Velocity of an ROV from Optical Flow and Stereo Cameras

Jill Alexander, Boise State University

Mentors: Dr. Kakani Katija, Dr. Joost Daniels, and Dr. Paul Roberts

Summer 2020

Keywords: Optical Flow, Stereo Vision, Midwater Navigation, OpenCV

ABSTRACT

This paper describes a method to find the velocity of an underwater vehicle using stereo camera data and optical flow data. The algorithm uses an optical flow function from the OpenCV library to find the velocity of particles on two different camera feeds. A stereo solution is then used to find a resulting 3D velocity vector. Results show that this method can return accurate velocity measurements when appropriate correction factors are applied. This algorithm can be used in-situ or after data collection to generate velocity data for underwater vehicles equipped with stereo cameras.

INTRODUCTION

The ocean's midwater range is the largest ecosystem in the world. The part of the ocean between the seafloor and the surface, the midwater region of the ocean is home to the planet's largest animal communities. Navigating in the midwater region offers unique challenges. Far from the ocean floor, low light and lack of a steady reference frame makes it difficult for underwater vehicles in the midwater to get consistently accurate velocity and position readings. GPS, a method often used for gathering positioning data, is unusable underwater. Remotely operated vehicles (ROVs) often have to use acoustic measurements from launch vessels that can be inaccurate far away from the reference frame of the ocean floor.

There are some consistent features to be found in the midwater. Marine snow, small particles of organic matter that drift slowly down to the seabed, is found almost everywhere in the midwater. Marine snow falls slowly enough relative to an underwater vehicle that it can be assumed to be stationary. By tracking the movement of the marine snow past cameras attached to an underwater vehicle, we can estimate the velocity of the underwater vehicle. This movement can be estimated using a concept called optical flow.

Optical flow is the apparent change in motion of an object, in this case in between frames of a camera. Optical flow is a video processing technique useful for tasks such as object tracking. The optical flow functions built into the OpenCV library, an open source computer vision library, return 2D vectors showing the movement of objects in between frames of a camera feed. One drawback of this function is that it cannot return 3D velocity directly. Previous research has solved this problem by adding other velocity sensors or by using equipment such as a monocular camera (Ho, de Croon, & Chu, 2017).

In this paper, a pair of stereo cameras are used to create a stereo solution. A stereo solution uses known geometries of two cameras in order to generate 3D information about an object seen by both cameras. By using an optical flow function in conjunction with a stereo solution, we can create an algorithm that returns the 3D velocity of an ROV. This paper details the creation and testing of this algorithm. The algorithm was tested in a variety of simulations before being compared to in-situ data taken from a real ROV.

MATERIALS AND METHODS

ALGORITHM DESCRIPTION

The velocity algorithm employed in this paper uses the Farneback optical flow function from the OpenCV library. This function estimates motion between two frames using approximation of each neighborhood of both frames using quadratic polynomials (Farnebäck , 2003). The function returns a matrix containing change in x and y coordinates for each neighborhood of the frame. In order to simplify this matrix, the average change in x and y over a specific region of interest (ROI) was taken. The ROI for each frame was chosen for areas where the field of view was best lit by the attached cameras. The coordinate system for the 3D space was defined with the XY plane parallel to the left camera, and the Z axis perpendicular to the left camera, positive in the forward direction.

The stereo solution used was previously developed by Mike Risi and Steve Rock, and was refined for this purpose by Paul Roberts. The stereo solution was used to find the 3D coordinates of the total centroid of the space bounded by the ROI. The change in X and Y from the local centroid on each camera was calculated for each frame, and the resulting left and right coordinate pairs were input into the stereo solution. The total centroid was then subtracted from the 3D points returned from the stereo solution, resulting in a total X, Y, and Z change in mm/frame. This change in X, Y, and Z was then multiplied by the framerate in order to get a 3D velocity. This calculation was done for each frame, returning the X, Y, and Z velocity over time.

ROV EQUIPMENT

The in-situ data was collected using the MiniROV, an electrically powered ROV equipped with a stereo camera pair (shown in Figures 1 and 2) as well as an Autonomous Doppler Velocimeter (ADV).

