

MAPPING AND VISUALIZING ANCIENT WATER STORAGE SYSTEMS
WITH AN ROV – AN APPROACH BASED ON FUSING STATIONARY
SCANS WITHIN A PARTICLE FILTER

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

William McVicker

December 2012

© 2012

William McVicker

ALL RIGHTS RESERVED

COMMITTEE MEMBERSHIP

TITLE: Mapping and Visualizing Ancient Water Storage Systems with an ROV – An Approach Based on Fusing Stationary Scans within a Particle Filter

AUTHOR: William McVicker

DATE SUBMITTED: December 2012

COMMITTEE CHAIR: Dr. Christopher M. Clark

COMMITTEE MEMBER: Dr. Zoë Wood

COMMITTEE MEMBER: Dr. Christopher Lupo

ABSTRACT

Mapping and Visualizing Ancient Water Storage Systems with an ROV – An Approach Based on Fusing Stationary Scans within a Particle Filter

William McVicker

This paper presents a new method for constructing 2D maps of enclosed underwater structures using an underwater robot equipped with only a 2D scanning sonar, compass and depth sensor. In particular, no motion model or odometry is used. To accomplish this, a two step offline SLAM method is applied to a set of stationary sonar scans. In the first step, the change in position of the robot between each consecutive pair of stationary sonar scans is estimated using a particle filter. This set of pair wise relative scan positions is used to create an estimate of each scan's position within a global coordinate frame using a weighted least squares fit that optimizes consistency between the relative positions of the entire set of scans. In the second step of the method, scans and their estimated positions act as inputs to a mapping algorithm that constructs 2D octree-based evidence grid maps of the site.

This work is motivated by a multi-year archaeological project that aims to construct maps of ancient water storage systems, i.e. cisterns, on the islands of Malta and Gozo. Cisterns, wells, and water galleries within fortresses, churches and homes operated as water storage systems as far back as 2000 B.C. Using a Remotely Operated Vehicle (ROV) these water storage systems located around the islands were explored while collecting video, still images, sonar, depth, and compass measurements. Data gathered from 5 different expeditions has produced maps of over 90 sites. Presented are results from applying the new mapping method to both a swimming pool of known size and to several of the previously unexplored water storage systems.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0966608.

I would like to thank the following people for making this research possible and providing me with the necessary expertise along the way.

Dr. Chris Clark (Advisor)

Dr. Zoë Wood (Interim Advisor/Committee Member)

Dr. Jane Lehr (Global Engineering Advisor)

Dr. Timothy Gambin

Keith Buhagiar

Other Maltese and Sicily Correspondences

Jeffrey Forrester

International Computer Engineering Experience (ICEX) program

ICEX 2011 members

ICEX 2012 members

Dr. Chris Lupo (Committee Member)

and

several others

Table Of Contents

List of Tables	vii
List of Figures	viii
1 Introduction	1
2 Background	4
3 Map Generation Pipeline	8
3.1 Sonar Scan Retrieval	9
3.1.1 Data Collection	9
3.1.2 Data Processing	12
3.2 Octree-based Evidence Grid Generation	14
3.2.1 Pair Wise Scan Matching	14
3.2.2 Scan Localization	18
3.2.3 Map Generation	20
3.3 Map Visualization	22
4 Results	23
5 Conclusions and Future Works	30
5.1 Conclusions	30
5.2 Future Works	31
Bibliography	32

List of Tables

3.1	Pair Wise Scan Matching Algorithm	15
3.2	Particle Filter Robot Localization	16
3.3	Part of the Resample Particles Algorithm	19
3.4	Mapping Algorithm	21
4.1	Mean Dimension Differences (Grid vs. Mosaic)	29

List of Figures

1.1	The VideoRay Pro 3 GTO is an underwater micro-ROV with dimensions 36.8cm x 28.9cm x 21.6cm. It is equipped with depth and compass sensors along with a front and a rear video camera. A Tritech Micron scanning sonar was attached to the top of the ROV.	2
3.1	The pipeline created to generate 3D maps	9
3.2	Data Collection process. The top left pane demonstrates the surveying of the site. The keyhole cistern found at the Citadel on Gozo is seen in the right image. The bottom left image shows the typical set up of the equipment during ROV deployment.	10
3.3	This sketch was taken while collecting data at the Mdina Gatto Pardo Bistro cistern in Mdina, Malta. The numbered circles with arrows represent the estimated position of the ROV during sonar scan # facing in the direction of the arrow. Refer to Figure 4.4(g) for the generated map of this cistern.	11
3.4	A representation of the Tritech SeaSprite sonar head on the ROV and its corresponding scan plane.	12
3.5	The particles' weights are calculated by taking the difference between the distance to a wall in the robot's map and in the range data with respect to the robot	16
4.1	Pool at Cal Poly's LAIR	24
4.2	Mdina private home cistern pipeline walk-through. In (a), the manual sonar mosaic is shown. The auto-generated occupancy grid is shown (b). The 2.5D model and the surface extracted, textured model is shown in (c) and (d) respectively.	26

4.3	Reconstruction of the Gozo Citadel "Keyhole" cistern. In (a) and (b), the auto-generated occupancy grid and the visualized model are shown respectively.	27
4.4	Pair wise scan matching algorithm on the Gatto Pardo Bistro cistern. Parts (a)-(f) demonstrate each of the scans and where the particles converged. In (g) and (h), the auto-generated occupancy grid is shown as well as the manually crafted sonar mosaic respectively.	28

Chapter 1

Introduction

Underwater robots are used to explore harsh environments, dangerous caves, and underwater domains. Remotely Operated Vehicles (ROV) allow researchers to safely study these places remotely by capturing video, images, acoustic data, and measurements from underwater sensors. The field of underwater robotics in the past decade has made substantial progress in the areas of localization and mapping. Sonar technology has been the primary choice of equipment to assist in mapping underwater environments because sonar waves propagate through water better than light. Recently in [19], an ROV was used to inspect 1 km of an underwater tunnel operated by Electricité de France for reasons of availability, safety, accessibility, and diagnostic quality. Another example of underwater ROV research includes implementing vision systems for underwater applications, to support shared control, and 3D mapping [15].

More relevant, micro-sized ROVs have been developed to help improve maneuverability within tight passages for a variety of applications including the exploration of sensitive ecosystems that one may not want to disrupt [13]. Other environments ideal for micro-ROVs include ancient cisterns, wells, and water gal-



Figure 1.1: The VideoRay Pro 3 GTO is an underwater micro-ROV with dimensions 36.8cm x 28.9cm x 21.6cm. It is equipped with depth and compass sensors along with a front and a rear video camera. A Tritech Micron scanning sonar was attached to the top of the ROV.

leries. The water storage systems found in Malta are difficult to access due to their size (e.g. a typical opening diameter of 0.3 m). To explore such environments, a VideoRay Pro 3 ROV equipped with an underwater scanning sonar head, depth sensor, and two video cameras, (seen in Fig. 1.1), was used to generate 2D and 3D sonar based maps. This paper proposes a new mapping and localization technique used to reconstruct the explored cisterns as well as document results from a recent expedition, (i.e. Spring of 2012).

Since the sites explored in this research have tunnels of limited size and accessibility, the ROV sensor payload must be minimal. Much research has focussed on developing complex motion models to decrease uncertainty, accurately model the ROV's motion, and improve localization. Instead of increasing the sensor payload of the ROV, this paper proposes a 2D localization algorithm to be run offline that does not require a motion model, but instead uses a particle filter to calculate relative translations between pairs of sonar scans that are used to construct a global map of the environment.

This paper is sectioned into five chapters. In chapter II, a brief background is provided on similar mapping and localization techniques. Chapter III explains the mapping reconstruction pipeline used to generate a map. In chapter IV, the results are presented followed by the conclusions and future works in chapter 5.

