Sidharth Shanmugam

Initial Project Report

Machine Vision-Based Anti-Backscatter Lighting System for Unmanned Underwater Vehicles

---

Submitted 7 March, 2024

4th Year Initial Project Report for Degree of
MEng in Electronic and Computer Engineering

School of Physics, Engineering and Technology,
University of York

Supervisors:

Prof. Paul D Mitchell, Prof. Andy M Tyrrell

Ethical Considerations

After consideration of the University of York's code of practice and principles for good ethical governance, I have identified no related issues in this project.

Contents

1 Introduction **3**

2 Background Information **4**

2.1 Computing Platforms 4

2.2 Specialised Light Source 5

2.3 Specialised Camera 6

2.4 Machine Vision Technologies for Backscatter Detection and Tracking 6

2.4.1 Automated gas bubble imaging at sea floor - a new method of in situ gas flux quantification [1] 7

2.4.2 Gas Bubble Shape Measurement and Analysis [2] 8

2.4.3 The Python Programming Language and OpenCV 8

2.5 Real-time Systems 9

2.5.1 Understanding Linux real-time with PREEMPT_RT training [3] 9

2.5.2 Raspberry Pi 4B: Real-Time System using Preempt-RT (kernel 4.19.y) [4] 10

3 Project Objectives **10**

3.1 Reliable and Accurate Backscatter Cancellation 10

3.2 Architectures and Methodologies for Real-Time 11

3.3 Optimisations Using Predictive Systems 11

4 Approach Description **12**

4.1 Milestone 1: Basic Canny-Based Backscatter Segmentation (Software V1) 12

4.2 Milestone 2: Advanced Canny-Based Backscatter Segmentation With Tracking (Software V2) 13

4.3 Deliverable 1: Demonstration of Software V2 13

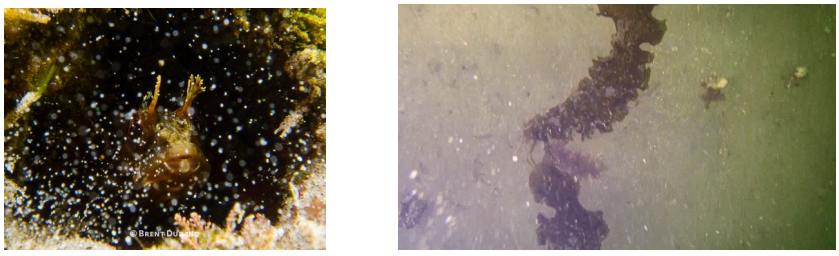
4.4 Milestone 3: Predictive Backscatter Segmentation Approaches (Software V3) . . 13

4.5 Deliverables 2 & 3: The Final Report, Project Presentation, and Viva Voce . . . 14

5 Project Timetable **14**

1 Introduction

Underwater imaging has long been crucial in various fields, such as marine research, environmental monitoring, underwater archaeology, and offshore industries. Underwater imaging has long been crucial in various fields, such as marine research, environmental monitoring, underwater archaeology, and offshore industries. For example, scientists use underwater photography with quadrats to audit the abundance of coral over time at several reef locations [5]. Unmanned Underwater Vehicles (UUVs), a type of submersible vehicle, are pivotal in advancing underwater imaging capabilities. Often housing vast arrays of sophisticated sensors, these vehicles enable the end user to explore and analyse underwater environments with unprecedented accuracy and efficiency. Their autonomous or remote-controlled nature advocates the ideal platform for conducting missions of extended duration in hazardous conditions, impossible for direct human presence and intervention. With the benefits of UUV deployment, they have become common as a safer and cheaper alternative to manned vehicular operations in the vast range of underwater imaging-related industries and applications, such as intelligence surveillance and reconnaissance in defence, inspection and identification of defects or foreign objects in maritime, and oceanography and hydrography in marine research [6].



(a) (b)

Figure 1: (a) Backscatter forms as the particles of sand drift between the aquatic animal and the camera [7]. (b) A captured frame in GoPro footage from a UUV of the seabed with backscatter formed as the propellors disrupt the sand.

The lack of light attenuation at greater sea depths is a critical challenge in underwater imaging, and to tackle this, an external high-power light source is often tied to a camera to ensure a well-lit scene. However, this produces an adverse side-effect: backscatter, shown in Figure 1, where suspended particles in water scatter light in an inhomogeneous manner, reflecting the light emitted by the light source back into the camera, creating exponentially bright spots and often saturating the image and degrading the quality. While there are a few simple and universal techniques to eliminate backscatter, such as reducing the separation between the subject and camera, fine-tuning the light source position such that only the edge of the light cone illuminates the subject, or achieving perfect buoyancy to minimise creating clouds of sand and debris [7], they lack viability for UUVs due to the erratic backscatter formation from the continuous and arbitrary motion of the propellors and general vehicular movement.

The ultimate goal of this project is to develop a novel light source system capable of aiding the generation of high-quality underwater images, of mainly the sea floor, from UUVs

without backscatter interference. This project first aims to research systems and develop reliable backscatter detection and elimination capabilities, with a specialised projector serving as a dynamic light source of tailored light patterns for selective scene illumination to mitigate backscatter effects. The second research aim will look into architectures and methodologies to optimise the system for real-time to ensure adherence to stringent requirements for predictability, stability, efficiency, and reliability, ultimately allowing for imaging performance control in dynamic underwater environments whilst maintaining requirements. The final research aim is delving into the engineering of a predictive system for anticipating the future positions of detected backscatter particles, enabling proactive elimination strategies without the need for continuous, computationally intensive machine vision processing, for efficient and preemptive adjustments to the light projection patterns for optimal backscatter suppression. These three aims represent critical trade-offs that must be carefully balanced to achieve the overarching objective, which this research project aims to establish through systematic exploration and optimisation to lead towards a cutting-edge framework for enhancing underwater imaging capabilities.