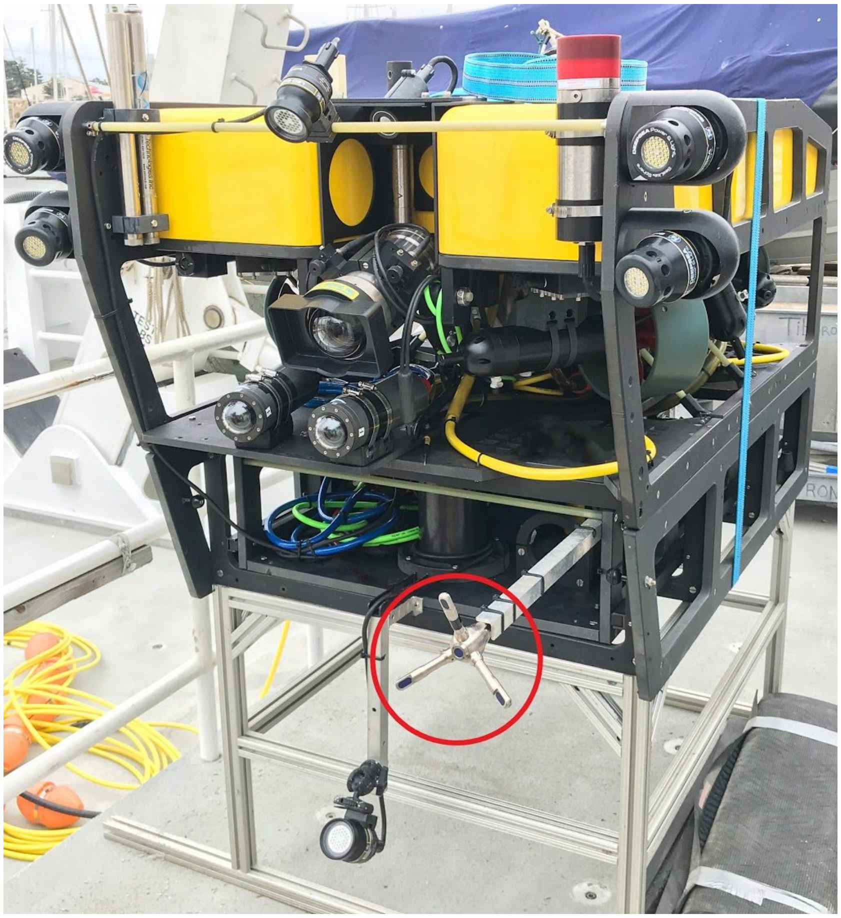

Figure 1: MiniROV with ADV circled in red.

The ADV, shown circled in Figure 1, is an acoustic device that uses particles in the water to measure velocity. The ADV generates velocity data based off of particles, similar to the optical flow method, which makes it appropriate for comparison data.

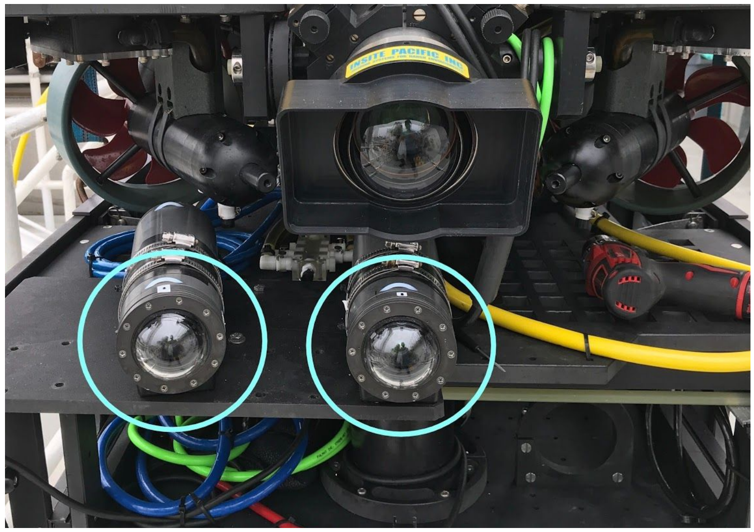

Figure 2: Close up of MiniRov showing stereo camera pair (circled in blue).

RESULTS

SIMULATION VALIDATION

Initial test simulations were generated using Blender, an open-source D computer graphics software package. A particle field of approximately 1000 particles/m^3 was generated and various movements past a stereo camera pair were tested. Speeds between -200 mm/s and 500 mm/s were chosen and tested along the three different axes. Movement in all three directions at the same time was also tested. An example of the velocity output over time for a 200 mm/s lateral move is shown below. As shown in Figure 3, the optical flow output shows a large X velocity and relatively low Y and Z, as expected. However, there is a clear offset from the simulation input, as well as a significant spurious Z.

