The Malta Cistern Mapping Project: Underwater Robot Mapping and Localization within Ancient Tunnel Systems

• •

Cory White and Daniel Hiranandani

Computer Science Department, California Polytechnic State University, San Luis Obispo, California 93407
e-mail: corybwhite@yahoo.com, danielhira@gmail.com

Christopher S. Olstad

Marine Resources Development Foundation, 51 Shoreland Drive, Key Largo, Florida 33037
e-mail: chris@mrdf.org

Keith Buhagiar

Department of Classics and Archaeology, University of Malta, MSD 2080, Malta
e-mail: keithbuhagiar@onvol.net

Timmy Gambin

AURORA Special Purpose Trust, Malta
e-mail: tgambin@hotmail.com

Christopher M. Clark

Computer Science Department, California Polytechnic State University, San Luis Obispo, California 93407
e-mail: cmclark@calpoly.edu

This paper documents the development of an underwater robot system enabled with several mapping and localization techniques applied to a particular archaeological expedition. The goal of the expedition was to explore and map ancient cisterns located on the islands of Malta and Gozo. The cisterns of interest acted as water storage systems for fortresses, private homes, and churches. Such cisterns often consisted of several connected chambers, still containing water. A sonar-equipped remotely operated vehicle (ROV) was deployed into these cisterns to obtain both video footage and sonar range measurements. Six different mapping and localization techniques were employed, including (1) sonar image mosaics using stationary sonar scans, (2) sonar image mosaics using stationary sonar scans with Smart Tether position data, (3) simultaneous localization and mapping (SLAM) while the vehicle was in motion, (4) SLAM using stationary sonar scans, (5) localization using previously created maps, and (6) SLAM while the vehicle was in motion with Smart Tether position data. Top-down-view maps of 22 different cisterns were successfully constructed. It is estimated that the cisterns were built as far back as 300 B.C., and few records of their size, shape, and connectivity existed before the expedition. © 2010 Wiley Periodicals, Inc.

1. INTRODUCTION

This project concerns the development of an underwater robot system capable of mapping and navigating underwater tunnel systems. The target environments for this project are cistern networks found in the lower chambers of fortresses and churches across the country of Malta.

In contrast to its closest neighbor, Sicily, from where the island's first inhabitants originated, Malta is dry with very limited seasonal rainfall. Over the past 8,000 years, the capture and storage of water has been of paramount importance to the islanders as this permitted the survival of relatively large communities on an offshore island with no natural resources other than limestone. The importance of water is supported from an archaeological perspective as wells and cisterns have been discovered on numerous sites including some dating back to circa 300 B.C. In the

Punic and Roman periods, an increase in population meant that water management became more extensive and complex. However, the reutilization of urban spaces throughout the past 2,000 years has meant that many of the ancient wells and systems have been integrated into more modern buildings. The current project allows the study of numerous wells and cisterns that have, due mainly to their inaccessibility, remained unexplored. By systematically surveying these sites, one is able to better understand the origin of Malta's ancient wells and cisterns as well as study how they were developed and integrated into more recent water management systems. On the basis of the survey results, one can create a geographic information system–based map that will also contribute to the valorization and protection of these hidden wonders.

Archaeologists looking to study and document such systems have found it too expensive and difficult to use
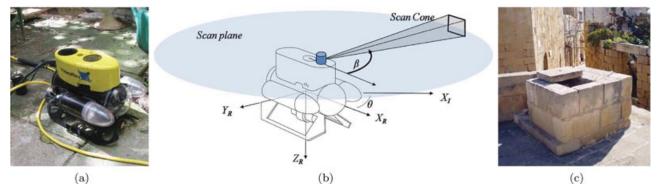
Figure 1. (a) The VideoRay Pro III Micro ROV with a Tritech SeaSprite sonar module and KCF Smart Tether. A depiction of the ROV mounted sonar and its scan plane is shown in (b). In (c), a typical cistern access point is shown.

people. Furthermore, the human exploration of these subterranean water storage systems is limited by safety and physical constraints and could possibly result in irreversibly damaging the site under study.

A small underwater robot, or specifically a VideoRay Micro remotely operated vehicle (ROV), was used as seen in Figure 1(a). Investigators first lowered the ROV down well access points until it was submerged in the cistern. The investigators then teleoperated the robot to navigate the tunnels. Top-down-view maps [i.e., two-dimensional (2D) maps across the horizontal plane] of the cisterns were created using a Tritech SeaSprite scanning sonar mounted on top of the ROV. The sonar was mounted as shown in Figure 1(b), with a sonar beam that rotates 360 deg around a horizontal scan plane. These sonar measurements were used in six ways to develop cistern maps and conduct localization. Although a goal was to conduct simultaneous localization and mapping (SLAM) of the cisterns in real time, the primary objective was to provide accurate cistern maps to archaeologists.

The paper is presented as follows. Section 2 presents related robot mapping techniques. A description of the hardware implementation is given in Section 3. Section 4 explains the methodology used during the expedition. In Section 5, details are provided of the specific mapping and localization techniques. Results from these experiments are shown in Section 6, followed by conclusions in Section 7.

2. BACKGROUND

Several methods exist for mapping subaqueous environments when using underwater robots. The maps constructed can be used both for the application at hand (e.g., marine biology, archaeology) and to improve the navigation capabilities of the robot itself.

When the robot is localized with respect to some inertial coordinate frame (i.e., the robot's position is known), mapping while in motion is a much simpler task. An approach typically used when operating wheeled robots

within indoor environments is an occupancy grid map that is updated via the log likelihood approach that assigns a probability of occupation for each cell in the grid (Moravec, 1988).

