Swimming Tango:
An Underwater Visual Odometry Mechanism for Deep-Sea Divers

ECE4012 Senior Design Project

Section L7A, Swimming Tango Team
Project Advisor: Dr. Mick West

Team Leader: Anupam Goli (agoli3@gatech.edu)
Team Member: Rishabh Ananthan (rishabh111@gatech.edu)

Submitted

May 4, 2016

Executive Summary

The project is aimed at creating an underwater means of visual odometry that also has ability to communicate via text for the U.S. Department of Defense. The goal is to limit the size of the hardware to within the profile of a handheld tablet. This is to help them link divers and topside personnel for increased diver safety and navigational accuracy. Our team will focus on the visual odometry aspect, with another team handling the communications aspect of this device. The navigation component will include implementing sensors such as a 2D sonar scanner, an inertial measurement unit (IMU) and a pressure transducer. The sensors will measure and record acceleration and deceleration on the vertical, horizontal, and transverse planes, data about the landscape and topography of the environment, and information on obstacles and how to avoid them. The mapping component of the project will utilize sensory data as well as imaging data (such as images or pointclouds) and create a visual odometry implementation similar to that of the 'Explorer' app in Google's Project Tango. The sensory data will be analyzed and processed using algorithms such as the FAST corner detection algorithm. The end goal of the mapping would be to produce a visualization of the underwater environment, which can then be viewed and used for robotic purposes. This product would help deep-sea divers find their way back to their starting point should they get lost while under the water. The device would also be designed in such a way as to complement the divers' Underwater Breathing Apparatus and not interfere with their gear or buoyancy.

Table of Contents

Swimming Tango:
An Underwater Visual Odometry Mechanism for Deep-Sea Divers

1. Introduction

The Swimming Tango team aimed at designing a means of underwater visual odometry, similar in functionality to the ‘Explorer’ app on Google’s Project Tango. This mechanism would in the future be implemented in a device whose size is limited to that of a handheld tablet, such as the Nexus 7 or the iPad Mini.

Figure 1. Rear-view of a Project Tango device [1].

1.1 Objective

The team focused on designing a visual odometry mechanism for underwater purposes that replicates the functionality of the ‘Explorer’ app in Google’s Project Tango using either a SONAR or an underwater camera instead of the Kinect sensor that is used in Google’s Project Tango. This is because the Kinect does not function well under the low-light conditions found underwater. This mechanism will be part of a device that also enables divers to communicate with each other through text messaging. This device will be waterproof and will not interfere with the divers’ gear or buoyancy. It will complement their Underwater Breathing Apparatus [2]. This team handled solely the visual odometry aspect of the project, while another team handled the communications aspect of the project.

1.2 Motivation

Deep-sea divers face a lot of hazards when they venture out into the ocean. Using such a device would enable them to communicate with other divers in their vicinity, which could be helpful in case of an emergency. Further, the device also tracks the movement of the divers. This could also be used to make sure that the divers can be easily found in the event they lose their way.

1.3 Background

Visual odometry allows a robot (or any moving object) to track its motion trajectory in real time. The two major types of visual odometry are monocular visual odometry (which uses only one camera or image source) and stereo visual odometry (which uses two cameras). Regardless of number of cameras, feature detection is run on frames to find and locate key features in the image, such as corners. An example of an algorithm used for this purpose is the FAST Feature Detection Algorithm. Once this is done, the movements of features between frames is tracked using an algorithm called RANSAC, and finally calibration data is used to translate these movements into real units. A block diagram describing the process as a whole is shown in Figure 2.

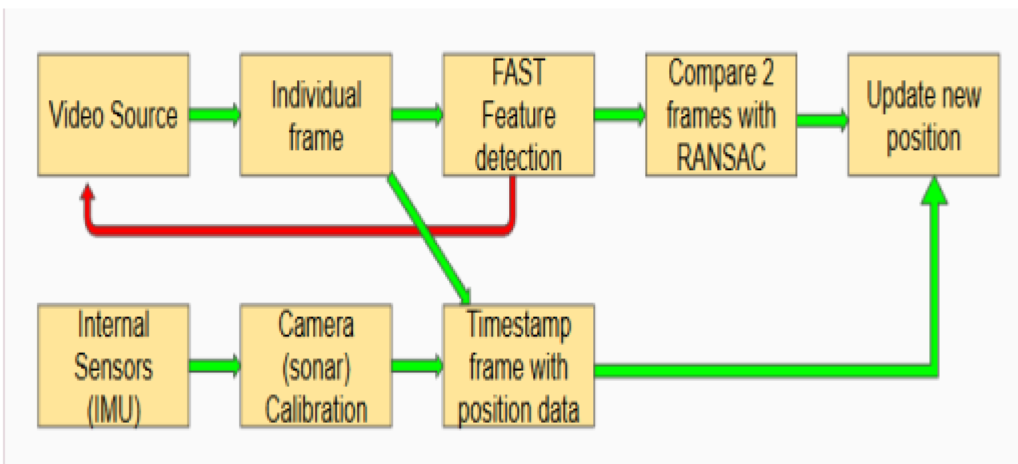

Figure 2. Block diagram showing the implementation of Visual Odometry.