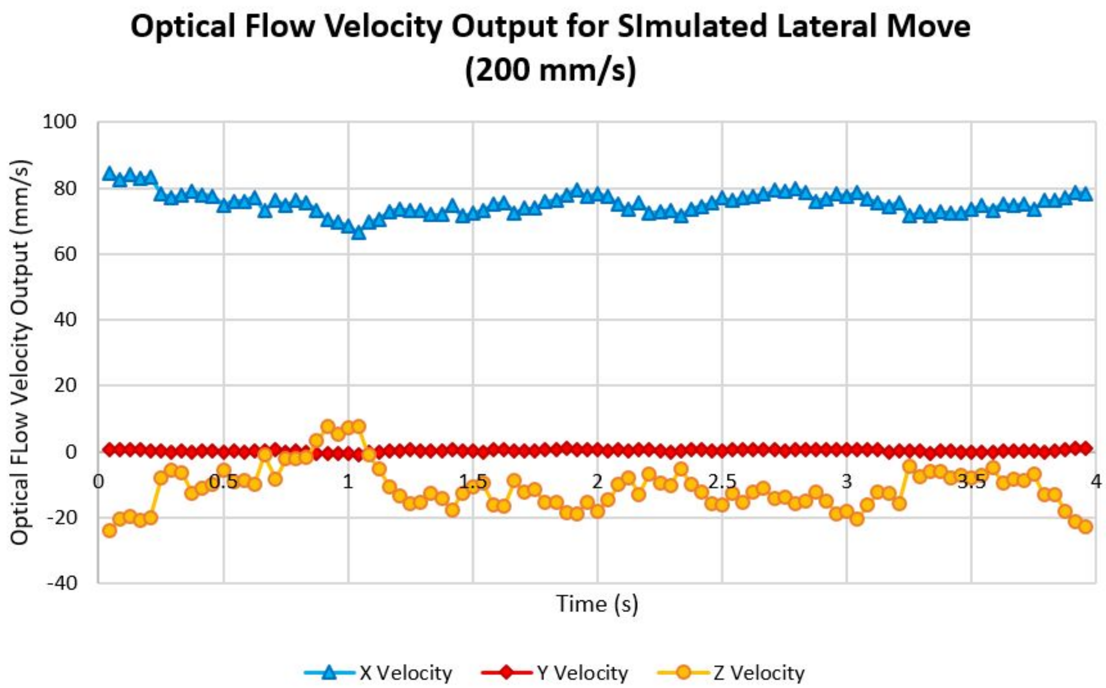

Figure 3: Optical flow velocity output over time for a simulated 200 mm/s lateral move. X and Y are measured parallel to the left camera plane, with the X axis positive to the right and the Y axis positive to the left. The Z axis is perpendicular to the left camera plane and is positive forward.

The results for different speeds and movement directions show a similar trend. Figure 4 shows the average velocity results for each simulation compared to the simulation input. There is a consistent offset from the actual simulation input. Some spurious Z results can also be observed when simulation Z input is equal to zero.

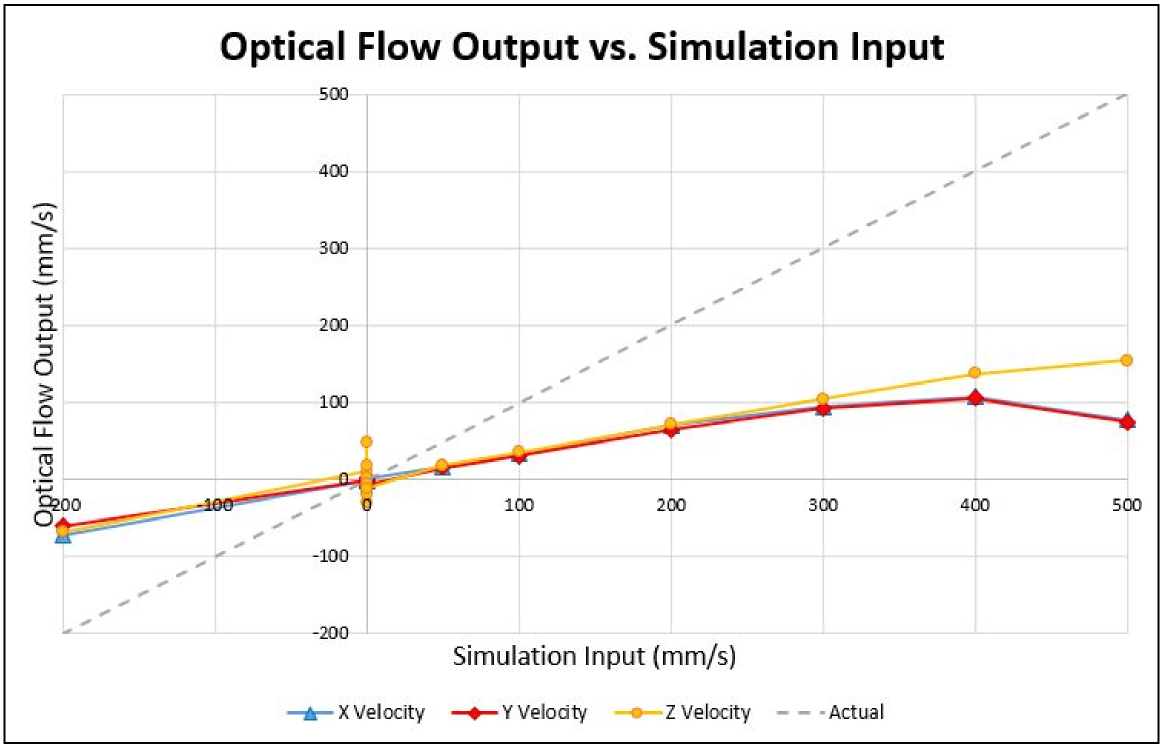

Figure 4: Graph showing the uncorrected optical flow results vs. the simulation input. Each point represents the average velocity output over a single simulation at constant velocity along either the X, Y, or Z axis. The gray line represents the ideal output (output exactly matching simulation input.)

However, despite the spurious Z, the results are relatively linear and consistent. This suggests that this offset results from some feature of the cameras and could be corrected using a constant factor. To calculate this correction factor, the average correction factor for each simulation along each axis was calculated, and then weighted using standard deviation. Simulation results with a low standard deviation were weighted higher than results that showed a high standard deviation. The optical flow function was shown to be more accurate at lower speeds, with lower standard deviations at speeds at and below 200 mm/s, so only simulation within those speeds were accounted for when calculating the correction factor. The final correction factors, along with their weights, are shown in Table 1.