This initial report forms a snapshot-based progress update on the project status. With the outline of the ultimate project goal and the aims, the report aims to summarise the background information in related theoretical realms considered thus far in the next section. The compiled background information will form the foundation of the subsequent sections, a discussion on the specifications to achieve the project goals with detail on hardware and software choices and a description of the intended project progression approach exemplified by a project schedule. Finally, a conclusion will close this report with a summary of details and any further thoughts.

2 Background Information

With the confirmation of the overall project goal and the set of aims for research, it is vital to consider the overall background information consisting of highly specialised technical subject theory to develop a rigorous theoretical basis for the project design and experimental work before considering project objectives and intended approach.

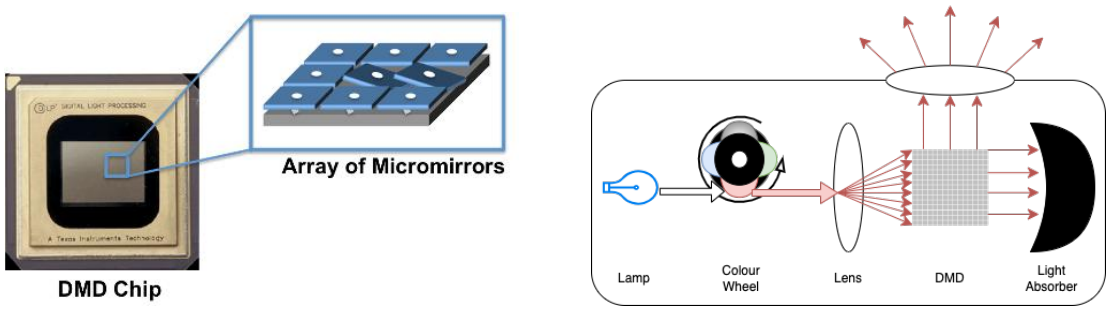
2.1 Computing Platforms

Field Programmable Gate Arrays (FPGAs) are semiconductor devices based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects, enabling the hardware reprogramming to desired application or functionality requirements after manufacturing [8]. The flexibility of FPGA-based systems allows for very diverse and capable solutions, with their parallelism allowing for efficient computation and re-programmability for efficient use of hardware without the need to dedicate hardware for a task that may only last a few seconds [9]. Unlike FPGAs, CPUs compute sequentially, breaking algorithms into operations by sequence, thus limiting execution to one operation at a time [10].

The Raspberry Pi (RPi) company has been designing single-board and modular computers built on the Arm architecture and running the Linux operating system since 2012, with a mission to put high-performance, low-cost, general-purpose computing platforms in the hands of enthusiasts and engineers worldwide [11]. The regular RPi product series features single-board computers powered by Broadcom-built Arm Cortex-based multicore processors. Perfect for a non-FPGA, CPU-based route, the newer RPi models implement various features such as built-in RAM up to 8GB, a dedicated GPU and video decoder, HDMI output interfaces, dual-band WiFi with Bluetooth capabilities, a Gigabit Ethernet interface, USB 3.0 and 2.0 ports, MIPI-compatible serial interfaces for cameras and displays, and an array of 40 GPIO pins [12, 13].

2.2 Specialised Light Source

The Digital Light Processing (DLP) chipset, created by Larry Hornbeck of Texas Instruments in 1987, consists of a Digital Micromirror Device (DMD) that houses millions of reflective aluminium mirrors, usually a few microns wide [14, 15]. The DMD is a Micro-Opto-Electro-Mechanical System (MOEMS) Spatial Light Modulator (SLM) that uses digital signals to control the angle of each mirror, enabling the modulation and attenuation of an incident light beam [16].



(a) (b)

Figure 2: (a) An illustration of the microscopic mirror array within a DMD chip [15]. (b) A cross-section illustrating the internals of an example projector utilising DLP technology.

As [17] explains the intricacies and Figure 2 illustrates, a DLP projector implements this specialised technology: The beam of a high-intensity white lamp directs into a spinning colour-tinted lens wheel consisting of red, green, and blue, and a clear lens for a white lamp pass-through. An internal lens diffracts the coloured beam incident into the DMD, with the angle and time spent in each microscopic mirror controlling the individual pixel colour and intensity. The DMD directs the desired beams of each pixel into the diffraction lens to exit the projector. The DMD will not actuate for undesired pixel beams, thus sending the beam into a light-absorbent region to prevent leakage and image distortion.

2.3 Specialised Camera

The cameras in most traditional and consumer-grade electronic devices, such as phones, all employ a CMOS sensor with a rolling shutter. While these sensors are smaller and much cheaper, they cause distortion effects when capturing fast-moving subjects due to their line-by-line scan image-capturing characteristic.

