

Pipeline for Reconstruction and Visualization of Underwater Archaeology Sites using Photogrammetry

Sebastian M. T. Seibert von Fock*, Sara Bilich*, Katie Davis*, Vaibhav K. Viswanthan[†], Zayra Lobo[‡], Jessica Lupanow[†], Christopher Clark[†], Timmy Gambin[‡], Zoë Wood*

*Department of Computer Science, California Polytechnic State University - San Luis Obispo
San Luis Obispo, CA, 93401, USA
sseibertvonfock@gmail.com, sbilich@calpoly.edu, davis.kd.m@gmail.com, zwood@calpoly.edu

[†]Department of Engineering, Harvey Mudd College
Claremont, CA, 91711, USA

(vviswanathan, zlobo, jlupanow, clark)@g.hmc.edu

[‡]Department of Classics and Archaeology, University of Malta
Msida, Malta
timmy.gambin@um.edu.mt

Abstract

This paper presents a complete pipeline to create accurate 3D reconstructions of underwater archaeological sites from video data captured by an Autonomous Underwater Vehicle (AUV). The fieldwork and data gathering for this project was completed off the coast of the island of Malta. While in the past the data gathering process for 3D reconstructions of underwater sites has been accomplished via multiple divers and expensive photographic equipment, this project aims to automate that process. The general pipeline for creating the 3D reconstructions began with the AUV completing several low-pass missions over the site of interest (historical ship wrecks) to gather video footage from different angles. The video data was then processed through a variety of methods to generate multiple usable frames. Next, Agisoft Photoscan, a photogrammetry software, was used to create models which were stitched together or further edited for better results. Finally, the models were placed in a custom-developed 3D rendering and visualization software. These 3D reconstructions could then be used by archaeologists for study and the creation of educational visualizations of previously unmapped and hard-to-access archaeological sites. Educational visualizations can be used to excite the general public and encourage further funding of marine archaeology and preservation.

keywords: Photogrammetry, Reconstruction, Autonomous Vehicles, Marine Archaeology

1 Introduction

Underwater archaeological sites are of historical interest to marine archaeologists, and images and stories of

shipwrecks capture the imagination of the public. Being able to capture video footage and create 3D reconstructions allows archaeologists to study such sites and share their relevance with the public from the comfort of land. Video footage has been used to tell the story of such historical wrecks as the Titanic, while 3D reconstructions allow a user to revisit and explore a computer graphics model of such shipwrecks at their leisure. To create 3D reconstructions with a low implementation cost[7], the use of photogrammetry in underwater archaeology is undeniably an important tool. ROVs and divers have explored and captured video of wrecks[2], but both of these options require a large time and cost commitment. Autonomous Underwater Vehicles (AUVs) have been used in several studies and projects for accessing underwater archaeology sites because of their low time and cost commitment [5]; many of the pipelines using AUVs use photogrammetry to reconstruct the sites into usable 3D digital models. [6].

In this paper, a complete pipeline is described to create 3D reconstructions and visualizations of underwater archaeological sites. This work is focused on archaeological sites in the coastal waters of Malta, a country with rich maritime history, and it focuses primarily on the details and unique challenges of the reconstruction process using underwater video footage.

The pipeline can be broken into three larger categories:

- (1) Data Acquisition
- (2) Reconstruction of a Computer Graphics Model
- (3) Visualization and Applications

2 Data Capture

2.1 Hardware

The captured video data of the marine archaeological sites was gathered using one or two GoPro Hero 2 cameras attached to an OceanServer Iver2 Autonomous Underwater Vehicle (AUV). The AUV was ballasted by the research team so that it would have the proper buoyancy for the different amounts of salinity in the water.

2.2 Process

We captured data at two separate sites off the coast of Malta: the wreck of the X127 Lighter “Coralita” in the Marsamxett Harbour of Manoel Island (approx. 5m depth), and the wreck of a Bristol Beaufighter (approx. 30m depth) at the mouth of St. Julian’s Bay.

The AUV was deployed either directly off the coast or from a small boat. It received its mission trajectories through a Wi-Fi connection; this is also how the team communicated with the AUV while it was in the water to start and stop missions. Both traditional lawnmower paths (shown in Figure 1) and Rapidly-Exploring Random Tree (RRT) paths with new sampling methods (shown in Figure 2) [8] were used to dive over the sites.

After extended periods of time in the water (over 40 minutes), the camera’s SD card and battery would be replaced with new ones to minimize the risk of the camera either shutting down or to halting its recording during the missions.