	Weight (1/Std. Dev)			Correction Factor		
Actual Speed (mm/s)	X	Y	Z	X	Y	Z
-200	2.931	3.687	0.478	2.793	3.323	3.001
50	8.850	7.102	1.596	2.930	3.182	2.679
100	7.710	7.418	1.067	2.930	3.201	2.779
200	3.064	2.040	0.280	2.849	3.061	2.817
Average				**2.876**	**3.192**	**2.819**
Weighted Average				**2.901**	**3.203**	**2.766**

Table 1: Table showing the weights for each simulation (1/standard deviation) and the correction factor for each simulation. The total average and total weighted average correction factors are shown at the bottom of the table.

In order to further investigate the spurious Z, the Z output over time for several simulations was graphed. As shown in Figure 5, the spurious Z increased at faster speeds. A second order correction was applied to reduce spurious Z values.

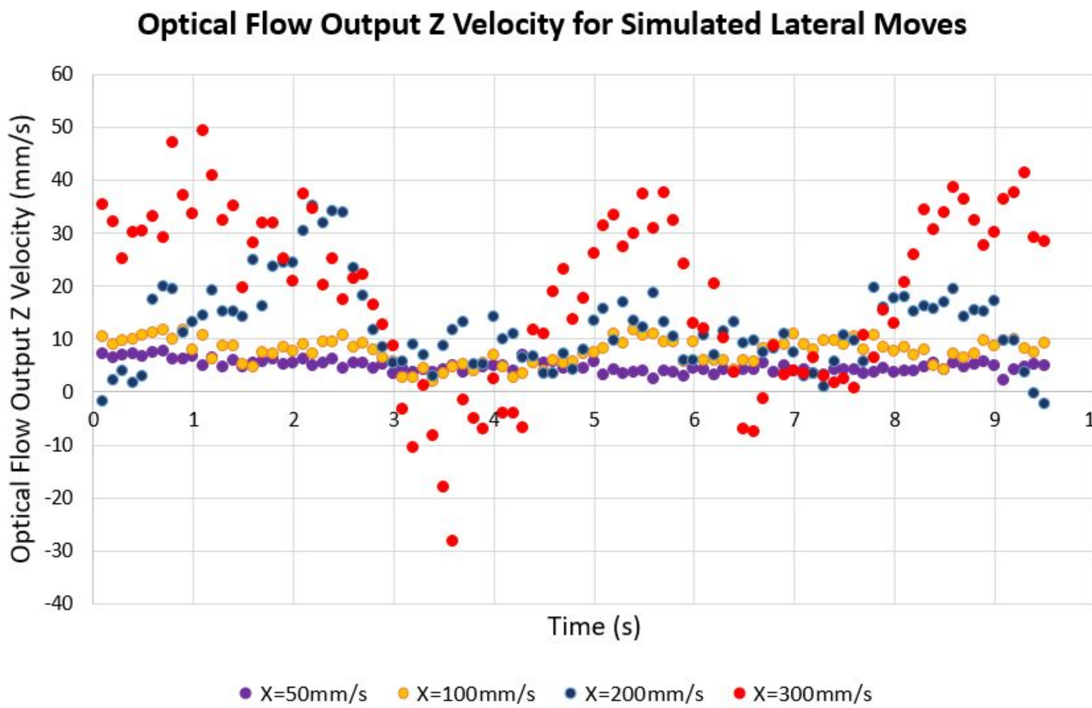

Figure 5: Optical flow Z output over time for four simulations with zero Z velocity input and various lateral speeds.

After applying the correction factors, as well as the Z correction, the optical flow output was much closer to the simulation input. As shown in Figure 6, the corrected optical flow output realistically represented the simulation speed.