A common method used for mapping underwater involves mosaicking bottom images obtained from different locations. Once combined, the resulting mosaic can be used as a map with which the robot can localize itself. A benefit of such mapping systems is they do not rely on the deployment of infrastructure such as acoustic positioning systems. For example, in Richmond and Rock (2006) an ROV was equipped with a real-time mosaicking system. Also, in Sakai, Tanaka, Mori, Ohata, Ishii, et al. (2004), video mosaicking was used for autonomous underwater vehicle (AUV) navigation.

In general, a large amount of research has been conducted in the area of SLAM, in which the robot's local environment is mapped while concurrently determining the robot's position within the map. SLAM techniques have been developed and modified for a large number of applications and environments. A good survey of the core techniques including both Kalman filtering and particle filtering can be found in Thrun, Burgard, and Fox (2005).

One example of robots conducting SLAM in tunnel systems is found in Baker, Morris, Ferguson, Thayer, Whittaker, et al. (2004), where the mapping of underground mines was conducted using an autonomous wheeled robot called "Groundhog." Although underwater, the cistern mapping is similar in that it uses particle filters. Also relevant is the SLAM reported in Sim, Elinas, and Little (2007), where an above-ground robot equipped with a vision system was used to construct occupancy grids. Other vision-based SLAM includes the underwater work accomplished in Eustice, Singh, Leonard, Walter, and Ballard (2005) and Eustice, Singh, and Leonard (2006), in which an ROV equipped with a camera was used for navigating the RMS Titanic shipwreck. Emphasis in Eustice et al. (2005) was on maintaining bounds on covariance with efficient Kalman updates, whereas work in Eustice et al. (2006) concentrated

on ensuring exact sparsity in an information matrix formulation to limit approximation errors.

One of the first instances of underwater robot SLAM is that of Williams, Newman, Dissanayake, and Durrant-Whyte (2000), where sonar scans were used to map and track features of the environment. Majumder, Rosenblatt, Scheding, and Durrant-Whyte (2001) proposed a framework for underwater navigation that utilizes data from various sensors into a single scene, which are combined to build and represent a map using probability theory. In Mahon and Williams (2004), SLAM was applied to an underwater vehicle within a natural environment and built environment models for data procured from the Great Barrier Reef in Australia.

Successful three-dimensional (3D) tunnel mapping in underwater environments was demonstrated in Fairfield, Kantor, and Wettergreen (2006). Additionally, the mapping of marinas via underwater SLAM was successful, as shown in Ribas, Ridao, Neira, and Tardós (2006). Following this work, the most recent publications of underwater robots implementing SLAM in man-made structured environments include Ribas, Ridao, Neira, and Tardós (2007) and Ribas, Ridao, Tardós, and Neira (2008). In these works, a mechanically scanned imaging sonar is used in combination with a line-feature extraction algorithm to gather information about the environment, and experiments are conducted in a marina in order to show the feasibility of the approach.

Unlike the work in Fairfield et al. (2006), Ribas et al. (2006, 2007), and Ribas et al. (2008), this paper describes applications that permit only the passage of small-scale robot systems (i.e., passage-opening diameters on the order of 0.3 m). Furthermore, the ROV was equipped only with a depth sensor, compass, scanning sonar, and a KCF Smart Tether that measures the location of the ROV relative to a stationary global positioning system (GPS) receiver location. The Smart Tether is not restricted to this cistern mapping domain. In other work, it was used along with the VideoRay underwater robot for hull inspection of the U.S.S. John F. Kennedy (http://www.kcftech.com/jfk.shtml) as well as to search for submerged unexploded

ordnance of a former U.S. Navy bombing range (http://wwv.videoray.com/PR/KCFOperation.html).

To overcome the limitations in sensing with a small-scale robot, a dynamic model of the ROV was used for the prediction step of both the SLAM and particle filter localization algorithms. However, a major issue associated with this approach is that tether snags and collisions with walls are not considered in a typical dynamic model. Such occurrences are accounted for, and it is shown that when implemented within a particle filter–based SLAM approach, i.e., FastSLAM (Thrun et al., 2005), mapping is possible even when frequent tether snags or collisions occur.

3. HARDWARE IMPLEMENTATION

A typical setup of the above-water equipment is shown in Figure 2(a). At the heart of the setup is the ROV control box (3), which is connected to the ROV via a tether. A joystick in the control box can be used to control the ROV, or a PC (5) can be connected to the control box via a serial RS-232 port. Also connected to the PC is the Smart Tether (1) via an interface box (2). A separate joystick (6) can also be used with this PC. Finally, our setup used another computer (4) for recording all digital video clips.

The flow of data through this setup is as follows: Sensor measurements from the ROV (depth, bearing, sonar) go through the tether to the control box and then to the PC. A software application written in C++ runs on the PC and receives these sensor measurements and any user inputs from the graphical user interface (GUI) and joystick [see Figure 2(b)]. The software also receives measurements directly from the Smart Tether. ROV thruster and light control signals can be sent from computer software through the control box, down the tether, to the ROV.

The software application on the PC was developed with tunable joystick and autonomous depth/bearing controls. Adjustable gains on the proportional control sigmoid functions and joystick sensitivity were essential, especially when navigating through narrow passages. The

Figure 2. A typical experiment setup (a) and the mapping software GUI (b).

autonomous depth/bearing controllers were also useful when mapping the few cisterns that had variation in size and shape when depth was varied.

Note that the SeaSprite scanning sonar has a variety of parameters that can be set using the SeaNet Pro software. The following settings were used: 6.0-m range, 40-dB dynamic range, 70% gain, 700-kHz frequency, 360-deg scan, and low resolution. Measurements from the sonar come through the SeaNet Pro application to our C++ mapping software via sockets.