Chapter 2

Background

The recent advances of underwater robot sensing technology (e.g. sonar, imaging, Doppler velocity logging) have led to the ability to conduct Simultaneous Localization and Mapping (SLAM) [7] in the underwater domain [28]. In many SLAM algorithms, it is common to use robot odometry to predict the new robot position with respect to the map before using exteroceptive sensor measurements to correct the robot's position and update the map itself [4].

The traditional approaches to SLAM rely mainly on feature extraction from the robot's environment in order to correct for odometry errors as well as movement from external forces which are undetectable without inertial sensors. For example, Mahon and Williams [20] used an AUV equipped with a sonar and camera for data collection to map portions of the Great Barrier Reef. They developed a set of transient features to track short-term movement and a set of stable features to track landmarks. The developed feature-based algorithm was able to correct for the odometry errors and undetectable movement from the waves in the ocean which assisted in the reconstruction of the robot's natural underwater environment. Similarly, in [21] a combination of sonar scanners and a

vision system was used with their feature-based SLAM algorithm, which incorporated statistical models for filtering the data, on an ROV to assist in underwater navigation.

Recently, the analysis of raw measurements for mapping and localization has become of great interest. Analyzing raw measurements, as opposed to extracting geometric features from a map, is known as scan matching. A probabilistic algorithm called iterative-closest-points (ICP) has taken popularity among the robotics community in many different variations [14, 11, 22]. In [1], the sonar probabilistic model spIC was used to localize a mobile robot by analyzing raw sonar data to correct odometry errors for short robot trajectories. This helped minimize the displacement between noisy and sparse measurements.

Such applications have proven to work well in underwater environments. The core of the ICP algorithm matches two point clouds together in order to align the scans for map generation and/or vehicle localization. This algorithm has proven to work well with noisy data, but remains computationally heavy – $O(P^3)$ per iteration for P number of points. In [11], Fairfield and Wettergreen developed a variation to ICP and the Lucas-Kanade (LK) matching algorithm that is an efficient form of the *inverse compositional algorithm* called icLK to generate 3D maps of underground mines which reduced the complexity of ICP to $O(P^2)$ by thresholding the data; however, with 100 k points the approach remains burdensome.

Other recent research has focused on generating maps in real-time using a complex motion model in order to associate each sonar measurement with a corresponding location in the map. For example in [22], a pose-based algorithm was developed to map unstructured and unfamiliar environments using a probabilistic scan matching technique. The scan matching techniques that extract ranges from

sonar beams explained in [2, 3, 22] are most similar to the techniques developed in this paper.

Similarly in [10], a complex real-time mapping system was developed for exploration of underwater tunnels using octree evidence grids with a particle filter based SLAM approach similar to the algorithm presented in this paper. The main difference in their approach is the robot used, which benefitted by their full suite of underwater navigation sensors. Their robot has a diameter of 2m and a weight of 1.3 metric tons whereas the robot used for the research presented here has the dimensions 36.8cm x 28.9cm x 21.6cm. The basic algorithms developed in [10] are similar to the core scan matching techniques presented in this paper, but are executed in real-time with several feedback loops from the on-board navigation sensors. The expedition in [10] successfully created 3D maps of the Wakulla Springs cave system in Florida and the Sistema Zacaton in Mexico.

There has also been an increase in visual SLAM recently [9, 24, 25]. A technique called frameSLAM [18] uses bundle adjustment techniques to match point features along with stereo vision to track landmarks. The system developed was capable of autonomously navigating an off-road vehicle with only the use of stereo vision. Visual SLAM has proven to work well in terrestrial environments, but murky water (a common condition in cisterns) would likely decrease performance.

Similar to detecting frames and features, the incremental smoothing and mapping (iSAM) [16, 17] technique uses an informational filter to incrementally associate measurements in large-scale environments to solve the full SLAM problem. iSAM and frameSLAM are both feature-based, using landmarks such as trees to assist in localization, which differs from the scan matching solution based on raw measurements presented in this paper.

In the previous expeditions dedicated to mapping Maltese cisterns, [5, 6, 27], the mapping techniques included sonar mosaicking, and underwater robot fast-SLAM (presented in [26]) with both a stationary and moving robot. Sonar mosaics are images generated by piecing together different parts (scans) of the image to create a single image. This is a manual and time consuming job, but is able to successfully localized an ROV through manual calculations. For underwater robot SLAM, inadequate motion modeling led to reduced accuracy in robot localization and hence mapping [5]. To ensure highly accurate maps, the subsequent expeditions focussed on obtaining a series of stationary sonar scans from several positions in the tunnel [27].

The work reported here differs from the scan matching techniques developed in [2, 10, 22], in that our robot has a limited payload, and no motion model or odometry is used to predict the robot’s location with respect to a map being built. Instead, mapping is done offline. A particle filter is first used to calculate relative positions of the robot between consecutive pairs of stationary scans. Then, a weighted least squares approach uses these relative positions to calculate the absolute position of the robot for each individual sonar scan. To note, the algorithm improves the consistency of scan matching by considering how every tuple and every triplet of scans fit together. Finally, the scans and their estimated positions are used to assist the fastSLAM algorithm in converging to the most likely positions creating a 2D octree-based evidence grid map. A detailed description of this approach is presented in the following chapter.

Chapter 3

Map Generation Pipeline

In [5], it was proven difficult to construct maps using sonar data collected while moving. Localization uncertainty accumulated at a far higher rate than could be corrected with infrequent sonar measurements. This work attempts to accomplish the following: given a series of stationary sonar scans with corresponding depth and compass measurements, where each scan overlaps with at least the immediately following scan in the series, determine the locations of the ROV scans such that fastSLAM and visualizations can be accomplished with relative scan positions and scan data. fastSLAM, similar to [8], was used over traditional SLAM approaches because it doesn't require features.

The proposed solution uses the following stages to generate a map: Sonar Scan Retrieval, Pair Wise Scan Matching, Scan Localization, Map Generation, and Map Visualization. These stages are shown in Figure 3.1.

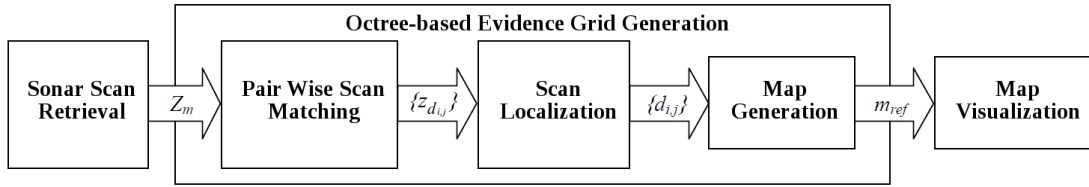


Figure 3.1: The pipeline created to generate 3D maps

3.1 Sonar Scan Retrieval

The range data used to generate the maps is created from sonar data. The sonar scanner uses a rotating mechanical head to sample its environment. A full 360 degree revolution of the sonar head returns of a set of 200 sonar beams. Each beam travels outwards from the robot where the window of time associated with capturing the return beam can be adjusted according to the beam range.

The sonar data is then filtered using image processing techniques to remove noise and multipaths commonly found in acoustic data. The resulting range data becomes the input data to the next stage of the map generation pipeline.

3.1.1 Data Collection

Data collection was performed using the VideoRay Pro 3 GTO seen in Figure 1.1. The VideoRay Pro 3 GTO is a professional remotely operated vehicle designed to assist in underwater exploration. The Pro 3 GTO model was designed for large payloads such as an external 2D imaging sonar. The dimensions of the Pro 3 GTO are 36.8cm x 28.9cm x 21.6cm (length, width, and height respectively) with a submersible weight of 4.5 kg. The internal sensors consist of a depth gauge, compass heading, a front, wide angle, color camera with a variable tilt, and a fixed, rear facing, black & white camera. To fly the ROV, two horizon-



Figure 3.2: Data Collection process. The top left pane demonstrates the surveying of the site. The keyhole cistern found at the Citadel on Gozo is seen in the right image. The bottom left image shows the typical set up of the equipment during ROV deployment.

tal propulsion thrusters, a single vertical thruster, two forward facing lights, and a rear LED array around the rear camera are used. The entire system includes a control box with the necessary remote controls which uses a 100 m tether to communicate with the ROV.

