2013

REAL-TIME FEATURELESS VISUAL ODOMETRY FOR SEA FLOOR IMAGING WITH A LAGRANGIAN FLOAT

David S. Casagrande
University of Rhode Island, davecasa@gmail.com

REAL-TIME FEATURELESS VISUAL ODOMETRY FOR SEA FLOOR IMAGING WITH A LAGRANGIAN FLOAT

BY

DAVID S. CASAGRANDE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
OCEAN ENGINEERING

UNIVERSITY OF RHODE ISLAND

2013

MASTER OF SCIENCE THESIS

OF

DAVID S. CASAGRANDE

APPROVED:

Thesis Committee:

Major Professor Chris Roman

Peter Swaszek

Jason Dahl

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2013

ABSTRACT

A Lagrangian float with bottom-imaging cameras is used for benthic surveys as it drifts a nominally constant altitude over the bottom. To maintain constant spatial sampling, the camera capture rate must be adjusted in real time based on vehicle speed. This speed is difficult to measure with conventional sensors, but can be found from the survey images using visual odometry. A featureless technique is used due to its increased robustness to noise and focus errors over feature-matching, along with a faster and more consistent computation time. A stereo pair of images taken at each vehicle location is used to find altitude. Then, the image from one camera is registered to the same camera's previous image with phase correlation, correcting for rotation and scale differences using a log-polar transformation. This registration is combined with known camera geometry to find vehicle motion and speed between imaging positions.

Registration is validated with float images having known offsets, and visual odometry is compared with ground-truthed ROV surveys. Odometry is performed successfully using data from float surveys. Low image overlap and high bottom roughness decrease the probability of successful matches, but these are overcome by slightly higher capture rates. Further, incorrect matches are easily identified and rejected, with minimal impact on the vehicle velocity estimate.

Image scheduling is simulated using a high framerate dataset and allowing the algorithm to select images taken at times separated approximately by its desired image period. Computation time is sufficiently short and consistent enough to keep up with image acquisition in real time. Average power and data storage requirements are decreased, allowing for longer and more frequent surveys.

ACKNOWLEDGMENTS

In addition to my major professor Chris Roman and my committee members Peter Swaszek and Jason Dahl, I would like to thank my labmates, especially Gabrielle Inglis and Ian Vaughn for their technical and moral support, along with Elizabeth Clark of NOAA Fisheries, and John King and Jeremy Collie of URI for their interest and participation in the camera float project. This work was partially funded by NOAA grant NA12NMF4550221.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction and Background

Lagrangian floats have become a popular platform for a wide variety of oceanographic studies in blue water (Dickey et al., 2008), and have more recently been adapted for near shore applications (Roman et al., 2011). These free-floating devices need no propulsion other than buoyancy control, which reduces cost and power consumption to allow for longer and more frequent deployments. As the floats drift, their motion can be used as a proxy for water particle motion to provide an estimate of ocean currents at a desired depth (Swallow, 1953).

High quality visual images of the sea floor benefit a number of research areas including fisheries assessments (Collie et al., 2000, Clarke et al., 2009, Howland et al., 2006), benthic habitat and fauna classification (Ferrini et al., 2006, Yoklavich and O'Connell, 2008), and coral habitat studies (Grasmueck et al., 2006, Singh et al., 2004). The imaging platforms commonly used for these surveys have active control of velocity, either with on-board thrusters or by towing from a ship. Traveling at a known velocity makes it possible to take images at a desired spatial sampling rate by controlling the frequency at which images are taken, the capture rate. Taking images too quickly and too close together depletes batteries and data storage space for no benefit, while taking them too slowly can alias the data or otherwise fail to accomplish the scientific goal.

While knowledge of vehicle velocity is critical to maintaining the desired capture rate, this information need not come from the vehicle's propulsion system. In the case of a Lagrangian imaging platform, there is no propulsion system at all as the vehicle is propelled entirely by water currents. Vehicle speed, and navigation information in general, must come from other sensors.

Underwater vehicle navigation is a difficult task. As GPS signals and other radio waves cannot travel through water, other methods such as long or ultra-short baseline acoustic tracking (LBL and USBL), Doppler velocity logs (DVL), and inertial dead reckoning (INS) must be used to provide estimates of position and motion (Kinsey et al., 2006). LBL requires the placement of acoustic beacons surrounding the site to be surveyed which significantly increases ship time. USBL needs a surface ship to constantly track the underwater vehicle, negating much of the benefit of an autonomous vehicle. DVL and INS provide velocity information without the need for ship support, but are subject to accumulation of random errors (drift) and are prohibitively expensive.

The Lagrangian imaging float (Figure 1) has none of these navigation sensors for position or velocity. It is equipped with only a magnetic compass to provide heading, and a depth sensor and sonar altimeter, required for the buoyancy control system (McGilvray, 2010). This buoyancy system can hold the float at a fixed altitude over the ocean bottom while it drifts, including when traveling over variable bathymetry. The float carries a pair of high quality bottom-imaging cameras for performing benthic surveys.

The images of the bottom taken by these cameras can be used for visual odometry, a common method of determining vehicle motion from the way in which subsequent images overlap (Botelho et al., 2010, Pizarro et al., 2003, Maimone et al., 2007). Odometry with a stereo camera pair allows the true scale of the motion and three-dimensional structure to be determined.

For previous deployments of the float, this odometry was performed after a mission, rather than in real time (Roman et al., 2011). The vehicle motion could be determined in post-processing only if the capture rate set prior to deployment was sufficiently high to provide enough image overlap. A high default capture rate

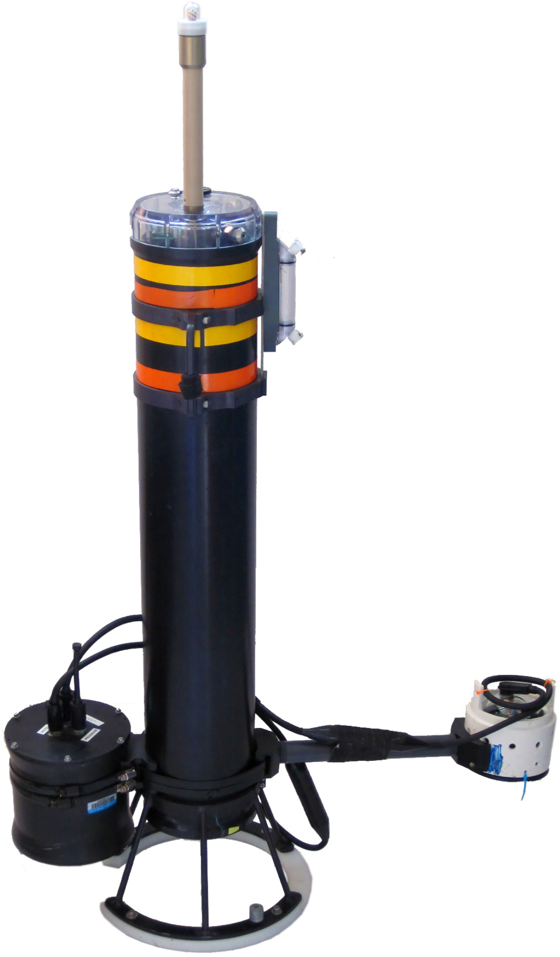


Figure 1. Lagrangian Float with Cameras and Strobe

to create more overlap does not solve the problem, as it reduces dive duration due to battery limitations and generates redundant data. Setting a fixed rate properly requires prior knowledge of the bottom currents, which are not generally available for most areas to high accuracy, and vary with time. Real-time drift speed estimates would provide a basis for adjusting the capture rate as float speed changes throughout the dive. Visual odometry can provide these measurements without any additional sensors.

1.1 Statement of the Problem

The goal of this project was to develop a real-time vision-based frame rate adjustment technique to maintain a desired spatial sampling rate for images taken with a freely drifting bottom-following Lagrangian float (Fig. 1). Maintaining this spatial rate requires estimating the translation speed of the float and adjusting the image timing to achieve a specified percent coverage or percent overlap along the drift track. Visual odometry was used to determine this motion, as the float is equipped with no other sensors for measuring lateral motion.

The algorithm for this purpose processes images as they are taken by the float's cameras to determine vehicle speed, updating the camera capture rate when appropriate to maintain the desired image spacing. The process needed to be fast enough to perform this calculation in real-time on the embedded computer, and be robust to common imaging problems such as turbidity in coastal areas, and out-of-focus images caused by variability in the float's altitude. Although accuracy is desirable, it is less crucial than robustness for real-time estimates.

1.2 Odometry

The float takes images of the ocean bottom at discrete time intervals, so its position is described only when these images are taken. These positions are called poses. Each pose can be represented by coordinates X, Y, and Z, and the angles *roll*, *pitch*, and *heading*, at the time an image was taken. Poses are sequentially numbered and are unique, even if two poses share the same position at different times. The difference between two of these poses is an odometry link, which can be described by six parameters: ΔX, ΔY, ΔZ, $\Delta roll$, $\Delta pitch$, and $\Delta heading$. Due to its design, the float drifts at a nominally constant attitude without significant changes in roll or pitch. This simplifies the odometry links, which can now be described by just four parameters: ΔX, ΔY, ΔZ, and $\Delta heading$.

As ocean bottom currents typically vary slowly (Swallow, 1953), estimating float velocity does not require that each pose in the sequence have an odometry link to those before and after it. Even a few occasional measurements of speed can be used to update the current estimate of velocity, although uncertainty in this estimate increases if fewer odometry links are used, or if a longer period of time has passed since one has been established.

One operating mode of the float, termed burst odometry, involves switching between relatively high overlap images taken at higher framerate to establish odometry links and improve the velocity estimate, and taking images which do not overlap sufficiently for odometry, or do not overlap at all. This mode can be useful for longer surveys where continuous image coverage is not needed so long as image spacing is predictable, and the decreased energy and storage use can extend dive duration.