Given the large number of connected components, along with a requirement for systems to be turned on in a specific order, a systematic hardware and software setup routine was developed and practiced. This routine involved powering on components, sensor and actuator functionality assessment, and sensor calibration. Additionally, during actual experiments the researchers progressed from implementing the easiest mapping techniques to the most difficult. These routines were essential given that site accessibility was often limited to 1–2 h.

4. EXPERIMENT DESCRIPTION

Twenty-four different sites in Malta and Gozo were visited in total. Six of these sites were visited in 2008, whereas the last 18 sites were visited most recently in 2009. The cisterns were estimated to be constructed between 300 B.C. and the 15th or 16th century. At each site, the ROV was initially lowered through a small opening and then down a 3–15-m-deep chute before submerging in the cistern water. As shown in Figure 3(a), several layers of construction can be observed with increasing depth. A reflection of the ROV's lights can be seen on the water's surface below as it descends down the chute (see center of image).

Once submerged, the ROV was piloted throughout the cistern, exploring any passageways and chambers. To accomplish this, operators used video from the onboard camera and a joystick controller. An example of one such video image is shown in Figure 3(b), where the ROV is traveling through a tight passage. Also note that the water clarity in this particular cistern allowed for a reflection on the water surface (as seen in the top half of the image).
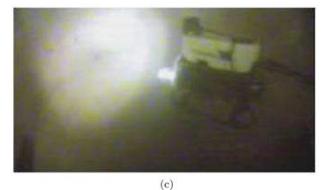

(a)

(b)

(c)

Figure 3. For each site, the ROV was initially lowered down a deep narrow chute (a). (b) An image obtained while returning through a tight passage. (c) The view from one ROV while it records images of another ROV inspecting the cistern wall.

To aid in SLAM experiments conducted in 2008, auto depth and auto bearing controllers were used. Shown in Figure 3(c) is an ROV using autonomous control methods to hold stationary while viewing a cistern wall. It should be noted that this image was obtained by lowering a second ROV into to the same cistern, allowing it to obtain video of the first ROV.

After video images of the cistern were recorded, stationary sonar scans were obtained while the ROV sat on the bottom of the cistern. As shown in Figure 1(b), scans captured ranges in the horizontal plane only. Because cisterns had relatively level floors and vertical walls, sonar scans gave ranges to cistern walls in the robot's local vicinity. For each scan, the ROV was positioned to ensure that scans would overlap each other to facilitate easy mosaicking.

Once a sufficient number of stationary scans were obtained, sonar scans were recorded while the ROV was in motion. Control signals and depth and heading measurements were also recorded for use with SLAM.

With regard to the work conducted in 2009, a more precise joystick control system was developed that allowed for nonlinear sensitivity adjustment. Additionally, the KCF Smart Tether was used to collect the robot's position data relative to the access point of a cistern. Thus, for most cisterns explored in 2009, sonar, Smart Tether, control signal, depth, and heading measurement data were collected.

5. MAPPING AND LOCALIZATION TECHNIQUES

Six different mapping and localization techniques were used, each producing maps qualitatively consistent with one another but of different formats.

5.1. Sonar Image Mosaics with Stationary Scans

The first approach taken was to mosaic several overlapping 360-deg sonar scans. Sonar images in .jpg format were simply dragged and dropped over one another using simple image manipulations tools (i.e., Microsoft PowerPoint). The order of the scans and the robot orientation were recorded during experiments so that they could easily be overlapped later.

Although not systematic or error-free, this method gave archaeologists and computer scientists a quick map of the entire cistern while still showing raw sonar image data. Figure 4 displays an example mosaic created from seven scans. Each scan on the mosaic has an obvious circle of high-strength returns indicating the robot's position within the scan. Note that the high quality of the images and the obvious correspondence allows for them to be easily fused by a human operator.

5.2. Stationary Sonar Image Mosaics Utilizing Smart Tether

On the 2009 expedition, a KCF Smart Tether was utilized to accumulate additional position data of the ROV. The Smart Tether records the orientation and position of the ROV by using acceleration, magnetic, and rate-gyro sensors [making it robust to noise, reflections, and obstructions (KCF Technologies, 2008)] distributed along the tether. Hence, in addition to collecting several overlapping 360-deg sonar scans, position data of the ROV were also recorded to mark the locations of the individual sonar scans, which assist in the creation of a final mosaic image. Specifically, the lat/long coordinates from the Smart Tether were used as initial locations of scans that were then aligned manually,
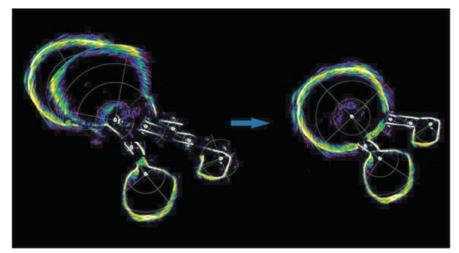

Figure 4. On the left is a collection of sonar scans obtained from a monastery in the fortress city of Mdina, Malta. On the right is the mosaic created from the scans. Note that scans are not transparent and are overlapping one another. For example, one of the centers of the seven scans is occluded in the right tunnel of the right image.
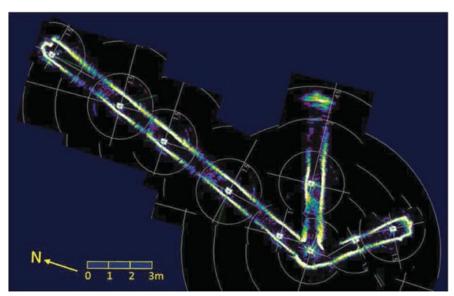
Figure 5. The cistern displayed in this image mosaic is located in a monastery courtyard in the city of Rabat, Malta. The mosaic was constructed by combining multiple independent sonar scans and using Smart Tether data to assist in positioning scans relative to each other.
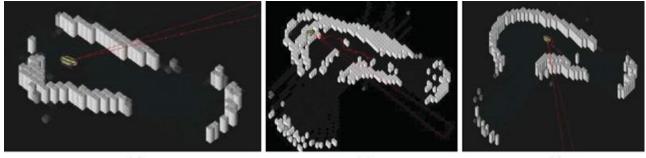
similar to the image mosaics created without a Smart Tether. Such a methodology became very beneficial for cisterns with notably long tunnels where features were very similar and potentially indistinguishable from each other. An example of this case is shown in Figure 5.