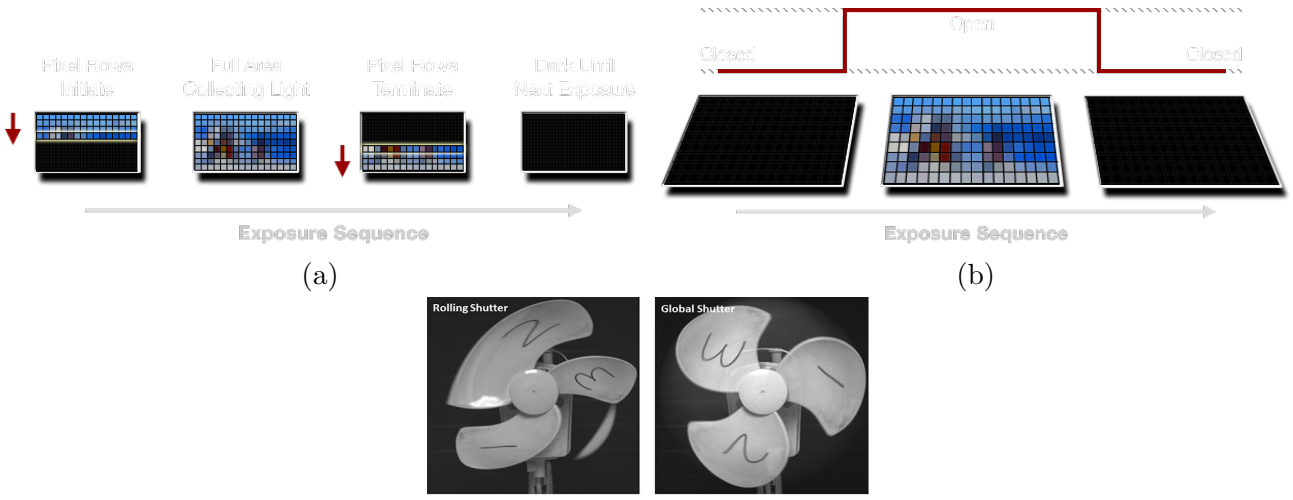


Figure 3: (a) An image capture timeline of a rolling shutter sensor, (b) An image capture timeline of a global shutter sensor [18], (c) The distortion in the image capture of a spinning fan due to a rolling shutter, compared to a non-distorted capture from a global shutter [19].

The article in [20] explains: Instead of having pixels from the top of the sensor switch on and work its way down like a scan, the global shutter sensor takes a snap of the scene using all of the pixels all at once. Figures 3a and 3b illustrate the differences in image capture, and Figure 3c illustrates the drastic differences in each shutter type. The Raspberry Pi company produces a global shutter camera featuring a 1.6MP Sony IMX296 sensor, with plug-and-play compatibility with RPi computers [21].

2.4 Machine Vision Technologies for Backscatter Detection and Tracking

Underwater backscatter has a mixture of many compositions, from bubbles and sand to all sorts of other marine debris. It is much easier to explore the image processing for one backscatter composition type and later expand to cover all types. Many environmental analysis fields employ systems for underwater bubble quantification to study gas seepages from the sea floor. These systems require implementations of bubble detection to calculate the exact outline and shape with bubble tracking to measure the movement to calculate the volume, flux, and overall chemical composition. Gas escape measurement and monitoring systems require high precision, extensive range, strong anti-interference, and low cost under complex underwater conditions [22]. Therefore, research into this field to harness techniques should theoretically provide a boost to help with the accurate backscatter cancellation project aim.

2.4.1 Automated gas bubble imaging at sea floor - a new method of in situ gas flux quantification [1]

This paper presents the design of a novel method for automated gas bubble imaging, its validation procedures and calibration experiments, with the primary aim of quantifying gas flux from gas release sites on the sea floor for environmental analysis. The authors have identified two main factors of the system: (a) segmentation by isolating the bubbles from the image background and (b) determining their position and sizes with bubble tracking in successive frames.

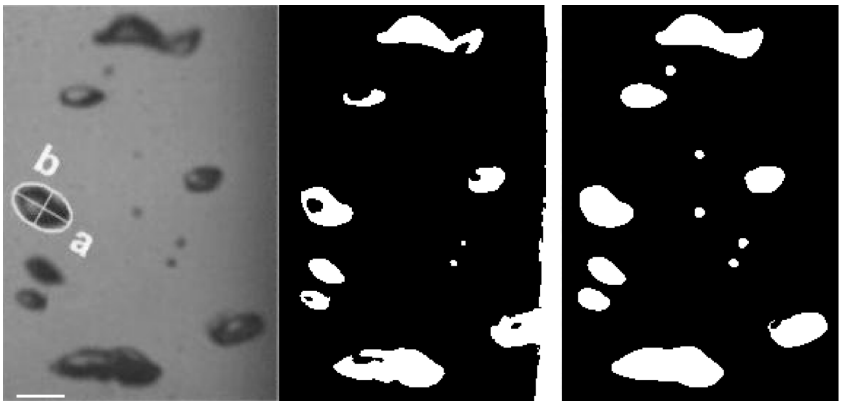


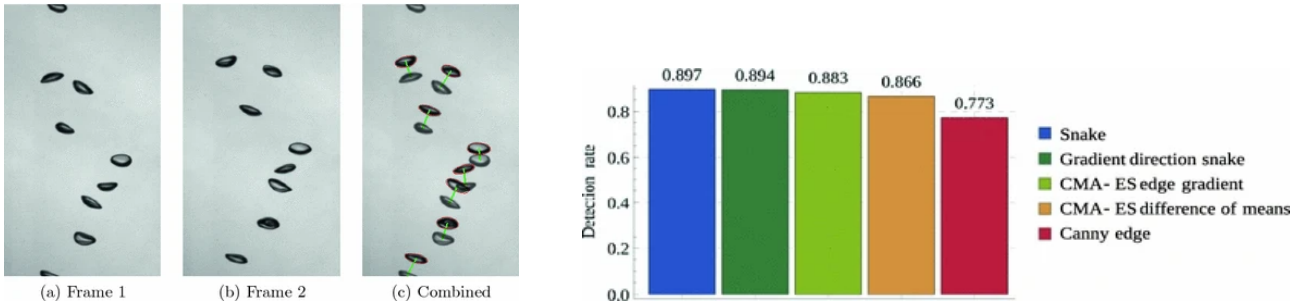
Figure 4: An illustration from the paper in [1] showing the original input in the left-most image, bubble segmentation using a simple thresholding method in the middle, and bubble segmentation using Canny in the right-most image.