Each mission was recorded with a start time, an end time, and whether it completed successfully. In addition, an identical surface mission was run before each dive mission to give a visual confirmation of a safe mission trajectory.

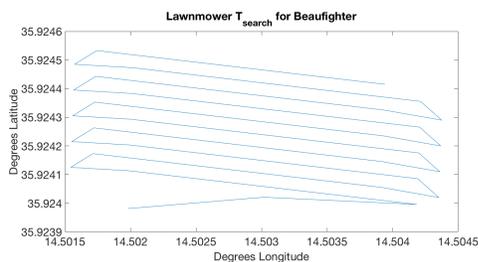


Figure 1: Lawnmower Path [8]

3 Reconstruction

3.1 Image Processing

Rough Editing After the video data was transferred from the SD cards, it had to be edited to obtain footage with the maximum number of views of the archaeological sites. If

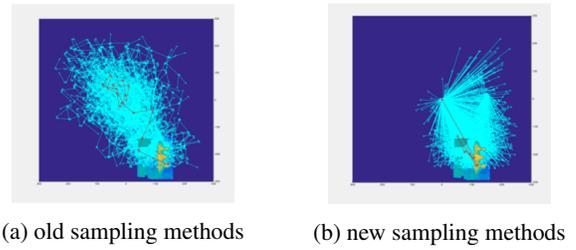


Figure 2: RRT Path [8]

the time in the footage when the site was visible was known, then the relevant part of the video could be easily selected. There were two main methods that were used to find the time in the footage (often over an hour long) during which the site was visible (often just several seconds).

The first method was to sample evenly-spaced frames from the footage and detect any view of the site using an algorithm. To accomplish this, a frame was pulled from the video footage every 5 seconds and numbered chronologically. An image processing program was used to take the average color of each of the frames and compare it to a random pixel in the frame, with any frame culled if Equation 1 were true.

$$\left| \frac{1}{n} \sum_{k=0}^{n-1} (\vec{C}_k) - \vec{C}_r \right| < t \quad (1)$$

Equation 1: Where n is the number of pixels in the frame, \vec{C}_k is the RGB vector of pixel index k , \vec{C}_r is a random pixel’s color, and t is some threshold value.

For any frame where the difference from the average pixel values was below a given threshold (signifying that the pixels were monochrome and most likely pure blue from open ocean footage), the frame would be deleted. The remaining frames were reviewed to see which contained part of the site rather than debris or seaweed. This method was especially useful in long lawnmower missions of deeper sites such as the Beaufighter.

The second method was to play the footage at three to four times normal speed and visually see which parts included sections of the site and which did not. This method was especially useful when the AUV was in shallow water since there was a lot of footage of debris on the seafloor (shown in Figure 3) which would trigger as an interesting site using the first algorithm.

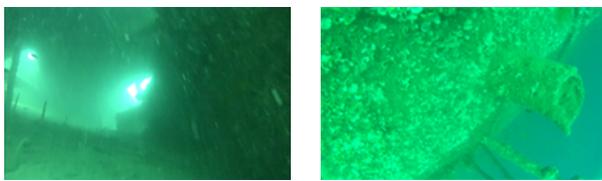
Sampling Frames Once short video files of dives over the sites were edited out of the longer video files, the short video files were broken down into frames that could be used for photogrammetry, with care paid that FFmpeg not automatically downscale the image’s resolution (PNG and TIFF were both acceptable). For most dives, the frames



Figure 3: Debris and rocks on seafloor

were pulled out at 2 FPS; for some of the more detailed quicker dives, the frames were pulled out at 10 FPS to get more detail.

Fine Editing All of the frames were then examined to identify any that were sub-optimal, thus lengthening the photogrammetry process without adding any data. Video data in an underwater setting from a moving vehicle creates a variety of unusual frames. The most common unusable frames were those that were too blurry, as can be seen in Figure 4a. Other unusable frames were those with insufficient light and lack of contrast.



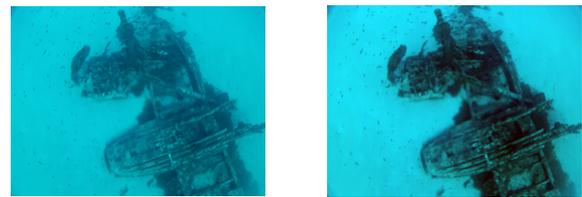
(a) Too blurry image (b) Good quality image

Figure 4: Different frames pulled from video data

Color Correction (optional) A final optional part of the image processing pipeline was color correction. Color correction is usually not recommended for photogrammetry; however, in our project it sometimes had a positive effect for more accurate texture generation while having no effect on the model generation. In general, the contrast of the images was raised, blues and greens were lowered, and brightness was increased. In addition, an S-curve was used for the exposure curve (darken shadows, raise highlights). This color correction made many images sharper and clearer; the difference can be seen in Figure 5.