Extensive research has been devoted to the development of navigation and mapping technologies. An important example is Google's Project Tango. Project Tango uses computer vision to enable mobile devices, such as smartphones and tablets, to detect their position relative to the world around them without using GPS or other external signals. This allows application developers to create user experiences that include indoor navigation, 3D mapping, measurement of physical spaces, recognition of known environments, augmented reality, and windows into virtual 3D worlds [1]. This is important because GPS signals are unable to provide location and time information underwater, since the electromagnetic signals from the orbiting satellites are heavily damped in water and hence cannot be detected by the receiver in most cases of interest [4].

One of the apps that comprises Project Tango is the 'Explorer' app, which implements visual odometry. The aim of this project is thus, to implement a visual odometry mechanism that functions in a similar way to the 'Explorer' app, with the exception that the one developed is for underwater usage. The visual odometry aspect of the device has an almost identical functionality to that of the 'Explorer' app (in particular, the 'Area Learning' mode) in Google's Project Tango. This app displays the path of a device as well as its current position and orientation. The trajectories displayed represent the motion tracking data based on two pairs of base and target frames:

- Device in respect to Start of Service, and
- Device in respect to a file that the user can load, called the Area Definition File (ADF).

Device in respect to ADF trajectory includes corrections based on area learning and if an ADF is loaded it will use the same origin as the ADF after the device has localized [3].

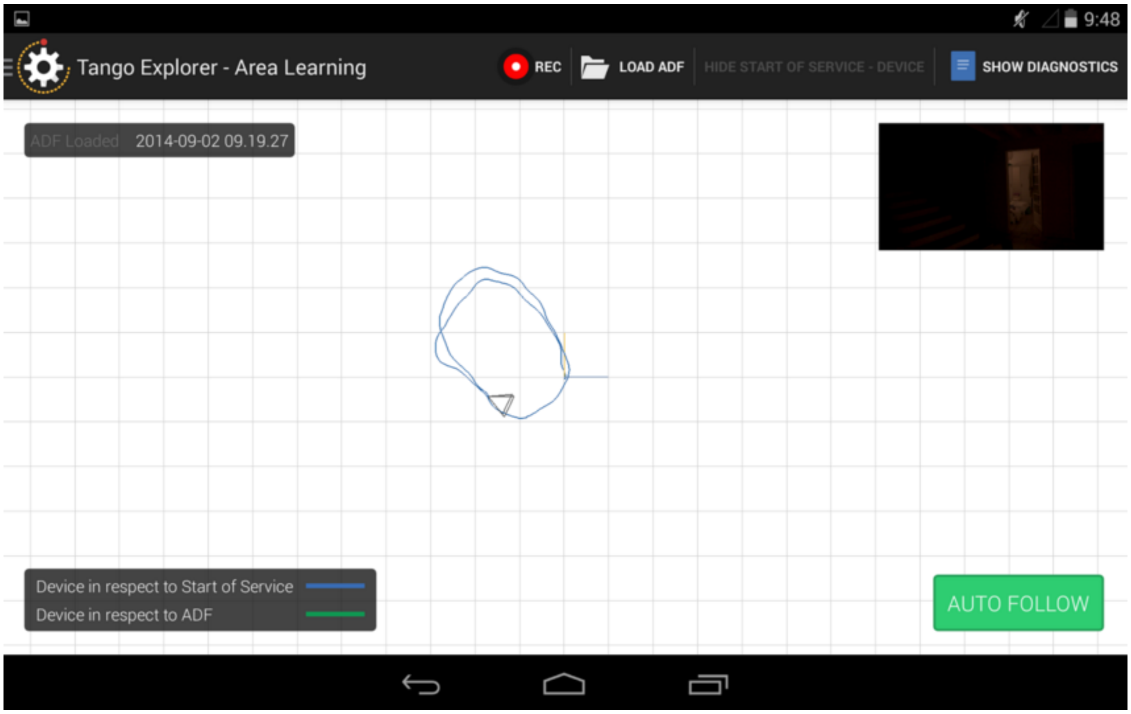

Figure 3. The Project Tango ‘Explorer’ app in ‘Area Learning’ mode [3]. The triangle represents the device.

