Article:

Henson, Benjamin Thomas and Zakharov, Yuriy orcid.org/0000-0002-2193-4334 (2018) Attitude-trajectory estimation for forward looking multi-beam sonar based on acoustic image registration. IEEE Journal of Oceanic Engineering. pp. 1-29. ISSN 0364-9059

Attitude-trajectory estimation for forward looking multi-beam sonar based on acoustic image registration

Benjamin Henson,
Department of Electronic Engineering
University of York
York, UK. YO10 5DD
bth502@york.ac.uk
Yuriy Zakharov,
Department of Electronic Engineering
University of York
York, UK. YO10 5DD
yury.zakharov@york.ac.uk

Abstract

This work considers the processing of acoustic data from a multi-beam Forward Looking Sonar (FLS) on a moving underwater platform to estimate the platform's attitude and trajectory. We propose an algorithm to produce an estimate of the attitude-trajectory for a FLS based on the *optical flow* between consecutive sonar frames. The attitude-trajectory can be used to locate an underwater platform, such as an Autonomous Underwater Vehicle (AUV), to a degree of accuracy suitable for navigation. It can also be used to build a mosaic of the underwater scene. The estimation is performed in three steps. Firstly, a selection of techniques based on the *optical flow* model are used to estimate a pixel displacement map (DM) between consecutive sonar frames represented in the native polar (range/bearing) format. The second step finds the best match between the estimated DM and DMs for a set of modeled sonar sensor motions. To reduce complexity, it is proposed to describe the DM with a small parameter vector derived from the displacement distribution. Thus, an estimate of the incremental sensor motion between frames is made. Finally, using a weighted regularized spline technique, the incremental inter-frame motions are integrated into an attitude-trajectory for the sonar sensor. To assess the accuracy of the attitude-trajectory estimate, it is used to register FLS frames from a field experiment dataset and build a high-quality mosaic of the underwater scene.

Index Terms

Forward Looking Sonar, registration, mosaic, navigation.

I. INTRODUCTION

The desire to explore and exploit the oceans, combined with a desire to reduce the risk to humans, has led to the use of robotics underwater. A fundamental requirement for a robotic system, whether autonomous or piloted, is the need to navigate. While there are many technologies to assist in navigation for land or air based vehicles, these technologies do not necessarily transfer well to the underwater environment. Global Positioning System (GPS) may be referenced using underwater acoustic beacons [1], however a system may not always be available, or the accuracy good enough, for tasks such as exploration [2] or underwater inspection [3][4]. Although inertial navigation systems are unaffected by being underwater, they are costly and subject to drift [5]. Optical vision systems have been effectively used for navigation on land [6], in air [7] and underwater [8]. Underwater, they have the advantage in providing an absolute position referenced to the seabed. However, underwater optical systems may suffer from poor visibility [9]. A practical alternative is acoustic imaging which being based on acoustic waves is less susceptible to propagation attenuation [3][10][11]. The use of acoustic waves also allows accurate estimation of the range to an object, though the angle estimation is typically more ambiguous than in optical systems.

With the advent of high resolution multi-beam forward looking sonars (FLSs) there has been an interest in using techniques akin to the optical vision systems for navigation underwater [3][10]. Fig. 1 shows an example field of view for a FLS and Fig. 2 shows an example application where the sonar is used to inspect the underside of a ship's hull.

The sonar produces a known bearing (Ψ) and range (r). However, vertically there is a single wide beam, which illuminates a large area but makes ambiguous the vertical angle of the reflection. Fig. 3 illustrates this ambiguity. This may be compared with optical imagery where the vertical and horizontal directions are known but not the range. Another difference is that for the sonar there is a single illumination point, which casts shadows seen in range, whereas most optical scenarios are in scattered light and therefore have uniform illumination.

FLS images have specific problems when trying to apply to them optical registration techniques. Acoustic images usually have a low resolution; for instance, a DIDSON 300 sonar

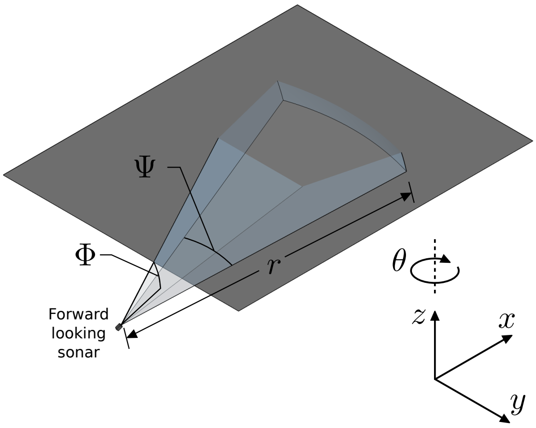


Fig. 1. Position of the FLS field of view relative to the underside of a ship's hull. The volume illuminated by the sonar is shown in light gray. Where it intersects an object, in this case a ship's hull represented by a plane, there is a reflection. It is the timing and direction of this reflection that is measured. The coordinate system used is also shown where r is the range, Ψ the field of view, Φ the elevation, z the altitude and θ the rotation around the z axis.