3.2 Photogrammetry

Load Images into Chunk There are many software programs available that can use photogrammetry to construct 3D models from frames. Agisoft Photoscan Professional is a common software that has been used by other similar projects[1][7]. Photogrammetry requires multiple pictures of the same object and uses triangulation to reconstruct the depth[4]. For each site, the frames were loaded into a chunk.



(a) Frame before color correction (b) Frame after color correction

Figure 5: Frame of Bristol Beaufighter

If the site had many distinct features or too many frames (significantly more than 100), multiple chunks were used.

Although Agisoft Photoscan Professional has an option to set the camera type to fisheye under the camera calibration option, this was not used on most of the reconstructions and had to be tested to be on a case-by-case basis even though all the footage was shot using a fisheye lens. This is most likely because the water neutralized the fisheye effect in most cases[1].

Masking (optional) By masking out the parts of the frame that were not part of the site (as can be seen in Figure 6) the extra amount of surface reconstructed can greatly be reduced. In this project, which used many low resolution frames, this method was not a viable option since masking takes a considerable amount of time to complete and there were often over 200 frames per site. However, this method has proven to be very useful for reconstruction in Photoscan, especially when using a few (below 30) high resolution images[3].



Figure 6: Masking out seafloor in Agisoft Photoscan

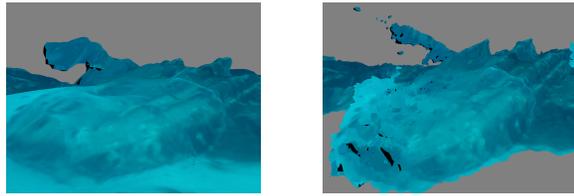
Align Photos To generate a point cloud the images first had to be aligned, and a tie point cloud had to be created. This was done under the workflow tab of Photoscan. In our project, we found the following settings to be the most efficient:

Accuracy	High
Pair Preselection	Disabled
Key Point limit	40000
Tie Point limit	0

Whenever 0 is used as a setting in Agisoft Photoscan, it means unlimited; in this case, we are not restricting the amount of tie points since it does not significantly increase the processing time.

Generate Point Cloud To build the dense point cloud the quality was set to Medium, while the Depth filtering was set to Aggressive.

For some sites the quality was raised to high or even ultra high. This would easily double or triple the amount of time to complete the whole pipeline, but in some cases was warranted to get more detailed reconstructions; the improvement in quality in one reconstruction of the Bristol Beaufighter can be seen in Figure 7.



(a) Medium quality setting (b) Ultra-high quality setting

Figure 7: Bristol Beaufighter reconstructions using different quality settings for generating Dense Point Clouds

Edit Point Cloud (optional) Since most of the sites’ point clouds had a strong color difference between the site and unwanted points, it was easy to edit the point cloud to remove the unwanted geometry. This was done in Agisoft Photoscan by selecting points of the site by color, inverting the selection, and then deleting them. This resulted in a reconstruction with much less unneeded geometry, making it possible to generate meshes at much higher resolutions while still keeping processing time low.

Generate Mesh To generate the mesh the most common settings for this project were the following:

Surface type	Arbitrary
Source data	Dense cloud
Face count	High (180,000)
Interpolation	Enabled (default)

Face count decimates the mesh after it is generated to the specified number of faces. Raising this decreases processing time in this step but may increase processing time in further work with the model. To turn off decimation, select custom and set it to 0 (unlimited).

Generate Texture Agisoft Photoscan provides the option to generate a texture based off of the pictures. In this project, these settings were used for all models:

Mapping mode	Arbitrary
Blending mode	Mosaic (default)
Texture size/count	4096 x 1
Color correction	Disabled

Merging Chunks Adding markers was necessary if more than one chunk was used in the reconstruction process. Markers had to be added to distinctive geometry or textures on each chunk that overlapped with some other chunk.

These markers were then used to align the chunks by going through the Align Chunks menu. “Fix scale” was selected to scale the different chunks so that their markers matched up.

Finally, the chunks that were then aligned had their dense point clouds, models, and markers merged. The texture, and sometimes the model, had to be generated again from the dense point cloud after merging.