1.3 Feature-Based Visual Odometry

A common way to find odometry links is to identify features which repeat in images taken from multiple poses (Eustice et al., 2004, Maimone et al., 2007, Inglis, 2013). A good feature should be unique and easily distinguishable from other features but able to be identified when it re-appears in another image taken from a different position, angle, and lighting conditions (Lowe, 2004).

Feature-based odometry can be implemented using a stereo pair of images from each pose. In each image of this pair, potential features are identified and described, then matched between the left and right images using a combination of feature descriptions and the inter-camera geometry (Hartley and Zisserman, 2004). The 3-dimensional position of each matched feature is calculated relative to the vehicle. Features, which now each represent a 3-dimensional point rather than one on the imaging plane, are then matched from one vehicle pose to the next by comparing their descriptions. Once the 3-dimensional feature points have been matched, the relative pose between the imaging locations can be computed (Horn, 1987).

If there is a prior estimate of the odometry link between the poses, such as with a constant velocity assumption, this information can be used to constrain the search for matches and reduce computation time (Pizarro et al., 2003). The constant pitch and roll of the float can also be of benefit in both matching features, and solving for the odometry link.

Choosing an appropriate feature detection and description algorithm is an important trade-off between feature uniqueness, repeatability and robustness to noise, imaging angle, lighting differences, etc., and computation time. Near one end of the spectrum are algorithms which produce a relatively small number of excellent features, such as the scale-invariant feature transform (SIFT) (Lowe, 2004) and speeded-up robust features (SURF) (Bay et al., 2008). These methods will take a long time to find and describe features, but the decreased number of features and more unique descriptions lead to faster matching between poses and fewer incorrect matches which must be rejected later. At the other end of this spectrum are the Harris and Stephens corner detector (Harris and Stephens, 1988) and the FAST algorithm (Rosten and Drummond, 2006, Rosten et al., 2010). Here, almost no time is devoted to finding and describing features, but significantly more must be spent matching them and rejecting incorrect matches. This typically becomes an important and computationally expensive operation (Fischler and Bolles, 1981). There is no best choice for all applications; many options must be evaluated.

There are a number of issues with feature-based odometry underwater. While sharp corners and edges make excellent features for detection and matching algorithms, these are rare in most underwater environments. The noise caused by high turbidity common in coastal areas can interfere with detection and create false positives. Finally, when the float altitude changes by too much, images can be out of focus due to a fixed focal range. This also detrimentally affects some feature detection and description algorithms.

1.4 Featureless Visual Odometry

Another visual odometry technique is to register images to each other using correlation, which does not depend on finding specific repeatable features and instead matches the image as a whole. This can be done for the full six degree of freedom odometry links,. However, unlike the feature-based technique where all degrees of freedom are solved for simultaneously, each additional degree of freedom must be considered individually in terms of how it changes the camera projection. Translation in X and Y simply shift the image by a number of pixels (u, v) determined by the imaging altitude and the camera geometry. Changes in heading cause the image to be rotated, and differences in altitude (Z) re-scale the image. Pitch and roll skew the image and are more difficult to recover (Wolberg and Zokai, 2000). As each degree of freedom is solved for, its effect on the projection can be removed until only translation in u and v remain, which can be found with a simple correlation. This method is most useful in highly constrained situations where the fewer degrees of freedom reduce computational complexity, and reduce the number of points of failure. In the case of the float, ΔX, ΔY, ΔZ, and $\Delta heading$ can be found in only two iterations (Reddy and Chatterji, 1996, Sarvaiya et al., 2012).

As this technique does not depend on repeatedly finding the same features and instead matches the image as a whole, it is extremely robust to noise and turbidity. For the same reason, it is also robust to out-of-focus images, and depending on the correlation function used, can even be robust to the focus being different between two poses.

The vast majority of execution time for featureless odometry is spent calculating correlation functions, in most cases by Fourier transforms. The runtime is independent of image contents or quality, and depends only on resolution which directly impacts the size of the Fourier transforms needed. A fixed execution time is beneficial in an embedded, real-time system as there is no risk of the program falling behind the data acquisition by solving for an especially slow link. While feature-based methods benefit from the highest resolution possible to find high quality features, correlation methods can still perform well at lower resolutions. Images can therefore be down-sampled before processing if necessary to provide an additional speed increase with little drawback.

CHAPTER 2

Methods

For reasons of robustness, speed, and consistency, a featureless algorithm is chosen. This chapter will discuss this algorithm and its implementation on the float. To solve the four degree of freedom motion between pose $n-1$ and pose n, all that is necessary is an image registration from pose n to pose $n-1$, and an altitude at pose $n-1$ to fix absolute scale. The altitude and its uncertainty are found by stereo matching. Then, one camera's image (eg. the left camera) from pose n is registered to the same camera's image from pose $n-1$ using phase correlation (Kuglin and Hines, 1975). Rotation and scale must be recovered and corrected using a log-polar transformation (Reddy and Chatterji, 1996). Finally, altitude, relative rotation and scale, and translation are combined with the known camera geometry to find X, Y, and Z motion, in addition to changes in heading.

2.1 Stereo Altitude

The left and right images from each pose are first corrected for lens distortion and stereo rectified such that a point (u, v) in the left rectified image has the same v coordinate in the right image. The difference in u coordinate, or pixel disparity, can be used to calculate the range from the cameras to these points as

$$range = \frac{T_x f_x}{disparity}, \tag{1}$$

where T_x is the rectified baseline distance between cameras and f_x is the rectified focal length in pixels (Hartley and Zisserman, 2004).

A representative altitude is found for each row of the image pair by correlating the rows from the left and right images, and using the peak correlation as the disparity in Equation 1. Phase correlation is used due to its speed of computation and robustness to noise and illumination differences (Kuglin and Hines, 1975). The phase correlation r_u is calculated as

$$r_u = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}_l\mathcal{F}_r^*}{|\mathcal{F}_l\mathcal{F}_r^*|}\right\}, \tag{2}$$

where the Fourier transform of the left row is $\mathcal{F}_l$ and the complex conjugate of the Fourier transform of the right row is $\mathcal{F}_r^*$. All matrix operations are performed element-wise, and $\mathcal{F}^{-1}$ is the inverse Fourier transform.

The peak location has multiple possible interpretations, as the correlation function is periodic. For example, with 1024 pixel-wide images, a peak location at $v = 20$ could also be interpreted as $v = 1048$ or $v = -1004$. In the rectified case of parallel stereo cameras, negative disparity or disparity larger than the camera's width are not possible, so the smallest positive value is used to calculate the altitude. In practice, a single pixel of error can result in a large error in altitude in cases of high image altitude or a very short camera baseline (Hartley and Zisserman, 2004). The peak correlation location is therefore found to sub-pixel precision by fitting a quadratic (Equation 3) to the correlation function in the region of the peak. In pixel coordinates, $x = -1, 0, 1$, and Δx is the amount by which the location of the peak should be moved. This method of sub-pixel interpolation gave similar results to increasing resolution without the associated decreased in performance, and the quadratic coefficients can be found with a single matrix multiplication.

$$\begin{gathered} \mathit{correlation\,magnitude} = ax^2 + bx + c \\ \Delta x = \frac{-b}{2a} \end{gathered} \tag{3}$$

An independent measurement of altitude is thus made for each row of the image pair. These measurements should roughly agree with each other if the scene is relatively flat, and any outliers greater than 3 standard deviations from the mean can be rejected. Their spread gives an estimate of scene structure or bottom roughness, measured by the variation in altitude. Figure 2 shows the calculated altitude for each row of an image pair (only the left image is shown), along with the mean altitude. The mean altitude is used to determine absolute scale, and its uncertainty is carried through the odometry calculation.

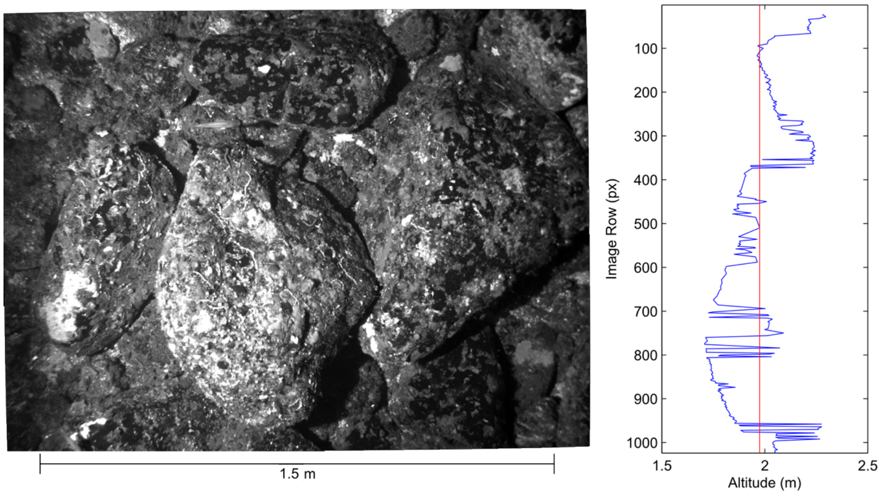


Figure 2. Stereo Altitude Example
Example left image, calculated altitude for each row, and mean altitude. Indicated scale is from stereo triangulation.

2.2 Image Registration

After the altitude is found, only one image of the stereo pair is used for each relative pose measurement. The black and white (left) image is used due to its higher resolution and longer depth of field. The image need only be undistorted, not rectified, but to save computation time the same rectified image from the altitude calculation is used.