5.3. SLAM with the ROV in Motion

One goal of this project was to implement SLAM in real time. The localization includes determining the robot state $\mathbf{x}_t = [x \ y \ z \ \theta \ \dot{x} \ \dot{y} \ \dot{z} \ \dot{\theta}]_t$ at each time step t. Here, the first three elements of the state vector correspond to Cartesian coordinates in an inertial coordinate frame. The fourth ele-

ment is the yaw angle, the robot's rotation about the vertical axis. Note that it is assumed that there is zero roll and pitch, which are valid assumptions for this vehicle in the relatively static cistern environments. The remaining four elements of the state vector are the time derivatives of the first four elements.

Because very little was known about the cisterns under investigation (i.e., size, types of features, number of features, etc.), an occupancy grid was used to represent the belief state of the environment (Moravec & Elfes, 1985). That is, the cistern model was discretized into square cells of equal size. Each cell was assigned a probability that it was occupied (e.g., by a wall). Figure 6 shows an occupancy grid

Figure 6. The ROV is mapping the cistern at site 8. (a) The ROV sits on the bottom and maps out the mound of silt just in front of it. (b) We see that without a model for tether snags, the predicted position of the robot is inaccurate, resulting in walls that are replicated several times in the map. Using the proposed model from Eq. (1), successful mapping is possible (c). The red line within the cistern indicates the path of the ROV. The two straight red lines indicate the direction of the current sonar measurement. Each cell is 0.20×0.20 m in size, and the height of the cell represents the likelihood of occupation.

map for site 8. Note that the lightness of color of the cell indicates probability of occupation.

The particular SLAM algorithm used in this project was FastSLAM for learning occupancy grids [presented in Thrun et al. (2005) and similar to that of Eliazar & Parr (2003)] because it does not require features like most SLAM algorithms. FastSLAM is a particle filter–based approach to SLAM, in which a collection of M particles denoted as X_t is used to model the belief state. For this case, the kth particle consists of an occupancy grid m_t, the robot's state $\mathbf{x}_t^k$, and a weight w_t^k that represents the likelihood that particle k represents the true state. As shown in Algorithm 1, the tth time step of the algorithm updates all particles as new sensor measurements z_t are observed.

The three key steps to this algorithm are on line numbers 4, 5 and 9 of Algorithm 1. The first, sample_motion_model, propagates the previous state $\mathbf{x}_{t-1}^k$ of the robot forward in time according to the control inputs u_t. A certain degree of randomness is added propagation, in accordance with the robot's motion model.

When the ROV is in motion, this function uses a dynamic model $\mathbf{x}_t^k = f(\mathbf{x}_{t-1}^k, u_t)$, which predicts the state of the ROV given the last state and current control signals. This model is based on that developed in Wang and Clark (2006). Whereas the model is nonlinear, it assumes zero pitch and zero roll and that the state dynamics are decoupled between the horizontal and vertical planes. These assumptions are valid when the ROV operates at low velocities in environments with no currents (e.g., within cisterns). Furthermore, the model in Wang and Clark (2006) does not consider the tether's effect on dynamics.

To account for both tether snags and the ROV's motion being obstructed by collision with walls, the propagation model was modified accordingly:

$$\mathbf{x}_t^k = f\left[\mathbf{x}_{t-1}^k, u_t(1+r_1) - \epsilon u_t(1+r_2)\right], \tag{1}$$

$$\epsilon = \begin{cases} 0 & \text{if } r_3 < \lambda \\ 1 & \text{else} \end{cases}. \tag{2}$$

In Eq. (1), r_1 and r_2 are normally distributed random variables. The value of ϵ is either 1 or 0, representing a tether snag or no tether snag, respectively. This is set according to a uniformly distributed random variable r_3 and a probability of tether snag or obstruction λ.

The next step in the algorithm invokes the measurement_model_map function, which calculates the weight of the kth particle. At a high level, the expected sonar measurement is calculated given the robot state $\mathbf{x}_t$ and the map m_{t-1}. This expected sonar measurement is compared with the actual measurement $z_{s,t}$. If the two measurements are similar, a high weight is returned; otherwise a low weight is returned.

To quantify this similarity, we first note that a sonar measurement z_s has the form $z_s = [\beta s^1 \ldots s^B]$, where β is the direction of the sonar head and s^i is the ith strength of return signal measured at a distance $i/\text{maxRange}$. To determine the weight of the particle, each strength of return s^i is converted to a corresponding occupancy probability according to a log odds mapping approach (Thrun et al., 2005) to yield $p_z = [p_z^1 \ldots p_z^B]$. If the map's cells that correspond with the B sonar measurement locations currently have occupation probabilities $p_m^k = [p_m^1 \ldots p_m^B]$, then the weight can be calculated using a Gaussian model as in Eq. (3), where σ_z is the standard deviation of the Gaussian model with expected probability p_m^k. The value for σ_z was calculated from various sonar scans taken in the California Polytechnic State University swimming pool:

$$w^k = \sum_{i=1}^{B} \frac{1}{\sigma_z \sqrt{2\pi}} \exp\left[\frac{-\left(p_m^k - p_z\right)^2}{2\sigma_z^2}\right]. \tag{3}$$