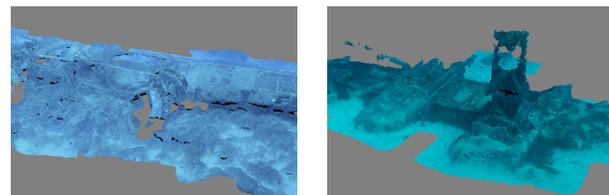
3.3 Reconstruction Results Compared to Traditional Means

Our current pipeline gave us very promising results, as can be seen in Figures 7, 8b, 9, and 10.

We compared these models to a reconstruction created from photographs taken by underwater SCUBA divers using the same software (Agisoft Photoscan). The two models can be seen in Figure 8. As can be seen in Figure 8, the model from the divers has slightly higher detail on the flat portions of the wing but is missing significant portions of the wreck, especially near the front part of the wreck and the landing gear.

We ran both of the models through the algorithm described in Section 4.2 of this paper and received a difference of approximately 1.55 units on a 10x10x10 3D grid scale. This is a significant difference, as the max difference possible is $2\sqrt{2}$ (around 2.8) as explained in Section 4.2.

We believe our reconstruction is superior to the traditional reconstruction since we are able to provide a more complete model that shows more significant parts of the wreck, as proven both visually and quantitatively.



(a) Reconstruction from Diver Pictures (b) Reconstruction from AUV Pictures

Figure 8: Bristol Beaufighter Reconstructions

4 Visualization and Application

4.1 Virtual Visualization Environment

Lighting and Setup Test By loading the reconstructions into Blender, an open-source 3D modeling and animation program, a pre-visualization step was used to test how the reconstructions would look in a variety of different arrangements and lighting setups.

In general, it was found that some strong point lights shining from one side of the hull of the reconstruction plus a weaker point light from the top gave the best results. This setup can be seen in Figure 9.

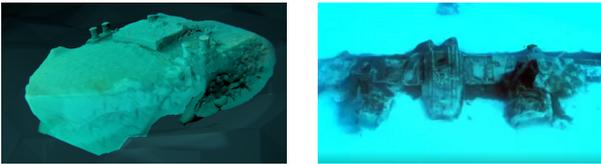


Figure 9: Lighting setup during test phase

Rendering and Camera To allow for control over the underwater visualizations and to create compelling visualizations that allow a user to experience the underwater environment, a custom world for the visualizations was created in OpenGL. Ultimately, this application was developed for the creation of educational visualizations that gave the user an understanding of the wreck’s underwater setting and the AUV data gathering process.

Underwater Environment Enhancements There were several enhancements made to the world to simulate an underwater environment. These included creating a geometric context for the wreck and several rendering effects to simulate underwater lighting such as adding caustics and blurring in the distance. The caustics were projected onto the surface from a virtual camera facing straight down and used several pre-rendered caustic texture files.

To achieve the effect of the decreased view that is experienced when underwater, a green-red color was subtracted more and more from the render the further the fragment was from the camera proportional to $\ln(d)$ where d is the distance from the camera.

4.2 Calculating Objective Parameter for Reconstruction Quality

Alignment This project is a part of a larger research project that includes autonomous path planning for the AUV. In order to improve the autonomous path planning, there was a need to calculate an objective parameter for the quality of



Figure 10: Final render of OpenGL visualization

a reconstruction (if one reconstruction of a site was “better” than other reconstructions of the same site). To accomplish this, we developed a program that would compare the average difference of model A, a base “perfect” model, and model B, the model obtained from the reconstruction.

The average difference between models was computed by looking at the vertex positions of model A and computing the length to the closest vertex of model B. Before the algorithm could run, the models were aligned and scaled manually to overlap as much as possible with each other using an external program; in this project we used Blender.

Spatial Data Structure Since the models that were generated often had a large number of vertices (over 1,000,000), an efficient spatial data structure had to be used. A uniform spatial grid composed of $10 \times 10 \times 10$ voxels was created for each model. These will be referred to as Grid A and Grid B. The vertex points of both models were scaled from 0 to 9.99 in each dimension. Each point would then be placed into a voxel by taking the floor of its location coordinate and placing it into the voxel with that index (e.g. point at (4.32, 0.19, 9.89) would be placed into voxel[4][0][9]).