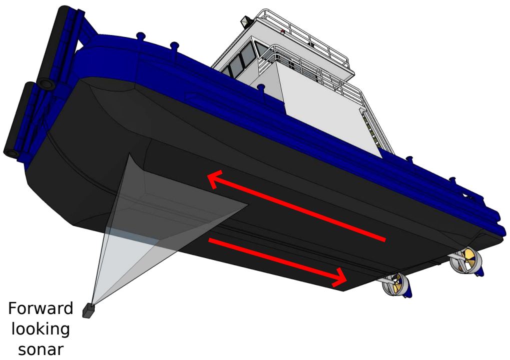


Fig. 2. An example application for a FLS is the inspection of the underside of a ship. An inspection robot with an FLS would follow multiple tracks along the ship's hull (illustrated here as red arrows) to ensure that whole surface is inspected.

frame might be of 96×512 'pixels' [12], whereas, an inexpensive optical sensor can produce an image of a much higher resolution, e.g. 3280×2464 pixels [13]. Optical registration techniques often select image segments assumed to be associated with a particular movement [14]. However, FLS images typically have a low signal to noise ratio (SNR). At a low SNR, the segment boundaries are difficult to identify and track between frames. There may also be fast signal intensity changes due to variation to the sensor attitude [15]. The intensity can also change rapidly due to the adaptive gain control (if not disabled) in the sonar.

With a single illumination point and the native polar format of FLS frames, for relatively simple sonar sensor motions, the map of pixel movements can become complicated [16]. For example, a pixel can appear or disappear rapidly due to acoustic shadows.

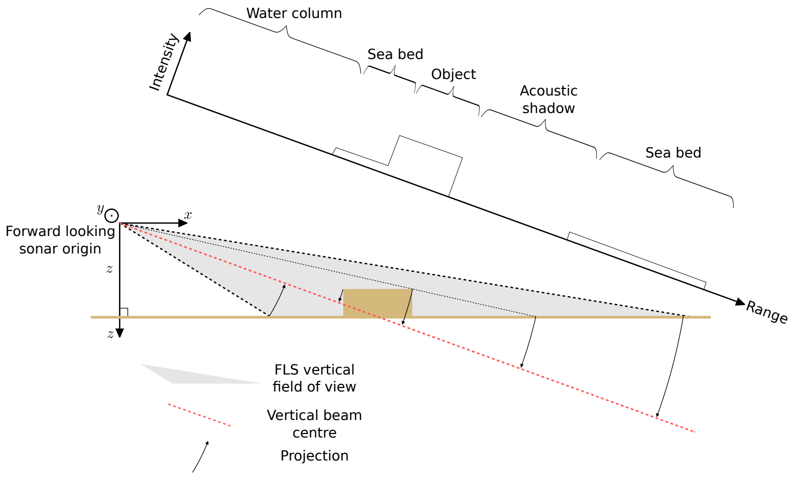


Fig. 3. An illustration of the projection of the reflections onto the beam center caused by a single wide vertical beam. Also shown is how the single point illumination casts shadows in the image.

The sonar sensor motion estimation can be based on comparison/registration of pairs of sonar frames [15]. A range of techniques have been used for registering FLS frames. One approach is to process a frame to identify distinct features that may be tracked between frames, such as a sharp transition in the image intensity, where an acoustic shadow occurs [3]. More sophisticated registration can be based on modeling of the geometry of features on the seabed [11]. However, these techniques rely on accurate extraction of the seabed features, which can be problematic at low SNRs. A technique developed in [10] does not rely on feature extraction, but instead processes the whole frame in the frequency domain. In each of these examples, FLS frames are preprocessed into Cartesian coordinates. The raw sonar data can be converted into a Cartesian format but this introduces an interpolation error at the very beginning of the processing pipeline, which is undesirable.