The next step is the measurement_smart_tether function, which uses the weight for each particle as calculated in measurement_model_map and executes only if the Smart Tether is used. Also from a high level, the expected Smart Tether measurement (within the map coordinate frame) is simply the particle position $\mathbf{x}^k$. The resulting expected Smart Tether measurement is then compared with the actual Smart Tether measurement of robot position z_{st}. If a high weight is calculated, the previously calculated weight from measurement_model_map is strengthened; otherwise the weight is reduced. Once the initial weight for the particle has been refined with the Smart Tether data, it is returned as the new weight:

$$w^k = w^k \frac{1}{\sigma_{st} \sqrt{2\pi}} \exp\left[\frac{-\left(x^k - z_{st}\right)^T \left(x^k - z_{st}\right)}{2\sigma_{st}^2}\right]. \tag{4}$$

Algorithm 1 FastSLAM

1: **Alg. FastSLAM_occupancy_grids**(X_{t-1}, u_t, z_t):
2:   $X_t' = X_t = 0$
3:   *for* $k = 1$ *to* M *do*
4:     $\mathbf{x}_t^k = $ **sample_motion_model**($u_t, \mathbf{x}_{t-1}^k$)
5:     $w_t^k = $ **measurement_model_map**($z_{s,t}, u_t, m_{t-1}^k$)
6:     *if (using_smart_tether) then*
7:       $w_t^k = $ **measurement_smart_tether**($z_{st,t}, u_t, m_{t-1}^k, w_t^k$)
8:     *endif*
9:     $m_t^k = $ **updated_occupancy_grid**($z_{s,t}, u_t, m_{t-1}^k$)
10:    $X' = X' + \{\mathbf{x}_t^k, m_t^k, w_t^k\}$
11:   *endfor*
12:   *for* $k = 1$ *to* M *do*
13:     *draw i with probability* $\sim w_t^i$ *from* X_t'
14:     *add* $\{\mathbf{x}_t^i, m_t^i\}$ *to* X_t
15:   *endfor*
16:   *return* X_t

In Eq. (4), σ_{st} is the standard deviation of the Smart Tether measurement obtained from the product specs.

The last core function of the algorithm, updated_occupancy_grid, updates the map with the new sonar measurements. Each return signal strength s^i is first mapped to a position according to the robot state and sonar heading β. The occupancy of the cell that corresponds to this position is updated, again according to the log odds mapping approach (Thrun et al., 2005). In general, a high signal return strength will result in a high probability of occupancy.

Lines 12–15 in Algorithm 1 correspond to the resampling phase of the algorithm. In this phase, a new collection of particles X_t is generated from X'_t. That is, particles are randomly selected from X'_t and added to X_t, giving higher likelihood of selection to those particles with higher weights.

An example of the effectiveness of the SLAM implementation is shown in Figure 6. In Figure 6(a), the ROV has conducted two sonar scans while resting motionless on the bottom. The ROV is sitting in front of a mound of sediment, resulting in a large number of strong sonar returns falsely indicating a wall just in front (i.e., just to the left of the robot in the image). Once the ROV rises off the floor of the cistern, sonar measurements reveal the absence of walls in front and the algorithm lowers the likelihood of occupation in corresponding cells. The ROV then moves forward (to the left). With no modeling of tether or collisions, the algorithm greatly overestimates the amount of motion the ROV travels, resulting in the mapping of several walls that replicate the original wall [Figure 6(b)]. In Figure 6(c), results when the tether is modeled are presented. The map shows no replication of walls and appears consistent with maps produced from other methods.

5.4. SLAM with Stationary Sonar Scans

When using stationary scans with FastSLAM, the sample_motion_model function does not use actual control inputs. Instead the translations and rotations required for mosaicking the stationary scans were recorded, i.e., they were manually determined with human in the loop. These transformations were easy to obtain but are subject to error. To model this error, a 2D Gaussian distribution was used, with a mean of 0 and a standard deviation σ_m. The value for σ_m was set according to the variation in transformations. Specifically, the operator transformed several sonar scan images, each 10 times, to fit within the mosaic. The standard deviations of each scan's x and y coordinates after the transformation were calculated. Of the scans transformed, the maximum value of $\sigma_m = 0.020$ m was obtained.

Figure 7 shows several maps constructed using the SLAM algorithm with stationary sonar scans. Note the ROV in each image marking the final scan position.

5.5. Localization Using Previously Constructed Maps

Once maps are constructed using any of these techniques, the robot can navigate using a localization algorithm to estimate the robot's state within the map. In this work, particle filter localization was implemented (Thrun et al., 2005). The algorithm was similar to the FastSLAM algorithm presented in Algorithm 1, with step 9 removed to leave the map unchanged over time.

5.6. SLAM with Smart Tether Data

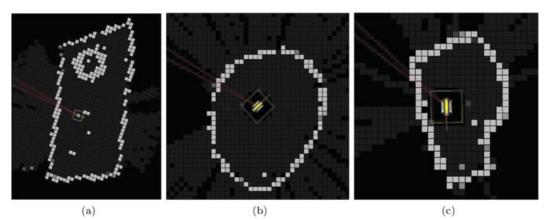This method incorporates the data collected from the sonar scans and Smart Tether into the SLAM algorithm that was

(a)   (b)   (c)

Figure 7. Examples of maps created by inputting static sonar scans into a SLAM algorithm. The ROV in each image indicates the state of the ROV during the final sonar scan. Each cell is 0.20×0.20 m in size, and the lightness of color of the cell represents the likelihood of occupation.
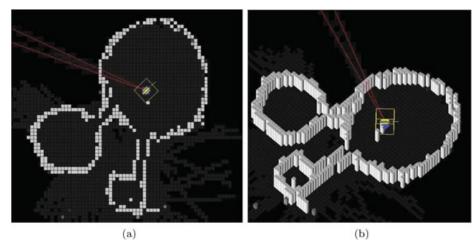
Figure 8. This sonar mosaic is of a cistern located in a priory courtyard in Rabat. This image was created through the use of the sonar scans and constructed to scale through the use of Smart Tether data. Each cell is 0.20 × 0.20 m in size, and the lightness of color (and height) of the cell represents the likelihood of occupation.

introduced in Section 5.3. In this case, using_smart_tether is set to true, so an additional correction step is made that utilizes data collected from the Smart Tether. So, in addition to the first correction step measurement_model_map, the measurement_smart_tether step strengthens the weights of the particles nearest to the true state of the robot. This improvement reduces the margin of error and allows for a more accurate mapping of environments with intricate details such as the cistern shown in Figure 8.