2. Project Description and Goals

The fundamental goal of the Swimming Tango team is to design a system that implements an underwater visual odometry mechanism that would enable deep-sea divers to keep track of their movements and help them find their way back to the starting point should they lose their way. This system is intended to be a part of a device that would also enable deep-sea divers to communicate with each other via text messaging. This device would be no bigger than a handheld tablet or smartphone. The device must complement the diver's Underwater Breathing Apparatus (UBA), and not interfere with the swimmer's gear or buoyancy. The features of the device include [2]:

- Portability

- Lightweight
- Waterproof
- Long range
- Links a large number of divers

3. Technical Specifications & Verification

The technical specifications of the device are listed in Table 1. Since the team was unable to fully develop the prototype, it was unable to meet these specifications for the most part.

Table 1. Technical specifications of the device [2].

Specification	Intended Values	Actual Values
Screen size	ideally between 4-7 inches	15 inches (Laptop screen)
Thickness	preferably not more than 9mm (excluding SONAR)	n/a
OS	Ubuntu 14.04	Ubuntu 14.04
Max. depth	66 Feat of Seawater (FSW)	n/a
Navigations equipment	SONAR, IMU	SONAR, IMU

A comparison of SONAR specifications is shown in Table 2.

Table 2. Comparison of Tritech Micron DST and Blueview P900 series [5,6].

Specification	**Tritech Micron DST [5]**	**Blueview P900 [6]**
Size	Smallest digital CHIRP SONAR in the world. Dimensions shown in Figure 4.	12.4 in x 5 in
Max. depth	750m (around 2460 feet)	4000m (13123 feet)
Max. range	75m	100m
Min. range	0.3m	2m
Power requirement	12-48V DC at 4 VA average power	Connects to computer via ethernet port
Communications protocols	RS-232, RS-485 (twisted pair)	Ethernet

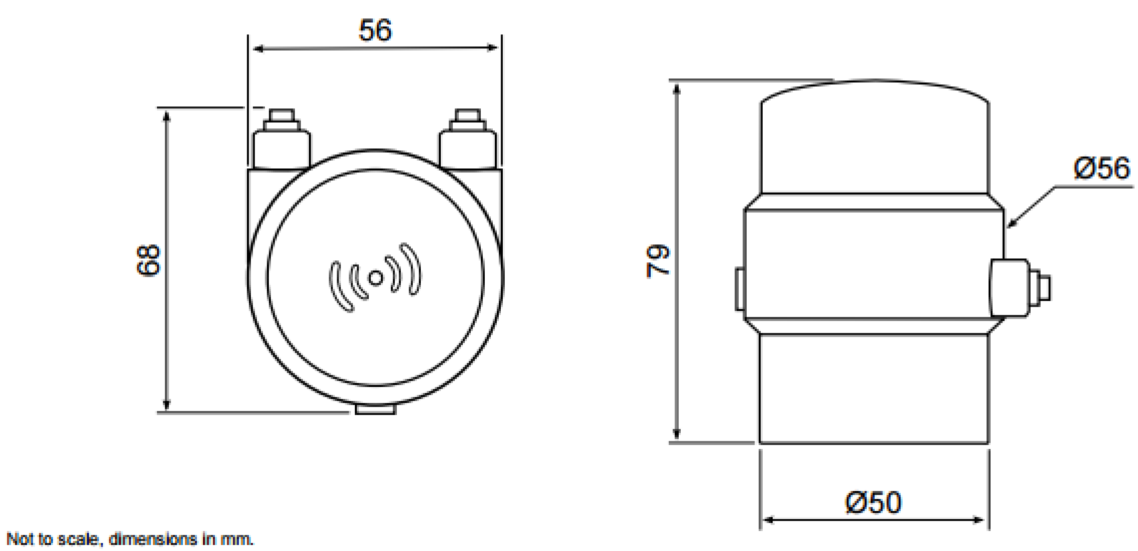

Figure 4. SONAR dimensions [5].

4. Design Approach and Details

4.1 Design Approach

The first step of the design was to list all of the desired specifications and objectives the design had to accomplish. The design constraints were derived from the objectives of the sponsor for this project. The constraints were as follows [2]:

- The product had to fit the hardware profile of a small tablet.
- The product had to work in deep sea environments with little to no clear light.
- The product should be able to provide the location of the diver at any time.
- The device needs to enable divers to communicate with each other.