This paper employs the Canny edge detection algorithm [23] for greyscale images. The Canny filter compares the value of each pixel relative to its neighbours, so when the gradient between two adjacent pixels is higher than a certain threshold, the algorithm sets the bordering pixel to a value of binary '1', otherwise '0', resulting in the formation of edges around objects. The paper compares the Canny segmentation approach with simple thresholding (blob detection approach) and deduces Canny as more accurate for segmentation, even in inhomogeneously illuminated areas. However, the disadvantages of Canny are the drastic increase in computing time and the requirement for implementing morphological techniques to account for the growing or shrinking of bubbles during segmentation.

The paper outlines a method for bubble tracking using the 'least distance' rule, which is the realisation that the distance a bubble travels in two successive frames is smaller than the distance to its closest neighbour. This assumption, although not valid for overlapping bubbles, very high bubble concentrations, and cases where the travel distance exceeds the neighbouring bubble distances, enables a computationally simple method for calculating positions in successive frames and the bubble rise velocity from the sea floor. While this paper explores methodologies perfect for a starting point, it does focus more on the system construction and quantifying gas flux measurements, thus skipping some information on vital aspects such as how the bubbles were perfectly highlighted after Canny in Figure 4, one can assume they are using logic to fill closed loop edges.

2.4.2 Gas Bubble Shape Measurement and Analysis [2]

Gas bubbles emerging from sea floor seepages rise at high speeds and may change in form, contour, and volume during their ascent. This paper aims to develop robust methods for automated image processing to extract the shape, motion, and volume from a sequence of image frames. The paper mentions the work in [1] to form a baseline. The sporadical false detection characteristic of the Canny edge detector due to light areas inside bubbles derives a motivation to search for more stable algorithms. This paper introduces methods for precise bubble stream quantification and accurate bubble elliptical fitting with snake-based methods, where a snake is an energy-minimising spline guided by external constraint forces and influenced by image forces that pull it towards features such as lines and edges, and the Covariance Matrix Adaptation - Evolution Strategy (CMA-ES) optimisation algorithm, which aids in solving non-linear, non-convex optimisation problems and is particularly effective for continuous optimisation tasks where the objective function may be multimodal, noisy, or complex.



(a) (b)

Figure 5: From [2]: (a) illustration of the movement of bubbles in between two frames at 100 fps. The combination shows frames 1 and 2 with the bubble detection in frame 2 in red and matching bubbles connected in green, (b) a graph comparing the detection rates for several selected methods on a high-quality GoPro image sequence with manual ground truth on 20 images.

The paper implements bubble movement tracking between the current and previous frames using a Kalman filter [24] to predict the bubble position in a subsequent frame from detection, employing the Hungarian method [25] for the minimum weighted matching of the predicted positions and the new detection of bubble positions. Figure 5a illustrates both the bubble detection and movement tracking between frames, notably observing the precise red border that follows the exact outline of the bubble in addition to the accurate tracking even in bubble-dense regions. The methods this paper explores are compared based on performance, illustrated in Figure 5b. The standard snake method achieves an 89.7% decision rate, a 12.4% advantage over the standalone Canny system.

2.4.3 The Python Programming Language and OpenCV

OpenCV is an open-source computer vision and machine learning software library with over 2500 optimised algorithms, including a comprehensive set of classic and state-of-the-art computer vision and machine learning algorithms [26]. Although native to C++, OpenCV features

interfaces for Python, Java and MATLAB, ensuring compatibility with Windows, Linux, Android, and macOS. OpenCV targets real-time vision applications by taking advantage of CPU accelerators, such as MMX and SSE instructions for parallel data processing, and GPU accelerators, with ongoing development of a full-featured CUDA and OpenGL interface.

2.5 Real-time Systems

A real-time operating system (RTOS) enables the compliance of a 'hard' real-time system, where there is a guarantee of the maximum time a task requires for completion. An RTOS is typically suited for low-power, embedded systems such as single-core STM32 microcontrollers due to their deterministic behaviour, low-latency interrupt handling, and low-complexity design. However, an STM32 microcontroller cannot handle the high computational requirements of machine vision tasks, thus invoking a preference towards an RPi, such as the Pi 4, or an FPGA-based system. All RPi's non-microcontroller product offerings are RTOS unsuitable due to high system complexity, as confirmed by [27]. Despite the presence of a FreeRTOS port for the Pi 4 [28], there are many limitations and thus will exponentially increase development times.

2.5.1 Understanding Linux real-time with PREEMPT_RT training [3]

This piece of literature is a 123-slide training presentation by Bootlin that covers RTOS fundamentals, Linux, and the PREEMPT-RT patch. Most multi-tasking OSes, such as Linux, are pre-emptive, meaning that when a task runs in user space mode and is interrupted and if the interrupt handler can wake up another task, the scheduler will schedule this task as soon as the interrupt handler returns. Since the Linux kernel does not support pre-emption due to the presence of spinlocks rather than mutual exclusion around critical sections, pre-emption logic falls apart when a user-space task calls kernel-specific functions, thus accepting a promotion to kernel-space as it receives an interrupt.