6. RESULTS

Twenty-two of the 24 sites visited had a sufficient water depth and were mapped, providing new and useful in-

formation for archaeological purposes. However, different levels of success were achieved, depending on the method used and the site in question.

The mosaics created for all sites provided information regarding the orientation, scale, and complexity of the cisterns. Figure 9 shows examples from three sites. As can be seen in Figure 9(a), a small rectangular chamber (bottom center of image) lies at the bottom of the access point to site 2 and is connected to a larger reservoir. This was observed in five of the sites.

In Figure 9(b), a tight passage connects two bulb-shaped chambers. The northeast chamber lies at the bottom of the access point of site 6. Upon visual inspection using the video camera, another access point (although covered) was found to be above the southwest chamber.
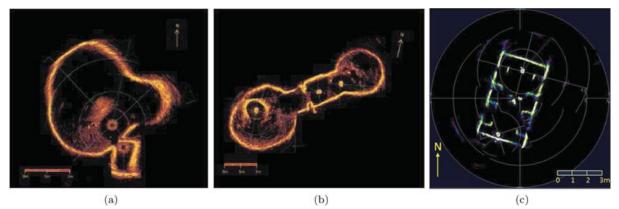

Figure 9. Examples of sonar mosaics created using stationary sonar scans. (a) A map of the cistern in site 2 (Gozo Citadel). (b) A map of the cistern from site 6 (private home in Mdina). (c) A map resembling a floor plan of a house or basement from site 24 (private home in Rabat).
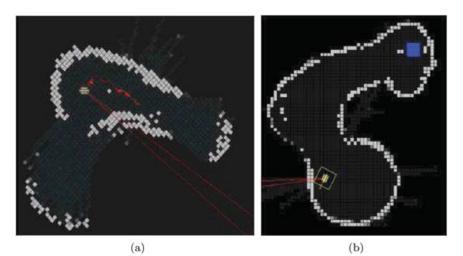
Figure 10. Example of a map created from implementing FastSLAM while in motion (a) and using static sonar scans (b). Each cell is 0.20 × 0.20 m in size, and the lightness of color of the cell represents the likelihood of occupation.

Figure 9(c) shows a more modern cistern found at site 24. This cistern resides underneath several private homes and was found to contain multiple access points from several of the residences. Arches separate the chambers in this cistern, which give it an appearance similar to that of a house or basement.

In validating the SLAM while-in-motion approach to mapping cisterns, data were obtained for only three of the sites. A significant issue that limited data was the inability to drive the ROV with complete control when running the online SLAM algorithm. When the algorithm is running, the ROV must be controlled via computer interface, which was not a problem in previous pool trials. However, navigating narrow passageways required the operator to navigate the robot at very low speeds. A dial was used to limit the joystick gains, which enabled smoother control, but even slower speeds were needed to capture more accurate SLAM data.

Despite these difficulties, it has been shown that the in-motion SLAM algorithm works well in mapping the cisterns. Figure 10(a) shows an occupancy grid map created for site 8. In this example, only 25 particles were used. To determine the number of particles, offline experiments with data obtained at the Cal Poly swimming pool were conducted, where the number of particles was set to 5, 10, 25, and 50. Using 5 or 10 particles resulted in inaccurate maps, whereas using 50 particles had little improvement in accuracy and significantly increased processing time. Using static sonar scans within the SLAM algorithm on the same cistern (site 8) proved effective, as shown in Figure 10(b).

To exemplify the differences between mapping techniques, length measurements of the maps were taken for sites 24 and 8 as shown in Table I. By *length* and *width*, the authors are referring to the length and width of the longest tunnel section in each site. Readers should note that stan-

dard deviations of these lengths were 0.33 m for SLAM in motion and 0.16 m for SLAM while static. The standard deviations are calculated using the fact that walls in the maps are represented with a high likelihood of occupation across three to four cells in wall width [see Figure 10(a)] for SLAM in motion and one to two cells in wall width for SLAM while static [see Figure 10(b)]. The human error (0.02-m standard deviation) was also included for calculating standard deviations in lengths for manual mosaic maps.

The differences in size are due to the number of sonar scans the robot is able to complete while in each position. While stationary, the robot is able to collect multiple sonar scans from a single location, which allows for features from previous scans to be compared with features from the current scan and increases map accuracy. On the other hand, in-motion SLAM prevents the robot from completing full sonar scans at each position, so the number of comparable features is reduced and the accuracy of the maps is degraded.

When referring to Table I, it must be observed that actual truth data for such lengths were not available. Aside from two of the cisterns mapped (sites 15 and 17 in

Table I. SLAM with stationary sonar scans vs. SLAM in motion.