2.2.1 Translation

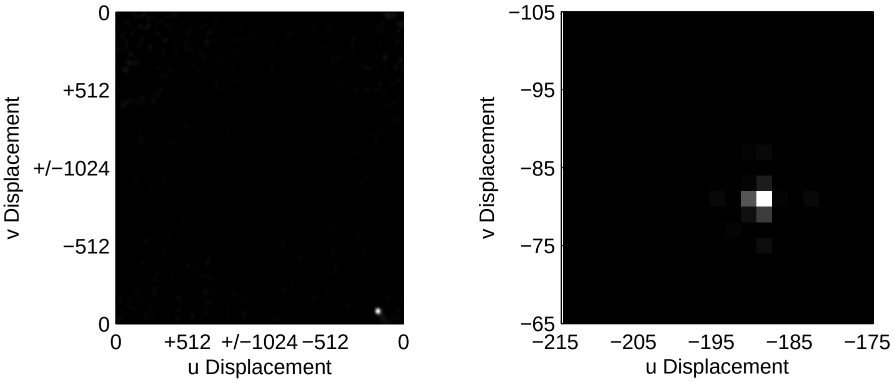
If the images from pose $n-1$ and n differ only in translation, they can be registered via two-dimensional phase correlation similar to how the average altitude is computed. The two-dimensional Fourier transforms from each pose $\mathcal{F}_{n-1}$ and $\mathcal{F}_n$ are used, the correlation function $r_{(u,v)}$ is calculated as

$$r_{(u,v)} = \mathcal{F}^{-1}\left\{\frac{\mathcal{F}_{n-1}\mathcal{F}_n^*}{|\mathcal{F}_{n-1}\mathcal{F}_n^*|}\right\}. \tag{4}$$

The peak location (u, v) is again found to subpixel precision by fitting a quadratic in the region of the peak. The peak location now has four interpretations due to the periodic nature of the correlation function. The ambiguity could be resolved by translating the image with each of the possibilities, then testing alignment a non-periodic method such as sum of squared differences. Testing all four cases is computationally expensive, however three of the possibilities represent image overlaps of less than 50%. These are unlikely to be valid matches (Section 3.3), so the vector (u, v) with the smallest norm is chosen as the most likely.

An example translation correlation function using the images from Section 3.1.1 is shown in Figure 3, with the axes indicating the most probable interpretation of the peak correlation location. The peak correlation value indicates how well the images are correlated. If they are correlated poorly, the match is likely to be incorrect and can be rejected with a threshold that is further discussed in Sections 3.1.2 and 3.3.

The translation correlation function from Section 3.1.1 is shown in Figure 3, with the axes indicating the most probable interpretation of the peak correlation location, a displacement of -189 pixels in u and -92 in v. The peak correlation *value* indicates how well the images are correlated: if they are correlated poorly, the match is likely to be incorrect and can be rejected. The rejection threshold is set in Sections 3.1.2 and 3.3.



(a)

(b)

Figure 3. Translation Correlation Function
Translation Phase Correlation Function (left) and detail (right). Axes indicate the peak is at $u = -189$ and $v = -92$.

2.2.2 Rotation and Scale

If the images from poses $n-1$ and n differ only in rotation and scale, they can be registered by remapping them to a log-polar representation before performing the 2D phase correlation as in Equation 4 (Reddy and Chatterji, 1996). Equation 5 performs the remapping from the original image coordinates (u_x, v_y) to the log-polar image coordinates (u_ρ, v_θ). Here c_x and c_y are the location of the optical center and *width* and *height* are the size of the image in pixels.

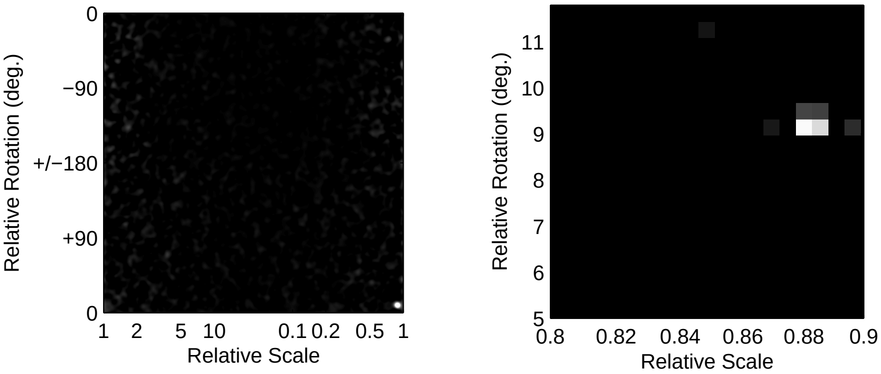
$$\begin{aligned} u_\rho &= \sqrt{(u_x - c_x)^2 + (v_y - c_y)^2} \times \frac{width}{\log\left(\frac{width}{2}\right)} \\ v_\theta &= \tan^{-1}\left(\frac{v_y - c_y}{u_x - c_x}\right) \times \frac{height}{360} \end{aligned} \tag{5}$$

As in the translation case, the peak location of the periodic correlation function has multiple possible interpretations. To find relative rotation, two of these possibilities are evaluated in Equation 6, and the value closest to zero is chosen. This gives the full rotation range of −180° to +180°. For relative scale, two possibilities are evaluated in Equation 7, and the value with its *logarithm* closest to zero is chosen. The maximum scale range is $\sqrt{2/width}$ to $\sqrt{width/2}$, although matches close to these extremes are likely to fail.

$$rotation = \begin{cases} \frac{360}{height} v \\ \frac{360}{height} \left(v - height\right) \end{cases} \tag{6}$$

$$scale = \begin{cases} exp\left(\frac{u}{width} \log\left(\frac{width}{2}\right)\right) \\ exp\left(-\frac{width-u}{width} \log\left(\frac{width}{2}\right)\right) \end{cases} \tag{7}$$

An example rotation and scale correlation function using images from Section 3.1.1 is shown in Figure 4. Unlike the translation case, the rotation and scale peak correlation value is not useful in rejecting poor matches.



(a) (b)

Figure 4. Rotation and Scale Correlation Function
Rotation and Scale Phase Correlation Function (left) and detail (right). The final rotation is 9.37° and the relative scale is 0.8763.

2.2.3 Translation with Rotation and Scale

Real-world images vary simultaneously in translation, rotation, and scale. Rotation and scale must be separated and corrected before translation can be found. This can be done using the fact that the magnitude of a Fourier transform is translation invariant, while preserving information about rotation and scale. For two images which differ in translation, rotation and scale, the magnitudes of their

Fourier transforms differ only in rotation and scale. Rotation and scale can then be recovered using the log-polar phase correlation technique, as in Section 2.2.2. The second phase correlation normalizes any differences in image lighting.

For real valued inputs (eg. images) rather than complex or imaginary, the transform has bilateral rotational symmetry. Only half of the transform contains unique information, so angles can be limited to 180 degrees rather than 360 in conversion to log-polar representation (Equation 5), and the angle resulting from the peak correlation location is also calculated using 180 degrees (Equation 6). The scale inversion has no effect on the log-polar remapping, but the scale found using Equation 7 must be inverted, ie. 1/scale.

After the rotation and scale are computed, the original image from pose n is rotated and scaled to correct for these differences, and now varies from pose $n-1$ only by (u, v) translation. This translation is found last as in Section 2.2.1.

2.2.4 Windowing and Filtering

The separation of translation, rotation, and scale is dependent on the ability to perform a phase correlation between the log-polar Fourier transform magnitudes of two images. These transforms must be free of any structure not caused by the image contents, such as persistent lighting artifacts, or this structure may correlate more strongly than the signal. Figure 5 shows this for three very different underwater test images, from Cordell Bank, California (left column), Eregli, Turkey (middle column), and Andvord Bay, Antarctica (right column).

The Fourier transforms are shown in the middle row. The strongest feature in all three is the white crossing pattern through the center, caused by the fact that the images are not periodic as assumed by the Fourier transform. This can be corrected by applying a window such as Hanning to each image before taking the Fourier transform. This window is shown in Figure 6(a), where black and white represent values of 0 and 1, respectively.

The other issue with these transforms is the greater intensity near the center, representing low frequencies. The lowest frequencies are caused by lighting artifacts, the effect of which changes when imaging from a different position. This information is wrong and should be eliminated. In addition, most images of the ocean bottom have a disproportionate amount of low frequency content. Registering images based only on the low frequencies is less precise and more likely to fail than registering after weighting all frequencies equally. To that end, a high-pass filter is applied to each Fourier transform magnitude. The cosine-squared filter from Reddy and Chatterji (1996) is used, with the modification that after the filter is created, all values less than one are replaced by their square root. This modification further smooths the transition between low and high frequencies, and allows scale to be found more reliably in cases of low overlap. The final filter is shown in Figure 6(b).

When the images have been windowed and their Fourier transforms filtered properly, the Fourier transforms represent only the contents of the image, and are clearly differentiable from each other as shown in the bottom row of Figure 5.

2.3 Odometry from Image Registration

Vehicle translation can be found from the image registration in pixel coordinates using the camera matrix K, using Equation 8. K^{-1} is the inverse of the camera matrix and (u, v) is the translation in pixel coordinates relative to the image center (c_x, c_y). u and v can be positive or negative. The result is X, Y, the distance the vehicle has moved in meters.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = altitude \times K^{-1} \begin{bmatrix} u + c_x \\ v + c_y \\ 1 \end{bmatrix} \tag{8}$$