The process of collecting data begins by visually investigating the site and sketching the surroundings in order for one to uniquely identify each site from a top-down perspective. The ROV is then lowered into the cistern for exploration. Figure 3.2 demonstrates these two processes as well as how the equipment is set up during deployment.

In order to collect data in the cisterns, the depth of the water in the cistern must be over a 0.5 m in order to fully submerge the ROV with the attached sonar head underwater as well as get a clear sonar image. Once lowered in the cistern,

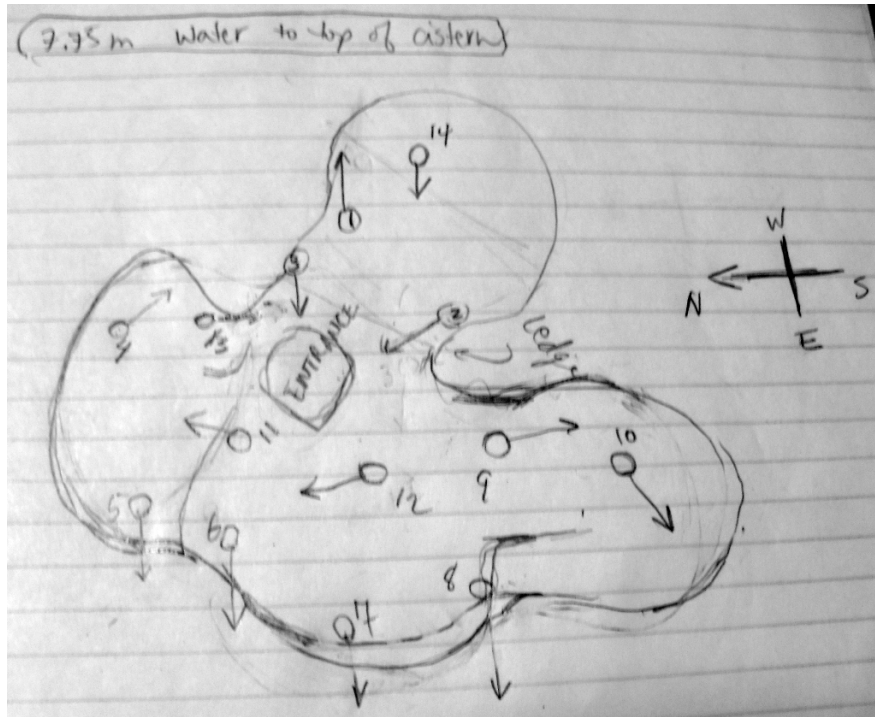


Figure 3.3: This sketch was taken while collecting data at the Mdina Gatto Pardo Bistro cistern in Mdina, Malta. The numbered circles with arrows represent the estimated position of the ROV during sonar scan # facing in the direction of the arrow. Refer to Figure 4.4(g) for the generated map of this cistern.

the ROV is driven around to create a sketch of the cistern's shape as seen in Figure 3.3. This sketch is used to manually indicate where each sonar scan was taken as well as a rough heading of the ROV for each scan. These sketches are used to verify the final map of the cistern as well as for manually mosaicking.

Using the sketched map, the driver can efficiently collect sonar data. The ROV data is collected as follows: land the ROV on the bottom or hover mid-depth without moving, log ROV depth and compass sensor measurements, initiate a sonar scan for one full revolution, move forward and repeat.

The collection of these measurements are defined as Z_m and collected at each time step t such that $Z_m = \{Z_m^t | t = 0 \dots T_{max}\}$ where Z_m^t represents all the sensor

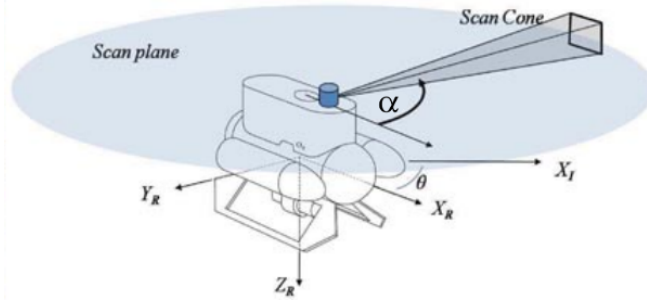


Figure 3.4: A representation of the Tritech SeaSprite sonar head on the ROV and its corresponding scan plane.

measurements collected at time t , i.e at a new scan location. These include robot yaw angle measurements z_θ , depth measurement z_d , and stationary sonar scans z_s .

$$Z_m^t = [z_\theta \ z_d \ z_s]^t \quad (3.1)$$

Each sonar scan z_s consists of a series of $j = 1 \dots A$ beam angles α_j , each with a corresponding vector of signal strengths $[ss_{j,i}]$ as seen in Figure 3.4. These signal strengths represent the signal strengths of the discretized sonar signal returned from a specific distance normal to the sensor referred to as the beam range that increases linearly with the value of $i = 1 \dots Num_Bins$ where Num_Bins is defined when configuring the sonar head. The most common beam range used was 5m.

$$z_s = \{[\alpha_j \ ss_{j,1} \ \dots \ ss_{j,Num_Bins}] \mid j = 1 \dots A\} \quad (3.2)$$

3.1.2 Data Processing

When collecting data with the Tritech scanning sonar head, there are several different settings that can be modified to enhance the quality of the sonar scans, e.g. adjust the signal strengths to reduce noise from debris in the water.

Additionally, signals will be stronger the closer the robot is to a wall due to the properties of the sonar wave. To compensate for these variations, the signal strengths need to be normalized to a standard range of values where a single wall threshold can be used for all the scans.

To normalize the signal strengths within the range $[0, 255]$, all the bins in the entire scan are iterated through, one beam at a time, to find the maximum signal strength ss_{max} . Then, all of the bins are divided by ss_{max} and multiplied by 255 (3.3). This allows for a more accurate comparison between scans with low returns and scans with high returns.

$$ss_{j,i_norm} = \frac{ss_{j,i}}{ss_{max}} \times 255 \quad (3.3)$$

At this stage, the sonar measurements may contain noise and multipaths which can negatively affect the particle filter, causing inaccurate alignments due to mistaking noise for a wall. A similar technique to beam segmentation in [3] was developed.

The signal strengths $ss_{j,i}$ are converted into range measurements $ss'_{j,i}$ by iterating through each beam of the scan and identifying the bin with the maximum signal strength that has at least one adjacent bin along the same beam with a signal strength value greater than zero. If the maximum intensity is above a set threshold (50 in our case), the bin's value along with the adjacent bins' values are set to 255 while all the other bins in the beam are set to 0. This approach removes most of the noise and allows for two scans with drastically different levels of signal strengths to be accurately compared.