With these constraints in mind, the team decided that a SONAR would be used to collect images of the underwater environment. These images would not be pictures; instead, it would capture pointcloud information. These pointclouds are then put through a visual odometry algorithm which tracks the relative position of the diver in real time. In addition to this, an IMU and a pressure transducer are used to make the relative position absolute. A block diagram depicting the execution of what the team wanted to develop is shown in Figure 5.

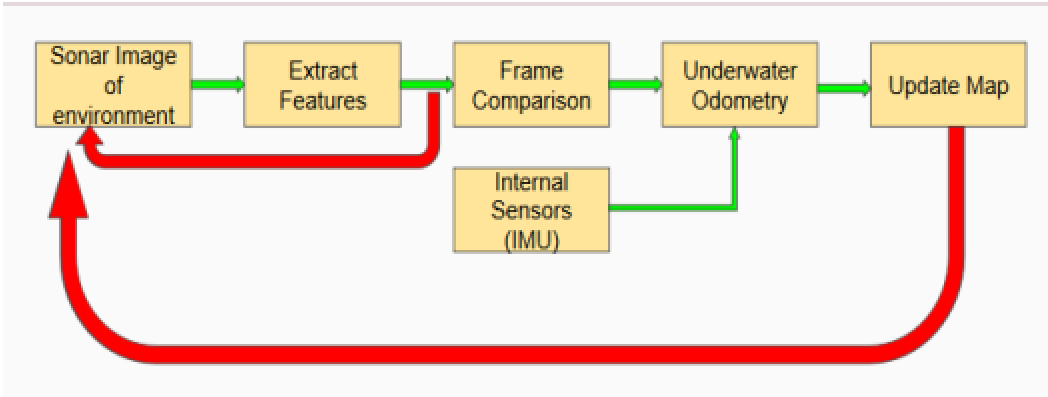

Figure 5. The concept of operations of Swimming Tango.

The team also decided to use a pressure transducer and a compass in order to determine the depth of the diver and his current heading. Their outputs were parsed using a Python script and converted into a human-readable format.

Owing to the various challenges faced by the team throughout the semester, it was not possible to completely implement the algorithm. The team was able to obtain a working visual odometry algorithm using image data taken on the surface; however, there wasn't sufficient time to implement the underwater aspect of the project. As such, the execution of the algorithm will be demonstrated using a dataset obtained from a project led by the Karlsruhe Institute of Technology and the Toyota Technological Institute in Chicago. This algorithm uses OpenCV. Individual frames are selected from images and the FAST feature detection algorithm is used to detect corners and turning points. The algorithm then uses the RANSAC algorithm to calculate the distance between features in frames. A screenshot of the execution of this project is shown in Figure 6.