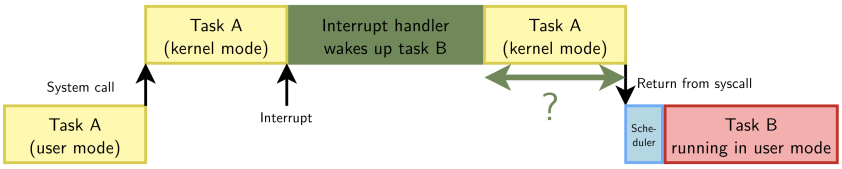


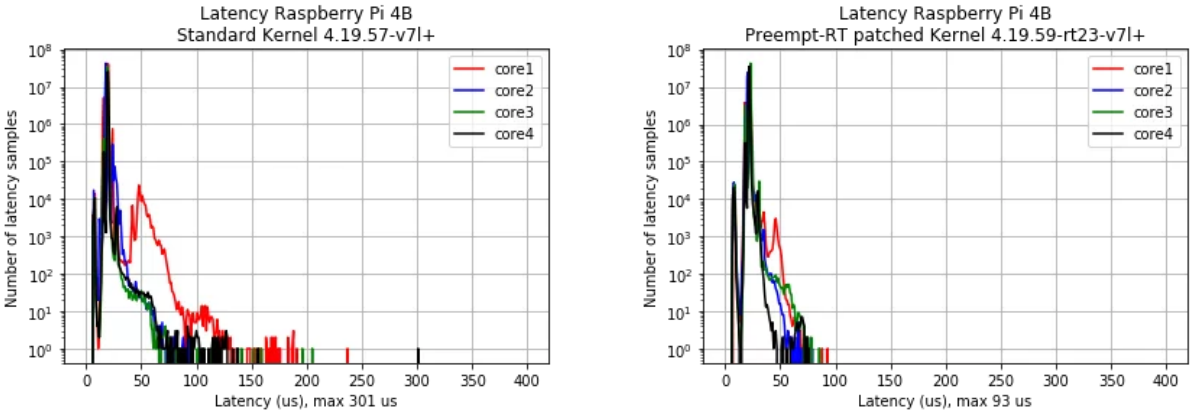
Figure 6: Illustration of the interrupt flow within two tasks, one in kernel space and another in user space [3].

Figure 6 illustrates this issue when a task promotes into kernel space. Critical sections are bound with spinlocks, ultimately preventing the immediate pre-emption of the scheduler by holding Task A to schedule Task B. When the interrupt handler wakes Task B, Task A resumes and will continue until the spinlock completes, resulting in monumental and unpredictable latencies. Aside from the kernel-space incompatibility, Linux already supports user-space pre-emption with task priority-based real-time scheduling to allow for an RTOS. The PREEMPT-RT is a kernel patch that aims to make all kernel-space code preemptible and deterministic. Although

insufficient to convert Linux into an RTOS, the PREEMPT-RT patch is a step in the correct direction to minimise task switching latencies. The presentation discusses the limitations of Linux and Linux-compatible hardware with RTOS: Linux will never be a formally proven RTOS, and the hardware Linux typically runs on isn't designed with RT in mind.

2.5.2 Raspberry Pi 4B: Real-Time System using Preempt-RT (kernel 4.19.y) [4]

This blog post details the installation steps for the PREEMPT-RT kernel patch in a Raspberry Pi 4 Linux system. The RT-Tests, a test suite that contains programs to test various real-time Linux [29], quantifies the latency reduction with the kernel modification, drawing comparisons with the non-RT kernel.



(a) (b)

Figure 7: Illustration from [4] showing the latency measurements following a Cyclictest execution from the RT-Tests suite for (a) the standard Linux RPi kernel and (b) the PREEMPT-RT patch Linux RPi kernel.

The Cyclictest, part of RT-Tests, accurately and repeatedly measures a thread's intended wake-up time and when it wakes up to provide statistics about the system's latencies, measuring latencies in real-time systems caused by the hardware, the firmware, and the operating system [30]. The maximum latency measurement with the standard kernel was 301 µs, while the PREEMPT-RT kernel was 93 µs, thus confirming a 3.23x latency reduction using the kernel patch.

3 Project Objectives

3.1 Reliable and Accurate Backscatter Cancellation

The system must be able to accurately and reliably pinpoint the location of backscatter particles as well as map out the shape of the outline, enabling the perfect segmentation of each particle for elimination. The work of the previous student on this project [31] outlines the backscatter detection and elimination system revolving around a simple blob detection algorithm. However, much industry-related literature refers to Canny, an edge-detection algorithm,

as the industry standard for bubble detection and characterisation for underwater environmental analysis systems. While simple blob detection is a viable method due to computational simplicity, as Shepherd mentions, for blob detection to work, the blobs must share at least one property, such as size or shape, to isolate these blobs. Underwater is a volatile environment where backscatter particles may be unique, so the system must revolve around the Canny algorithm, harnessing enhancements seen in [2] for utmost reliability and accuracy.

Despite many highly efficient intellectual property (IP) cores for re-usable FPGA-based logic to accelerate FPGA development, the monumental complexity curve persists due to the requirement of low-level hardware knowledge and familiarity with digital design concepts. Furthermore, FPGAs are, on average, much more expensive than traditional CPU-based computers. It is much easier to harness the simplicity of a CPU-based machine, a high-level programming language such as Python and heavily optimised libraries from OpenCV for rapid machine vision application development. Taking a simplistic approach with a non-FPGA system, an RPi 4 or 5 with an RPi Global Shutter camera will drive a DLP projector, utilising the particle centre location and outline mapping to project holes in the light beam in addition to the implementation of position calibration, which is crucial to minimise the parallax effect, the displacement of the target from the perceived position in the video capture, ultimately ensuring accurate backscatter elimination.