3.2 Octree-based Evidence Grid Generation

Using evidence grids to represent an environment was first introduced in 1983 [23]. This approach has since been developed, modified, widely accepted, and proven to work well for many different environments with sonar measurements, including underwater. Evidence grids discretize space into an occupancy grid map $m(c_{xy})$ (3.4) where x and y are indices to the grid and the value $p_{xy} \in [0, 255]$ of the cell c_{xy} indicates the likelihood that c_{xy} contains a wall. This probability is calculated using the log-odds method [26] to logarithmically diffuse the filtered signal strengths $ss'_{j,i}$ (range data) into the corresponding cell c_{xy} (3.5) which can ultimately generate a highly accurate and detailed representation of one's surroundings.

$$m(c_{xy}) = \{c_{xy} \mid x \in [x_{min}, x_{max}], y \in [y_{min}, y_{max}]\} \quad (3.4)$$

$$L(p_{xy}, ss'_{j,i}) = \log\left(\frac{p_{xy}}{1 - p_{xy}}\right) + \log\left(\frac{ss'_{j,i}}{1 - ss'_{j,i}}\right) \quad (3.5)$$

$$p_{xy} = \frac{1}{1 + e^{-L(p_{xy}, ss'_{j,i})}} \quad (3.6)$$

This approach was adopted and used to generate a single map from a set of sonar scans.

3.2.1 Pair Wise Scan Matching

Pair Wise Scan Matching takes the sensor measurements Z_m as input and outputs the measured position translations $z_{d_{ij}}$ of the robot between each pair of stationary scans i and j . To note, these relative translation vectors are aligned with a global coordinate frame that has the X-axis aligned with true North.

The pseudo code for Pair Wise Scan Matching is shown in Table 3.1. To begin, the robot’s position during scan s_i is assumed to be at the origin of a Cartesian coordinate frame and map m_i is generated using an evidence grid (line 2 of Table 3.1) within this coordinate frame with the log-likelihood approach [26].

On line 4 of Table 3.1, the location of the robot during scan j with respect to map m_i is calculated using an implementation of Particle Filter Localization [26]. Note, when $i = 1$ line 4 of Table 3.1 iterates only on $j = i + 1$ and when $i = Num_Scans$ the same line only iterates on $j = i - 1$.

In this Particle Filter, as described in Table 3.2, a collection of $k = 1 \dots Num_Particles$ particles is used to represent the robot state during scan j . Each particle k consists of the robot’s state $[x^k y^k z^k \theta^k]$, and a weight w^k that indicates how likely particle k represents the true state. The particle position states are initially sampled randomly (line 1 of Table 3.2) from a square uniform distribution centered on the origin of m_i and within the range of the sonar beam range r_s .

The Particle Filter iterates for $Num_Iterations$ or until the particles converge. At each iteration of the algorithm, the x , y , and θ of the particle’s state are propagated (line 4 of Table 3.2) by adding a sample drawn from a zero mean Gaussian distribution of variance σ_p^2 . This added randomness models errors introduced by drift, sensor measurements, and driver control.

Table 3.1: Pair Wise Scan Matching Algorithm

Calc_Translation_Measurments(s, Num_Scans)	
1:	<i>for</i> $i = 1$ <i>to</i> Num_Scans <i>do</i>
2:	$m_i = \mathbf{construct_map}(s_i)$
3:	<i>for</i> $j = i-1$ <i>to</i> $i+1$ <i>do</i>
4:	$d_{ij_m} = \mathbf{PF_Localization}(m_i, s_j)$
5:	<i>endfor</i>
6:	<i>endfor</i>

Table 3.2: Particle Filter Robot Localization

PF Localization(m_i, s_j)	
1:	initialize_particle_states ($$)
2:	<i>for</i> $i = 1$ to $Num_Iterations$ <i>do</i>
3:	<i>for</i> $k = 1$ to $Num_Particles$ <i>do</i>
4:	$X_k = \mathbf{propagate_robot_state}(X_k)$
5:	$w_k = \mathbf{calculate_weight}(m_i, s_j)$
6:	<i>endfor</i>
7:	resample_particles ($$)
8:	<i>if</i> particles_converged ($$) <i>break</i>
9:	<i>endfor</i>
10:	$d_{ij} = \mathbf{calculate_translation}(d_m, w_\mu)$

To calculate particle weights (line 5 of Table 3.2), scan s_j 's range data is compared to the map m_i . Particle k 's weight w_{ij}^k is calculated as shown in (3.8) by calculating the difference between δ_s , the distance to a wall according to the scan s_j 's range measurement beam angle α_j , and δ_m calculated as the distance from particle k 's robot state in the direction of α_j to the nearest occupied cell c_p with a probability above τ_w in map m_i as demonstrated in Figure 3.5.

This difference $\delta_s - \delta_m$ is plugged into the Gaussian function $\varphi(\delta_s, \delta_m)$ (3.7)

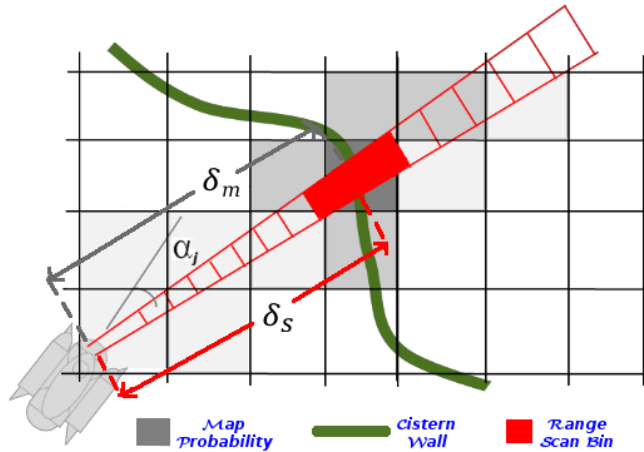


Figure 3.5: The particles' weights are calculated by taking the difference between the distance to a wall in the robot's map and in the range data with respect to the robot

where the standard deviation σ_c equals twice the map's cell size r_c with parameter a set to equate each of the Gaussian function's maximum values to one. If this difference is less than a set maximum distance, δ_{max} , then the particle's weight w_{ij}^k is equal to $\varphi(\delta_s, \delta_m)$; otherwise, w_{ij}^k is reduced by $1 - \varphi(\delta_s, \delta_m)$ (3.8) in order to include negative feedback.

$$\varphi(\delta_s, \delta_m) = \frac{1}{a\sqrt{2\pi}} e^{-\frac{(\delta_s - \delta_m)^2}{2\sigma_c^2}} \quad (3.7)$$

$$w_{ij}^k = \begin{cases} \varphi(\delta_s, \delta_m) & \text{if } \delta_s - \delta_m < \delta_{max} \\ w_{ij}^k - (1 - \varphi(\delta_s, \delta_m)) & \text{otherwise} \end{cases} \quad (3.8)$$

For every iteration, the algorithm creates a new set of particles (line 7 of Table 3.2) by randomly resampling from the current set of particles such that each particle has a probability of being selected proportional to its weight w_{ij}^k .

In order to determine if the particles have converged (line 8 of Table 3.2), an exponential average of the particles' weights $w_{ij-\mu}^t = \chi_w w_{ij}^t + (1 - \chi_w) w_{ij-\mu}^{t-1}$ and an exponential average of the standard deviation of the particles' states $\sigma_{p-\mu}^t = \chi_\sigma \sigma_p^t + (1 - \chi_\sigma) \sigma_{p-\mu}^{t-1}$ are calculated before the particles are resampled where $\chi_w, \chi_\sigma \in [0, 1]$ are constant smoothing factors. The algorithm iterates for scans i and j until the particles converge or *Num.Iterations* is exceeded. Convergence is determined when both of the following criteria are met:

- The exponential average of the standard deviation of the particles' states $\sigma_{p-\mu}^t$ is less than or equal to the robot's map cell size r_c .

$$\sigma_{p-\mu}^t \leq r_c \quad (3.9)$$

- The exponential average of the particles' weights $w_{ij-\mu}^t$ is greater than or

equal to an experimentally determined threshold τ_{pw} .

$$w_{ij-\mu}^t \geq \tau_{pw} \quad (3.10)$$

If the scans do not converge within $Num_Iterations$, then scan j is skipped and the next iteration begins. If the scans i and j converge, the translation vector $z_{d_{ij}} = \{z_{dx_{ij}}, z_{dy_{ij}}\}$ is calculated as the difference between the location of the ROV for scans i and j .