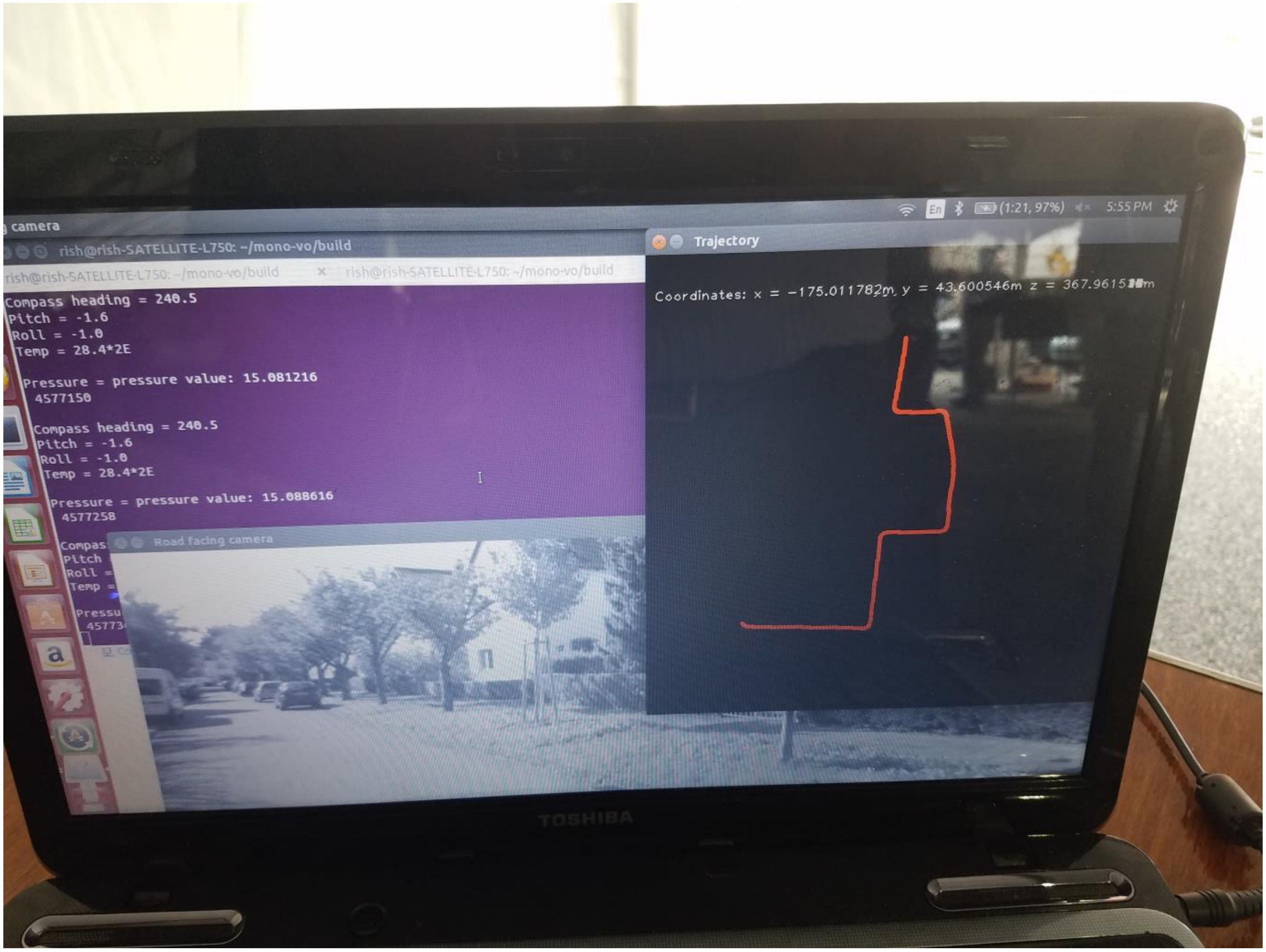

Figure 6. Execution of the project showing the visual odometry and the execution of the Python script.

4.2 Codes and Standards

The following codes and standards are the most relevant to the project:

- Universal Serial Bus (USB): It is a set of interface specifications for high speed wired communication between electronics systems peripherals and devices with or without PC/computer. The USB was originally developed in 1995 by many of the industry leading companies like Intel, Compaq, Microsoft, Digital, IBM, and Northern Telecom. The major goal of USB was to define an external expansion bus to add peripherals to a PC in easy and simple manner [7]. This standard was used when connecting the MBED microcontroller to the computer and had no effect on any design decisions,

- RS-232: It is a standard for serial communication transmission of data. It formally defines the signals connecting between a DTE (data terminal equipment) such as a computer terminal, and a DCE (data circuit-terminating equipment, originally defined as data communication equipment), such as a modem [8]. This standard was used to connect the compass to the computer. However, since the computer used did not have a serial port, a serial-to-USB converter had to be used.

- GNU General Public License (GPL): Over the course of implementing this project, a lot of open-source software was used, such as Ubuntu 14.04 LTS, Ubuntu ARM, Python, Eclipse and OpenCV, which required compliance with the GNU GPL.

4.3 Constraints, Alternatives, and Tradeoffs

Constraints

- Cost is one of the most important constraints for the design of any project, as it is the budget available that decides what parts can be purchased. This in turn would affect the implementation of the design. For example, one of the main aims of our project was portability, for which we wanted to use Tritech's DST Micron SONAR. However, this SONAR was well beyond our budget, and as such we could not afford it.

- Time is just about as important a design constraint as cost. Since the team just had one semester to work on this project, a prototype of the device could not be built from scratch.

- The final constraint would be team size. The team was slated to have four members initially, however this reduced to two members. Thus, it became hard to manage time, thereby limiting what the team could achieve over the course of the semester.

Alternatives

- The team initially wanted to use the Tritech Micron DST SONAR, due to its small size. However, this SONAR exceeded the budget by a significant amount. As a result, the team used the Blueview P900-5, as this was already available to the team.

