhibiting random motion in accord with the sensor noise, the average of landmark updates seems to be constrained to the line (x -axis). These symptoms manifest once the robot heading uncertainty becomes large, but they will not appear if the Jacobians are linearized about the true state. Note that these symptoms are not caused by numerical errors, since different numerical forms of the AEKF give identical results. These symptoms are simply due to a measurement trying to correct the robot heading, when the heading variance has been artificially reduced. Above all, it is evident that the AEKF-SLAM algorithm is much more reliable and efficient than the FastSLAM 2.0 for line mapping, since the AEKF-SLAM estimates both the landmark positions and the robot pose more precisely and faster than the classic FastSLAM 2.0. The comparisons of the computational complexity and the estimated landmark A , B , C and D positions in the simulated line map generated by the AEKF-SLAM and FastSLAM 2.0 are shown in Table 6.2.

Table 6.2 The comparisons of the computational time and the estimated landmark
A, B, C, D positions in the line map derived by the AEKF-SLAM and FastSLAM 2.0

	Computational Time [s]	Estimated Landmark A [m]	Estimated Landmark B [m]	Estimated Landmark C [m]	Estimated Landmark D [m]
AEKF-SLAM	99.837974	(885.5, -22.53)	(904.9, 6.886)	(964.4, 11.27)	(988, -24.21)
FastSLAM2.0	594.113594	(879.4, -72.63)	(902.6, -46.26)	(961.7, -48.6)	(981.5, -86.27)

As a consequence, in the line map, the standard deviations between the landmark positions A , B , C and D estimated by the AEKF-SLAM and the true ones are:

$$\sigma_A = 3.724, \sigma_B = 3.877, \sigma_C = 4.983, \sigma_D = 5.356. \quad (6.3)$$

The standard deviations between the landmark positions A , B , C and D estimated by the FastSLAM 2.0 and the true ones are:

$$\sigma_A = 54.099, \sigma_B = 35.715, \sigma_C = 32.697, \sigma_D = 67.693. \quad (6.4)$$

which are all at least 6.5 times larger than those of the presented AEKF-SLAM. Even for the landmarks A and D , the standard deviations estimated by the FastSLAM 2.0 are both more than 12.5 times higher than for our method. Moreover, the computational time of the conventional FastSLAM 2.0 is nearly 6 times higher than that of the AEKF-SLAM for line mapping. As a conclusion, for both dense loop mapping and line mapping experiments, the AEKF-SLAM approach has the best performances of localization and mapping accuracy with relatively low computational cost.

6.2. AEKF-SLAM dense loop map using landmarks detected by the improved Otsu TSM

The simulation environment for the AEKF-SLAM loop map, where the landmarks are detected by the improved Otsu TSM is defined as an area of $200\text{ m} \times 200\text{ m}$, containing 17 robot waypoints connected as a circle and 54 landmarks distributed near the robot path, as shown in Figure 6.3. Since the environment is an area of $200\text{ m} \times 200\text{ m}$ (from -100 m to 100 m), we set the centroid coordinates of the ship as (53.5, 60.3), (54.23, 65.39), the body centroids as $(-94.98, -66.29)$, $(-96.69, -66.06)$, $(-102.12, -61.57)$, $(-102.41, -70.3)$, $(-106.55, -81.13)$ and the branch centroid as (18.7, -11.6). Besides, those centroids of certain parts of the background detected by the improved Otsu TSM are also included. The velocity of the robot has been fixed to 3 m/s , its deviation to 0.3 m/s , and the steering angle error to $3\pi/180\text{ rad}$. The range and bearing of observation variance is 0.1 m/s and $\pi/$

180 *rad*, respectively. The sampling time is 0.0125 s and the robot measures an observation each 0.1 s, which means that the robot gets an image each 0.3 m. The maximum observation distance is 30 m.

Figure 6.3 demonstrates the 2D landmark map obtained by the AEKF-based SLAM algorithm, where the landmarks are detected by the improved Otsu TSM. The positions of sensor scans for the point landmarks are clearly visible, and few of these sightings are rejected based on statistical outlier rejection techniques [21], since they are out of the observation range of the robot. The rest are believed to represent a landmark and are added into the map. The true landmarks are shown as blue stars ('*'), green dots ('.') are the robot waypoints which are used for calculating the steering angle, and the red crosses ('+') are the estimated landmark positions. The ellipses around each red cross illustrate the uncertainty covariances for the landmark coordinate estimations. The estimated robot path is depicted as the solid black line, led by a cyan triangle. Around this cyan triangle we can see a red ellipse, which stands for the covariance of the AEKF estimate projected to the robot pose. The larger it is, the more uncertain the current robot pose is. The perceptions are drawn as cyan lines, in Figure 6.3 (a), the robot is observing the centroids of some parts of the body. After navigation, the robot has identified the centroid positions of different portions of the ship, branch and body, which are connected by the black line in Figure 6.3 (b).

(a)

(c)

Figure 6.3 (a) The robot is observing the centroids of certain parts of the body before loop closure; (b) The final AEKF-SLAM loop map where the landmarks are detected by the improved Otsu TSM

The landmark point positions of the ship, branch and body estimated by the AEKF and the real ones directed by the improved Otsu TSM are shown in Table 6.3.

Table 6.3 The landmark point positions of the ship, branch and body estimated by the AEKF and the true ones detected by the improved Otsu TSM

	Ship [m]		Branch[m]		Body [m]			
True	(53.5,	(54.23,	(18.73,	(-94.98,	(-96.69,	(-102.12,	(-102.41,	(-106.55,
	60.3)	65.39)	-11.56)	-66.29)	-66.06)	-61.57)	-70.3)	-81.13)
Estimated	(53.66,	(54.31,	(18.8,	(-94.99,	(-96.67,	(-102.2,	(-102.4,	(-106.4,
	60.23)	65.32)	-11.49)	-66.35)	-66.12)	-61.59)	-70.34)	-81.44)

Therefore, the standard deviation between landmark point positions of different features estimated by the AEKF and the true ones detected by the improved Otsu TSM is:

$$\sigma_1 = 0.124. \quad (6.5)$$

For comparison, the AEKF-SLAM loop map where 77 landmarks are detected by the maximum entropy TSM is shown in Figure 6.4.

Figure 6.4 (a) The robot is observing the centroids of certain parts of the body before loop closure; (b) The final AEKF-SLAM loop map where the landmarks are detected by the maximum entropy TSM

The observations of the body parts are shown in Figure 6.4 (a), while Figure 6.4 (b) shows as before, the corresponding final estimated robot path after navigation. Table 6.4 contains the data for the landmark point positions of the ship, branch and body estimated by the AKEF and the real ones detected by the maximum entropy TSM.