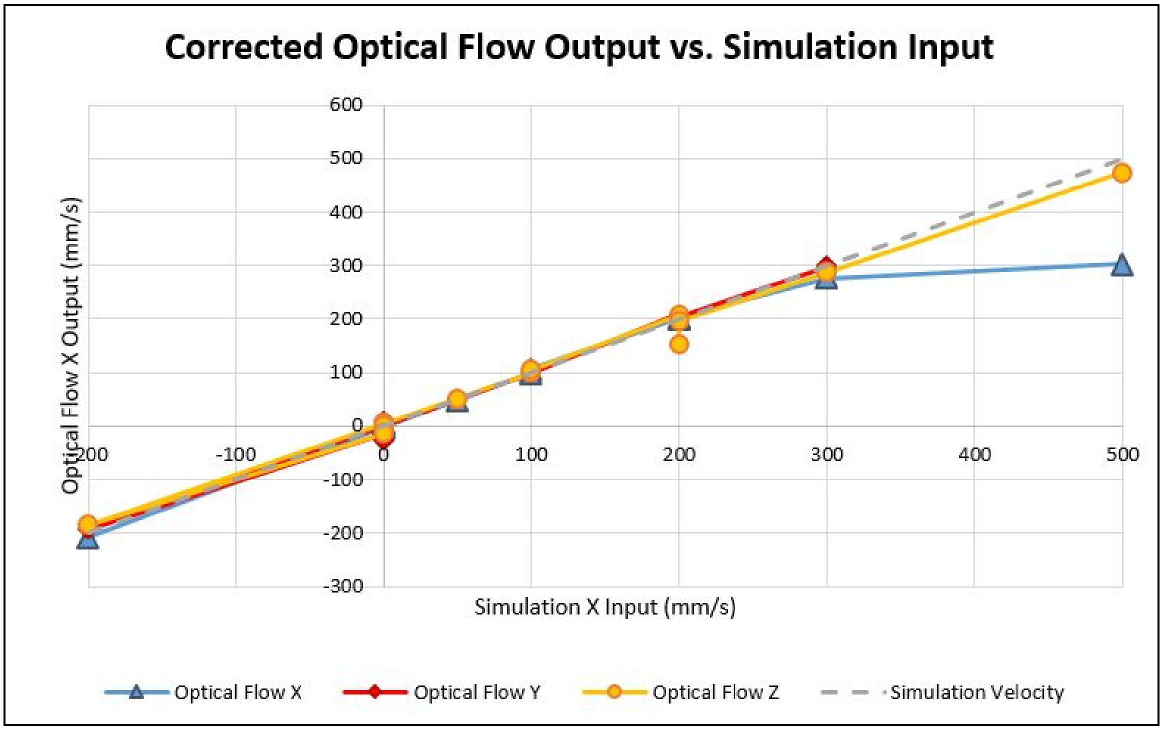

Figure 6: Graph showing the uncorrected optical flow results vs. the simulation input. Each point represents the corrected average velocity output over a single simulation in one direction. The gray line represents the ideal output (output exactly matching simulation input.)

IN-SITU DATA

The optical flow data was also compared to real world data. Real world data was taken during two scientific cruises in July 2020 on the Monterey Bay using the ROV and equipment described in Methods and Materials. Data was taken when the ROV was moving at a constant thruster setting, giving us periods of time when the ROV was moving at a semi-constant speed. Comparison data was chosen with thruster settings at 20 percent, 30 percent, and 50 percent, at a depth between 160m and 400m. A period of time when the ROV was ascending after a dive was also analyzed. The correction factor previously applied for the simulations was removed. Due to the change in camera set up and calibration, the correction factors previously calculated were determined to no longer be accurate. The ADV data and the optical flow output was smoothed to result in 1 Hz measurements. The ADV data was smoothed by taking the average of 30 samples over a single second. The optical flow results had a frequency of 10 Hz and were smoothed using a moving average with an interval of 10 samples. As shown in Figures 7 and 8, the optical flow data showed a clear correlation to the acoustic velocity readings taken by the ADV. After smoothing, the optical flow results often showed less noise than the ADV readings, which is especially apparent in Figure 8.

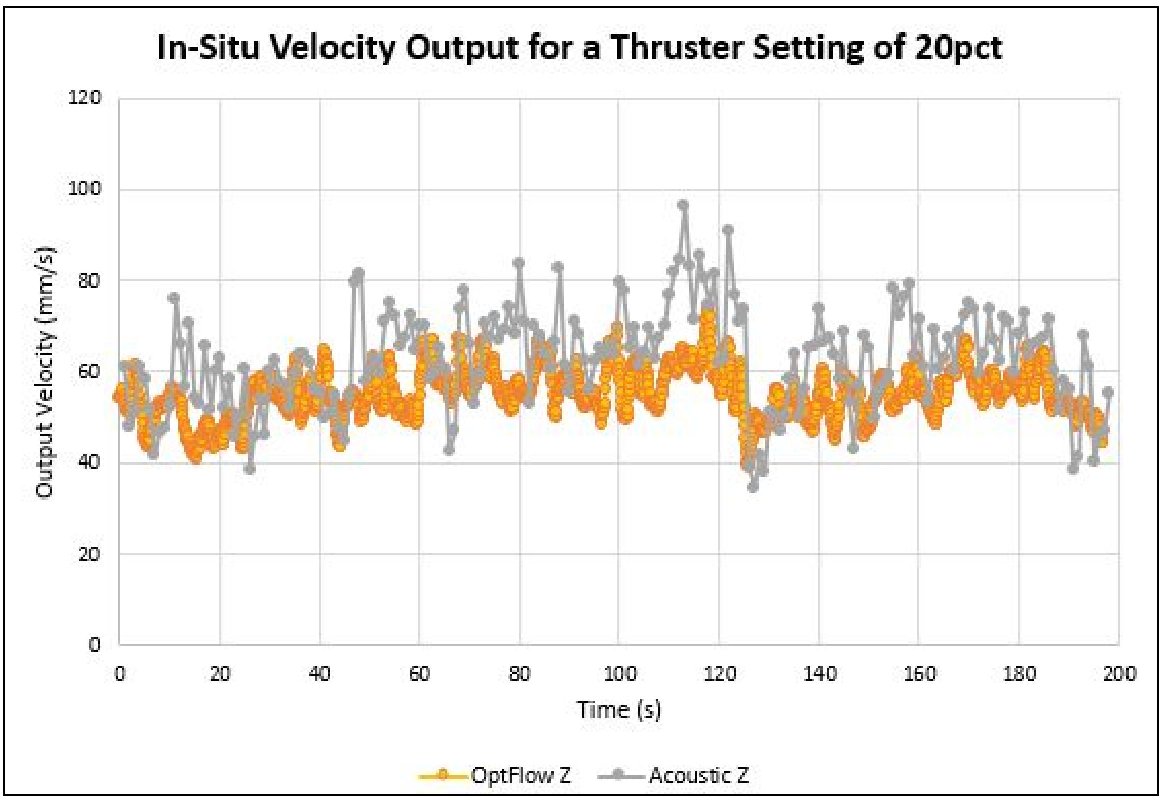

Figure 7: optical flow and acoustic ADV Z velocity results over time for a period of time when the thruster setting was 20 percent.