3.2 Architectures and Methodologies for Real-Time

Shepherd discusses the hindrance to the successful operation of the blob detection-based backscatter elimination software due to the effects of the Linux operating system (OS): Linux's ability to run tasks in the background while the program is running, interfered with the execution of the program, making the camera image buffer and freeze [31]. Due to the incompatibility of an RPi system with RTOS, the project will still be Linux-driven, though implementing the PREEMPT-RT kernel patch to minimise task switching latencies to mitigate the jitter issues as much as possible experienced by Shepherd. Sacrificing performance for rapid development, Python enables real-time development for less stringent requirements. Research to benchmark the system performance of Python can lead to the development of a C++-based system for low-level optimisation and compliance with even the most rigorous real-time requirements.

3.3 Optimisations Using Predictive Systems

Although reducing the resolution of high-quality image frame captures is viable to reduce the computational overheads with machine vision applications, it often trades off object detection and segmentation accuracy. A predictive system approach can mitigate the computationally intensive requirement of machine vision application to every frame of an input video feed. Accurate backscatter particle movement and future location prediction eliminate the requirement to apply machine vision-based technologies to every frame in the video feed to identify and segment the particles. There are two main approaches: (a) an artificial intelligence (AI) ap-

proach, harnessing supervised machine learning (ML) methods with a potential to incorporate unsupervised methods for greater accuracy, or (b) a non-AI/ML interpolative approach with an application of either a linear or polynomial interpolation algorithm to predict future positions concerning the tracked locations in a finite count of previous frames. There are significant trade-offs concerning accuracy, speed, and computational overheads for each approach that requires exploration.

4 Approach Description

The three project objectives form a waterfall-effect task cascade, powering the overarching software development process by providing major version milestones. Following the software version milestones, each development stint will consist of subtasks with a scope for agile methodologies. The prior completion of many project aspects, such as crucial background research, finalising and ordering parts, and some software development, makes room for much experimental work.

4.1 Milestone 1: Basic Canny-Based Backscatter Segmentation (Software V1)

With the technical background research in place, I have developed a software spike that uses the Canny edge detection algorithm. The spike utilises a GoPro footage of the sea floor. Due to the rolling-shutter sensor on board the GoPro and intense video compression, this footage has many artefacts and image distortion. The first step forward is to utilise the underwater testing facilities at the Institute for Safe Autonomy (ISA) to generate uncompressed footage of a stream of bubbles using the RPi global shutter camera system. So far, I have developed a Python script intended for the RPi computers with an installed RPi global shutter camera to record a video feed. Before recording, I must allocate sufficient time to update this script to record in raw format, as it currently uses H.264 encoding, and to set up the recording rig to use at the testing facilities. In parallel, I must research metrics for quantifying the real-time performance of the software to prepare for the V1 software requirements: (a) Canny-based backscatter segmentation, optimised with the test footage recording, and (b) real-time metrics tracking logic. There must be an excess buffer time to account for the paramount risk of insufficient optimisation time for the Canny algorithm and insufficient development time for modular software, otherwise resulting in a backlog of software development tasks, mainly for the real-time tracking metrics implementation. With the target for modular software development, future feature implementations will be seamless and will not require re-writing large code sections.

4.2 Milestone 2: Advanced Canny-Based Backscatter Segmentation With Tracking (Software V2)

Following the first milestone, the RPi test systems, consisting of the Pi 4 and the Pi 5, must contain the PREEMPT-RT kernel patch. The implementation must house simple logic for rapid switching between the PREEMPT-RT kernel and the standard kernel, aiding for quick parameterisation of performance. While this task should only take a day and has no paramount risks to cause overrun, as with all software, there must be a slight buffer. On the software side, the first requirement for V2 is the snake method implementation for a more accurate backscatter segmentation. In parallel, the second requirement is the research and development of the least distance rule and Kalman filter methodologies for backscatter tracking. In a fast-paced environment, such as underwater from a UUV, where the rapidly moving propellors disrupt backscatter particles at a snappy rate, it may be worth considering the computational performance and development time trade-offs with the Kalman filter-based tracking approach when the least distance approach may suffice, over-estimating the task duration to account for research time towards method evaluation to result in a single method deployment. The final requirement of this software version is the mitigation of parallax effects, in addition to implementing logic to drive the DLP projector, after which the real-time performance of backscatter cancellation between the standard and PREEMPT-RT kernel is ready for evaluation. Due to the nature of software development, there are many risks to consider for this milestone. Hence, a 1-week buffer is crucial before the milestone deadline.

4.3 Deliverable 1: Demonstration of Software V2

The completion of V2 from the previous milestone showcases a fully-fledged Canny edge detection-based backscatter cancellation system. A department-wide project demonstration to industry professionals on the 29th of April, 2024, forms the ideal platform to exhibit the project and to welcome feedback for improvement. Allocating some time to prepare for this presentation is vital, although over-estimating the time allocation to ensure there is sufficient buffer in case Milestone 2 is incomplete.