Table 6.4 The landmark point positions of the ship, branch and body estimated by the AEKF and the true ones detected by the maximum entropy TSM

	Ship [m]		Branch [m]		Body [m]	
True	(53.61, 60.18)	(54.22, 65.4)	(18.75, -11.55)	(-99.23, -67.7)	(-97.59, -72.08)	
Estimated	(54.24, 59.12)	(54.96, 64.35)	(18.62, -11.69)	(-100.1, -65.82)	(-98.58, -70.23)	

As a result, the standard deviation between landmark point positions of different features estimated by the AEKF and the real ones detected by the maximum entropy TSM is:

$$\sigma_2 = 1.37, \quad (6.6)$$

which is much larger than that of the improved Otsu TSM. Thus, the AEKF achieves more accuracy and estimates the robot pose more robustly due to the landmark positions obtained by our improved Otsu TSM, than those detected by the maximum entropy TSM. Although the maximum entropy TSM achieves the highest segmentation precision over other three classic segmentation methods compared above, it has much lower precision than our improved Otsu TSM, and it also needs much more computation time. Moreover, it can be found that since the position of the first observed landmark point is known with high accuracy, the uncertainty in the estimated robot pose will decrease significantly when the robot completes the loop and revisits this landmark. Thus, in turn, the uncertainties of previously detected landmarks are reduced. Also, the AEKF-based SLAM algorithm achieves reliable detection of cycles in the map and consistent map update on loop closure.

6.3. SWARMs seabed mapping use case

This thesis is carried out in the context of the European Smart and Networking Underwater Robots in Cooperation Meshes (SWARMs) project [133]. Motivated by one of the goals defined in the SWARMs project, it aims to find efficient solutions for robotic perception, autonomous localization and mapping, so that more complex missions could be accomplished in cooperation of underwater vehicles. Our presented AEKF-based robotic underwater SLAM algorithm is tested and validated in Trondheim Fjord (Norway), June 2018. The depth of the Trondheim Fjord is around 92 meters, in June the water temperature is 7.69 °C, the water salinity 35.07 g/L, the sound speed 1483.25 m/s, and the sea bottom material is medium sand with sand. In our case, the DeepEye 680D Side Scan Sonar (SSS) is embedded into the Alister 9 AUV provided by ECA Gruop company, one of our SWARMs project partner. The DE680D SSS is a chirp sonar working at the frequency of 680 kHz with an optimized resolution of 1 cm. The sonar beams reach from 10 m to 200 m and the maximum operation depth is 100 m.

The primary goal of the SWARMs project is to facilitate the conception, planning and execution of the maritime and offshore operations and to make underwater and surface vehicles further accessible and useful, which includes: (1) Enabling Autonomous Underwater Vehicles (AUVs)/Remote Operating Vehicles (ROVs) to work in a cooperative mesh thus opening up new applications and ensuring reusability; (2) Increasing the autonomy of AUVs/Unmanned Surface Vehicles (USVs) and improving the usability of ROVs for the execution of simple and complex tasks. This reduces the operational costs, enhances the safety of mission tasks and of involved individuals, and expands the offshore sector. All the underwater vehicles including the AUVs and ROVs that are used in the SWARMs project are shown in Table 6.5.

Table 6.5 The parameters of the underwater vehicles (AUVs and ROVs) employed in the SWARMs environmental sensing mission

Platform	Circular Length (m)	Circular Width (m)	Circular Height (m)	Weight on Air (kg)
Alister9AUV	2.0	0.22	0.22	70
IXN AUV	1.9	0.5	0.3	150
Naiad AUV	0.84	0.6	0.25	30
SAGA ROV	0.42	0.33	0.27	10

Localization and mapping is one of the key components that enables the autonomy of AUVs. Many sensor modalities can be used in this task, and the INS is one of the most common approaches. For precise maneuvering, an INS on board of the AUV calculates through dead reckoning the AUV position, acceleration, and velocity. Estimates can be made using data from an Inertial Measurement Unit (IMU). However, the INS suffers from a drift problem, which should be mitigated for a long-term operation of AUVs.

In Figure 6.5, the following vehicles employed in the SWARMs project can be individually selected within the simulation environment and can be controlled by using a standard game controller: Alister 9 AUV provided by the ECA Group (Paris, France) [82], IXN AUV provided by the Ixion Industry & Aerospace SA company (Madrid, Spain) [132], Naiad AUV provided by Mälardalen University, SAGA ROV provided by Desistek Robotik Elektronik Yazilim company (Ankara, Turkey) [89], and an imaginary RexROV with a subsea manipulator, which is modelled similar to real existing ROVs. We have proved that it is possible and computationally feasible to concurrently simulate all the vehicles mentioned above in real time.

Figure 6.5 The simulated SWARMs vehicles

There are several use cases involved in the SWARMs project, (<http://www.swarms.eu/usecases.html>), including use case No. 1 Corrosion Prevention in Offshore Installation; No. 2 Monitoring of Chemical Pollution; No. 3 Detection, Inspection and Tracking of Plumes; No. 4 Berm Building; and No. 5 Seabed Mapping. Our presented work is related to the fifth one, Seabed Mapping, which mainly consists of five procedures.

First of all, the Unmanned Surface Vehicles (USVs) will pre-survey the underwater area of interests before the AUVs are deployed. The survey path of the USV could be coordinated with the mother-ship or another USV to maximize the operational efficiency and the optimal multibeam overlap. Then, we deploy the AUVs over the seabed to be mapped. Afterwards, the AUVs monitor and map the underwater environments with the assistance of their on-board navigation sensors to measure their own depth and the distance from the AUVs to the seabed. The AUVs work cooperatively and follow the preplanned trajectories in order to cover the interested exploration areas. Subsequently, the derived sonar data is transmitted from the AUVs to the ROV, which retransmits the perceived information via cable to the surface support vessel. Eventually, real-time data mapping and monitoring is accomplished. Real-time maps could be presented to the operator computer. Operator intervention, such as reducing the velocity of the AUVs, the resolution of certain detecting area will be enhanced. The procedures of the SWARMs seabed mapping mission is clarified in Figure 6.6.

Figure 6.6 Link all the actors for landmark localization and seabed mapping in the SWARMs project

Here, the support vessels transport all the required equipments and the robotic vehicles. They supervise all the operations. The USVs perform a quick bathymetric pre-survey, so that the mission manager could better be better prepared on the conditions for the USVs may thereafter be used to assist the planed track for more detailed and high-resolution mapping performed later by the AUVs. As for the AUVs, they are in charge of mapping the entire seabed as well as to characterize the subsea environment. The ROV plays a role as a relay between the USVs and the AUVs to ease all underwater to surface communications. The ROV and USVs serve as a base station and communication hub for multiple AUV's equipped with any sensor for seabed monitoring and mapping. With a relatively short distance between the ROV and the AUVs, a faster real-time seabed mapping and inspection is enabled.

6.4. Summary

In this section, on the one hand, several MATLAB simulated experiments have been performed for both AEKF-based SLAM and classic FastSLAM 2.0 for dense loop mapping and line mapping, which are executed by a generic autonomous robot measuring the environmental landmark points (including not only the calculated centroids of the shipwreck, branch and plastic mannequin, but also those centroids of certain parts of the background detected by the formerly presented improved Otsu TSM in Chapter 3). The simulation results prove that for both dense loop mapping and line mapping experiments, the AEKF-SLAM based robotic navigation approach has the best performances of localization and mapping accuracy with relatively low computational cost. Moreover, the AEKF achieves more robust and accurate estimations of the robot pose and the landmark positions due to the landmark positions detected by our improved Otsu TSM, than those obtained by the maximum entropy TSM. Although the

maximum entropy TSM achieves the highest segmentation accuracy over other the other three conventional segmentation methods (the Traditional Otsu TSM, the Local TSM and the Iterative TSM) compared in Section 3, it has much lower precision than our improved Otsu TSM, and it also consumes much more computational time. As a conclusion, our presented AEKF-based robotic underwater SLAM algorithm achieves reliable detection of cycles in the map and consistent map update on loop closure.