Trade-offs

- The team had to make a choice between portability and functionality. Initially, it was decided that a Tritech Micron DST SONAR would be used, with a Raspberry Pi as the development platform. However, the Tritech Micron DST exceeded the budget by a significant amount and the team had issues installing some essential software on the Raspberry Pi. As such, it was decided that the Blueview P900-5 SONAR would be used, along with a laptop as the development platform. While this SONAR is significantly bulkier than the Micron DST, it is more accurate. Also, the team was able to run all software on the laptop.

- The team also had to decide between focusing on the underwater aspect of the project or the visual odometry implementation first. It was decided that visual odometry would be implemented first, since this is the essence of the project. However, shortage of time combined with the shortage of manpower meant that the team was unable to implement the algorithm for underwater purposes. As such, the project was demonstrated by showing the implementation of the algorithm using an on-the-surface image feed.

5. Schedule, Tasks, and Milestones

The overall timeline of the project development is shown in the Gantt Chart in Figure 7. Both team members contributed equally to all aspects of the project, such as the development, the presentations and this report. Since the team only comprised of two members, there was no division of tasks, as both members worked together on all tasks. All tasks were of high priority, since each was essential in order to best develop the project and meet all the requirements of the course.

	Task Name	Start	Finish
1	Prepare for Proposal Presentation	Tue 01/12/16	Thu 01/21/16
2	Proposal Presentation	Thu 01/21/16	Thu 01/21/16
3	**Set up Research Environment**	**Tue 01/26/16**	**Tue 02/02/16**
4	Install Ubuntu on netbook	Tue 01/26/16	Wed 01/27/16
5	Install ROS	Wed 01/27/1€	Thu 01/28/16
6	Clone McGill University GitHub repository	Thu 01/28/16	Fri 01/29/16
7	Prepare for Sponsor Presentation 1	Fri 01/29/16	Tue 02/02/16
8	Sponsor Presentation 1	Tue 02/02/16	Tue 02/02/16
9	**Test parts**	**Tue 02/09/16**	**Thu 03/03/16**
10	SONAR	Tue 02/09/16	Thu 03/03/16
11	IMU	Tue 02/09/16	Sat 02/27/16
12	Pressure Transducer	Tue 02/16/16	Sat 02/27/16
13	Set up Raspberry Pi	Thu 03/03/16	Fri 03/18/16
14	SPRING BREAK	Sat 03/19/16	Sun 03/27/16
15	Write code for pressure transducer and test it out	Tue 03/29/16	Tue 04/05/16
16	Obtain and compile BlueView SDK	Tue 03/29/16	Tue 04/19/16
17	**Test Visual Odometry algorithm**	**Tue 04/19/16**	**Tue 04/26/16**
18	Final Sponsor Presentation	Tue 04/19/16	Tue 04/26/16
19	Expo	Tue 04/26/16	Tue 04/26/16

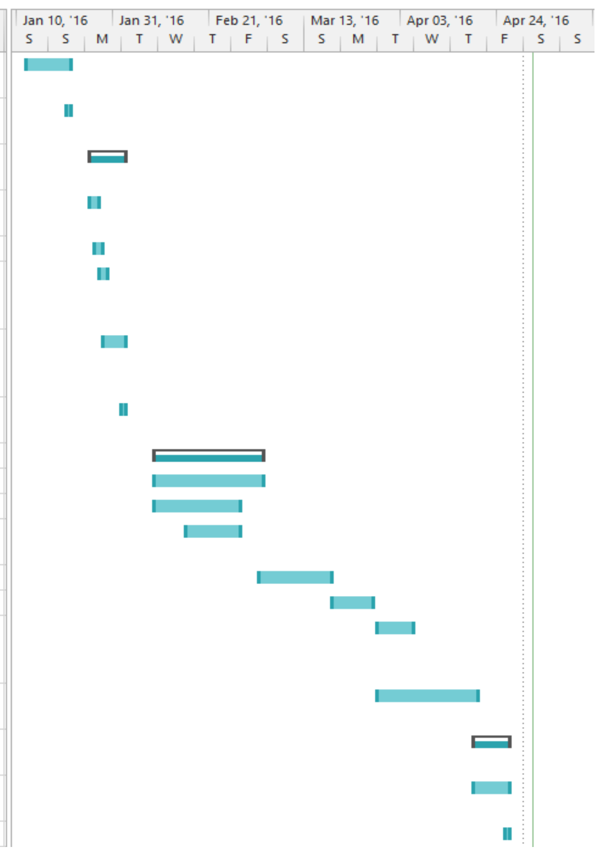

Figure 7. Gantt Chart depicting the project timeline.

