

# Structured Light Profilometry for Obstacle Detection and Navigation in Underwater Autonomous Vehicles

Ari Porad  
RoboLab, Olin College of Engineering

Spring 2021

## Abstract

I researched the use of structured light profilometry for underwater autonomous navigation and obstacle detection. The project progressed to the prototype phase, where I built a desktop test bed that contained a line laser, a camera (with a bandpass filter), and a Raspberry Pi for computation. The system was capable of successfully detecting the position of the laser beam within the camera frame, but more development is needed to properly derive physical object positions, and to differentiate between multiple objects.

## Source Code

All source code for this project can be found at: <https://github.com/ariporad/robofish-SP21>.

## 1 Introduction

The Olin RoboLab is currently engaged in a multi-year effort to develop a versatile, efficient, and bio-realistic autonomous robotic fish for littoral surveying. A significant area of research for this project is the development of a system for navigation and obstacle avoidance. Our target underwater terrain is frequently dark, rendering traditional computer vision systems inoperable. Any form of broad active illumination to supplement a traditional computer vision system is infeasible due to power limitations.

Structured light profilometry offers a promising alternative to these traditional vision systems. By using a focused laser beam, the power requirements are drastically reduced without sacrificing illumination distance. While a line laser is only capable of calculating depth data in one dimension (therefore producing a two-dimensional result), the forward motion of the fish can be used to recreate the third dimension over time. This technology has a wide-ranging array of possible uses, from underwater surveying to submarine navigation.

Structured light is often used in highly-controlled factory contexts to measure surfaces without touching them (Salvi et al. 2010; Tsai and Hung 2005). It was also previously researched by students in the Olin RoboLab on a previous incarnation of the robot fish project (Barrett, Vandor, and Kohler 2018). However, it has to the best of the author’s knowledge never been perfected in an underwater, uncontrolled context. This report details my research into this topic over the course of the Spring 2021 semester. While much progress was made, far more is yet to be done to continue validating this promising technology.

## 2 Methods

### 2.1 Software

#### 2.1.1 Platform

I used a Raspberry Pi 4 Model B as the system’s controller. The Pi 4 is affordable, powerful, and easy to use. It has a robust camera ecosystem (see 2.2.1), a mechanism for interfacing with electronics directly (via GPIO pins), and is generally well-suited to robotic experimentation.

All code for this project was written in MATLAB. MATLAB is well suited to rapid prototyping, on account of being rather flexible and having a large set of (mostly) easy-to-use and turn-key toolboxes. For example, I was able to use MATLAB’s built in camera calibrator to correct for lens distortion in less than an hour. However, MATLAB is generally poorly suited to writing large amounts of scalable or performant software (it’s very much a mechanical engineer’s programming language).<sup>1</sup>

Additionally, MATLAB has a [built-in Raspberry Pi integration](#). With it, all code runs on a separate laptop and simply communicates with the Pi over Wi-Fi. This configuration prevents the Pi’s limited processing power from being a bottleneck, and makes development drastically easier (no need to write code over SSH, or synchronize files, etc.).

## 2.2 Hardware

### 2.2.1 Optics: Camera

For the system’s main camera, I used a Raspberry Pi [High Quality Camera](#). I chose the HQ Camera for its simple integration with the Raspberry Pi (and with MATLAB) and its flexibility: I anticipated wanting to experiment with various lenses, and so having a removable lens was enticing. Additionally, having a standard-size removable lens opened the possibility of using a [C-Mount bandpass filter](#) (see below), which would have been extremely clean and convenient. However, I wasn’t able to successfully source a C-Mount lens—nor did I need to use different lenses—so the simplicity of a standard Raspberry Pi [Camera Module](#) would have been preferable.

I used a standard Raspberry Pi HQ Camera [6mm Wide Angle CS-mount lens](#). In front of the lens was a [535nm bandpass filter](#)<sup>2</sup>, which filtered out all light other than the laser light (or other light the same color). This resulted in the camera’s image being essentially 1 bit—if there was any light at all, then it had to be the laser(see [2.2.1](#)). In practice, this was decent but imperfect because the laser was bright enough that it would also illuminate other things (especially the floor, see [Figure 5a](#)), so there still needed to be some filtering of laser vs. not laser in software.

### 2.2.2 Optics: Laser

A [relatively-inexpensive line laser](#) was used. While this particular part was easy to get started with, its circular body made it difficult to mount precisely (this resulted in some issues with the laser beam not being perfectly horizontal, which made processing more complicated—see [3.2](#)). Additionally, it was far brighter than required for desktop testing, which possibly contributed to the aforementioned light bleed.<sup>3</sup> It also wasn’t eye safe, which made testing procedures substantially more cumbersome (see [2.3](#)).