On the other hand, as a part of the European Smart and Networking Underwater Robots in Cooperation Meshes (SWARMs) project, we have introduced several use cases, especially the No. 5 Seabed Mapping, which our presented work is mostly concerned with.

7

Conclusions and future work

To conclude this thesis, the last chapter briefly summarizes the research work that has been done and points out future works that are worth to explore. The underwater robotic mapping and autonomous navigation problems are going to be everlasting and evolving research topics, which will deserve much more efforts in the future. The specific problem this thesis tried to address, namely underwater robotic SLAM based navigation, is only a small fraction among the issues many researchers have been working on. The approaches involved in the thesis could be considered as novel trials of sonar image processing and wireless sensor network techniques for the coming era of the robotic underwater perception and mapping, and the experimental results would be hopefully useful for researchers who are interested in similar problems.

7.1. Conclusions

In this work, we focus on key topics related to robotic SLAM applications in underwater environments. First and foremost, existing underwater navigation approaches, including inertial navigation, acoustic navigation and geophysical techniques have been briefly introduced and carefully compared. Also, a detailed overview of currently used and popular solutions for the underwater SLAM problem has been presented including the Kalman filter (KF) and its variations, Expectation Maximization (EM) methods and Particle filter (PF) based FastSLAM; characteristics like accuracy, robustness, computational complexity, etc. have been compared. These new developments aim to reduce computational efforts for the underwater SLAM problem, increase capabilities for large-scale maps and improve data association for loop-closing, long-term underwater vehicles operation. Besides, different types of commonly used map representations have been compared regarding their suitability for a priori map localization, which are computational requirements, reliability, robustness, etc. In our case, the landmark map is chosen to represent the exploring underwater region. Consider the sparse spatial distribution of the marine features, being the metric landmark map the most suitable one to represent the underwater environment.

Besides, we have proposed the improved iterative Otsu Threshold Segmentation Method (TSM), which constrains the search range of the ideal segmentation threshold of sonar images. Combined with the Canny contour detection algorithm, the foreground objects of interest (a shipwreck from a high-resolution SSS image, a branch from a low-resolution SSS image and a plastic mannequin from a FLS image) have been separated more accurately than with other conventional TSMs like the traditional Otsu method, the local TSM, the iterative TSM and the maximum entropy TSM. Compared with the traditional Otsu method, our presented method could keep more complete information and details of objects after segmentation, also the holes and gaps in objects are reduced. What's more, the computational cost of our proposed improved Otsu TSM is much lower compared to the maximum entropy TSM, which has the highest segmentation accuracy of the above four classic methods.

Moreover, the improved Otsu TSM has much lower False Positive Rate (FPR) and much higher precision rate on the detected feature centroids than those values of the maximum entropy TSM in all SSS and FLS images presented in this work. In conclusion, the improved Otsu TSM achieves much more precise and fast segmentation performances than other methods for detecting underwater objects of different shapes.

The task of normalizing the sonar images with regard to illumination is challenging due to inhomogeneous echo decay in acoustic shadows. Therefore, an optimal solution which employs the MIXed exponential Regression Analysis (MIRA) method has been proposed for handling the echo decay in sonar images. The Structural Similarity Index Metric (SSIM) and the normalized cross-correlation have been used to evaluate the similarity between a landmark and its 180° - rotated counterpart. Our MIRA model has been compared with the Dark Channel Prior (DCP) model, which is an adaption to well-known fog removal technique to sonar imaging, in terms of one ping, local similarity and global quality analysis. The experimental results showed that our proposed MIRA method has superior normalization performances.

Moreover, a detailed state of the art underwater feature matching methods have been summarized and compares, including the traditional Harris, Speeded Up Robust Features (SURF), Binary Robust Invariant Scalable Keypoints (BRISK), Scale Invariant Feature Transform (SIFT) and Affine Scale Invariant Feature Transform (ASIFT) feature detector. Consider their suitability for subsea sonar map fusion, in terms of computational requirements, reliability, accuracy, etc. For the high-resolution sonar images, where contain particular and distinguishable features, we can detect their SURF feature matches with comparatively low computational requirements. Nevertheless, due to various sonar image resolutions, view angle and illumination changes in the perceived sonar data, underwater features are blurry, few and difficult to be detected, for this case, the conventional keypoint matching approaches fail to detect keypoint pairs. Therefore, we have introduced the underwater wireless sensor network (UWSN) into the sonar map fusion task, and combined with the Delaunay Triangulation technology, the underwater wireless sensor network-based Delaunay Triangulation (UWSN-DT) algorithm has been proposed for improving the performances of sonar map fusion accuracy with low computational complexity, in which the wireless nodes are considered as underwater feature points. Since nodes could provide sufficiently useful information for the underwater map fusion, such as the location. In the above-simulated experiments, it shows that our presented UWSN-DT algorithm works efficiently and robustly, especially for the subsea environments where there are few distinguishable feature points. To the best of our knowledge, this is the first algorithm which adopts the underwater wireless sensor network into the application of sonar map fusion.

The robotic localization and map building process consists of generating the best estimation of the

system states given the information available to the system. This paper also has proposed a novel approach to perform underwater SLAM by using a Mechanically Scanned Imaging Sonar (MSIS). This method starts by processing the underwater sonar data so that range information is extracted from the acoustic profiles. Besides, the range data is grouped in scans while taking into account that the underwater vehicle moves during the sonar data gathering. Afterwards, the scans are stored and the relative robot motions between consecutively gathered scans are included in the state vector. Therefore, robotic underwater SLAM is performed by using the whole sequence of robot motions where scans have been gathered. Moreover, this sequence is stored starting in the current robot pose, so that AEKF linearization errors can be reduced, similarly to the robocentric SLAM approach.