The heading at pose n is equal to the heading at pose $n - 1$ plus Δheading,

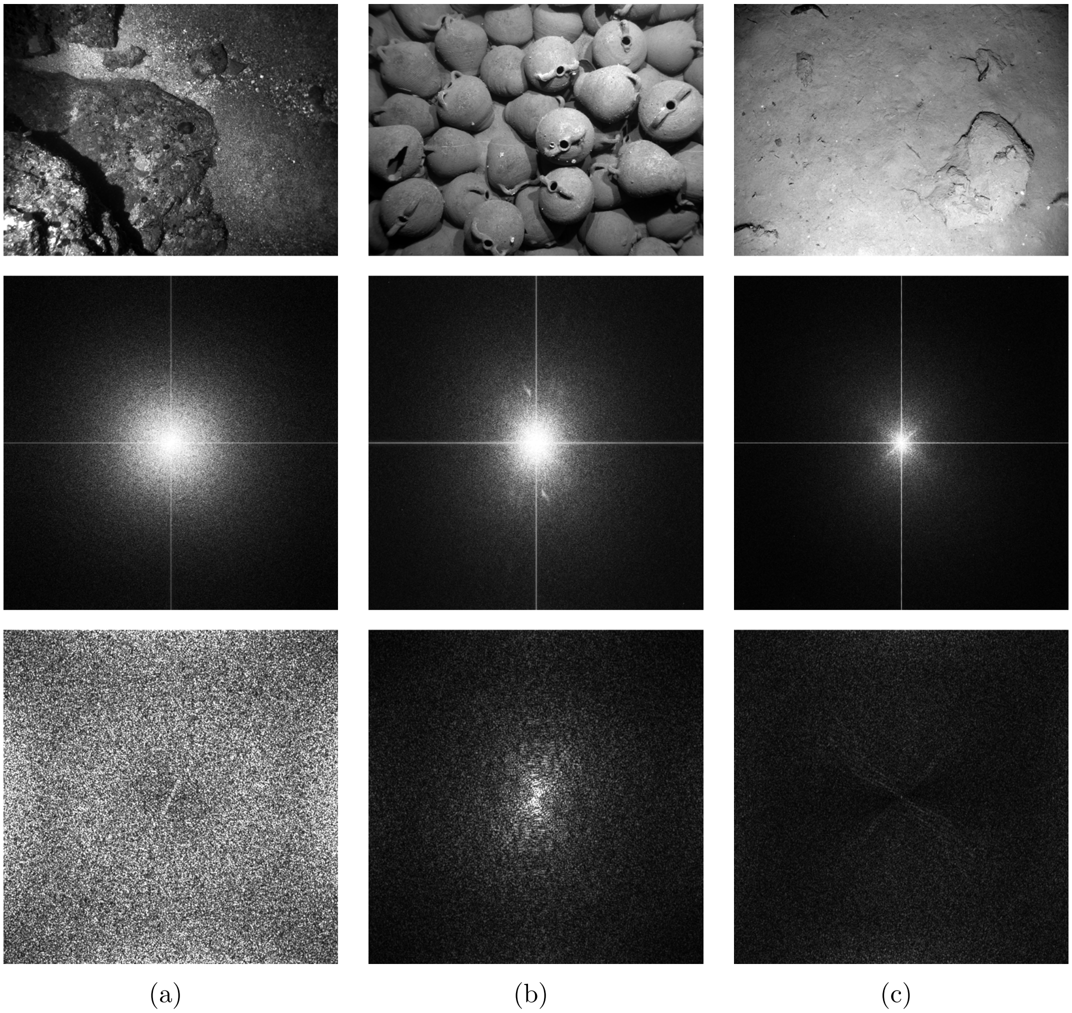
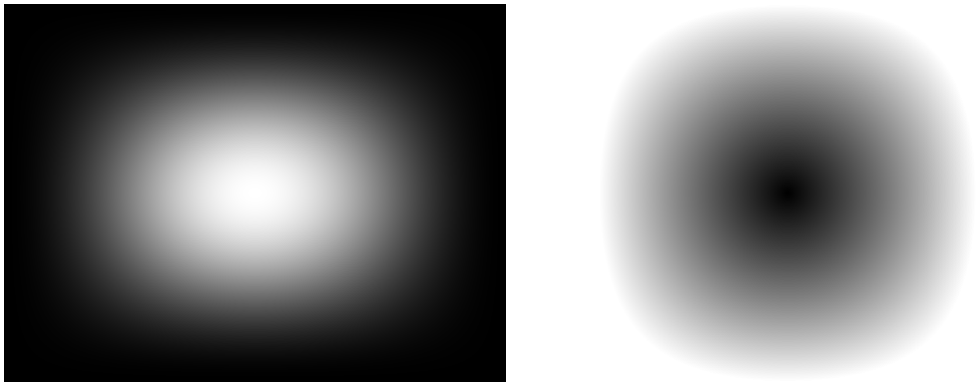


Figure 5. Example Images and FFTs
Example images (top row), FFTs with improper window/filter (middle row), and FFTs with proper window/filter (bottom row) from Cordell Bank California (a), a shipwreck near Eregli Turkey (b), and Andvord Bay, Antarctica (c)

and the altitude at pose n is equal to the altitude at pose $n-1$ times the relative scale. This calculated altitude should agree with the altitude from stereo at pose n.

2.4 Implementation and Performance Improvements

The methods described in Sections 2.2 and 2.3 find the relative motion between two vehicle poses. There are several ways in which its performance can be improved



(a) (b)

Figure 6. Window and High-Pass Filter
Window (left) applied to images before FFT to remove aperiodic artifacts.
High-pass filter (left) applied to FFTs to de-emphasize low frequency

for implementation in an embedded system with limited memory and processing power.

2.4.1 Sequential Caching

For bottom surveys, computing the position between two poses is not done in isolation. Pose $n-1$ must be linked to pose n, pose n to pose $n+1$, and so on. The Fourier and inverse Fourier transform operations are computationally expensive, so intermediate results are cached whenever possible to minimize re-computation, as well as memory and copying overhead. Figure 7 shows this process for sequential images.

2.4.2 Down-sampling Images

The majority of execution time is spent performing two-dimensional Fourier and inverse Fourier transforms, which take $\mathrm{O}\left(n^2 \log(2n)\right)$ time for an image of width n. Down-sampling the images to reduce resolution prior to performing these transforms greatly reduces computation time, at a loss of precision due to pixel quantization error. In addition, smaller images increase noise in the correlation function, which has an impact on rejection thresholds and can lead to false posi-

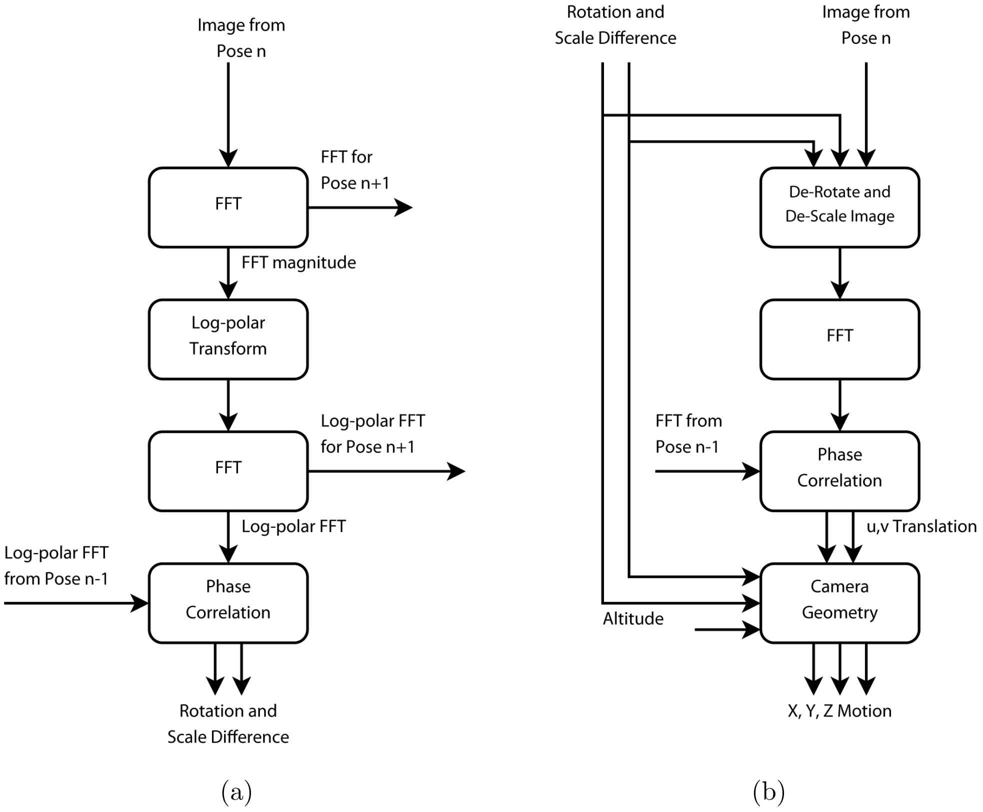


Figure 7. Sequential Registration Algorithm
Algorithm for registering sequential images in scale and rotation (a), followed by translation (b). Inputs from Pose $n-1$ are cached from the last iteration, and outputs to Pose $n+1$ will be cached for the next.

tives.

2.5 Vehicle Position and Speed

Each odometry link is treated as a noisy measurement of X,Y motion with uncertainty related to the variation in altitude found in Section 2.1. These measurements are inputs to a Kalman filter to estimate vehicle drift speed. The state vector x consists of long-track distance and speed, and is initialized to a distance of 0 and a speed of 150 mm/s, a common drift speed for float surveys. The measurement matrix H reflects that only measurements of distance are taken. The initial covariance P indicates low confidence in this speed. For uncertainty in speed to be meaningful, long-track acceleration must be used as the random variable, resulting in the process model described by F_n and G_n. The timestep Δt_n is variable and determined by the image timing, so these matrices must be calculated at each iteration.

$$x = \begin{bmatrix} distance \\ speed \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \end{bmatrix}, P_0 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, F_n = \begin{bmatrix} 1 & \Delta t_n \\ 0 & 1 \end{bmatrix}, G_n = \begin{bmatrix} \frac{\Delta t_n^2}{2} \\ \Delta t_n \end{bmatrix}$$

The measurement noise is scaled by a factor of $R_\sigma = 100$ to give realistic uncertainty in speed. The process noise Q_σ is adjusted to make the filter behave as a low-pass, giving a relatively constant value for vehicle speed which can be used to set the camera framerate for some time horizon. $Q_\sigma = 0.02$ results in a time constant around five minutes. Equation 9 updates the state vector x_n and covariance P_n at each time step n with inputs Δt_n, Δs, and $\sigma_{\Delta s}$, the time step, change in distance, and uncertainty in this distance, respectively.

$$\begin{aligned}
R &= R_\sigma \left(\sigma_{\Delta s}\right)^2 \\
\bar{x} &= F_n x_{n-1} \\
Q &= Q_\sigma G_n {G_n}' \\
\bar{P} &= F_n P_{n-1} {F_n}' + Q \\
z &= x_{n-1}[0] + \Delta s \\
y &= z - H\bar{x} \\
S &= H\bar{P}H' + R \\
K &= \bar{P}H'S^{-1} \\
x_n &= \bar{x} + Ky \\
P_n &= (I - KH)\bar{P}
\end{aligned} \tag{9}$$

The filter's estimate of state at any time can be found by evaluating Equation 9 up to $\bar{x}$. An advantage of this filter is that it behaves well with a highly variable sampling rate, such as when no odometry links are made for as long as several minutes. This is particularly advantageous for the burst odometry mode, where overlapping images are only taken periodically.