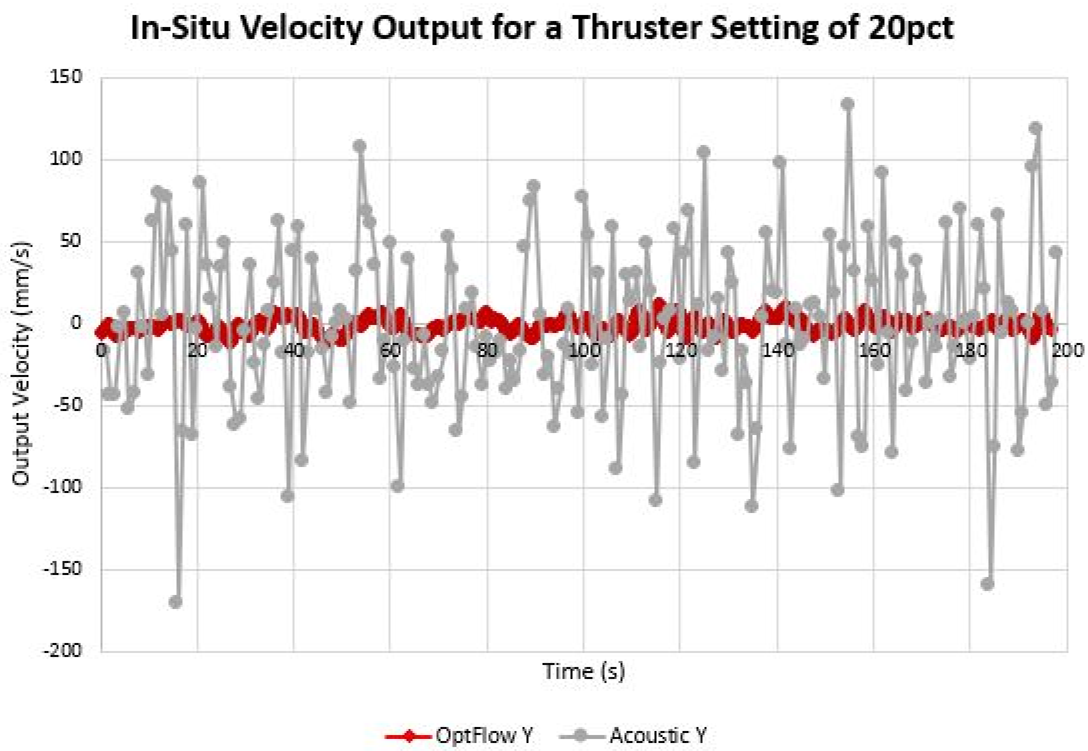

Figure 8: optical flow and acoustic ADV Y velocity results over time for a period of time when the thruster setting was 20 percent.

The optical flow results were significantly more noisy at times when the ROV was moving faster. As shown in Figure 9, the optical flow data was far less accurate and much more noisy at a thruster setting of 50 percent, which translated to a velocity of around 300 mm/s.