Most importantly, a new method, the robotic underwater AEKF-SLAM based navigation algorithm is provided in this thesis. The main contributions of this work are to demonstrate this estimation theoretic solution to the underwater SLAM based navigation problem and to elucidate upon its mathematical structure. It is performed by storing the robot pose and the map landmarks in a single system state vector, and estimating the state parameters via a recursive, iterative, three-stage procedure comprising a prediction, an observation process (as in conventional EKF) and an augmentation state. The prediction phase deals with robot motion based on the incremental dead reckoning estimates, and it also increases the uncertainty of the robot pose estimate. The update stage occurs with the re-observation of already stored landmarks, improving the overall state estimate. When a landmark is detected for the first time, however, it is added to the state vector through an initialization process called state augmentation. The AEKF-SLAM based robotic underwater navigation algorithm emphasizes the identification of each sensor perception as a new landmark or a re-visited observed one. It can distinguish new landmarks from those already detected ones. With the assistance of the presented AEKF approach, the underwater robot achieves a more accurate and robust self-localization and mapping of the surrounding landmarks. Several MATLAB simulated experiments for both the proposed AEKF-SLAM based navigation algorithm and the classic FastSLAM 2.0 approach for both dense loop mapping and line mapping are performed, experimental results prove that the AEKF-SLAM approach has the best performances of localization and mapping accuracy with relatively low computational cost. Moreover, during navigation, the robot localizes the centroids of different segments of features in recorded sonar images, which are detected by our improved Otsu TSM, as point landmarks. Using them with the AEKF achieves more robust and accurate estimations of the robot pose and the landmark positions, than with those obtained by the conventional maximum entropy TSM. Although the maximum entropy TSM achieves the highest segmentation precision over other three classic segmentation methods, namely the Traditional Otsu TSM, the Local TSM and the Iterative TSM compared in Section 3, it has much lower precision than our improved Otsu TSM, and it also needs much more computational time. As a conclusion, together with the landmarks identified by the proposed segmentation approach, our presented AEKF-based

robotic underwater SLAM algorithm achieves reliable detection of cycles in the map and consistent map update on loop closure.

7.2. Future work

Based upon the conclusions above and considering the limitations of the work existed, further enhancements of our research can be carried out in the following areas:

- (1) Employing other forms of target objects for the detection and tracking purpose, devising parametric feature models for describing general objects, and more complex scenarios with multiple distinct features will also be included.
- (2) Investigating more complicated vehicle model, such as 6-DOF kinematic model. Therefore, as the underwater robot navigates, we can perform the proposed feature detection algorithm on the acquired images exactly when the 3D object is detected by the sonar.
- (3) Establishing a computationally tractable robotic underwater SLAM based navigation algorithm, which maintains the properties of being consistent and nondivergent. The use of the full map covariance matrix at each stage in the underwater map generating process will lead to substantial computational problems. Hierarchical SLAM or sub-mapping methods build local maps of limited size, which bound the covariances and thus the linearization errors. Afterwards, by incorporating the local maps into a global map or a hierarchy of global maps, the AEKF-SLAM based robotic navigation applications in large scenarios can be possible.
- (4) Studying other state-of-the-art approaches to the underwater SLAM problem for suitability, such as the graph-based SLAM and optimization-based SLAM. Afterwards, comparing their localization and mapping performances in terms of accuracy, robustness, computational complexity, etc. with those of the proposed AEKF-SLAM based robotic navigation algorithm.
- (5) Acquiring only one map of some part of the regions may not depict the global topology of the whole surveying area correctly, on account of the imaging geometry of the mapping devices. Incorporating data measurements derived from the sonars and cameras to generate a practical 3D subsea map, such as seafloor, exploring environment and artefacts. In our case, we will use the depth logger based on pressure for navigation and the DeepEye680D SSS as perception sensor to get horizontal positions of features of interest, therefore by integrating with the vertical positioning data obtained through pressure sensor, an underwater 3D map could be created.

- (6) Simplifying and fusing two different resolution sonar maps by transform based algorithms. Employing the large-scale medium-resolution map to trigger detailed investigations of regions of interest in the local high-resolution maps.

References

- [1] Warren, M.; Coke, P.; Pizarro, O.; Williams, S.; Upcroft, B. Visual sea-floor mapping from low overlap imagery using bi-objective bundle adjustment and constrained motion. In Proceedings of the 2012 Australasian Conference on Robotics and Automation, Wellington, New Zealand, 3–5 December 2012; pp. 1–10.
- [2] Hogue, A.; German, A.; Jenkin, M. Underwater environment reconstruction using stereo and inertial data. In Proceedings of the 2007 International Conference on Systems, Man and Cybernetics (ISIC), Montreal, Canada, 3–5 October 2007; pp. 2372–2377.
- [3] Clark, C. M.; Olstad, C. S.; Buhagiar, K.; Gambin, T. Archaeology via underwater robots: Mapping and localization within maltese cistern systems. In Proceedings of the 2018 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), Hanoi, Vietnam, 17–20 December 2008; pp. 662–667.
- [4] Yoerger, D. R.; Bradley, A. M.; Walden, B. B.; W.; Singh, H.; Bachmayer, R. Surveying a subsea lava flow using the autonomous benthic explorer (ABE). *Int. J. Syst. Sci.* **1998**, *29*, 1031–1044.
- [5] Zhou, M.; Bachmayer, R. Working towards single beam acoustic iceberg profiling using active roll control on a slocum glider. In Proceedings of the 2001 IEEE Symposium on Underwater Technology (UT) and 2011 Workshop on Scientific Use of Submarine Cables and Related Technologies (SSC), Tokyo, Japan, 3–5 April 2011; pp. 1–7.
- [6] Thrun, S. Simultaneous localization and mapping. In *Robotics and Cognitive Approaches to Spatial Mapping*; Jefferies, M.E., Ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 13–41.
- [7] Guth, F.; Silveira, L. Underwater SLAM: Challenges, state of the art, algorithms and a new biologically-inspired approach. In Proceedings of the 2014 5th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), São Paulo, Brazil, 12–15 August 2014; pp. 981–986.
- [8] Yuan, X.; José-Fernán, M.; Martina, E.; Lourdes, L.-S. An improved Otsu threshold segmentation method for underwater simultaneous localization and mapping-based navigation. *Sensors* **2016**, *16*, 1148 (1–31).
- [9] Yuan, X.; José-Fernán, M.; José Antonio, S. F.; Martina, E. AEKF-SLAM: a new algorithm for robotic underwater navigation. *Sensors* **2017**, *17*, 1174 (1–30).
- [10] Yuan, X.; José-Fernán, M.; Juana, S. S.; Martina, E. AEKF-SLAM: a new algorithm for robotic underwater localization and landmark mapping. *Auton. Robot.* (April **2018**, Accepted for publication).
- [11] Yuan, X.; José-Fernán, M.; Li, N. Speeded up robust feature (SURF) matching method and underwater wireless sensor network-based Delaunay triangulation (UWSN-DT) algorithm for sonar map fusion. *IEEE Access* (**2018**, Under the second round review).