CHAPTER 3

Results and Analysis

The presented algorithm has been tested with three types of data. The first are benthic surveys performed by the float. One of these is from near Block Island, Rhode Island and has a very flat bottom but high turbidity. This is a typical environment for the float, which was able to hold a relatively constant altitude of 1 to 2.5 meters. The final dataset is from the Cordell Bank off of California, with very clear water but large boulders on the bottom and complex currents. Altitude for this dive ranges from 1 to 5+ meters.

The second type of data is ancient shipwrecks surveyed by the ROV Hercules near Eregli and Knidos, Turkey. These sites have a flat bottom with some three-dimensional scene structure provided by the ship remains, primarily amphora, surveyed from a constant altitude of 3 meters. The ROV's Doppler velocity log and other sensors are combined to create a navigation solution (Kinsey and Whitcomb, 2003) which is used as an independent ground-truth for visual odometry. The ROV's camera system is similar to the float's and the navigation solution provides the same pose information generated by the odometry program, making this an easy comparison.

Finally, images of the seafloor in Antarctica taken at high framerate with another similar camera system are used to simulate the image scheduling portion of the algorithm.

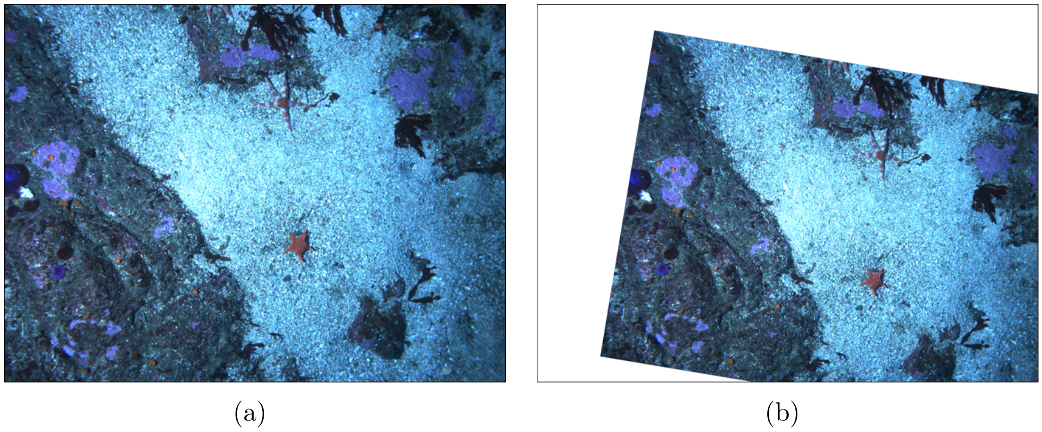
Before images are registered, they are first down-sampled by a factor of two in height and width. This operation decreases computation time significantly while increasing robustness, without any measurable loss in precision. This is discussed further in Section 3.3.

3.1 Validation

There are two ways in which the odometry algorithm is tested. The first confirms that images are registered accurately and consistently. Vehicle speed computed by odometry is then compared to the ROV ground-truth.

3.1.1 Registration

Image registration can be tested by transforming images by randomly generated but known rotation, scale, and (u, v) translation, then attempting to recover these values by registering the transformed image to the original. Figure 8 shows an example original image and the transformed version. Table 1 shows the known and recovered values for this specific test (left), and statistics for 100 random cases (right). In each case, the known values are recovered well, and the corresponding error in the calculated distance traveled would be less than 1%. This shows that images differing by the 4 degrees of freedom expected for the float's motion can be registered accurately using the log-polar transformation and phase correlation.



(a)

(b)

Figure 8. Image Registration Validation Example
Original (a) and transformed (b) images for registration test. The transformed image has been scaled, rotated, and translated by known values. The algorithm should recover these values.

	Example Test Case			100 Test Cases	
	Known Value	Recovered Value	Error	Mean Error	Std. Dev. Err.
Rotation	9.44°	9.37°	0.067°	0.0083°	0.045°
Scale	0.8763	0.8763	-0.000046	0.000082	0.00035
u	189.1 px	189.0 px	0.096 px	-0.013 px	0.14 px
v	82.3 px	81.7 px	0.55 px	0.093 px	0.46 px
% Dist. Err.	0.15%			0.034%	0.36%

Table 1. Image Registration Validation Results
Known and recovered values for image registration tests. A single example from Figure 8 is on the left, and statistics for 100 random cases are on the right. In general, subpixel accuracy is achievable.

3.1.2 Odometry

The second test is whether accurate image registration carries through to accurate odometry links. To test this, speed estimates from visual odometry are compared with the ground-truth for a portion of an ROV dive on a shipwreck near Eregli, Turkey in Figure 9. At most times the speeds agree well, but there are some outliers.

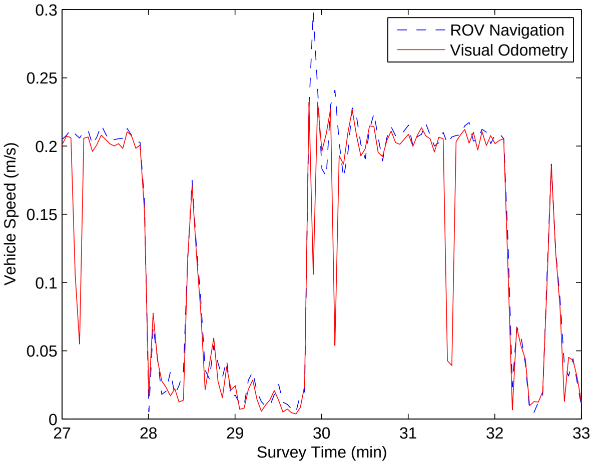


Figure 9. ROV Comparison With Outliers
Speed from ROV Hercules navigation and visual odometry.

Many of these outliers can be rejected using the peak correlation value of the phase correlation function. Figure 10 compares the odometry error in meters for two ROV surveys with the peak correlation value. Most outliers fall below a predictable threshold, while nearly all valid links are above it. This threshold of 0.015 can therefore be used to reject poor odometry links.

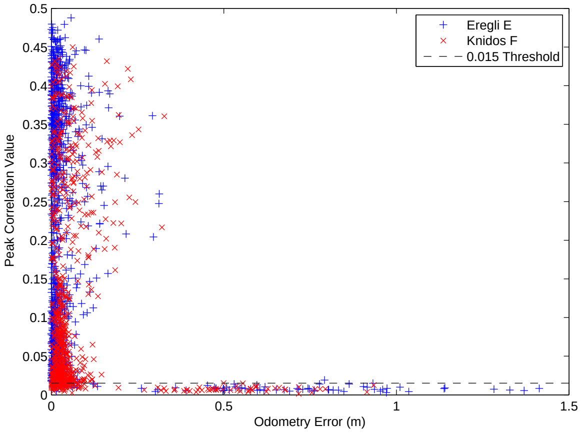


Figure 10. Peak Corr. Value vs. Odometry Error
Odometry error and peak correlation value for two ROV surveys. This relationship suggests a simple threshold can be used to eliminate many of the poor links.

When this peak correlation value threshold is applied to the Eregli ROV survey, all extreme outliers are rejected (Figure 11). The agreement between the vehicle speed found by visual odometry and the ROV's ground truth demonstrates that the method can accurately find vehicle speed for surveys similar to these.

3.2 Float Odometry

The data in this section are from a float dive on the Cordell Bank off of Northern California. Figure 12(a) shows the float speed across the bottom for the full survey of 1260 image pairs. Initially, this appears to be a noisy measurement of a slowly-varying or constant drift speed, but this is not the case. Figure 12(b) shows a short section of this survey, with the 95% confidence interval indicated. There is a clear oscillation in drift speed. This oscillation is confirmed by assembling a photo mosaic from the images in this section of the survey (Figure 13). The float position in each image is indicated by a red dot, with the distance between them

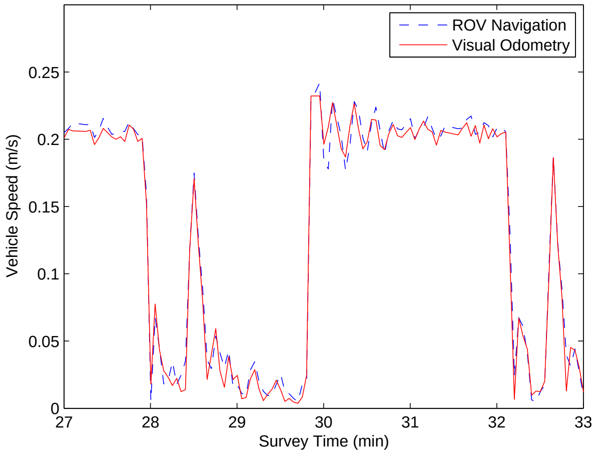
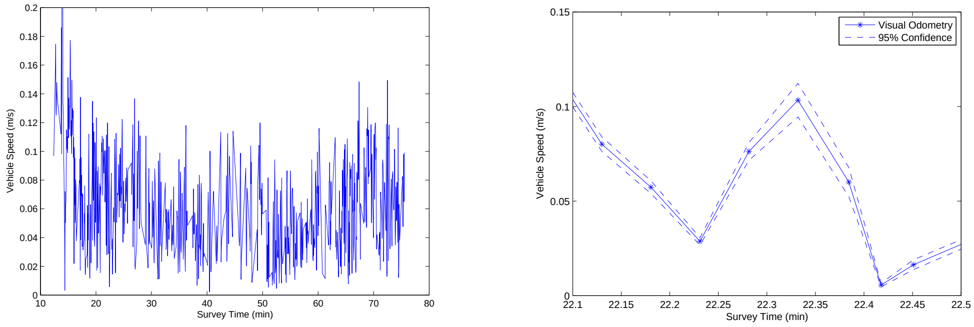


Figure 11. ROV Comparison Without Outliers
Speed from ROV Hercules navigation and visual odometry, after rejecting bad odometry links based on peak correlation value.

shown in white. The images were taken at a constant time interval of 3 seconds. The accurate alignment of images in the mosaic indicates that the registration portion of the program performed well, and the previous ROV comparison lends confidence that the resulting links are valid as well.