### 2.2.3 Testbed

I built a 3D-printed testbed, which held the laser and camera at a precise angle (along with a the Raspberry Pi itself). The laser was held in place by a wood screw at the manufacturer’s recommendation<sup>4</sup>. The Pi and Camera were intended to be held in place by screws, but due to tolerance errors had to be held in place with tape. The filter was pinned in place by the mount and the camera.

## 2.3 Testing

Because the laser I used wasn’t eye safe<sup>5</sup>, all testing was conducted in an isolated environment—affectionately referred to as the *Laser Cave*.<sup>6</sup> The Laser Cave was constructed by duct-taping cardboard around the feet of a table, ensuring that it was appropriately and fully sealed. The Laser Cave was also outfitted with a means of measuring the distance between the prototype and the target, and with a power supply.

---

<sup>1</sup>I had a brief foray into using C++ and OpenCV instead of MATLAB, which did indeed produce better code at the expense

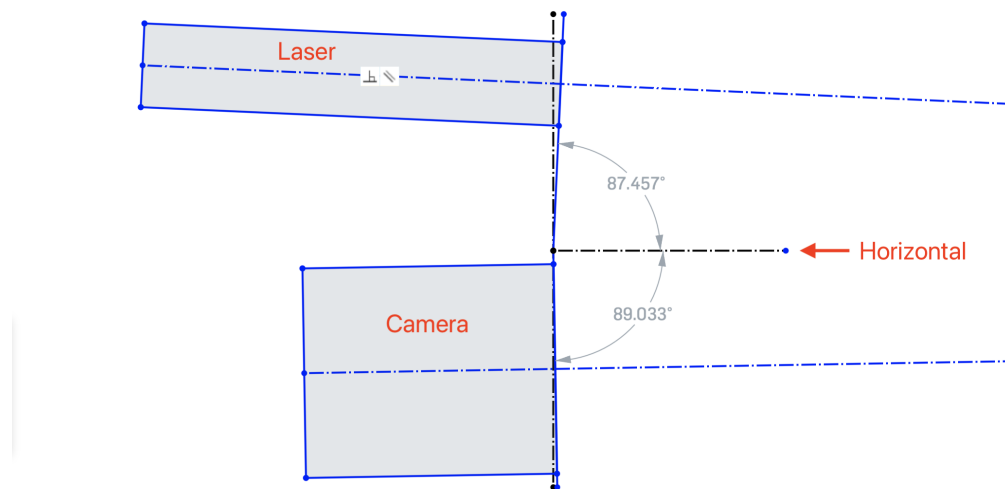


Figure 1: CAD rendering of testbed geometry, showing the precise angles of the laser and the camera. No attempt was made to quantify the precision or tolerance of the 3D printer as it pertains to these angles. Blue lines extending to the right represent the center of the camera’s view and the center of the laser beam, and intersect at 1 meter (not shown).

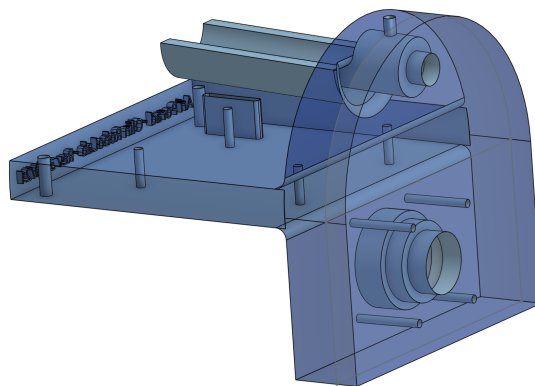


Figure 2: Transparent CAD rendering of the testbed, showing its internal geometry. Camera, Laser, Pi, and screws are not shown.

of having a much lower development velocity. I ultimately returned to MATLAB in the name of efficiency.

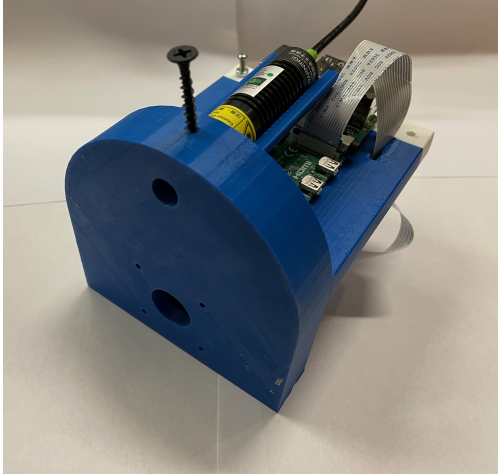
<sup>2</sup>This is the same model I had hoped to get in a C-Mount configuration, in a different size.