- [12] Al-Rawi, M.S.; Adrián, G.; Yuan, X.; Martina, E.; José-Fernán, M.; Fredrik, E.; Baran, C.; Jonathan, R.; Joaquin, B.; Marc, P. Intensity normalization of sidescan sonar imagery. In Proceedings of the 2016 6th International Conference on Image Processing Theory, Tools, and Applications (IPTA), Oulu, Finland, 12–15 December 2016; pp. 1–6.
- [13] Al-Rawi, M.S.; Fredrik, E.; Mirgita, F.; Baran, C.; Yuan, X.; José-Fernán, M.; Joaquin, B.; Jonathan, R.; Marc, P. Algorithms for the detection of first bottom returns and objects in the water column in sidescan sonar images. In Proceedings of the 2017 Oceans' 17 Marine Technology Society (MTS)/IEEE Oceanic Engineering Society (OES), Aberdeen, Scotland, 19–22 June 2017; pp. 1–5.
- [14] Vinod, C.; Steve, E.; Anthony, N. Detection of mines in acoustic images using higher order spectral features. *IEEE J. Ocean. Eng.* **2002**, *27*, 610–618.
- [15] Edward, T.; James, R.; Daniel, T. Automated optimisation of simultaneous multibeam and sidescan sonar seabed mapping. In Proceedings of the 2007 IEEE Conference on Oceans-Europe, Scotland, UK, 18–21 June 2007; pp. 1–6.
- [16] Allotta, B.; Costanzi, R.; Ridolfi, A.; Pascali, M.A.; Reggiannini, M.; Salvetti, O.; Sharvit, J. Acoustic data analysis for underwater archaeological sites detection and mapping by means of autonomous underwater vehicles. In Proceedings of the Oceans 2015 Genova, Genova, Italy, 18–21 May 2015; pp. 1–6.
- [17] Zeyneb, K.Y.; Sirma, Y. A comparison of EKF, UKF, FastSLAM 2.0 and UKF-based FastSLAM algorithms. In Proceedings of the 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), Lisbon, Portugal, 13–15 June 2012; pp. 37–43.
- [18] Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; The MIT Press: Cambridge, MA, USA, 2005; pp. 1–480.
- [19] Zeyneb, K.Y.; Sirma, Y. Improvement of the measurement update step of EKF-SLAM. In Proceedings of the 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), Lisbon, Portugal, 13–15 June 2012; pp. 61–65.
- [20] Thrun, S. Probabilistic algorithms in robotics. *AI Magazine* **2000**, *21*, 93–109.
- [21] Dissanayake, G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241.
- [22] Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In Proceedings of the 2003 18th International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 9–15 August 2003; pp. 1151–1156.
- [23] Montello, D. R. Navigation. In *The Cambridge Handbook of visuospatial thinking*; Shah, P., Miyake, A., Eds. The Cambridge University Press: New York, NY, USA, 2005; PP. 257–294.
- [24] Levis, T. S.; Lawton, D. T. Qualitative navigation for mobile robots. *Artif. Intell.* **1990**, *44*, 305–360.

- [25] Wolter, D.; Richter, K.-F. Schematized maps for robot guidance. In Proceedings of the 2004 16th European Conference on Artificial Intelligence (ECAI) - 4th International Workshop on Cognitive Robotics (CogRob), Valencia, Spain, 23–24 August 2004; pp. 71–76.
- [26] Eustice, R. M.; Whitcomb, L. L.; Singh, H.; Grund, M. Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In Proceedings of the 2007 7th IEEE International Conference on Robotics and Automation (ICRA), Rome, Italy, 10–14 April 2007; pp. 4257–4264.
- [27] Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I the essential algorithms. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110.
- [28] Bailey, T. Mobile robot localization and mapping in extensive outdoor environments. Ph.D. Thesis, Australian Center for Field Robotics, University of Sydney, Sydney, Australia, August 2002.
- [29] Zelinsky, W. *The Cultural Geography of the United States*, 2nd ed.; Prentice-Hall: Englewood Cliffs, NJ, USA, 1992.
- [30] Elfes, A. Occupancy grids: a probabilistic framework for robot perception and navigation. Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, PA, USA, 1989.
- [31] Moravec, H. P. Sensor fusion in certainty grids for mobile robots. *AI Magazine* **1988**, *9*, 61–74.
- [32] Hähnel, D. Mapping with mobile robots. Ph.D. Thesis, Department of Applied Science, University of Freiburg, Baden-Württemberg, Germany, December 2004.
- [33] Thrun, S.; Beetz, M.; Bennewitz, M.; Burgard, W.; Cremers, A.B.; Dellaert, F.; Fox, D.; Hähnel, D.; Rosenberg, C.; Roy, N.; Schulte, J.; Schulz, D. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *Int. J. Robot. Res.*, **2000**, *19*, 972–999.
- [34] Chatila, R.; Laumond, J.-P. Position referencing and consistent world modeling for mobile robots. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation (ICRA), St. Louis, MO, USA, 25–28 March 1985; pp. 138–145.
- [35] Mataric, M. J. A distributed model for mobile robot environment-learning and navigation. Master Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, MA, USA, January 1990.
- [36] Benjamin, K.; Yung-Tai, B. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *J. Robot. Auton. Syst.* **1991**, *8*, 47–63.
- [37] Smith, R.; Self, M.; Cheeseman, P. Estimating uncertain spatial relationships in robotics. In *Autonomous Robot Vehicles*; Ingemar, J.C., Gordon, T.W., Eds.; Springer: New York, NY, USA, 1990; pp. 167–193.
- [38] Smith, R.C.; Cheeseman, P. On the representation and estimation of spatial uncertainty. *Int. J. Robot. Res.* **1987**, *5*, 56–68.
- [39] Dissanayake, G.; Durrant-Whyte, H.; Bailey, T. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In Proceedings of the 2000 IEEE

- International Conference on Robotics & Automation, San Francisco, CA, USA, 24–28 April 2000; pp. 1009–1014.
- [40] Durrant-Whyte, H.; Majumder, S.; Thrun, S.; Battista, M.; Scheduling, S. A Bayesian algorithm for simultaneous localization and map building. In Proceedings of the 10th International Symposium of Robotics Research (ISRR'01), Lorne, Australia, 9–11 November 2001.
 - [41] Hidalgo, F.; Bräunl, T. Review of underwater SLAM techniques. In Proceedings of the 6th International Conference on Automation, Robotics and Applications, Queenstown, The New Zealand, 17–19 February 2015; pp. 305–311.
 - [42] Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II the state of the art. *IEEE Robot. Autom. Mag.* **2016**, *13*, 108–117.
 - [43] Mallios, A.; Ridao, P.; Ribas, D.; Maurelli, F.; Petillot, Y. EKF-SLAM for AUV navigation under probabilistic sonar scan-matching. In Proceedings of the 2010 IEEE /RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4404–4411.
 - [44] Dempster, A.P.; Laird, A.N.; Rubin, D.B. Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. Ser. B-Stat. Methodol.* **1997**, *39*, 1–38.
 - [45] McLachlan, G.J.; Krishnan, T. *The EM Algorithm and Extensions*. Wiley Series in Probability and Statistics: NY, USA, 1997; PP. 1–369.
 - [46] Lingemann, K.: Schnelles Pose-Tracking auf Laserscan-Daten für autonome mobile Roboter. Master Thesis, Department of Information and Communication Technology, University of Bonn, North Rhine-Westphalia, Germany, July 2004.
 - [47] Zeyneb, K.Y.; Sirma, Y. A comparison of EKF, UKF, FastSLAM 2.0 and UKF-based FastSLAM algorithms. In Proceedings of the 2012 IEEE 16th International Conference on Intelligent Engineering Systems (INES), Lisbon, Portugal, 13–15 June 2012; pp. 37–43.
 - [48] Durrant-Whyte, H.F. Uncertain geometry in robotics. *IEEE J. Robot. Autom.* **1988**, *4*, 23–31.
 - [49] Ayache, N.; Faugeras, O. Maintaining representations of the environment of a mobile robot. *IEEE Trans. Robot. Autom.* **1989**, *5*, 804–819.
 - [50] Stachniss, C. Class lecture: Robot mapping-WS 2013/14 short summary. In *Autonomy Intelligent System*; University of Freiburg: Freiburg, Germany, 2013; Available online: <http://ais.informatik.uni-freiburg.de/teaching/ws13/mapping/> (accessed on 28 June 2018).
 - [51] Paz, L. M.; Guivant, J.; Tardos, J. D.; Neira, J. Divide and conquer: EKF SLAM in $O(n)$. *IEEE Trans. Robot.*, **2008**, *24*, 1107–1120.
 - [52] Julier, S. J. A sparse weight Kalman filter approach to simultaneous localisation and map building. In Proceedings the 2011 IEEE/RJS International Conference on Intelligent Robots and Systems, HI, USA, 29 October–3 November 2011; pp. 1251–1256.
 - [53] Tardos, J. D.; Neira, J.; Newman, P. M.; Leonard, J. J. Robust mapping and localization in indoor environments using sonar data. *Int. J. Robot. Res.* **2002**, *21*, 311–330.