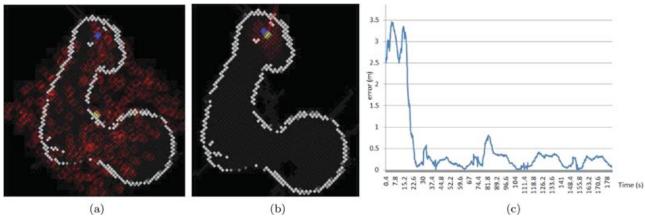| Map type	Site 24		Site 8		
	Length (m)	Width (m)	Length (m)	Width (m)	Std. dev. (m)
Manual mosaics	5.6	1.4	8.9	2.3	0.18
Stationary SLAM	5.4	1.2	8.9	2.3	0.16
SLAM in motion	5.1	1.0	9.6	2.1	0.33

Figure 11. An example of particle filter localization being conducted with data from site 8. Initially, the robot has no idea where it is located, as shown by the randomly distributed set of red particles (a), After a few sonar scans (b), the robot can successfully localize itself with respect to the actual position (blue square). Each cell is 0.20 × 0.20 m in size, and the lightness of color of the cell represents the likelihood of occupation. In (c), the error in position is plotted.

St. Angelo's Fortress), the maps created in this project are the only known maps of the ancient cisterns. Even for sites 15 and 17, the maps were in the form of old blueprints that did not have accurate scale. However, the blueprints did confirm the shapes and relative scale of the maps created using mosaics and SLAM methods.

Finally, the particle filter implementation showed positive results in that the robot always converged to within 0.5 m of the actual location, despite having no knowledge of the initial state. To determine this accuracy, the robot was flown to the location directly below the access point, which is visible by a human operator and designated as the origin of the coordinate frame attached to the map.

An example is provided in Figure 11. In Figure 11(a), 500 particles are shown: each represents a possible state of the robot. The robot's state estimate is calculated as the weighted average of all particle states and is shown in the center of the image. The actual position is shown as a blue square. Despite the fact that the robot has not moved, it can localize itself with only two scans of the area, as shown in Figure 11(b). Figure 11(c) shows the localization error as a function of time.

Tables II and III are provided to summarize characteristics of the 24 sites visited. It can be observed that several sites could not be fully explored due to their being dry. On the other hand, only a few sites had cistern shapes that

Table II. 2008 site characteristics.

Site no.	No. of chambers	Access points	Max dimension (m)	Notes
1	1	1	1.2	Small rectangular cistern with wide access point.
2	2	1	5.0	Small rectangular chamber attached to a large bulb-shaped chamber. A 0.4-m-wide passage connecting the two chambers made navigation difficult.
3	?	?	?	With less than 0.05 m of water depth, it was impossible to fly the ROV.
4	3	2	5.0	Cistern had two small chambers (with associated access points) attached to one large oval chamber of greater depth (~6.5 m). Two ROVs were deployed simultaneously. Without 3D scanning capabilities, it was difficult to obtain usable 2D maps. Poor visibility.
5	3	2	4.0	Cistern had two small chambers (with associated access points) attached to one large circular chamber.
6	2	2	7.0	This dumbbell-shaped cistern had two bulb-shaped chambers connected by a small passage. Much debris made sonar returns noisy. Mapping was difficult.
7	?	?	?	With less than 0.05 m of water depth, it was impossible to fly the ROV.
8	3	2	5.0	This cistern had three circular chambers connected with tunnels not much smaller in width than chamber diameters (~2.5 m). Visibility was excellent.

Table III. 2009 site characteristics.

Site no.	No. of chambers	Access points	Max dimension (m)	Notes
15	1	1	6.0	This cistern was oval in shape and easy to map because a single scan reached all walls.
16	1	1	4.0	This cistern was an old guard tower filled with water. Semicircular in shape, the cistern was difficult to navigate because multiple levels existed. Autonomous depth control made mapping easier.
17	1	1	10.0	This parallelogram-shaped cistern was large and easy to navigate within. A 2.0-m-wide depression in the floor caused sonar returns.
18	1	1	4.0	This cistern was oval in shape and easy to map because a single scan reached all walls.
19	1	2	6.0	This cistern was oval in shape and easy to map because a single scan reached all walls.
20	1	1	1.8	This cistern was circular in shape and possibly much larger than was accessible. A large pile of broken pottery littered the floor of the cistern. Several pieces were extracted using the ROV gripper (for later examination).
21	1	2	6.0	This cistern turned out to be the same as site 19 (approached from another access point).
22	1	1	1.5	This cistern, although attached to a larger system, was almost completely dry, which severely limited the ROV's ability to maneuver.
23	1	1	2.0	This cistern was almost completely dry, severely limiting the ROV's ability to maneuver.
24	1	1	5.0	A rectangular chamber of dimensions 2.5 × 5.0 m, this cistern had excellent visibility, which allowed operators to see a series of arches not seen in any other cistern. Unfortunately, the arch pillars made mapping via sonar difficult.
25	1	1	1.5	This cistern, although attached to a larger system, was almost completely dry, which severely limited the ROV's ability to maneuver. Worse, the access point was very small, making it difficult to enter the cistern.
26	1	1	2.0	This cistern was almost completely dry, severely limiting the ROV's ability to maneuver.
27	1	1	2.0	This cistern was almost completely dry, severely limiting the ROV's ability to maneuver.
28	1	1	2.0	This cistern was almost completely dry, severely limiting the ROV's ability to maneuver.
29	3	3	10.0	One large circular chamber was connected to one smaller square chamber and one smaller circular chamber via tunnels.
30	2	2	7.0	This dumbbell-shaped cistern had two bulb-shaped chambers connected by a small tunnel. Similar to site 29 in the same location, mapping was relatively easy.
31	0	1	15.0	This cistern was a well access point acting as a hub for three tunnels of 1.0-m width. Tunnels were long and featureless, making them difficult to map without the aid of a Smart Tether.
32	1	1	30.0+	This cistern started as a long tunnel that went farther than the tether's length, making it impossible to map the entire length. Making it more difficult was the fact that the Smart Tether was not working and the tunnel walls were featureless aside from one 90-deg bend. A final difficulty occurred when the tether became snagged 15.0 m down the tunnel in a bottleneck caused by two rocks.

varied in depth, making the methods used both possible and useful. However, archaeologists made it clear that having 3D maps in such situations would be beneficial. Another issue that arose in a few of the sites was the presence of long featureless tunnels within which SLAM became inaccurate. In these instances, having the Smart Tether was helpful.