In analyzing the pair wise scan matching algorithm for runtime complexity, we find that the most computation occurs in the resampling stage of the particle filter which occurs once per eight iterations. The pseudo code with the highest runtime complexity is provided in Table 3.3. The worst case occurs when *rand_weight* equals *sum_weights*, which causes the inner most for loop to iterate $Num_Particles$ times. As seen in Table 3.1, the **PF_Localization()** is called for each pair of scan, $2 \times Num_Scans$, and **resample_particles()** is called $Num_Iterations/8$, where $Num_Iterations$ is a constant. Therefore, the runtime complexity for the pair wise scan matching algorithm is $O(Num_Scans \times Num_Particles^2)$. This is an area for future work. Specifically for a real-time system, resampling should be improved to a $O(N \log N)$ solution.

3.2.2 Scan Localization

The Scan Localization stage of the pipeline inputs the set of measured translation vectors $z_{d_{ij}}$ from the Pair Wise Scan Matching stage, and aims to output the translation estimates d_{ij} for scans i and j . This is accomplished by solving a weighted least squares minimization that aims to reduce the cost function S

in (3.11).

$$S = \sum_i \sum_{j=i-1}^{i+1} \sum_{n=1}^4 w_{ij,\mu,n}^t (d_{ij} - \hat{d}_{ij,n})^2 \quad (3.11)$$

In this cost function, d_{ij} is the relative translation vector between scans i and j being estimated. The variable $\hat{d}_{ij,n}$ represents $n = 1 \dots 4$ possible measurements extracted from various combinations of d_{ij_m} as outlined below:

- $\hat{d}_{ij.1} = z_{d_{ij}}$
- $\hat{d}_{ij.2} = -z_{d_{ji}}$, noting that $z_{d_{ji}} \neq z_{d_{ij}}$
- $\hat{d}_{ij.3} = z_{d_{ik}} + z_{d_{kj}}$, a reciprocal pseudo measurement
- $\hat{d}_{ij.4} = -z_{d_{jk}} - z_{d_{ki}}$, a reciprocal pseudo measurement

Each translation vector has a corresponding weight that represents the likelihood of that measurement being the true measurement. The weight w_{ij}^t is the exponential average of the particles' weights at the time t the particles converged. The pseudo measurements' weights $w_{ij,\mu,n}^t$ are calculated in the same manner as the pseudo translate vector measurements.

Table 3.3: Part of the Resample Particles Algorithm

resample_particles()	
1:	...
2:	<i>sum_weights</i> = sum_particles_weights ()
3:	<i>for</i> $i = 1$ <i>to</i> <i>Num_Particles</i> <i>do</i>
4:	<i>rand_weight</i> = <i>sum_weights</i> * rand (0, 1)
5:	<i>for</i> $i=0$ <i>to</i> <i>Num_Particles</i> <i>do</i>
6:	<i>sum</i> += <i>particle</i> [i]. <i>weight</i>
7:	<i>if</i> ($sum \geq rand_weight$) <i>break</i>
8:	<i>endfor</i>
9:	<i>new_particle_set</i> . push (<i>particle</i> [i])
10:	<i>endfor</i>
11:	...

The minimizing function reduces the error among the four types of $\hat{d}_{ij,n}$.

$$\min \left\{ S = \sum_i \sum_{j=i-1}^{i+1} \sum_{n=1}^4 w_{ij-\mu-n}^t (d_{ij} - \hat{d}_{ij,n})^2 \right\}, \quad (3.12)$$

$$\frac{dS}{d(d_{ij})} = 0, \quad (3.13)$$

$$d_{ij} = \frac{\sum_n^4 (w_{ij-\mu-n}^t \times \hat{d}_{ij,n})}{\sum_n^4 w_{ij-\mu-n}^t} \quad (3.14)$$

If neither permutations of scans i and j converge then there is no possible path between scans i and j .

The weights w_{ij}^t represent the uncertainty of the translation vectors. This uncertainty is reduced by the weighted least squares calculation and becomes apparent in the final map by how accurate the walls line up. Further analysis could calculate a weight of the final translation vectors d_{ij} and then adjust the final occupancy grid cells' probabilities based on these weights. Basically, when adding the range data to the final map in the Map Generation stage, the weight of the scans translation vector could be used to adjust the probability of the cells added to the map.

3.2.3 Map Generation

To create the robot's map m_{ref} , we fuse the set of sonar scans together using the log-odds approach previously explained. Since many of the sites explored contain multiple passageways, the maps can include large empty sections of space. Additionally, when exploring water galleries, the tunnels can extend for hundreds of meters and include side passages along the way. To avoid allocating memory for every cell in the map at runtime, we instead use an octree-based evidence

grid that dynamically allocates memory for a cell upon insertion of a new cell in the map. By reducing memory consumption, we were able to quadruple the resolution of the map from $100 \text{ cm}^2/\text{cell}$ to $25 \text{ cm}^2/\text{cell}$ without a performance hit.

The mapping algorithm, shown in Table 3.4, moves the robot to the localized positions according to the calculated translation vectors d_{ij} for scans i and j from the *Scan Localization* step, and then calls FastSLAM to update the robot’s map with the new sonar scan.

In generating the map, a single reference scan ref is chosen to be positioned at the center of the robot’s map based on the first scan in the series that converges with another scan. The remaining scans are added using the translation vectors calculated in Section 3.2.2. If there is no path between scans ref and i , then the remaining scans are discarded.

The pair wise scan matching algorithm explained in this paper is currently only capable of matching scans for 2D maps. Since many of the cisterns explored were bell-shaped, the 2D maps generated are only rough estimates of the curved walls. Future work will focus on converging scans along the vertical axis to generate true 3D maps from a set of 2D maps

Table 3.4: Mapping Algorithm

Generate_ROV_Map(d_{ij}, Num_Scans)	
1:	$s_{ref} = \mathbf{choose_reference_scan}(d_{ij})$
2:	$m_{ref} = \mathbf{draw_map}(s_{ref})$
3:	<i>for</i> $i = 1$ to Num_Scans <i>do</i>
4:	<i>if</i> ($s_i \neq s_{ref}$)
5:	$X_{ref,i} = \mathbf{set_ROV_state}(d_{ref,i})$
6:	FastSLAM ($X_{ref,i}$, s_i)
7:	<i>endif</i>
8:	<i>endfor</i>
9:	<i>return</i> m_{ref}

3.3 Map Visualization

After an octree-based evidence grid representation of the environment is generated, the map is visualized into a 3D model using isosurface extraction and then textured and visualized as described in [\[12\]](#).

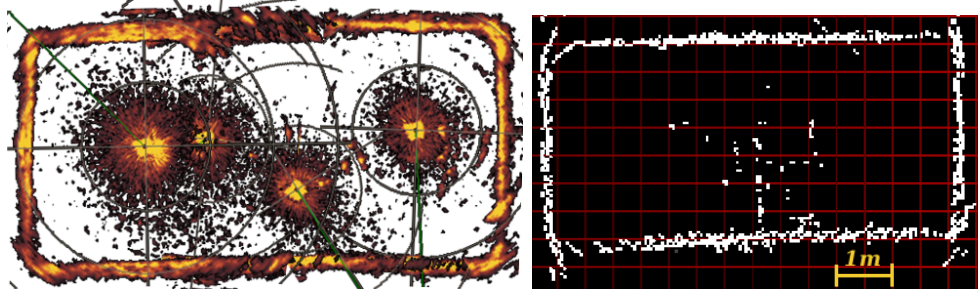
Chapter 4

Results

The pair wise scan matching algorithm presented above was applied in two scenarios. First, a map of a swimming pool located at Cal Poly's Lab for Autonomous and Intelligent Robotics (LAIR) was generated. Second, the algorithm was used to create maps of several different ancient cisterns found at archaeological sites in Malta.