- [54] Folkesson, J., Christensen, H.I.: Outdoor exploration and slam using a compressed filter. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA), Taipei, Taiwan, 14–19 September 2003; pp. 419–427.
- [55] Williams, S.B.; Dissanayake, G.; Durrant-Whyte, H. An efficient approach to the simultaneous localisation and mapping problem. In Proceedings of the 2002 2nd IEEE International Conference on Robotics and Automation (ICRA), DC, USA, 11–15 May 2002; pp. 406–411.
- [56] Eduardo Nebot, J. G.; Nieto, J.; Masson, F. Navigation and mapping in large unstructured environments. *Int. J. Robot. Res.* **2004**, *23*, 449–472.
- [57] Nieto, J. I.; Guivant, J. E.; Nebot, E. M. The hybrid metric maps (HYMMs): A novel map representation for dense SLAM. In Proceedings of the 2004 4th IEEE International Conference on Robotics and Automation (ICRA), LA, USA, 26 April –1 May 2004; pp. 391–396.
- [58] Burgard, W.; Fox, D.; Jans, H.; Matenar, C.; Thrun, S. Sonar-based mapping of large-scale mobile robot environments using EM. In Proceedings of the 1999 16th International Conference on Machine Learning (ICML), Bled, Slovenia, 27–30 June 1999; pp. 1–10.
- [59] Chen, Z. H.; Samarabandu J.; Rodrigo, R. Recent advances in simultaneous localization and map-building using computer vision. *Adv. Robot.* **2012**, *21*, 233–265.
- [60] Montemerlo, M.; Thrun S. *FastSLAM: A Scalable method for the simultaneous localization and mapping problem in robotics*, Springer: Berlin/Heidelberg, Germany, 2007, pp. 1–109.
- [61] Davide, S.; Nicolas, C.; Agostino, M.; Roland, S. A robust descriptor for tracking vertical lines in omnidirectional images and its use in mobile robotics. *Int. J. Robot. Res.* **2009**, *28*, 149–171.
- [62] Allotta, B.; Caiti, A.; Costanzi, R.; Fanelli, F.; Fenucci, D.; Meli, E.; Ridolfi, A. A new AUV navigation system exploiting unscented Kalman filter. *Ocean Eng.* **2016**, *113*, 121–132.
- [63] Kohlbrecher, S.; Meyer, J.; Petresen, K.; Graber, T. Hector SLAM for robust mapping in USAR environments. ROS RoboCup Rescue Summer School Graz 2012. Available online: http://tedusar.eu/cms/sites/tedusar.eu.cms/files/Hector_SLAM_USAR_Kohlbrecher_RRSS_Graz_2012.pdf (accessed on 28 June 2018).
- [64] Balasuriya, B.L.E.A.; Chathuranga, B.A.H.; Jayasundara, B.H.M.D.; Napagoda, N.R.A.C.; Kumarawadu, S.P.; Chandima, D.P.; Jayasekara, A.G.B.P. Outdoor robot navigation using Gmapping based SLAM algorithm. In Proceedings of the 2016 IEEE Moratuwa Engineering Research Conference (MERCon), Moratuwa, Sri Lanka, 5–6 April 2016; pp. 403–408.
- [65] Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In Proceedings of the 2002 American Association for Artificial Intelligence (AAAI-02), Edmonton, AB, Canada, 28 July–1 August 2002; pp. 593–598.
- [66] Neira, J.; Tardos, J. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robot. Autom.* **2001**, *17*, 890–897.
- [67] Thrun, S. Robotic Mapping: a survey. In *Exploring artificial intelligence in the new millennium*. Morgan Kaufmann Publisher: CA, USA, 2003; pp. 1–35.

- [68] Wijk, O.; Jensfelt, P.; Christensen, H.I. Triangulation based fusion of ultrasonic sensor data. In Proceedings of the 1998 IEEE International Conference on Robotics and Automation, Lueven, Belgium, 16–20 May 1998; pp. 3419–3424.
- [69] Hans, M. Sensor fusion in certainty grids for mobile robots. *AI Mag.* **1988**, *9*, 61–74.
- [70] Hahne, D. Mapping with Mobile Robots. Ph.D. Thesis, University of Freiburg, Freiburg, Germany, December 2004.
- [71] Siegwart, R.; Nourbakhsh, I. R. *Introduction to Autonomous Mobile Robot*. The MIT Press: Cambridge, MA, USA, 2004.
- [72] Guivant, J.; Nebot, E.; Durrant-Whyte, H. Simultaneous localization and map building using natural features in outdoor environments. *Intell. Auton. Syst.* **2000**, *6*, 581–586.
- [73] Royer, E. Cartographie 3D Et Localisation Par Vision Monoculaire Pour La Navigation Autonome D'un Robot Mobile. Ph.D. Thesis, Université Blaise Pascal-Clermont II, Aubière, France, September 2006.
- [74] Gil, A.; Reinoso, O.; Ballesta, M.; Julia, M. Multi-robot visual SLAM using a Rao-Blackwellized particle filter. *Robot. Auton. Syst.* **2010**, *58*, 68–80.
- [75] Blackman, S.; Popoli, R. *Design and Analysis of Modern Tracking Systems*; Artech House: Boston, MA, USA, 1999.
- [76] Williams, S.B. Efficient solutions to autonomous mapping and navigation problems. Ph.D. Thesis, Australian Center for Field Robotics, University of Sydney, Sydney, Australia, September 2001.
- [77] Petillot, Y.; Tena Ruiz, I.; Lane, D.M. Underwater vehicle obstacle avoidance and path planning using a multi-beam forward looking sonar. *IEEE J. Ocean. Eng.* **2001**, *26*, 240–250.
- [78] Wang, X.; Wang, H.; Ye, X.; Zhao, L.; Wang, K. A novel segmentation algorithm for side-scan sonar imagery with multi-object. In Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics(ROBIO), Sanya, China, 15–18 December 2007; pp. 2110–2114.
- [79] Cervenka, P.; Moustier, C. Sidescan sonar image processing techniques. *IEEE J. Ocean. Eng.* **1993**, *18*, 108–122.
- [80] Ribas, D.; Ridao, P.; Tardos, J.; Neira, J. Underwater slam in manmade structured environments. *J. Field Robot.* **2008**, *25*, no. 11-12, 898–921.
- [81] *Deep Vision AB Company*. Available online: <http://deepvision.se/> (accessed on 28 June 2018).
- [82] *ECA Group Company*. Available online: <http://www.ecagroup.com/en/defence-security> (accessed on 28 June 2018).
- [83] Otsu, N. A threshold selection method from gray-level histogram. *IEEE Trans. SMC* **1979**, *9*, 62–66.
- [84] Abber, G.G. Contour Tracing Algorithms. Available online: http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/alg.html (accessed on 28 June 2018).
- [85] Kang, L.; Zhong, S.; Wang, F. A new contour tracing method in a binary image. In Proceedings of the 2011 International Conference on Multimedia Technology (ICMT), Hangzhou, China, 26–28 July 2011; pp. 6183–6186.