6. Final Project Demonstration

The project was demonstrated at the Capstone Design Expo on 26 April 2016. The following components were demonstrated:

- A shell script was written to run the visual odometry algorithm multiple times, thereby eliminating the need to run the program every time execution terminated.

- The pressure sensor was connected to an mbed microcontroller on a breadboard, which was then connected to the computer via a USB cable. The compass was connected to the computer using a serial-to-USB cable. Data from the pressure transducer and compass was parsed using a Python script written by the team, which then displayed the output on the terminal in a human readable format. A second version of the script was also written that would log the output to a file instead of displaying it on the terminal.

- A poster was also prepared to explain the project in detail and how the team went about attempting to solve the problem. The poster provided a brief introduction on what objectives the team intended to achieve and block diagrams explaining how the team intended to achieve these objectives.

- Owing to time constraints, the underwater aspect of the project was not developed. It was decided to work on the visual odometry aspect first, in order to have a presentable aspect for the Expo. As such, the visual odometry algorithm was demonstrated using an image feed obtained from an online dataset.

Appendix A contains a list of all supplementary material that would be useful to operate this project. In particular, it contains a user guide, the poster, links to the algorithm and datasets, photographs taken at the Expo and also the Python script that was used to parse data from the pressure transducer and the compass and convert it into a human readable format. Two versions of this script are provided in the link; Version 1 displays all output on the terminal, while Version 2 logs all output to a file.

7. Marketing and Cost Analysis

7.1 Marketing Analysis

The target market of this product is primarily the U.S. Department of Defense. However, it can be used by any deep-sea diver. While the team was unable to develop the prototype this semester, it is anticipated that a fully functioning device would enable underwater visual odometry. While visual odometry has been implemented by Google in the Project Tango Explorer app, an underwater implementation is a first. Further, when completed, the prototype will also enable divers to communicate with each other. Underwater Technologies Center (UTC) has developed a line of smartwatch-like devices called UDI. These devices allow divers to communicate with each other through text messages, just like this product [9]. However, this product allows more divers to communicate with each other on the same channel, thereby avoiding the need to use multiple channels. In addition, this product provides real-time tracking of a diver’s position, since it is fitted with a Micron DST SONAR. The product will however be slightly larger than the UDI devices in order to be able to accommodate the extra sensors.

7.2 <u>Cost Analysis</u>

Since the team was unable to develop a prototype this semester, the values in the following section are the costs that the team expects to be incurred during the development process. Table 3 shows a breakdown of the material costs of the prototype. Since the exact components have not been decided yet, the costs mentioned in this section are just estimates. However, the team anticipates that these would be reasonably close to the true cost. The most expensive equipment is the SONAR.

Table 3. Prototype Equipment Costs.

Product description	Quantity	Unit Price ($)	Total Price ($)
Tritech Micron DST SONAR	1	$7000	$7000
Triple Axis Accelerometer and Gyro	1	$39.95	$39.95
Project Tango Development Tablet	1	$512	0 (received for free)
TOTAL COST			$7039.95

The total development costs shown in Table 4 were determined with an assumed labor cost of $40 per hour. Fringe and overhead costs are factored into the higher costs and will be amortized over all units produced. The values in the labor hours field are just estimates.

Table 4. Development Costs

Project Component	Labor hours	Labor Cost	Part Cost	Total Component Cost
DEVICE DEVELOPMENT				
Assembly, OS Development and Install	100	$4,000	$39.95	$4,039.95
Testing	50	$2,000		$2,000
SWIMMING TANGO IMPLEMENTATION				
Algorithm formulation	80	$3,200		$3,200
Coding	120	$4,800		$4,800
Debugging and Testing	120	$4,800		$4,800
DEMO PREPARATION	120	$4,800		$4,800
GROUP MEETINGS	300	$12,000		$12,000
PAPERS AND PRESENTATIONS	100	$4,000		$4,000
TOTAL LABOR	990	$39,600		
TOTAL PART COST			$7039.95	
TOTAL COST (LABOR + PART)				**$46,639.95**

Using the fringe benefit as 30% of total labor and overhead costs as 120% of material and labor costs, the total development cost for the device is $121,743.89, as shown in Table 5.

Table 5. Total Development Costs

Component	Cost
Parts	$7039.95
Labor	$39,600
Fringe benefits	$11,880
Subtotal	$51,519.95
Overhead costs	$70,223.94
TOTAL COST	**$121,743.89**

8. Conclusion

The main goal for the semester was to develop an underwater implementation of visual odometry. Unfortunately, the team fell short on this regard, as it was unable to implement the algorithm using the SONAR. The SONAR uses pointcloud data, while the algorithm uses images. The team was however, able to lay a foundation for a future senior design team to build on. The python script that was written in order to translate the outputs from the compass and pressure sensor into a human readable format also adds a timestamp every time the data is recorded. This data can be stored in a file and the timestamps can be very useful in helping stitch pointclouds together from the SONAR, thereby enabling the algorithm to be expanded for underwater usage.