<sup>3</sup>It’s possible this wouldn’t be a concern when testing the system underwater (ie. without a floor), but the necessity of desktop testing makes the issue worth addressing irregardless.

<sup>4</sup>I know, I know, using a wood screw to pin a cylinder in place is bad.

<sup>5</sup>While I had appropriate eye protection, it was infeasible to provide adequate protection to everyone in the RoboLab.

<sup>6</sup>Naming Credit: Esme Abbot, 2021



(a) Front

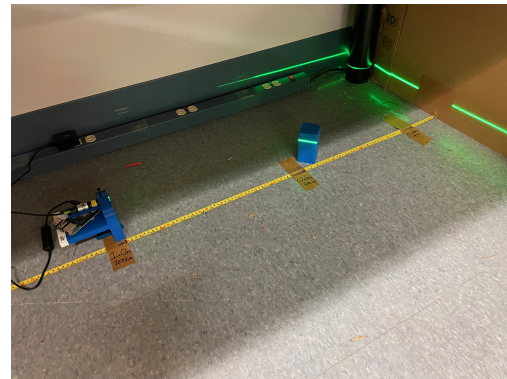


(b) Back

Figure 3: 3D-printed testbed prototype. White section is due to running out of filament. The screw in the back-right was to hold up the Pi plate—printing a foot would have required significant amounts of support.



(a) Outside



(b) Inside

Figure 4: The Laser Cave isolated testing environment, constructed from a table, cardboard, and duct-tape.

## 3 Results

### 3.1 Platform

The Raspberry Pi was a versatile and highly effective development platform, and I would endorse continuing to use it. The Raspberry Pi HQ Camera, however, ended up being rather difficult to properly focus (especially considering its labels are rather counter-intuitive). Especially considering our inability to source a C-Mount filter, evaluating simpler options for a camera might be worthwhile.

Additionally, MATLAB’s Raspberry Pi integration leaves much to be desired. While it does make it easy to work with images, it’s slow and buggy (in addition to MATLAB’s general poor performance and instability, especially on macOS). In the future, investigating a middle ground between C++ and MATLAB (perhaps Python and OpenCV) would be worthwhile.

### 3.2 Filtering & Laser Detection

The bandpass filter was highly effective at filtering out non-laser light, however reflections/glow from laser light often bounced against other surfaces and passed through the filter, thereby requiring additional processing (see [Figure 5](#)). Several steps of additional processing were used (see `camera.m`):



- MATLAB’s camera calibration toolkit was used to account for and correct camera lens distortion.
- The image was rotated slightly to account for the laser beam being imperfectly horizontal (see 2.2.2). This correction was imperfect, and better aligning the laser would be a significant improvement.
- Only the green channel of the image was used, because the laser used was green.
- The image was thresholded, where each pixel was considered either **on** or **off**, depending on if its green value was greater than 75 (out of 255).
- For each column of pixels, the center position of the laser was calculated by finding the center of the largest block of continuous **on** pixels.
- Columns with zero **on** pixels were ignored.