4.4 Milestone 3: Predictive Backscatter Segmentation Approaches (Software V3)

In parallel with the demonstration day preparation for milestone 2, there must be a focus on the third and final software iteration. Consisting of researching, implementing, and evaluating non-ML approaches, such as interpolative methodologies, with ML-based approaches, considering unsupervised and supervised learning methodologies, fulfils the requirements of the final software version. Research into layering the predictive logic on machine-vision technologies to enhance system efficiency by assessing real-time metrics will complete the third objective of this project. As with all software-related tasks, there must be a sufficient buffer in the outcome

of unsuccessful deadline completion. It is increasingly paramount when incorporating ML technologies as not only is fine-tuning a requirement, but the training between each trial run can take a very long time. With the completion of V3, underwater testing can occur. Based on buffer consumption, not just limited to the underwater testing tank at the ISA, but with scope for real-life application with system deployment into a lake.

4.5 Deliverables 2 & 3: The Final Report, Project Presentation, and Viva Voce

Only when the final testing runs and the acquisition of project evaluation material, such as real-time metrics, is complete can writing the final project report proceed. Due to the underestimated difficulty and time consumption of articulating metrics, results, and summaries, the plan must consider this task to be of monumental duration. The same applies to the final presentation and project viva. There must be sufficient buffer time on occasions where the pushback of a deadline is the only option, such as to mitigate an illness.

5 Project Timetable

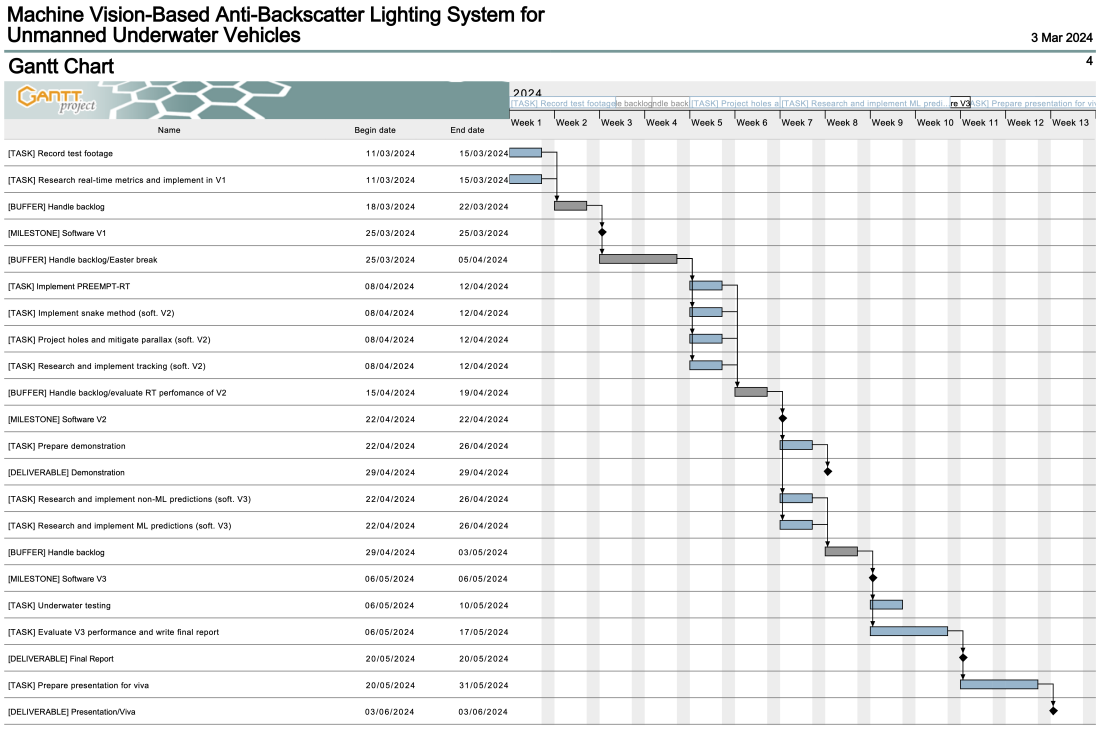


Figure 8: A Gantt chart illustrating the overall project schedule plan.

Figure 8 portrays a Gantt chart outlining the approach from the section above. At the end of each week on Friday, in parallel with the weekly email updates, I will evaluate this schedule to update the chart with any progress changes. I intend to approach this schedule as a 'live' document with constant change by preserving a complete version history.

References

[1] K. Thomanek, O. Zielinski, H. Sahling, and G. Bohrmann, "Automated gas bubble imaging at sea floor - a new method of in situ gas flux quantification," *Ocean Science*, vol. 6, no. 2, pp. 549–562, Jun. 2010. [Online]. Available: https://os.copernicus.org/articles/6/549/2010/ [Accessed February 24, 2024].

[2] C. Zelenka, "Gas Bubble Shape Measurement and Analysis," in *Pattern Recognition*, X. Jiang, J. Hornegger, and R. Koch, Eds. Cham: Springer International Publishing, 2014, vol. 8753, pp. 743–749. [Online]. Available: https://link.springer.com/10.1007/978-3-319-11752-2_63 [Accessed February 26, 2024].

[3] Bootlin, "Understanding Linux real-time with PREEMPT_RT training," Jan. 2024, [Accessed January 31, 2024].

[4] Mauro Riva, "Raspberry Pi 4B: Real-Time System using Preempt-RT (kernel 4.19.y)," Sep. 2019. [Online]. Available: https://lemariva.com/blog/2019/09/raspberry-pi-4b-preempt-rt-kernel-419y-performance-test [Accessed March 1, 2024].