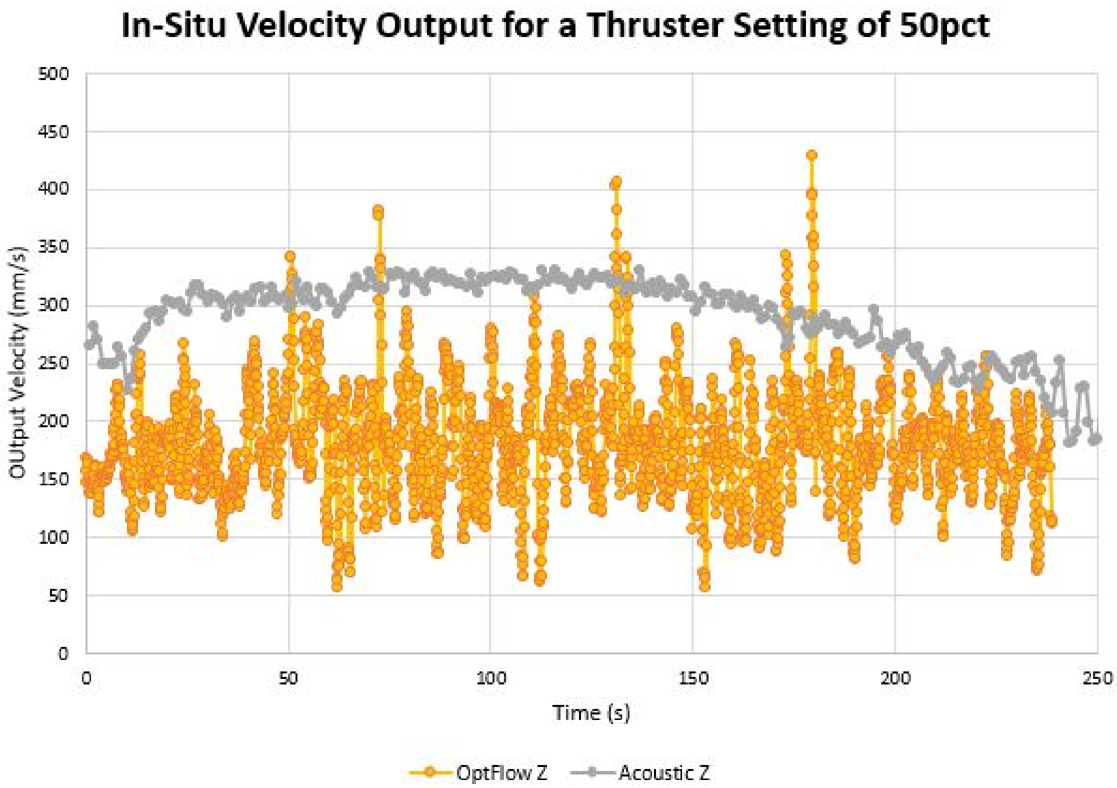

Figure 9: optical flow and acoustic ADV Z velocity results over time for a period of time when the Thruster setting was 50 percent.

Data collected during a period of time when the ROV was ascending after a dive was also inaccurate and noisy. The optical flow output during the ascent was either significantly noisy and showed a significant offset (Figure 10c) or failed to register the velocity at all (Figures 10a and 10b).

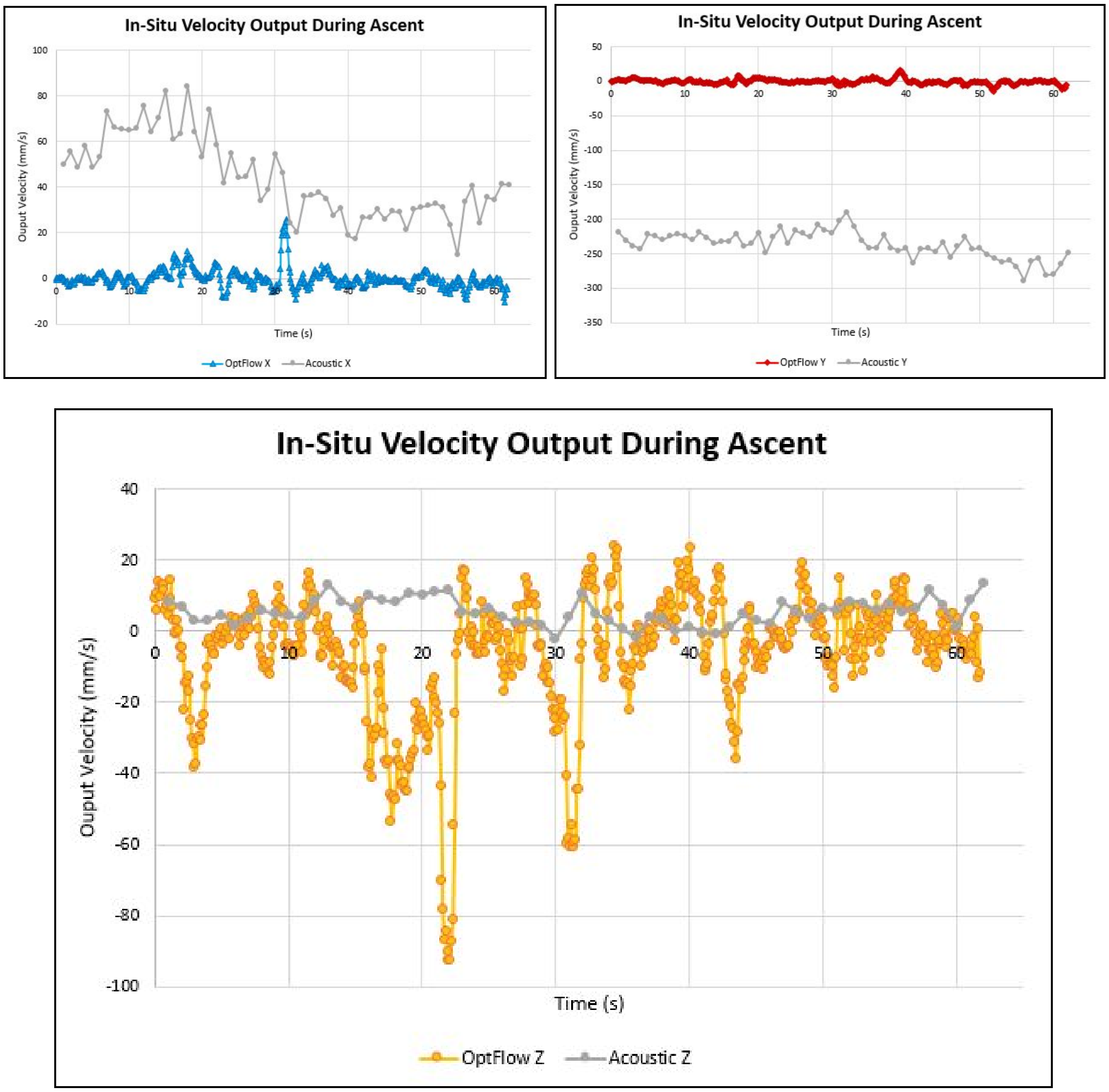

Figure 10a, 10b, 10c: optical flow and acoustic ADV X (a), Y (b), and Z (c) velocity results over time for a period of time when the ROV was ascending very quickly.