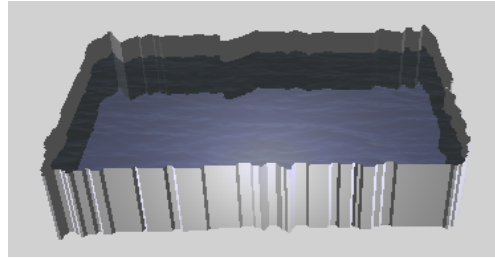
The swimming pool located at the LAIR was used to statistically validate the accuracy of the pair wise scan matching algorithm by comparing the true dimensions of the pool to the measured dimensions of the evidence grids. The true dimensions of the pool were obtained using a measuring tape that is accurate to $\pm 0.025\text{m}$. The shape of the pool curves inward at the bottom of the pool. When measuring, a straight edge was used to measure the width and length at the center depth of the pool. This best represents where the sonar beams would hit. The true width of the pool is 3.61m and the length is 7.21m. Four stationary sonar scans of the pool were obtained using the procedure for data collection explained in Section 3.1.1 with the sonar beam range r_s set to 5m. The mosaic of these scans can be seen in Figure 4.1(a). The measurements Z_m were passed into

the map generation pipeline and the evidence grid with cell size 0.05m x 0.05m seen in Figure 4.1(b) was generated.



(a) Swimming pool sonar mosaic

(b) Evidence grid representation



(c) Model from visualization

Figure 4.1: Pool at Cal Poly’s LAIR

The accuracy of the mapping algorithm was determined by comparing the true dimensions of the pool to dimensions extracted from the 2D map. The cells inside the map were counted from one inside edge to the other. This resulted in map estimated dimensions of 3.47m in width and 7.27m in length. Since each cell is 0.05m in width and length, the pools dimensions can not be accurate to better than $\pm 0.05m$. The Percent Difference PD of the ratio of the mean width and length to the ratio of the true width and length was calculated as -4.78% (4.1).

$$PD = \frac{r_{\mu} - r}{(r_{\mu} + r)/2} \times 100\% \quad (4.1)$$

$$PD = \frac{3.47m/7.27m - 3.61m/7.21m}{(3.47m/7.27m + 3.61m/7.21m)/2} \times 100\% = -4.78 \quad (4.2)$$

The standard deviation of the pair wise scan matching algorithm was calculated by measuring the width of the pool at 10 different locations and then comparing those 10 different measured widths and lengths to the true width and length of the pool. The calculated standard deviation of the width was 0.119m and length was 0.060m.

This algorithm was then applied to several cisterns explored between 2008 and 2012. Results from applying the algorithm to sensor measurements taken from three different cisterns are provided here. The constants that vary when generating the maps include τ_{pw} and $Num_Particles$. The following constants and their corresponding values were used when generating the three cistern maps and the swimming pool:¹ $\chi_w = \frac{2.0}{6.0+1.0}$, $\chi_\sigma = \frac{2.0}{4.0+1.0}$, $r_c = 0.05m$ (map cell size), and $Num_Iterations = 1200$.

The first map presented represents the Gatto Pardo Bistro cistern found in the city of Mdina, Malta. The value for τ_{pw} was experimentally determined as 8.0, and 800 particles were used. Figure 4.4 presents the different pairings of sonar scans during the Pair Wise Scan Matching stage. The position of the ROV for the first scan used to create the map (line 2 of Table 3.4) is represented by the yellow robot model. The localized ROV position corresponding to the second scan to be matched with the first is represented by the green and white particles, where the whiteness of the particle corresponds to the magnitude of the weight (as calculated using line 4 of Table 3.1) where the white particles have higher weights than the green particles.

The second map presented (Site 8) was explored in 2008 at a private home in the city of Mdina seen in Figure 4.2. This cistern was mapped with the new

¹ χ represents the degree of weighting decrease. Alternatively, $\chi = \frac{2.0}{N+1.0}$ where N represents the number of previous measurements that have the greatest impact on the exponential average.

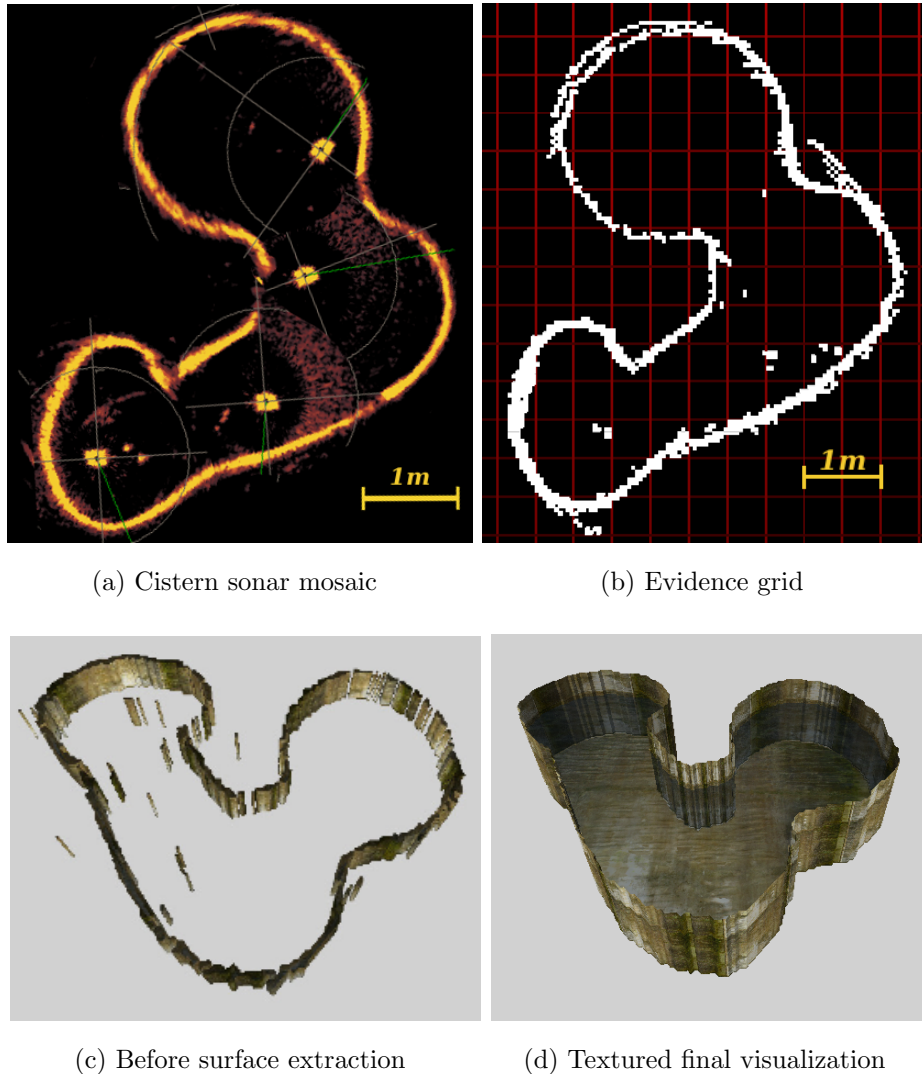


Figure 4.2: Mдина private home cistern pipeline walk-through. In (a), the manual sonar mosaic is shown. The auto-generated occupancy grid is shown (b). The 2.5D model and the surface extracted, textured model is shown in (c) and (d) respectively.

algorithm using 800 particles with a τ_{pw} value of 8.0. This cistern demonstrates the ability to converge a set of scans where the end point scans have limited overlap due to the robot being in two separate chambers.