[5] University of Hawai'i, "Practices of Science: Underwater Photography and Videography." [Online]. Available: https://manoa.hawaii.edu/exploringourfluidearth/physical/ocean-depths/light-ocean/practices-science-underwater-photography-and-videography [Accessed February 13, 2024].

[6] Yannick Allard and Elisa Shahbazian, *Unmanned Underwater Vehicle (UUV) Information Study.* Defence Research & Development Canada, Atlantic Research Centre, Nov. 2014. [Online]. Available: https://apps.dtic.mil/sti/pdfs/AD1004191.pdf [Accessed February 14, 2024].

[7] Brent Durand, "Easy Ways to Eliminate Backscatter in your Photos," Oct. 2013. [Online]. Available: https://www.uwphotographyguide.com/eliminate-backscatter-underwater [Accessed February 14, 2024].

[8] "What is an FPGA? Field Programmable Gate Array." [Online]. Available: https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html [Accessed February 23, 2024].

[9] C. Villalpando and R. Some, "Reconfigurable machine vision systems using FPGAs," in *2010 NASA/ESA Conference on Adaptive Hardware and Systems*, Jun. 2010, pp. 31–35. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5546238?casa_token=YZkc1g9nyG8AAAAA:qaUYE7iAG0x2pnYW2X1EGUwYbQPc6B_ScahtXONJg3QzfRqGqnunPldYwFM0obJQSgb7UVRQwA [Accessed February 23, 2024].

[10] Alex Liang, "Boosting Machine Vision with Built-in FPGA Image Preprocessing," Measurement & Automation Product Segment, ADLINK Technology, White Pa-

per, May 2016. [Online]. Available: https://qnv.com/wp-content/uploads/catalogos/BoostingMachineVisionwithBuilt-inFPGAImagePreprocessing_webversion.pdf [Accessed February 23, 2024].

[11] Raspberry Pi, "Raspberry Pi - About us." [Online]. Available: https://www.raspberrypi.com/about/ [Accessed February 23, 2024].

[12] Raspberry Pi Ltd, "Buy a Raspberry Pi 4 Model B." [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ [Accessed February 23, 2024].

[13] Raspberry Pi Ltd, "Buy a Raspberry Pi 5." [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-5/ [Accessed February 23, 2024].

[14] "Digital Light Processing," *Wikipedia*, Feb. 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Digital_Light_Processing&oldid=1201748148 [Accessed February 24, 2024].

[15] "How does a DLP projector work?" [Online]. Available: https://www.projectorscreen.com/blog/How-does-a-DLP-projector-work [Accessed February 24, 2024].

[16] "DLP4500 Digital Micromirror Device - TI | Mouser." [Online]. Available: https://www.mouser.co.uk/new/texas-instruments/ti-dlp4500-micromirror-device/ [Accessed February 24, 2024].

[17] "How DLP projector technology works - YouTube." [Online]. Available: https://www.youtube.com/ [Accessed February 24, 2024].

[18] RED Digital Cinema, "Global & Rolling Shutters." [Online]. Available: https://www.red.com/red-101/global-rolling-shutter [Accessed February 24, 2024].

[19] "Rolling Shutter vs Global Shutter sCMOS Camera Mode." [Online]. Available: https://andor.oxinst.com/learning/view/article/rolling-and-global-shutter [Accessed February 24, 2024].

[20] U. D. last updated, "What is a global shutter – and why is it so important?" Feb. 2021. [Online]. Available: https://www.digitalcameraworld.com/news/what-is-a-global-shutter-and-why-is-it-so-important [Accessed February 24, 2024].

[21] Raspberry Pi Ltd, "Buy a Raspberry Pi Global Shutter Camera." [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-global-shutter-camera/ [Accessed February 24, 2024].

[22] Y. Zhang, Y. Yu, X. Rui, Z. Feng, J. Zhang, Y. Chen, L. Qi, X. Chen, and X. Zhou, "Underwater bubble escape volume measurement based on passive acoustic under noise factors: Simulation and experimental research," *Measurement*, vol. 207, p. 112400, Feb. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224122015974 [Accessed February 24, 2024].

[23] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986. [Online]. Available: https://ieeexplore.ieee.org/document/4767851 [Accessed February 24, 2024].

[24] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960. [Online]. Available: https://asmedigitalcollection.asme.org/fluidsengineering/article/82/1/35/397706/A-New-Approach-to-Linear-Filtering-and-Prediction [Accessed February 28, 2024].

[25] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, Mar. 1955. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/nav.3800020109 [Accessed February 28, 2024].

[26] OpenCV, "About." [Online]. Available: https://opencv.org/about/ [Accessed March 1, 2024].

[27] "Free RTOS on the Raspberry Pi 4 - Kernel," Aug. 2022. [Online]. Available: https://forums.freertos.org/t/free-rtos-on-the-raspberry-pi-4/15659 [Accessed February 29, 2024].

[28] TImada, "TImada/raspi4_freertos," Feb. 2024. [Online]. Available: https://github.com/TImada/raspi4_freertos [Accessed February 29, 2024].

[29] Costa Shul, "RT-Tests," Sep. 2023. [Online]. Available: https://wiki.linuxfoundation.org/realtime/documentation/howto/tools/rt-tests [Accessed March 1, 2024].

[30] Costa Shul, "Cyclictest," Aug. 2023. [Online]. Available: https://wiki.linuxfoundation.org/realtime/documentation/howto/tools/cyclictest/start [Accessed March 1, 2024].

[31] Katie Shepherd, "Machine Vision Based Underwater Anti-Backscatter Lighting System," MEng Project Report, University of York, York, UK, 2023, [Accessed September 1, 2023].