DISCUSSION

The optical flow algorithm showed a clear correlation to the actual velocity in both the simulations and during real world testing. The simulation results needed to be corrected with a linear correction factor before they could be considered accurate, but returned consistent results. The in-situ optical flow output was uncorrected, but much more closely followed the ADV data. However, it is still possible that a correction factor could improve these results, and should be more thoroughly investigated in the future.

The optical flow function demonstrated a drop in accuracy at higher speeds. In the simulation results, this drop is clearly visible at around 300 mm/s. The in-situ data showed a similar trend, with significantly less accuracy when thrusters were set at 50 percent, translating to a velocity of around 250 - 300 mm/s. However, below 200 mm/s is a realistic speed for an ROV. As shown in Figure 11, the ROV rarely reached speeds higher than 200 mm/s.

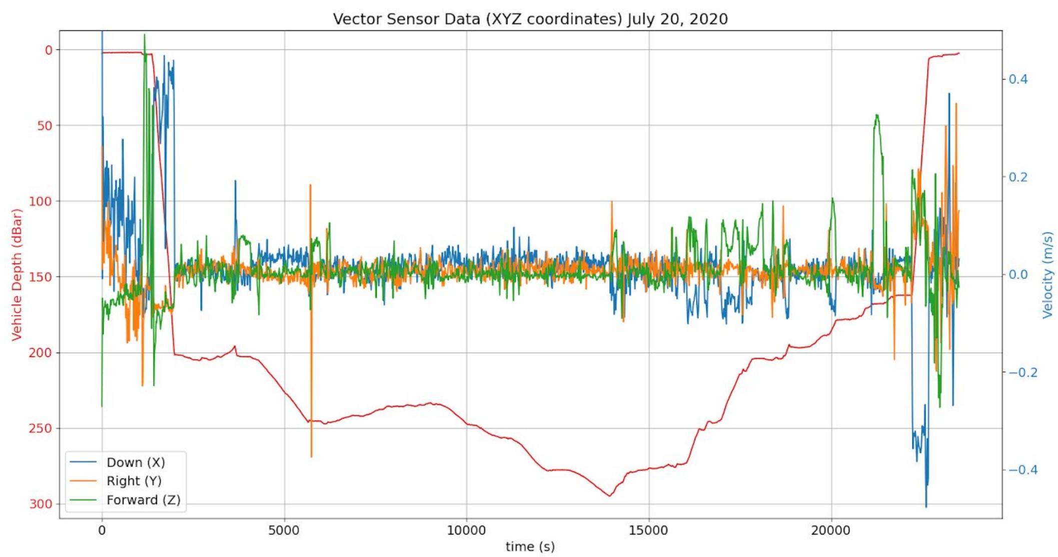

Figure 11: Speed and depth measurements over time of the MiniROV during a July cruise. Note that the velocity scale is shown on the right side, in m/s.

Time spent at speeds higher than 200 mm/s were typically during the beginning descent and final ascent of the ROV, when knowing the speed of the ROV is much less important for navigation. Due to the amount of time spent at lower speeds, this algorithm is still practical for use in ROV midwater navigation.

CONCLUSIONS/RECOMMENDATIONS

Based on our results, it can be concluded that the optical flow output can represent actual velocity when the appropriate correction is applied. This function has been shown to be most accurate at speeds at or below 200 mm/s for a framerate of 10 Hz. Although accuracy could be improved with an increased framerate, 200 mm/s is a realistic speed for an underwater vehicle.

While this method can accurately measure velocity, it is still heavily reliant on the density of particles, due to the method of averaging that is used and the structure of the optical flow function. However, there are other optical flow functions in the OpenCV library that are less reliant on particle density that could improve the algorithm in the future. Using other optical flow functions could allow for more filtering by particle size or particle density, which could be useful for removing noise and generating more accurate results.

In addition, the correction factor relied heavily on during the simulations phase could also be investigated more heavily in the future. Determining the exact relationship between calibration of the stereo cameras and the correction factor will allow data to be corrected at collection time, rather than after analysis of the data.

While future improvements should be made before practical use, velocity results from this method are consistent and show a clear representation of the actual data. The algorithm used is especially suited to the unique features of the midwater and the low speed movement of the ROV used. The combination of optical flow and a stereo solution is a promising method for getting velocity data in the midwater and is worth further research.

ACKNOWLEDGEMENTS

I would like to give my thanks to my mentor at MBARI, Dr. Kakani Katija, Dr. Joost Daniels, and Dr. Paul Roberts from the Bioinspiration Lab for their support and guidance. I'd also like to thank the staff of MBARI, and intern coordinators Dr. George Matsumoto and Megan Bassett, whose hard work and support made this paper possible. I'd also like to thank the sponsors of the MBARI summer internship program, the Dean and Helen Witter Family Fund, the Rentschler Family Fund, and the David and Lucile Packard Foundation, for their generous funding.

References:

Farnebäck, Gunnar, (2003). Two-Frame Motion Estimation Based on Polynomial Expansion. In: Bigun J., Gustavsson T. (eds) Image Analysis. SCIA 2003. Lecture Notes in Computer Science, vol 2749. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45103-X_50

Ho, H. W., de Croon, G. C., & Chu, Q. (2017). Distance and velocity estimation using optical flow from a monocular camera. International Journal of Micro Air Vehicles, 198–208. https://doi.org/10.1177/1756829317695566