Lastly, the map shown in Figure 4.3 represents a cistern found on the island of Gozo at the Gozo Citadel. Since this site had much smaller dimensions, the

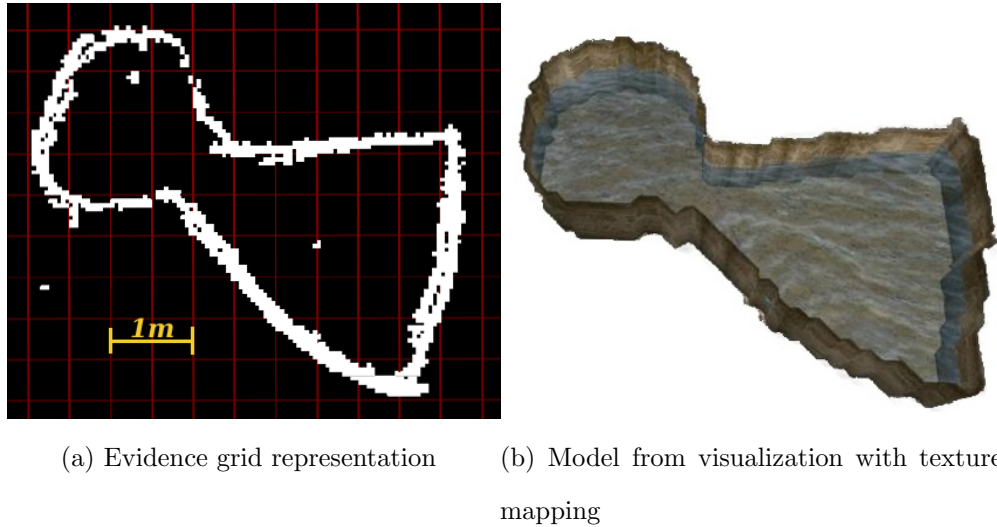


Figure 4.3: Reconstruction of the Gozo Citadel "Keyhole" cistern. In (a) and (b), the auto-generated occupancy grid and the visualized model are shown respectively.

sonar signal strengths were higher. This resulted in a τ_{pw} value of 20.0 and the use of 1000 particles. The higher τ_{pw} value is experimentally determined based on the exponential average of the particles' weights when the first pair of scans converge.

Since no truth data regarding the size and shape of the explored cisterns were available, mosaics of raw sonar scan images were manually created. The result of this manual mosaic process for the Gatto Pardo Bistro cistern is shown in Figure 4.4h. To note, the average particles' positions obtained from the automated pair wise scan matching (the green and white particles) converged to locations that correspond well to the localized ROV positions in the manually created sonar mosaic (orange circles seen in Figure 4.4). The orange circles in the raw sonar scans are believed to be the acoustic reflections from the robot itself. The holes in the map are attributed to the mapping algorithm's method of fusing scans together by adding the log-odds of each cell. These holes are filled during the Map Visualization step of the pipeline covered in [12].

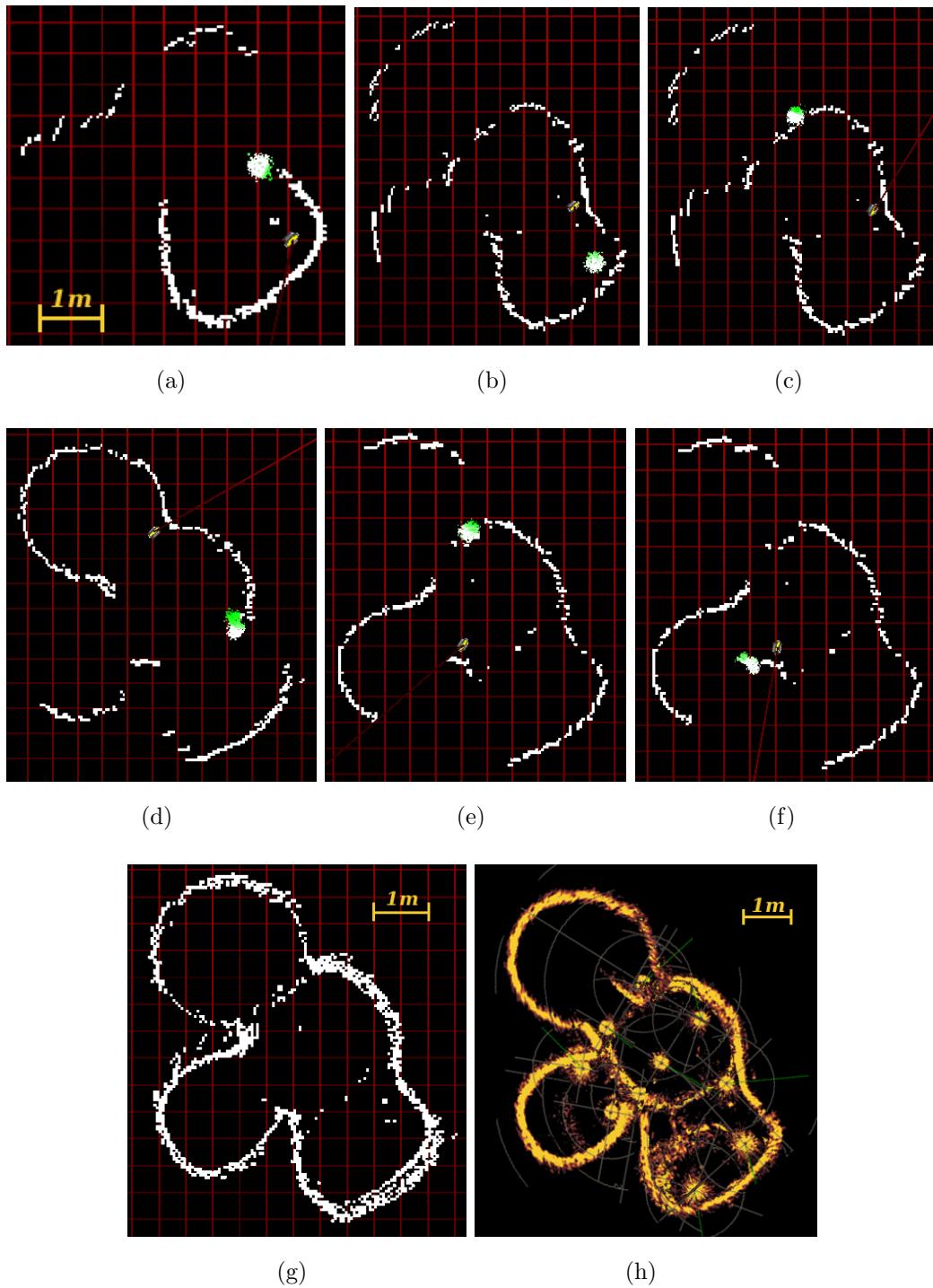


Figure 4.4: Pair wise scan matching algorithm on the Gatto Pardo Bistro cistern. Parts (a)-(f) demonstrate each of the scans and where the particles converged. In (g) and (h), the auto-generated occupancy grid is shown as well as the manually crafted sonar mosaic respectively.

Table 4.1: Mean Dimension Differences (Grid vs. Mosaic)

Site Name	Width (m)	Length (m)
Mdina Home (Site 8)	0.000	-0.022
Gatto Pardo Bistro	-0.026	0.033
Swimming Pool	0.105	-0.025

Further comparison of three evidence grid maps was done by comparing the width and length of the mosaics to the width and length of the evidence grids. Since all the sites are not linear in shape, width and length measurements were chosen based on distinct features in the map, i.e. corners and tunnels. Table 4.1 outlines the difference in width and length measurements between the mosaics and evidence grids of three different maps.