(a)

(b)

Figure 12. Float Odometry on Cordell Bank, CA
Float speed from visual odometry for a full dive on Cordell Bank, CA (a), and a short portion of this dive (b).

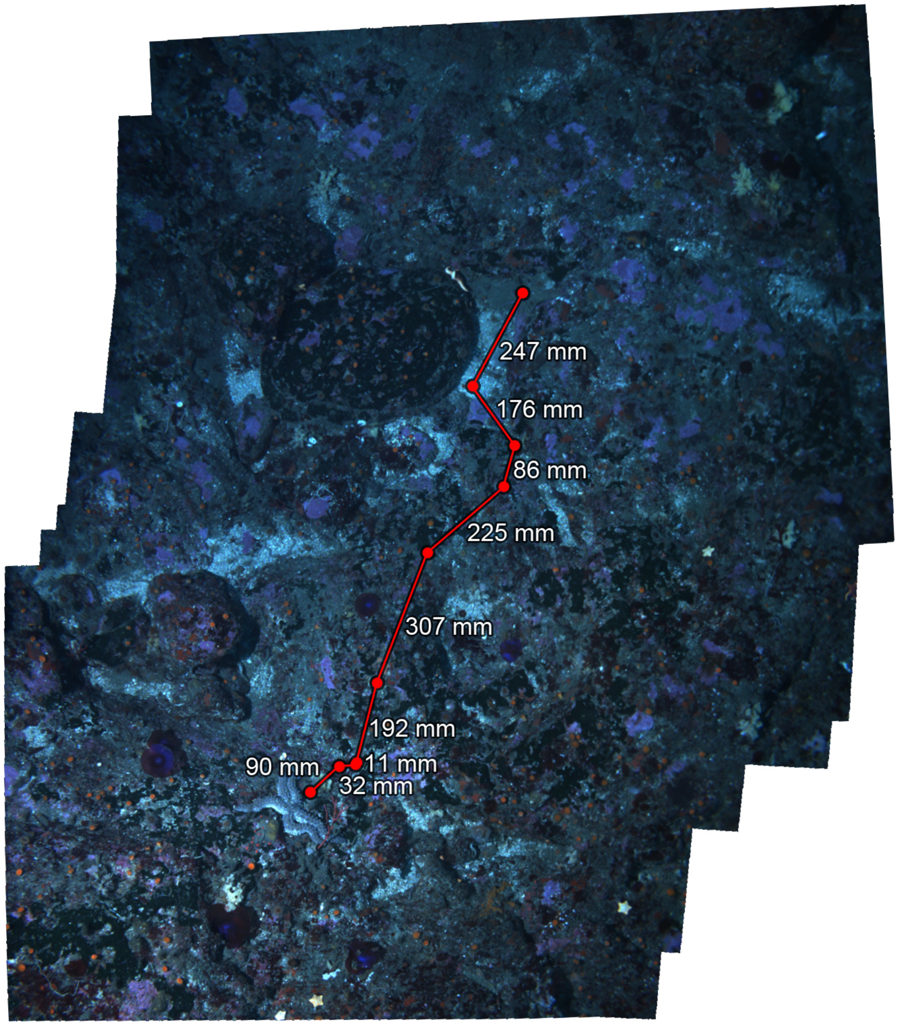


Figure 13. Photo Mosaic, Cordell Bank, CA
Photo mosaic from Cordell Bank, CA, assembled with the images from Figure 12(b).

3.3 Image Overlap, Bottom Roughness, and Down-sampling

Correlation-based registration methods are known to require relatively high overlap between images. Overlap in this case is defined as the percentage of pixels from the first image which also appear in the second. Figure 14 shows how the peak correlation value varies with decreasing overlap for two ROV surveys and two float surveys. For all but the Cordell Bank float survey, peak correlation value has fallen to approximately the minimum threshold around 50% overlap. For the Cordell Bank survey, the threshold is reached at approximately 60% overlap. This result demonstrates that the simplifying assumption made in Section 2.2.1 of choosing the translation vector (u, v) with the minimum norm is valid. A pair of images for which any other (u, v) vector correctly represents their overlap would have a peak correlation value indistinguishable from noise.

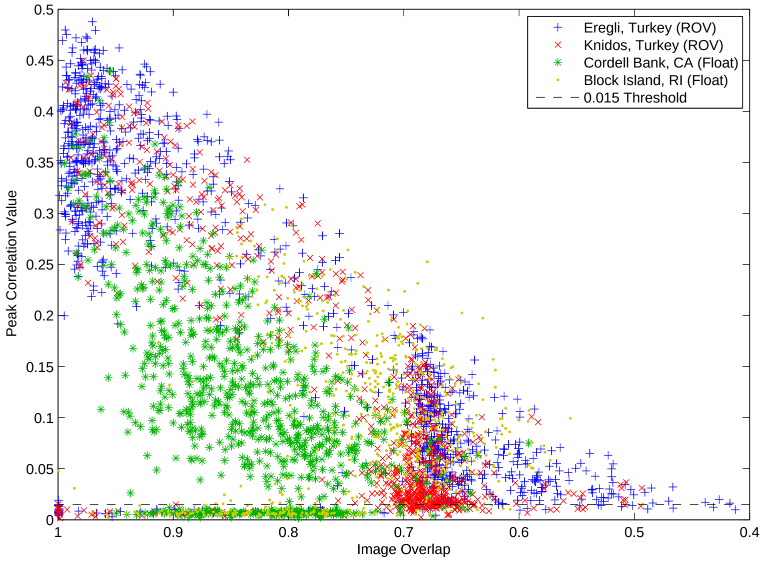


Figure 14. Peak Corr. Value vs. Image Overlap
Image overlap and peak correlation value for two ROV surveys and two float surveys. Peak correlation value reaches the aforementioned minimum threshold at around 50% overlap.

The primary difference between the Cordell Bank dataset, which shows the quickest delay with overlap, and the other three is the extreme bottom roughness in this area. Figures 8(a) and 13 show boulders approximately 0.5 to 1 meters in diameter. This violates the flat bottom assumption. Figure 15 shows the bottom roughness as calculated in Section 2.1 compared with peak correlation value. While peak correlation value is affected by many factors, a rougher bottom as seen in the Cordell Bank dataset most often results in lower peak correlation values. The algorithm still functions on this rough terrain, although the image overlap requirement to exceed the peak correlation value threshold is increased by approximately 10%.

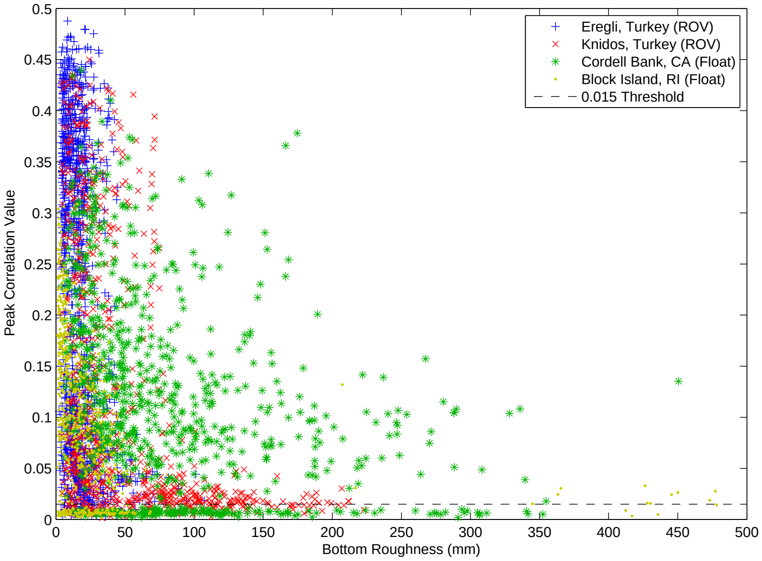


Figure 15. Peak Corr. Value vs. Bottom Roughness
Bottom roughness and peak correlation value for two ROV surveys and two float surveys. In general, peak correlation value is lower over a rough bottom.

Before images are registered, they can be down-sampled to reduce resolution and therefore computation time, at the cost of increased uncertainty due to larger pixels, and greater difficulty in localizing correlation peaks. Figure 16 shows speed calculated for the Eregli ROV survey with full resolution images, 1/2 resolution, 1/4 resolution, and 1/8 resolution. Outliers have been rejected based on a different minimum peak correlation threshold for each resolution. All agree well with the ROV's ground-truth, and the predicted loss in precision at lower resolutions is not readily apparent.