Difference Computation The program would loop over each voxel in Grid A and then loop through all the points in that voxel. For each point in that voxel in Grid A, the program first checked if there were any points in the voxel of the same index in Grid B. If there were, it would calculate the distances from the point in Grid A to all the points in the voxel in Grid B, as can be seen in Figure 11a. It would then choose the shortest of these distances to store in a variable, *pointDistance*. If there were no points in the voxel of the same index in Grid B, the program would check all 26 neighboring voxels in 3D space for any points. It would then calculate the distance to these points from the point in Grid A, as can be seen in Figure 11b. Again it would choose the shortest one and store it in *pointDistance*. If there were no points in any neighboring voxels either, *pointDistance* would be assigned a static value of $2\sqrt{2}$.

After calculating *pointDistance* for all points in each voxel, the program averaged the *pointDistance* for each voxel and stored this value in a variable, *voxelDistance*.

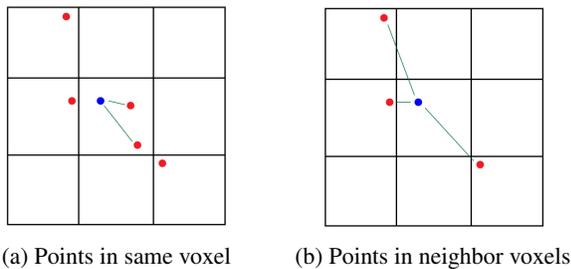


Figure 11: Visualization of how the distance algorithm works in 2D space

These *voxelDistance* were then averaged again to compute the overall distance between model A and model B. This metric could then be used in the path planning process to see which mission path would allow the AUV to collect data for the most accurate reconstruction possible.

Results For example, we compared the geometry of the Bristol Beaufighter reconstructions from a model generated from two lawnmower passes vs. four lawnmower passes. The difference between the two reconstructions was approximately 0.31 units on a 10x10x10 3D scale. This shows that having more than two passes over important features of the wreck is not significantly beneficial, as there is very little difference between the two models.

4.3 Future Goals

We have presented our pipeline and results for reconstructing and visualizing underwater archaeological sites. This work was conducted in the coastal waters of Malta, and results from two differing wreck sites are shown. Using an existing photogrammetry software, we discussed the data gathering and pipeline configurations needed to produce 3D reconstructions. We also discussed the two applications created for this project, an underwater visualization software and a simple reconstruction quality metric. As work continues on this project, these two applications will be improved and the pipeline further automated.

The goal of this project is to produce final visualizations that can be used by archaeologists to better study specific marine archaeological sites. Using the entire pipeline, the program gives archaeologists a view of how the sites currently look. This information can then be used for archaeologists to study the sites on their own time in an easier-to-access environment rather than having to invest in expensive dive missions. In addition, this method gives archaeologists a broad overview of the full scope of the sites. Finally, the development of this pipeline can be used to show the state of underwater archaeological sites over time to determine what effect the water, commercial fishing, or divers have had in deteriorating the sites.

To make this an even more useful tool for archaeologists, there are still several key features to be implemented. These include to-scale models with accurate measurement tools, a time history to see changes in the sites over time, and an easier-to-navigate interface showing multiple sites.

In addition to being useful to archaeologists, the reconstructions and visualizations could be used to raise awareness of marine archaeology. By providing the public with an interactive experience where they can “dive” through these sites as they appear on the ocean floor, they would feel more involved in the rich history of these sites and better understand the need for further exploration and preservation. Future work includes improving the visual appeal of the underwater environment, automated fly-throughs that focus on the most visually appealing and interesting sections of the reconstruction, and a 360 degree view of the visualization in a virtual reality environment.

References

- [1] T. Van Damme. Computer vision photogrammetry for underwater archaeological site recording in a low-visibility environment. In *The International Archives of the Photogrammetry*, volume XL-5/W5, Apr. 2015.
- [2] T. Gambin. A phoenician shipwreck off gozo, malta. 10, 2011.
- [3] A. Koutsoudidis. Multi-image 3d reconstruction data evaluation, 2014.
- [4] Gustavo Olague and Roger Mohr. Optimal camera placement for accurate reconstruction. In *RR 3338, INRIA, 1998*, 1998.
- [5] B. Bingham *et al.* Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle, 2010.
- [6] B. Foley *et al.* The 2005 chios ancient shipwreck survey: New methods for underwater archaeology. 78.2:30351–30384, 2009.
- [7] P. Drap *et al.* Underwater photogrammetry and object modeling: A case study of xlendi wreck in malta. In *Sensors*, pages 30351–30384, 2015.
- [8] V. Viswanathan *et al.* Auv motion-planning for photogrammetric reconstruction of marine archaeological sites. 2016. 2017 IEEE International Conference on Robotics and Automation.