Chapter 5

Conclusions and Future Works

5.1 Conclusions

This work demonstrates the ability to conduct offline mapping of underwater cisterns with robots that have low payloads and then apply surface extraction techniques to further enhance the generated maps for archaeological research. Specifically, maps were constructed offline without the use of robot odometry or a motion model. By applying the newly developed pair wise scan matching algorithm, octree-based evidence grid representations of such tunnels, cisterns, wells, and water galleries were produced with a best fit solution and then visualized using surface extraction techniques.

The results of four generated maps demonstrate that for the application at hand, the pair wise scan matching algorithm is accurate up to a standard deviation of 0.119m calculated from the evidence grid of the mapped swimming pool. The percent difference between the dimensions of the pool's evidence grid map and the truth data was also measured as -4.78%. To reinforce these statistics

and the accuracy of the pair wise scan matching algorithm, the dimensions of three manually created mosaic maps were compared to their corresponding evidence grid maps generated from the pair wise scan matching algorithm seen in Table 4.1 and resulted in the highest difference being just over twice the cell size of the evidence grids.

5.2 Future Works

Some limitations of the pair wise scan matching algorithm include the need for unique features that overlap across scans. Additionally, this algorithm is negatively affected by noise that introduces false positives due to the misclassification of debris in the water and multipaths as a wall. Lastly, if one of the scans does not overlap with either the scan before it or after it, then that scan along with the remaining scans in the set are eliminated. This last limitation is avoided by collecting more scans that have ideally 50% or more overlap. These three limitations are the focus for further research.

The pair wise scan matching algorithm could also be expanded to support localizing an ROV in the third dimension. This is important when working in an underwater environment with nonlinear walls like those found in many Maltese cisterns.

Lastly, a necessary improvement for future work is the addition of a loop closure algorithm. A simple approach would be to modify the pair wise scan matching algorithm to use the distance to a previous scan as the determining factor for comparing two scans with the particle filter. Ideally, this should reduce the localization error when for example the last scan overlaps with the first scan.

Bibliography

- [1] A. Burguera, Y. Gonzalez, and G. Oliver. Probabilistic Sonar Scan Matching for Robust Localization. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 3154 –3160, April 2007.
- [2] A. Burguera, Y. Gonzalez, and G. Oliver. Underwater SLAM with Robocentric Trajectory Using a Mechanically Scanned Imaging Sonar. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 3577–3582, Sept. 2011.
- [3] A. Burguera, G. Oliver, and Y. Gonzalez. Range Extraction From Underwater Imaging Sonar Data. In *Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on*, pages 1 –4, Sept. 2010.
- [4] J. Castellanos, J. Tardos, and J. Neira. Constraint-based Mobile Robot Localization. In *Proceedings of the International Workshop on Advanced Robotics and Intelligent Machines (ARIM)*, 1996.
- [5] C.M. Clark, D. Hiranandani, C. White, M. Boardman, M. Schlachtman, P. Phillips, J. Kuehn, T. Gambin, and K. Buhagiar. The Malta Cistern Mapping Project: Expedition II. In *Proc. of the Unmanned Untethered Submersible Technology (UUST 09)*, 2009.

- [6] C.M. Clark, C. Olstad, K. Buhagiar, and T. Gambin. Archaeology via Underwater Robots: Mapping and Localization within Maltese Cistern Systems. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 662 –667, Dec. 2008.
- [7] H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *Robotics Automation Magazine, IEEE*, 13(2):99 –110, June 2006.
- [8] A. Eliazar and R. Parr. DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks. In *in Proc. 18th Int. Joint Conf. on Artificial Intelligence (IJCAI-03*, pages 1135–1142. Morgan Kaufmann, 2003.
- [9] P. Elinas, R. Sim, and J. Little. σ SLAM: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1564 –1570, May 2006.
- [10] N. Fairfield, G. Kantor, and D. Wettergreen. Real-time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels. *Journal of Field Robotics*, 24:3–21, 2007.
- [11] N. Fairfield and D. Wettergreen. Evidence Grid-based Methods for 3D Map Matching. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1637 –1642, May 2009.
- [12] C. Forney, J. Forrester, B. Bagley, W. McVicker, J. White, T. Smith, J. Batoryn, A. Gonzalez, J. Lehr, T. Gambin, C.M. Clark, and Z.J. Wood. Surface Reconstruction of Maltese Cisterns Using ROV Sonar Data for Archeological Study. In *Proceedings of the 7th international conference on Advances in vi-*

- sual computing - Volume Part I*, ISVC'11, pages 461–471, Berlin, Heidelberg, 2011. Springer-Verlag.
- [13] A. Goldstein and S. Bentley. Use of Highly Portable Micro-sized Remotely Operated Vehicles for Environmental Monitoring and Mapping. In *OCEANS 2010*, pages 1–6, Sept. 2010.
- [14] E. Hernández, P. Ridaó, D. Ribas, and J. Batlle. A Probabilistic Scan Matching Algorithm Using a Mechanical Scanned Imaging Sonar. *Journal on Physical Agents (JoPhA)*, 3(1):3–12, 2009.
- [15] P. Jasiobedzki, S. Se, M. Bondy, and R. Jakola. Underwater 3D Mapping and Pose Estimation for ROV Operations. In *OCEANS 2008*, pages 1–6, Sept. 2008.
- [16] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast Incremental Smoothing and Mapping with Efficient Data Association. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1670–1677, April 2007.
- [17] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental Smoothing and Mapping. *Robotics, IEEE Transactions on*, 24(6):1365–1378, Dec. 2008.
- [18] K. Konolige and M. Agrawal. FrameSLAM: from Bundle Adjustment to Realtime Visual Mapping. *Robotics, IEEE Transactions on*, 24(5):1066–1077, Oct. 2008.
- [19] F. Loisy, P. Franois, G. Douchet, P. Hope-Darby, K. Shimmin, T. Bonner, E. Laurent, and R. Colin. Underwater Inspection Experiment for a Long Tunnel of EDF's Hydroelectric Facilities. In *Applied Robotics for the Power*

- Industry (CARPI), 2010 1st International Conference on*, pages 1 –4, Oct. 2010.
- [20] I. Mahon and S. Williams. SLAM Using Natural Features in an Underwater Environment. In *Control, Automation, Robotics and Vision Conference, 2004. ICARCV 2004 8th*, volume 3, pages 2076 – 2081 Vol. 3, Dec. 2004.
- [21] S. Majumder, J. Rosenblatt, S. Scheduling, and H. Durrant-whyte. Map Building and Localization for Underwater Navigation. In *In International Symposium On Experimental Robotics*, 2000.
- [22] A. Mallios, P. Ridaou, M. Carreras, and E. Hernández. Navigating and Mapping with the Sparus AUV in a Natural and Unstructured Underwater Environment. In *OCEANS 2011*, pages 1 –7, Sept. 2011.
- [23] M. C. Martin and H. P. Moravec. Robot Evidence Grids. Technical report, 1996.
- [24] R. Sim, P. Elinas, and M. Griffin. Vision-based SLAM using the Rao-Blackwellised Particle Filter. In *In IJCAI Workshop on Reasoning with Uncertainty in Robotics*, 2005.
- [25] R. Sim, P. Elinas, and J. J. Little. A Study of the Rao-Blackwellised Particle Filter for Efficient and Accurate Vision-based SLAM. *Int. J. Comput. Vision*, 74(3):303–318, Sept. 2007.
- [26] S. Thrun. *Probabilistic Robotics*. MIT Press, Cambridge, Mass, 2005.
- [27] C. White, D. Hiranandani, C. Olstad, K. Buhagiar, T. Gambin, and C.M. Clark. The Malta Cistern Mapping Project: Underwater Robot Mapping and Localization with Ancient Tunnel Systems. *Journal of Field Robotics*, 2010.

- [28] S. Williams, P. Newman, G. Dissanayake, and H. Durrant-Whyte. Autonomous Underwater Simultaneous Localisation and Map Building. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, volume 2, pages 1793 –1798 vol.2, 2000.