7. CONCLUSIONS AND FUTURE WORK

The two cistern mapping expeditions in Malta and Gozo successfully constructed maps for use in archaeological studies of these ancient water storage systems. In each cistern, a small ROV was deployed that collected Smart Tether and sonar data from various positions in the cistern.

Using these data sets, six methods for mapping and localization were investigated. Stationary scan methods, including scan mosaicking and FastSLAM, worked well. Implementing FastSLAM while moving had success but was validated by only a few data sets. Particle filter localization also worked very well in that state estimates converged to actual states despite there being no knowledge of initial conditions. Whereas the core FastSLAM and particle filter algorithms were not changed for this research, the use of the ROV's dynamic model, the sonar sensor model, and tether model in these algorithms was new.

In the future, scalability of the FastSLAM implementation will be improved. The current implementation requires large memory constraints, which could be remedied with multiresolution grids (e.g., octrees). Work done in Fairfield et al. (2006) provides guidance on this issue and demonstrates that there is a realistic solution. Second, a sonar module will be placed on the side of the robot such that the scan plane is perpendicular to the vertical axis. This will provide sonar scans across the vertical plane and enable the construction of 3D maps. In this scenario, position keeping while the robot rotates on the spot (a current ability in stagnant environments) is required. A hurdle to overcome for 3D mapping will again be dealing with scalability. Third, new cisterns will be visited across Malta and possibly Italy, resulting in a variety of previously unencountered environment features (e.g., multifloor chambers) and related issues to be resolved.

ACKNOWLEDGMENTS

REFERENCES

Baker, C., Morris, A. C., Ferguson, D., Thayer, S., Whittaker, C., Omohundro, Z., Reverte, C., Whittaker, W. L., Haehnel, D., & Thrun, S. (2004, April). A campaign in autonomous mine mapping. In Proceedings of IEEE Conference on Robotics and Automation (ICRA), New Orleans, LA.

Eliazar, A., & Parr, R. (2003, August). DP-SLAM: Fast, robust simultanous localization and mapping without predetermined landmarks. In Proceedings of International Joint Conference on Artificial Intelligence, Acapulco, Mexico (pp. 1135–1142).

Eustice, R., Singh, H., & Leonard, J. (2006). Exactly sparse delayed-state filters for view-based SLAM. IEEE Transactions on Robotics, 22(6), 1100–1114.

Eustice, R., Singh, H., Leonard, J., Walter, M., & Ballard, R. (2005, June). Visually navigating the RMS Titanic with

SLAM information filters. In Proceedings of Robotics Science and Systems, Cambridge, MA (pp. 57–74).

Fairfield, N., Kantor, G., & Wettergreen, D. (2006). Real-time SLAM with octree evidence grids for exploration in underwater tunnels. Journal of Field Robotics, 24, 3–21.

KCF Technologies (2008). The new Smart Tether from KCF Technologies. Navigation & positioning (pp. 80–82). State College, PA: KCF Technologies.

Mahon, I., & Williams, S. (2004, December). SLAM using natural features in an underwater environment. In The Eighth International Conference on Control, Automation, Robotics and Vision, Kunming, China (pp. 2076–2081). IEEE.

Majumder, S., Rosenblatt, J., Scheding, S., & Durrant-Whyte, H. (2001). Map building and localization for underwater navigation. In ISER '00: Experimental Robotics VII (pp. 511–520). London, UK: Springer-Verlag.

Moravec, H. P. (1988). Sensor fusion in certainty grids for mobile robots. AI Magazine, 9, 61–74.

Moravec, H. P., & Elfes, A. E. (1985, March). High resolution maps from wide angle sonar. In Proceedings of the 1985 IEEE International Conference on Robotics and Automation, Sacramento, CA (pp. 116–121).

Ribas, D., Ridao, P., Neira, J., & Tardós, J. D. (2006, October). SLAM using an imaging sonar for partially structured underwater environments. In IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China.

Ribas, D., Ridao, P., Neira, J., & Tardós, J. D. (2007, October). Underwater SLAM in a marine environment. In IROS, San Diego, CA (pp. 1455–1460).

Ribas, D., Ridao, P., Tardós, J. D., & Neira, J. (2008). Underwater SLAM in man-made structured environments. Journal of Field Robotics, 25(11–12), 898–921.

Richmond, K., & Rock, S. M. (2006, September). An operational real-time large-scale visual mosaicking and navigation system. In Proceedings of MTS/IEEE OCEANS Conference, Boston, MA.

Sakai, H., Tanaka, R., Mori, T., Ohata, S., Ishii, K., & Ura, T. (2004, April). Underwater video mosaicing using AUV and its application to vehicle navigation. In Proceedings of International Symposium on Underwater Technology, Taipei, Taiwan.

Sim, R., Elinas, P., & Little, J. (2007). A study of the Rao–Blackwellised particle filter for efficient and accurate vision-based SLAM. International Journal of Computer Vision, 74(3), 303–318.

Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. Cambridge, MA: MIT Press.

Wang, W., & Clark, C. M. (2006, May). Modeling and simulation of the VideoRay Pro III underwater vehicle. In Proceedings of IEEE OCEANS Asia Pacific IEEE Conference, Singapore.

Williams, S. B., Newman, P., Dissanayake, D., & Durrant-Whyte, H. (2000, April). Autonomous underwater simultaneous localisation and map building. In IEEE International Conference on Robotics and Automation, San Francisco, CA.