The team faced a lot of challenges over the course of the semester that hindered progress. Probably the most important one would be the fact that the team was initially slated to have four members, but this dropped down to two. As such, the team was short of manpower for the entirety of the development process. Despite this, the best efforts were made by both team members. Another challenge faced was that initially, the Tritech Micron DST SONAR was to be used. However, there was a lot of uncertainty regarding funding, which was only resolved in the first week of March. As such, it was decided that the Blueview P900-5 would be used for development purposes, since this SONAR was already available to the team. Finally, the SONAR software development kit (SDK) would not compile on the Raspberry Pi that the team was using for development. Thus, it had to be compiled on a laptop. This occurred during the first week of April, and thus the team had to set up all the required software packages yet again.

The team has developed the foundations for a future team to carry on and implement the underwater visual odometry mechanism. The mechanism could first be implemented on the Blueview P900-5 and then could be ported to the Tritech Micron DST. Once this implementation is complete, it can be combined with a device that would enable divers to communicate with each other. This was the ultimate aim of the project; to create a device that would enable divers to communicate with each other as well as keep track of their movements while underwater. The visual odometry aspect can also be optimized further. Another potential area of future research would be to create a 3-D mapping of the underwater terrain using the SONAR, by identifying key features underwater and stitching multiple images together.

9. References

[1] Google, 'Project Tango – Google', 2015. [Online].

Available: https://www.google.com/atap/project-tango/about-project-tango/

[2] Capstonemarketplace.org, 'Projects', 2015. [Online].

Available: http://www.capstonemarketplace.org/projects/view/44

[3] Google Developers, ‘Project Tango Explorer’,2015. [Online].

Available: https://developers.google.com/project-tango/tools/explorer#area_learning

[4] G. Taraldsen, T. Reinen and T. Berg, 'The underwater GPS problem', *OCEANS 2011 IEEE - Spain*, 2011.

[5] Tritech, ‘Micron Sonar Datasheet’, tritech.co.uk, 2015. [Online]

Available:

http://www.tritech.co.uk/media/products/small-rov-mechanical-sector-scanning-sonar-tritech-micron.pdf

[6] Blueview, ‘PSeries-Deepwater-Datasheet-v5’, blueview.com, 2015. [Online]

Available: http://www.blueview.com/assets/Uploads/downloads/PSeries-Deepwater-Data-Sheet-v5-web1.pdf

[7] EEHerald.com, ‘USB interface tutorial’, 2015. [Online]

Available: http://www.eeherald.com/section/design-guide/esmod14.html

[8] Radio-Electronics.com, ‘RS-232 Serial Interface Tutorial’, 2015. [Online]

Available: http://www.radio-electronics.com/info/telecommunications_networks/rs232/rs232-serial-interface-basics-tutorial.php

[9] i. xnet, 'Overview', *Utc-digital.com*, 2015. [Online].

Available: http://utc-digital.com/product_overview.asp?cid=80&pr=38

APPENDIX - A

Listing of all supplementary material

1. **User Guide**

   Available:

   http://www2.ece.gatech.edu/academic/courses/ece4012/16spring/p51/User%20Guide.pdf

2. **Poster**

   Available:

   http://www2.ece.gatech.edu/academic/courses/ece4012/16spring/p51/Capstone%20Poster.pdf

3. **Visual Odometry GitHub repository**

   Available: https://github.com/avisingh599/mono-vo

4. **Visual Odometry Datasets**

   Available: http://www.cvlibs.net/datasets/kitti/eval_odometry.php

5. **Python Scripts to Parse Serial Data**

   Available: http://www2.ece.gatech.edu/academic/courses/ece4012/16spring/p51/serial_read.zip

6. **Project Photographs**

   Available:

   http://www2.ece.gatech.edu/academic/courses/ece4012/16spring/p51/Expo%20Photos.zip

7. **Tritech Micron DST Datasheet**

Available:

http://www.tritech.co.uk/media/products/small-rov-mechanical-sector-scanning-sonar-tritech-micron.pdf

8. **Blueview Series Datasheets**

Available:

http://www.blueview.com/assets/Uploads/downloads/PSeries-Deepwater-Data-Sheet-v5-web1.pdf