Figure 17 shows that while down-sampling does not greatly increase odometry error, the minimum peak correlation threshold used to reject outliers must be adjusted. A higher threshold means that more valid links will be rejected because

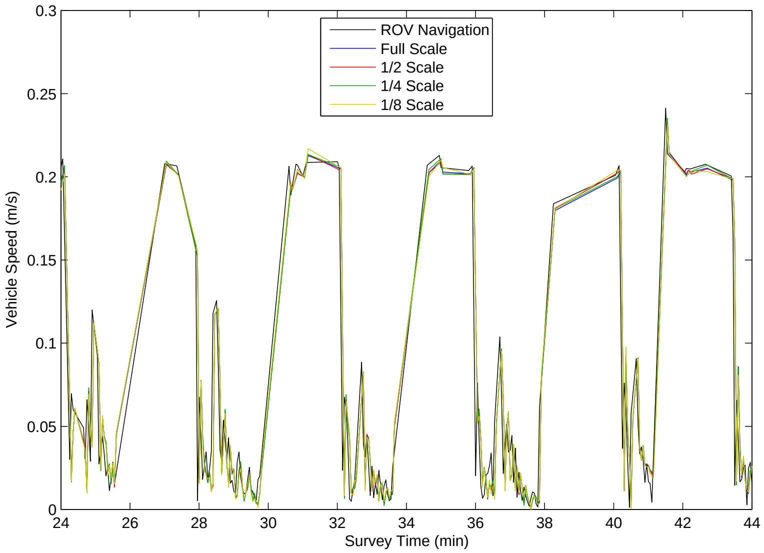


Figure 16. Reduced Resolution Odometry Comparison
Speed from full and reduced resolution images compared with ROV navigation for Eregli survey.

they fall under the threshold. As with the rough terrain, higher image overlap is required to remain above the threshold. Surprisingly, the minimum threshold value is for one-half resolution images rather than full. One-half resolution images are therefore used by default, as they provide a more reliable result with decreased computation time.

3.4 Simulation

A dataset taken at 10 Hz was available from a similar set of cameras, taken from a towed body 2-3 meters off the bottom of Anvord Bay, Antarctica. This high framerate allows the real-time odometry algorithm to be simulated in post processing. After processing an image pair, the algorithm will determine the period it would like to wait before the next image (eg. 5.32 seconds), and the image closest to this time (5.3 seconds) will be used. This allows the image scheduling portion of the algorithm to be evaluated using existing data sets.

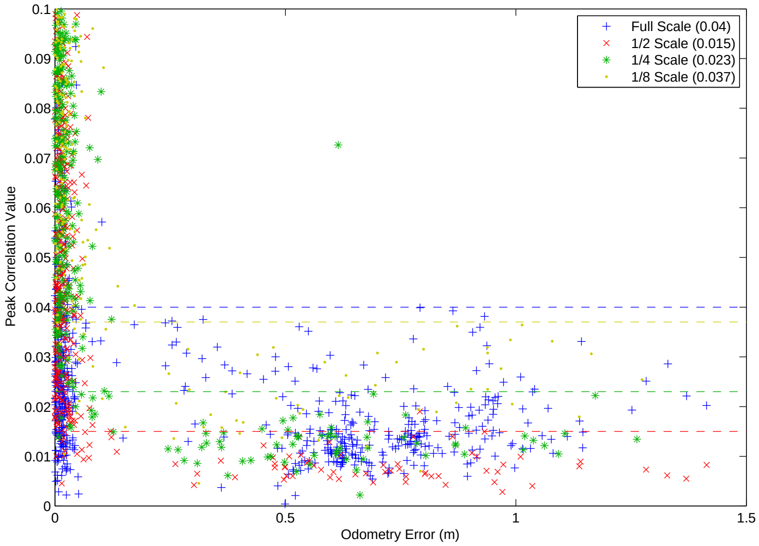


Figure 17. Reduced Resolution Error vs. Peak Corr. Value
Odometry error for full and reduced resolution images compared with peak correlation value. Minimum thresholds for each resolution are indicated.

The vehicle dynamics for the towed system are very different from the float and reflect abrupt ship movements, so the filtering of speed measurements is not evaluated, only the image scheduler. The process noise Q_σ is increased greatly to remove the low-pass characteristic of the filter and to minimize its latency.

Figure 18 shows that the real-time odometry algorithm and image scheduler performed better than a constant period appropriate for the nominal speed at maintaining the desired 0.4 m spacing between image centers. The two spikes around 21 minutes are caused by an abrupt change from one relatively constant speed to another, a behavior which has not been observed in the float. Figure 19 compares the period calculated by the image scheduler using real-time odometry with the ideal period given perfect and instantaneous knowledge of the vehicle speed. The two periods agree well, differing only by a latency of 1-2 image periods. Where the vehicle speed is relatively constant, from 24 to 29 minutes, the scheduler performed extremely well.

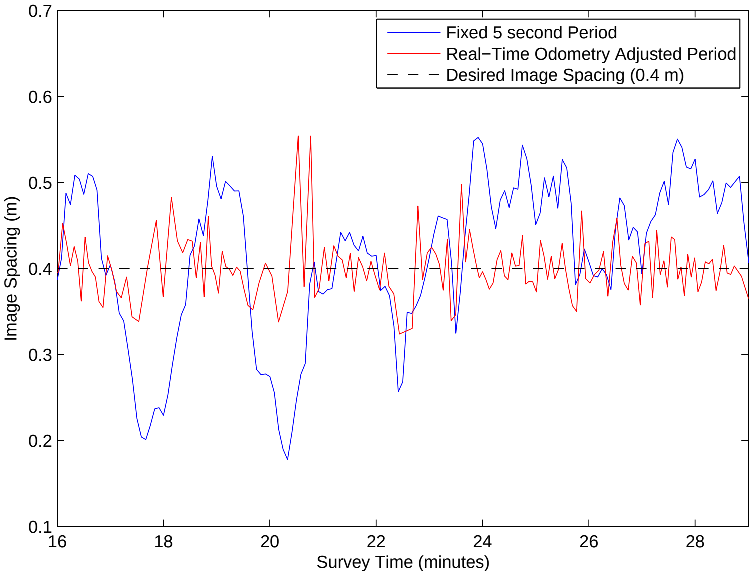


Figure 18. Simulated Image Spacing
Image spacing resulting from images taken at times determined by the real-time image scheduler (red) compared with spacing resulting from a constant image period set by an a priori speed estimate of 0.08 m/s.

3.5 Float Considerations

The camera system's embedded computer has limited processing power, with a single core 32-bit processor running at 1.6 GHz. The current algorithm takes an average of 1.47 seconds to process each image pair on this computer, with a standard deviation of 0.11 seconds, and a worst case out of 1000 image pairs of 1.90 seconds. Since the strobe is unable to maintain a period of less than approximately 2.5 seconds, the algorithm is sufficiently fast for the current float camera system.

The maximum dive duration of the float, currently limited to about 10 hours by battery life, can benefit from real-time image scheduling. The total power usage of the float is around 30 watts on a typical dive, which is not measurably increased by the computational load of running these calculations regardless of image resolution used. The strobe consumes roughly 15 Joules of energy each time it fires, which for an imaging rate of 2.5 to 5 seconds can be 10-20% of the total

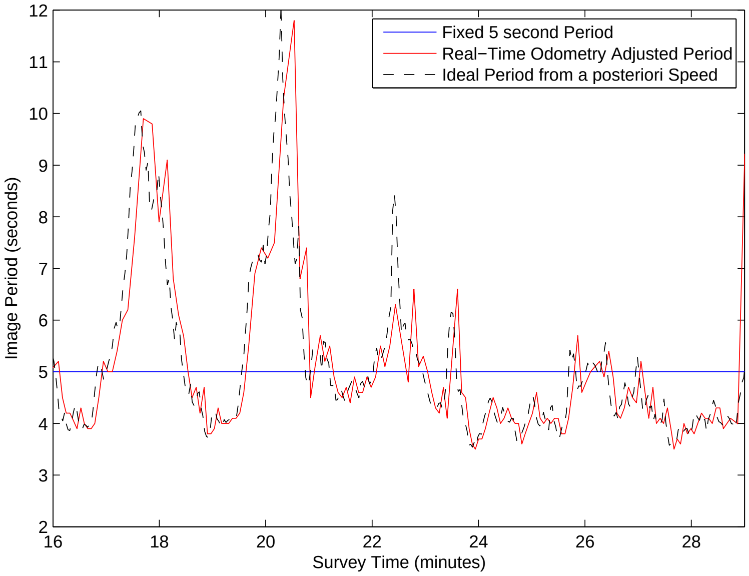


Figure 19. Simulated Image Period
Image period from the real-time image scheduler (red) compared with ideal spacing determined a posteriori using instantaneous knowledge of vehicle speed

system power. This percentage is increased at shallow depths, where the float's bouyancy control system is more efficient. The Cordell Bank dive in Section 3.2 was near the bottom for 63 minutes and took 1260 pairs of images, while traveling 153 meters. If images were taken every 0.5 meters, approximately every 12 seconds on average, total float power consumption would have been reduced by 13%, and image storage requirements would have been reduced by 76%. This would have resulted in sufficient sampling, without oversampling. As both energy and image storage can be limiting factors on dive duration, these reductions have a direct benefit to maximum dive duration, and for shorter dives, how quickly the float can be turned around to start the next survey.

CHAPTER 4

Conclusions and Future Work

This thesis presents a method for finding Lagrangian float drift speed using visual odometry on pictures of the ocean bottom. This is motivated by the need for consistent spatial sampling in benthic surveys, which requires knowledge of vehicle velocity. The float does not control its velocity and has no sensors to directly measure it, making visual odometry a good source of this information.

Phase correlation, a featureless image registration technique, is chosen due to its robustness to noisy and out-of-focus images, along with its predictable and short computation time, a significant benefit for embedded systems. The float's altitude and heading vary throughout the survey, affecting image scale and rotation, as does the direction of travel. The registration algorithm therefore measures translation in two dimensions, along with differences in scale and heading. These values are combined with the known camera geometry to find vehicle motion in X, Y, and Z.

The registration and odometry algorithms are validated using float images with known differences, and ROV surveys with ground-truth navigation data. In both of these tests the method performs well. With data from float surveys, the algorithm is also successful at finding odometry links. Image overlap and bottom roughness have significant impacts on the likelihood of successful matching. A small increase in capture rate can overcome both of these issues. Incorrectly linked poses are easily identified by the peak correlation value, and discarded with little impact on the vehicle's velocity estimate. A simulation of the image scheduler using high framerate images from a similar camera system mounted on a towed body indicates that using real-time odometry to adjust the capture rate can give more consistent image spacing than a fixed framerate.