- [86] Zhang, Y.; Wu, L. Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach. *Entropy* **2011**, *13*, 841–859.
- [87] Flandrin, P.; Rilling, F.; Goncalves, P. Empirical mode decomposition as a filter bank. *IEEE Signal Process. Lett.* **2004**, *11*, 112–114.
- [88] Liu, L.; Bian, H.; Yagi, S.I.; Yang, X. A configuration-conjunct threshold segmentation method of underwater linear object detection for forward-looking sonar. In Proceedings of the Symposium on Ultrasonic Electronics, Tsukuba, Japan, 5–7 November 2015.
- [89] Desistek Robotik Elektronik Yazilim Company. Available online: <http://www.desistek.com.tr/> (accessed on 28 June 2018).
- [90] Forsyth, D.A.; Ponce, J. Computer vision: a modern approach, 2nd ed.; Prentice-Hall Professional Technical Reference, 2002.
- [91] Hodges, R. P. *Underwater acoustics: analysis, design and performance of sonar*, Wiley Press: NJ, USA, 2010; pp. 1–353.
- [92] Chang, Y.-C.; Hsu, S.-K.; Tsai, C.-H. Sidescan Sonar image processing: correcting brightness variation and patching gaps. *J. Mar. Sci. Technol.-Taiwan*, **2010**, *18*, 785–789.
- [93] Capus, C. G.; Banks, A. C.; Coiras, E.; Ruiz, I. T.; Smith, C. J.; Petillot, Y. R. Data correction for visualisation and classification of sidescan SONAR imagery. *IET Radar Sonar Navig.* vol. **2008**, *2*, 155–169.
- [94] He, K.; Sun, J.; Tang, X. Single image haze removal using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2341–2353.
- [95] Tarel, J.-P.; Hautiere, N. Fast visibility restoration from a single color or gray level image. In Proceedings of the 2009 12th IEEE International Conference on Computer Vision. Kyoto, Japan, 29 September–2 October 2009; pp. 2201–2208.
- [96] Galdran, A.; Vazquez-Corral, J.; Pardo, D.; Bertalmio, M. Enhanced variational image dehazing. *SIAM J. Imaging Sci.* **2015**, *8*, 519–546.
- [97] Lu, Y.; Xie, F.; Wu, Y.; Jiang, Z.; Meng, R. No reference uneven illumination assessment for dermoscopy images. *IEEE Signal Process. Lett.* **2015**, *22*, 534–538.
- [98] Yves, D.; Cordelia, S.; Radu, H. Matching images with different resolutions. In Proceedings of the 2000 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CPVR), Hilton Head Island, SC, USA, 15–15 June 2000; pp.1–7.
- [99] Wang, Y.M. A consistent decision algorithm for N frames continuous scene matching. *Chin. J. of Comput.* **2005**, *28*, 1032–1035.
- [100] Harris, C.; Stephens, M. A combined corner and edge detector. In Proceedings of the Fourth Alvey Vision Conference, University of Manchester, United Kindom, 31 August–2 September 1988; pp. 147–151.
- [101] Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision.* **2004**, *60*, 91–110.

- [102] Krystian, M.; Cordelia, S. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615-1630.
- [103] Guo, S.Y.; Morel, J. M. ASIFT: an algorithm for fully affine invariant comparison. *Image Processing On Line.* **2011**, *1*, 11-38.
- [104] Stefan, L.; Matgarita, C.; Roland, Y.S. Binary Robust Invariant Scalable Keypoints. In Proceedings of the 2011 IEEE International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
- [105] Rao, M.; Kamila, N.K.; Kumar, K.V. Underwater wireless sensor network for tracking ships approaching harbor. In proceedings of the International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs), Paralakhemundi, India, 3-5 October 2016; pp. 1872-1876.
- [106] Li, N.; Curuklu, B.; Bastos, J.; Sucasas, V.; Fernandez, J.A.S.; Rodriguez, J. A probabilistic and highly efficient topology control algorithm for underwater cooperating AUV networks. *Sensors.* **2017**, *17*, 1-23.
- [107] Ayaz, B.; Allen, A.; Wiercigroch, M. Improving routing performance of underwater wireless sensor networks. In proceedings of the MTS/IEEE Oceans 2017, Aberdeen, United Kingdom, 19-22 June 2017; pp. 1-9.
- [108] Yadav, S.; Kumar, V. Optimal clustering in underwater wireless sensor networks: Acoustic, EM and FSO Communication Compliant Technique. *IEEE Access.* **2017**, *5*, 12761-12776.
- [109] Shen, Y.; Xu, Q.; Zhang, J.; Zhang, R.; Li, J. Target tracking with energy efficiency using robotic fish-based sensor networks. In proceedings of the 13th International Conference on Embedded Software and Systems (ICESS), Chengdu, China, 13-14 August 2016; pp. 24-29.
- [110] Maqsood, H.; Javaid, N.; Yahya, A.; Ali, B.; Khan, Z.A.; Qasim, U. MobiL-AUV: AUV-aided localization scheme for underwater wireless sensor networks. In proceedings of the 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Fukuoka, Japan, 6-8 July 2016; pp. 170-175.
- [111] Kang, J.; Jin, R.; Li, X. Regression kriging-based upscaling soil moisture measurements from a wireless sensor network and multiresource remote sensing information over heterogeneous cropland. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 92-96.
- [112] Dasgupta, Y.; Darshan, P.M.G. Application of wireless sensor network in remote monitoring: water-level sensing and temperature sensing, and their application in agriculture. In proceedings of the first International Conference on Automation, Control, Energy and Systems (ACES), Hooghly, India, 1-2 February 2014; pp. 1-3.
- [113] Kim, Y.; Evans, R.G.; Iversen, W.M. Remote sensing and control of an irrigation system using a distributed wireless sensor network. *IEEE Trans. Instrum. Mea.* **2008**, *57*, 1379-1387.
- [114] Freeman, W. T.; Adelson, E.H. The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 891-906.