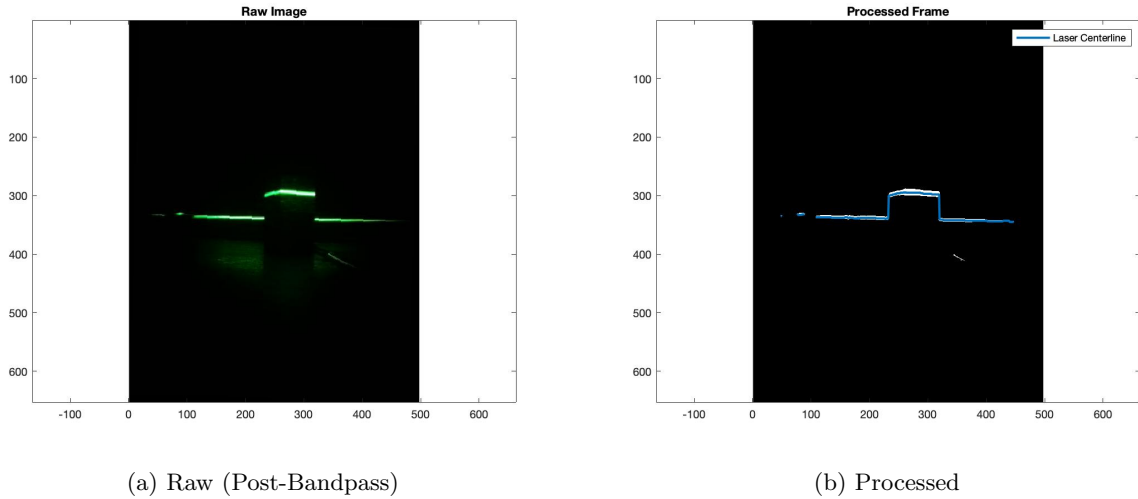


Figure 5: Post-bandpass raw and fully-processed images of the laser. Reflection of laser light (in this case, off the ground) can be seen.

### 3.3 Distance Measurement

Once the center position of the laser was determined, it was translated into a set of distances from the camera (again, on a column-by-column basis) using Equation 1, where  $Y_0$  is the pixel row index where the laser beam intersects the center of the camera’s vision,  $\theta$  represents the angle between the laser and the horizontal (as calculated in CAD), and  $\alpha$  represents an experimentally-determined constant roughly corresponding to half the distance between the center of the camera and the center of the laser, in pixels.

$$\alpha (Y_0 - y) \arctan(\theta) \quad (1)$$

$Y_0$  was calibrated by positioning the testbed 1 meter from the wall (the theoretical distance at which the laser beam and the camera’s view intersect) and setting  $Y_0$  to the pixel row index of the laser center line.  $\alpha$  was determined (after calibrating  $Y_0$ ) by moving the testbed to various distances from the wall and adjusting  $\alpha$  to result in the most accurate results for all values. The system was then tested by placing it at as-yet-uncalibrated distances from the wall and comparing the calculated distance to the known distance.

This system was rather inaccurate, and needs substantial iteration (it’s unclear if the algorithm is inaccurate, the calibration mechanism is inaccurate, or both). Idrobo-Pizo, Motta, and Sampaio 2019 showed great promise in the area of calibration, but unfortunately I ran out of time to fully explore it.

## 4 Conclusion

I developed the first stages of an underwater structured light profilometry system. At time of writing, the project is in the early prototype stage, where I built a simple testbed that holds the camera, laser, and supporting electronics. The prototype can accurately filter and detect laser light, however more research is needed into proper calibration.

## 5 Acknowledgements

Thank you to Dave Barrett and the Olin RoboLab for making this research possible—I absolutely could not have accomplished this project without his support and guidance. Thank you also to all of my fellow researchers in the RoboLab, who have helped me countless times throughout this semester.