Imaging at a consistent desired spacing can reduce average power and data storage requirements, while ensuring that data are not aliased. Computation time per image pair on the float's embedded computer is consistently less than the minimum strobe period, allowing the algorithm to run in real-time and process images as they are taken.

4.1 Future work

The camera-float system would benefit from further integration between the cameras and the float control system. The float's magnetic compass and sonar altimeter could be incorporated into the odometry navigation solution, and in the case of the compass, give absolute heading. Combining speed, direction, and a known starting location would allow estimates of absolute position (latitude and longitude, or Northing and Easting). However, with nothing to bound the accumulation of random navigation error, these position estimates could drift considerably.

The sonar altimeter can also be used to only acquire images when the float is within visual range of the bottom, roughly 5 to 10 meters depending on location. If images of the water column are not required, this saves further energy and data storage.

A field test of the float running the real-time visual odometry algorithm is scheduled for September 2013. The float should be able to maintain nominally constant image spacing across the bottom.

Although the odometry algorithm performed well on ground-truthed ROV surveys and towed transects using similar camera systems, the float's motion differs significantly from both the ROV's and the towed body's. To date, there are no float datasets with high quality, independent navigation data. A dataset using a USBL or preferably LBL system could confirm the float's oscillatory motion.

LIST OF REFERENCES

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359.

Botelho, S., Oliveira, G., Drews, P., Figueriredo, M., and Haffele, C. (2010). *Visual Odometry and Mapping for Underwater Autonomous Vehicles*, volume Robot Localization and Map Building. Intech.

Clarke, M. E., Tolimieri, N., and Singh, H. (2009). Using the seabed auv to assess populations of groundfish in untrawlable areas. In Beamish, R. J. and Rothschild, B. J., editors, *The Future of Fisheries Science in North America*, volume 31 of *Fish & Fisheries Series*, pages 357–372. Springer Netherlands.

Collie, J. S., Escanero, G. A., and Valentine, P. C. (2000). Photographic evaluation of the impacts of bottom fishing on benthic epifauna. *ICES Journal of Marine Science*, 57:987–2000.

Dickey, T. D., Itsweire, E. C., Moline, M., and Perry, M. J. (2008). Introducton to the limnology and oceanography special issue on autonomous and lagrangian platforms and sensors (alps). *Limnology and Oceanography*, 53(5-2):2057–2061.

Eustice, R., Pizarro, O., and Singh, H. (2004). Visually augmented navigation in an unstructured environment using a delayed state history. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 25–32, New Orleans, USA.

Ferrini, V., Singh, H., Clarke, M., Wakefield, W., and York, K. (2006). Computer-assisted analysis of near-bottom photos for benthic habitat studies. In *OCEANS 2006*, pages 1–4.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):391–395.

Grasmueck, M., Eberli, G. P., Viggiano, D. A., Correa, T., Rathwell, G., and Luo, J. (2006). Autonomous underwater vehicle (auv) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the straits of florida. *Geophysical Research Letters*, 33:616.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642.

Howland, J., Gallager, S., Singh, H., Abrams, L., Griner, C., Taylor, R., and Vine, N. (2006). Development of a towed survey system for deployment by the fishing industry. In *OCEANS 2006*, pages 1–5.

Inglis, G. (2013). *Hybrid Optical Acoustic Seafloor Mapping*. PhD thesis, University of Rhode Island.

Kinsey, J., Eustice, R., and Whitcomb, L. (2006). A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft, Lisbon, Portugal*, pages 1–12.

Kinsey, J. C. and Whitcomb, L. L. (2003). Preliminary field experience with the dvlnav integrated navigation system for manned and unmanned submersibles. Technical report, In: Proceedings of the 1st IFAC Workshop on Guidance and Control of Underwater Vehicles, GCUV 03.

Kuglin, C. D. and Hines, D. C. (1975). The phase correlation image alignment method. In *IEEE Int. Conf. Cybernet. Society*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics, Special Issue on Space Robotics*, 24:2007.

McGilvray, B. (2010). Control system performance and efficiency for a mid-depth lagrangian profiling float. Master's thesis, University of Rhode Island.

Pizarro, O., Eustice, R., and Singh, H. (2003). Relative pose estimation for instrumented, calibrated imaging platforms. In *Proceedings of Digital Image Computing Techniques and Applications*, pages 601–612, Sydney, Australia.

Reddy, B. S. and Chatterji, B. N. (1996). An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5:1266–1271.

Roman, C., Inglis, G., and McGilvray, B. (2011). Lagrangian floats as sea floor imaging platforms. *Continental Shelf Research*, 31(15):1592–1598.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443.

Rosten, E., Porter, R., and Drummond, T. (2010). Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119.

Sarvaiya, J., Patnaik, S., and Kothari, K. (2012). Image registration using log polar transform and phase correlation to recover higher scale. *Journal of Pattern Recognition Research*, 7:90–105.

Singh, H., Armstrong, R., Gilbes, F., Eustice, R., Roman, C., Pizarro, O., and Torres, J. (2004). Imaging coral i: Imaging coral habitats with the seabed auv. *Subsurface Sensins Technologies and Applications*, 5(1):25–42.

Swallow, J. (1953). A neutral-bouyancy float for measuring deep currents. *Deep Sea Research*, 3:74–81.

Wolberg, G. and Zokai, S. (2000). Robust image registration using log-polar transform. In *In Proc. IEEE Int. Conf. image processing*, pages 493–496.

Yoklavich, M. M. and O'Connell, V. (2008). *Twenty Years of Research on Demersal Communities Using the Delta Submersible in the Northeast Pacific*. Alaska Sea Grant.

BIBLIOGRAPHY

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359.

Botelho, S., Oliveira, G., Drews, P., Figueriredo, M., and Haffele, C. (2010). *Visual Odometry and Mapping for Underwater Autonomous Vehicles*, volume Robot Localization and Map Building. Intech.

Clarke, M. E., Tolimieri, N., and Singh, H. (2009). Using the seabed auv to assess populations of groundfish in untrawlable areas. In Beamish, R. J. and Rothschild, B. J., editors, *The Future of Fisheries Science in North America*, volume 31 of *Fish & Fisheries Series*, pages 357–372. Springer Netherlands.

Collie, J. S., Escanero, G. A., and Valentine, P. C. (2000). Photographic evaluation of the impacts of bottom fishing on benthic epifauna. *ICES Journal of Marine Science*, 57:987–2000.

Dickey, T. D., Itsweire, E. C., Moline, M., and Perry, M. J. (2008). Introducton to the limnology and oceanography special issue on autonomous and lagrangian platforms and sensors (alps). *Limnology and Oceanography*, 53(5-2):2057–2061.

Eustice, R., Pizarro, O., and Singh, H. (2004). Visually augmented navigation in an unstructured environment using a delayed state history. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 25–32, New Orleans, USA.

Ferrini, V., Singh, H., Clarke, M., Wakefield, W., and York, K. (2006). Computer-assisted analysis of near-bottom photos for benthic habitat studies. In *OCEANS 2006*, pages 1–4.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):391–395.

Grasmueck, M., Eberli, G. P., Viggiano, D. A., Correa, T., Rathwell, G., and Luo, J. (2006). Autonomous underwater vehicle (auv) mapping reveals coral mound distribution, morphology, and oceanography in deep water of the straits of florida. *Geophysical Research Letters*, 33:616.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642.

Howland, J., Gallager, S., Singh, H., Abrams, L., Griner, C., Taylor, R., and Vine, N. (2006). Development of a towed survey system for deployment by the fishing industry. In *OCEANS 2006*, pages 1–5.

Inglis, G. (2013). *Hybrid Optical Acoustic Seafloor Mapping*. PhD thesis, University of Rhode Island.

Kinsey, J., Eustice, R., and Whitcomb, L. (2006). A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conference of Manoeuvering and Control of Marine Craft, Lisbon, Portugal*, pages 1–12.

Kinsey, J. C. and Whitcomb, L. L. (2003). Preliminary field experience with the dvlnav integrated navigation system for manned and unmanned submersibles. Technical report, In: Proceedings of the 1st IFAC Workshop on Guidance and Control of Underwater Vehicles, GCUV 03.

Kuglin, C. D. and Hines, D. C. (1975). The phase correlation image alignment method. In *IEEE Int. Conf. Cybernet. Society*.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *Journal of Field Robotics, Special Issue on Space Robotics*, 24:2007.

McGilvray, B. (2010). Control system performance and efficiency for a mid-depth lagrangian profiling float. Master's thesis, University of Rhode Island.

Pizarro, O., Eustice, R., and Singh, H. (2003). Relative pose estimation for instrumented, calibrated imaging platforms. In *Proceedings of Digital Image Computing Techniques and Applications*, pages 601–612, Sydney, Australia.

Reddy, B. S. and Chatterji, B. N. (1996). An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, 5:1266–1271.

Roman, C., Inglis, G., and McGilvray, B. (2011). Lagrangian floats as sea floor imaging platforms. *Continental Shelf Research*, 31(15):1592–1598.

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443.

Rosten, E., Porter, R., and Drummond, T. (2010). Faster and better: A machine learning approach to corner detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32:105–119.

Sarvaiya, J., Patnaik, S., and Kothari, K. (2012). Image registration using log polar transform and phase correlation to recover higher scale. *Journal of Pattern Recognition Research*, 7:90–105.

Singh, H., Armstrong, R., Gilbes, F., Eustice, R., Roman, C., Pizarro, O., and Torres, J. (2004). Imaging coral i: Imaging coral habitats with the seabed auv. *Subsurface Sensins Technologies and Applications*, 5(1):25–42.

Swallow, J. (1953). A neutral-bouyancy float for measuring deep currents. *Deep Sea Research*, 3:74–81.

Wolberg, G. and Zokai, S. (2000). Robust image registration using log-polar transform. In *In Proc. IEEE Int. Conf. image processing*, pages 493–496.

Yoklavich, M. M. and O'Connell, V. (2008). *Twenty Years of Research on Demersal Communities Using the Delta Submersible in the Northeast Pacific*. Alaska Sea Grant.