- [115] Delaunay, B. Sur la sphère vide. A la mémoire de Georges Voronoï, Bulletin de l'Académie des Sciences de l'URSS, classe des sciences mathématiques et naturelles, 1934; Issue 6, pp. 793–800.
- [116] Leonard, J.; Durrant-Whyte, H. Directed sonar sensing for mobile robot navigation. In *The Springer International Series in Engineering and Computer Science*; Springer: Boston, MA, USA, 1992.
- [117] Menegatti, E.; Zanella, A.; Zilli, S.; Zorzi, F.; Pagello, E. Range-only slam with a mobile robot and a wireless sensor networks. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 8–14.
- [118] Aulinas, J.; Lladó, X.; Salvi, J.; Petillot, Y.R. Selective submap joining for underwater large scale 6-DOF SLAM. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2552–2557.
- [119] Ribas, D.; Ridao, P.; Tardós, J.D.; Neira, J. Underwater SLAM in a marina environment. In Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November, 2007; pp. 1455–1460.
- [120] Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV navigation and localization: a review. *IEEE J. Ocean. Eng.* **2014**, *39*, 131–149.
- [121] Mahon, I.; Williams, S. B.; Pizarro, O.; Johnson-Roberson, M. Efficient view-based SLAM using visual loop closures. *IEEE Trans. Robot.* **2008**, *24*, 1002–1014.
- [122] Gustafsson, F. Particle filter theory and practice with positioning applications. *IEEE Aerosp. Electron. Syst. Mag.* **2010**, *25*, 53–82.
- [123] Hiebert-Treuer, B. An introduction to robot SLAM (Simultaneous Localization And Mapping). Bachelor Thesis, Computer Science, 2007.
- [124] Folkesson, J.; Christensen, H.I. Closing the loop with graphical slam. *IEEE Trans. Robot.* **2007**, *23*, 731–741.
- [125] Moreno, L.; Garrido, S.; Blanco, D.; Muñoz, M.L. Differential evolution solution to the slam problem. *Robot. Auton. Syst.* **2009**, *57*, 441–450.
- [126] Williams, B.; Cummins, M.; Neira, J.; Newman, P.M.; Reid, I.D.; Tardos, J.D. A comparison of loop closing techniques in monocular slam. *Robot. Auton. Syst.* **2009**, *57*, 1188–1197.
- [127] Csorba, M. Simultaneous localisation and map Building. Ph.D. Thesis, University of Oxford, Oxford, UK, 1997.
- [128] Wang, H.J.; Wang, J.; Qu, L.P.; Liu, Z.Y. Simultaneous localization and mapping based on multilevel-EKF. In Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation, Beijing, China, 7–10 August, 2011; pp. 2254–2258.
- [129] Estrada, C.; Neira, J.; Tardos, J.D. Hierarchical SLAM: Real-time accurate mapping of large environments. *IEEE Trans. Robot.* **2005**, *21*, 588–596.
- [130] Kang, J.G.; Choi, W.S.; An, S.Y.; Oh, S.Y. Augmented EKF based SLAM method for improving the accuracy of the feature map. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3725–3731.

-
- [131] Ribas, D. Towards simultaneous localization & mapping for an AUV using an imaging sonar. Ph.D. Thesis, University de Girona, Girona, Spain, June 2005.
 - [132] Ixion Industry & Aerospace SA Company. Available online: <http://ixion.es/en/> (accessed on 28 June 2018).
 - [133] Smart and Networking Underwater Robots in Cooperation Meshes (SWARMS) project. Available online: <http://www.swarms.eu/> (accessed on 28 June 2018).
 - [134] Norgren, P.; Skjente, R. A multibeam-based SLAM algorithm for iceberg mapping using AUVs. *IEEE Access* **2018**, *6*, 26318–26337.
 - [135] Clayton Gregory, K. Autonomous underwater vehicle navigation and mapping in dynamic, unstructured environments. Ph.D. Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, MA, USA, July 2012.
 - [136] Li, J.; Kaess, M.; Eustice, R. M.; Johnson-Roberson, M. Pose-Graph SLAM using forward-looking sonar. *IEEE Robot. Autom. Lett.* **2018**, *3*, 2330–2337.
 - [137] Melo, J.; Matos, A. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Eng.* **2017**, *139*, 250–264.
 - [138] Jung, J.; Li, J.-H.; Choi, H.-T.; Myung, H. Localization of AUVs using visual information of underwater structures and artificial landmarks. *Intell. Serv. Robot.* **2017**, *10*, 67–76.

List of Author's Publications and Projects

Most of the results presented in this thesis were published in the following journals indexed in Journal Citation Reports (JCR) and international conferences:

- [1] **Xin Yuan**, José-Fernán Martínez, Martina Eckert and Lourdes López-Santidrián. An improved Otsu threshold segmentation method for underwater simultaneous localization and mapping-based navigation. *Sensors* 16, 7 (July 2016), 1148. DOI: <https://doi.org/10.3390/s16071148>. (**JCR Q2, Impact Factor=2.677**).
- [2] **Xin Yuan**, José-Fernán Martínez-Ortega, José Antonio Sánchez Fernández and Martina Eckert. AEKF-SLAM: a new algorithm for robotic underwater navigation. *Sensors* 17, 5 (May 2017), 1174. DOI: <https://doi.org/10.3390/s17051174>. (**JCR Q1, Impact Factor=2.475**).
- [3] **Xin Yuan**, José-Fernán Martínez, Juana Sendra Pons and Martina Eckert. AEKF-SLAM: a new algorithm for robotic underwater localization and landmark mapping. *Auton. Robot.*, In Press (Accepted on 2nd April 2018). (**JCR Q2, Impact Factor=2.244**).
- [4] **Xin Yuan**, José-Fernán Martínez, Ning Li. Speeded up robust feature (SURF) matching method and underwater wireless sensor network-based Delaunay triangulation (UWSN-DT) algorithm for sonar map fusion. *IEEE Access*, Accepted with change (2018). (**JCR Q1, Impact Factor=3.557**).
- [5] MS Al-Rawi, Adrián Galdrán, **Xin Yuan**, Martina Eckert, José-Fernán Martínez, Fredrik Elmgren, Baran Cürüklü, Jonathan Rodriguez, Joaquim Bastos and Marc Pinto. Intensity Normalization of Sidescan Sonar Imagery. In *Proceedings of the 2016 6th International Conference on Image Processing Theory, Tools, and Applications (IPTA)*, 12–15 December 2016, Oulu, Finland, DOI: <https://10.1109/IPTA.2016.7820967>.
- [6] Mohammed Al-Rawi, Fredrik Elmgren, Mirgita Frasheri, Baran Cürüklü, **Xin Yuan**, José-Fernán Martínez, Joaquim Bastos, Jonathan Rodriguez and Marc Pinto. Algorithms for the Detection of First Bottom Returns and Objects in the Water Column in Sidescan Sonar Images. In *Proceedings of the 2017 Oceans' 17 Marine Technology Society (MTS)/ IEEE Oceanic Engineering Society (OES)*, 19–22 June 2017, Aberdeen, Scotland, DOI: <https://10.1109/OCEANSE.2017.8084587>.

Participation in European research projects:

- SWARMs–Smart and Networking Underwater Robots in Cooperation Meshes, under Grant Agreement n. 662107-SWARMs-ECSEL-2014-1, partially supported by the ECSEL JU and the Spanish Ministry of Economy and Competitiveness (Ref: PCIN-2014-022-C02-02).