## References

- Alexander Duda (2021). *3D Seafloor Mapping with Structured Light*. Schmidt Ocean Institute. URL: <https://schmidtoccean.org/cruise-log-post/3d-seafloor-mapping-structured-light/> (visited on 01/31/2021).
- Barrett, David, Isaac Vandor, and Emily Kohler (Oct. 2018). “Applying Structured Light Laser Imaging to Underwater Obstacle Avoidance and Navigation”. In: *OCEANS 2018 MTS/IEEE Charleston*. OCEANS 2018 MTS/IEEE Charleston. Charleston, SC: IEEE, pp. 1–6. ISBN: 978-1-5386-4814-8. DOI: [10.1109/OCEANS.2018.8604751](https://ieeexplore.ieee.org/document/8604751/). URL: <https://ieeexplore.ieee.org/document/8604751/> (visited on 05/07/2021).
- Creath, Katherine (July 7, 2000). “Choosing a phase-measurement algorithm for measurement of coated LIGO optics”. In: International Symposium on Optical Science and Technology. Ed. by Malgorzata Kujawinska, Ryszard J. Pryputniewicz, and Mitsuo Takeda. San Diego, CA, pp. 46–55. DOI: [10.1117/12.498386](http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=916362). URL: <http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=916362> (visited on 05/07/2021).
- Digital micromirror device (Dec. 21, 2020). In: *Wikipedia*. Page Version ID: 995523360. URL: [https://en.wikipedia.org/w/index.php?title=Digital\\_micromirror\\_device&oldid=995523360](https://en.wikipedia.org/w/index.php?title=Digital_micromirror_device&oldid=995523360) (visited on 02/10/2021).
- Distortion — Edmund Optics (2021). Edmond Optics. URL: <https://www.edmundoptics.com/knowledge-center/application-notes/imaging/distortion/> (visited on 02/18/2021).
- Idrobo-Pizo, Gerardo Antonio, José Maurício S. T. Motta, and Renato Coral Sampaio (Apr. 14, 2019). “A Calibration Method for a Laser Triangulation Scanner Mounted on a Robot Arm for Surface Mapping”. In: *Sensors* 19.8, p. 1783. ISSN: 1424-8220. DOI: [10.3390/s19081783](https://www.mdpi.com/1424-8220/19/8/1783). URL: <https://www.mdpi.com/1424-8220/19/8/1783> (visited on 05/07/2021).
- Liu, Hongyu et al. (Feb. 2003). “Calibration-based phase-shifting projected fringe profilometry for accurate absolute 3D surface profile measurement”. In: *Optics Communications* 216.1, pp. 65–80. ISSN: 00304018. DOI: [10.1016/S0030-4018\(02\)02290-3](https://linkinghub.elsevier.com/retrieve/pii/S003040180222903). URL: <https://linkinghub.elsevier.com/retrieve/pii/S003040180222903> (visited on 05/07/2021).
- Nabil Madali (Aug. 1, 2020). *Structured Light Scanning*. Medium. URL: <https://towardsdatascience.com/structured-light-scanning-c125d5e06c41> (visited on 02/10/2021).
- Rosales-Guzmán, Carmelo, Aniceto Belmonte, and Juan P. Torres (Mar. 2017). *Optical Metrology Techniques Harness Structured Light Beams*. Photonics Media. URL: [https://www.photonics.com/Articles/Optical\\_Metrology\\_Techniques\\_Harness\\_Structured/a61643](https://www.photonics.com/Articles/Optical_Metrology_Techniques_Harness_Structured/a61643) (visited on 02/10/2021).
- Salvi, Joaquim et al. (Aug. 2010). “A state of the art in structured light patterns for surface profilometry”. In: *Pattern Recognition* 43.8, pp. 2666–2680. ISSN: 00313203. DOI: [10.1016/j.patcog.2010.03.004](https://linkinghub.elsevier.com/retrieve/pii/S003132031000124X). URL: <https://linkinghub.elsevier.com/retrieve/pii/S003132031000124X> (visited on 05/07/2021).
- Sansoni, G., L. Biancardi, et al. (Aug. 1994). “A novel, adaptive system for 3-D optical profilometry using a liquid crystal light projector”. In: *IEEE Transactions on Instrumentation and Measurement* 43.4, pp. 558–566. ISSN: 00189456. DOI: [10.1109/19.310169](http://ieeexplore.ieee.org/document/310169/). URL: <http://ieeexplore.ieee.org/document/310169/> (visited on 05/07/2021).

- Sanconi, G., M. Carocci, and R. Rodella (June 2000). “Calibration and performance evaluation of a 3-D imaging sensor based on the projection of structured light”. In: *IEEE Transactions on Instrumentation and Measurement* 49.3, pp. 628–636. ISSN: 00189456. DOI: [10.1109/19.850406](https://doi.org/10.1109/19.850406). URL: <http://ieeexplore.ieee.org/document/850406/> (visited on 05/07/2021).
- Sensoni, G. (1999). “3D optical whole-field range sensor: development of procedures for the automatic set-up of the measurement and the calibration of the system”. In: *IMTC/99. Proceedings of the 16th IEEE Instrumentation and Measurement Technology Conference (Cat. No.99CH36309)*. IMTC/99. 16th IEEE Instrumentation and Measurement Technology Conference. Measurements for the New Millennium. Vol. 2. Venice, Italy: IEEE, pp. 1154–1159. ISBN: 978-0-7803-5276-6. DOI: [10.1109/IMTC.1999.777038](https://doi.org/10.1109/IMTC.1999.777038). URL: <http://ieeexplore.ieee.org/document/777038/> (visited on 05/07/2021).
- Tsai, Ming-June and Chuan-Cheng Hung (Oct. 2005). “Development of a high-precision surface metrology system using structured light projection”. In: *Measurement* 38.3, pp. 236–247. ISSN: 02632241. DOI: [10.1016/j.measurement.2005.07.014](https://doi.org/10.1016/j.measurement.2005.07.014). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0263224105000746> (visited on 05/07/2021).
- Van der Jeught, Sam and Joris J.J. Dirckx (Dec. 2016). “Real-time structured light profilometry: a review”. In: *Optics and Lasers in Engineering* 87, pp. 18–31. ISSN: 01438166. DOI: [10.1016/j.optlaseng.2016.01.011](https://doi.org/10.1016/j.optlaseng.2016.01.011). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0143816616000166> (visited on 05/07/2021).