One influential idea for optical image registration (pixel displacement estimation) is based on the *optical flow*, which relates to the brightness variation within a scene and in time [17]. *Optical flow* has been successfully applied to sonar images for object tracking [18] and vehicle motion estimation [19] with the assumption that over the image as a whole there is sufficient consistency of the intensity between frames to give meaningful results. More recent developments for optical registration use techniques to select image segments considered to have similar movement [20]. Alternatively, pixels may be grouped by particular attributes to form super pixels [21]. Matching the elements between frames is then achieved through minimization of a suitable cost function [14].

This paper proposes an algorithm to estimate the motion of an underwater platform based on the *optical flow* between consecutive FLS frames. Firstly, the work proposes and details a method for estimating the pixel displacement map (DM) between a pair of FLS frames. The proposed estimator does not rely on identifying particular features in the sonar frames, thus making it more versatile. The processing consists of two stages, coarse estimation to pixel precision and fine estimation to sub-pixel precision. Using a sparse motion representation, the coarse estimation stage produces and compares forward-backward displacement estimates, allowing artifacts caused by revealed or occluded pixels to be identified. The fine estimation stage, based on adaptive filtering along a Hilbert space filling curve over the frame, identifies the sub-pixel displacements. The coarse estimate allows the aperture of the adaptive filter to be reduced to a few pixels, thus reducing the complexity and improving the accuracy at this stage.

Secondly, this work proposes a method for estimation of the sonar sensor inter-frame motion. This estimation is based on converting a DM into a small set of summary statistics. This set is compared to statistics pre-computed for modeled motions within an expected motion range. The best match identifies the sensor motion between the pair of frames.

Thirdly, a method is proposed that uses weighted regularized splines to integrate the inter-frame motions into an attitude-trajectory estimate for the sonar sensor.

Although the purpose of this paper is to propose and investigate techniques that would provide accurate localization of an FLS platform, significant attention was paid to the complexity of the implementation. In particular, the inter-frame DM estimator is partitioned into coarse and fine estimation stages; this partitioning allows a reduction in the aperture of the adaptive filter thus significantly reducing the complexity of the DM estimator. A downsampling scheme is used for the coarse estimation that significantly reduces the complexity. The coarse estimation is based on the Orthogonal Matching Pursuit (OMP) algorithm, which can be efficiently implemented in the form of the Dichotomous Coordinate Decent (DCD) OMP [22]. The Recursive Least Squares (RLS) adaptive filter used for sub-pixel estimation can be efficiently implemented as the DCD-RLS algorithm [23]. For the inter-frame motion estimation, a compact descriptor and a fast search algorithm are used. B-splines used for transforming the inter-frame motions into the attitude-trajectory estimate have local support thus making possible on-line tracking of the FLS platform and also allowing an efficient implementation.

The result for an example FLS dataset is validated by producing a mosaic built by using the attitude-trajectory estimate. Note that in some applications the final aim is the production of a seabed mosaic [24], therefore techniques proposed in this paper can be used not only for navigation itself but also for building such mosaics. Throughout this text we will be using a numerical example related to a FLS inspection of a ship's hull. The dataset, captured by Bluefin Robotics Corp. [25], was using a hull inspection procedure described in [4]. In this context, the 'seabed' is the underside surface of a ship's hull.

The remainder of this paper is arranged as follows: Section II is a description of an *optical flow* algorithm for producing a DM that shows the estimated movement of each individual pixel. Section III describes the estimation of the sonar sensor motion from a sequence of DMs. The process of fitting a modeled movement to the DM is described and the post processing and smoothing of the estimated attitude-trajectory is detailed. Sections IV-A and IV-B present the attitude-trajectory estimates of two datasets. Finally, Section V gives some concluding remarks.

II. ESTIMATION OF INTER-FRAME DISPLACEMENT MAP

In this section, we describe the method for estimating an inter-frame DM. This is based on a selection of *optical flow* estimation techniques applied to two frames represented in the native polar (range/bearing) format. The frames are processed without extraction of seabed features. The processing is divided into two stages, as illustrated in Fig. 4, namely the coarse and fine displacement estimation. Based on sparse motion representation, the coarse estimation produces two (forward and backward) initial DMs with pixel precision. These maps are compared to divide the frame into areas with trusted displacement estimates and areas with artifacts, e.g. caused by revealed or occluded pixels due to acoustic shadows. Displacement estimates for areas with artifacts are extrapolated from the accurate estimates in the vicinity. The fine estimation is based on adaptive filtering along a Hilbert space filling curve over the frame; it identifies the motion with a sub-pixel precision. Finally the inter-frame DM is refined by removing outliers using median filtering.

A. Coarse displacement estimation

1) Signal model and sparse recovery: Consider two frames, with a reference image $i_r(\xi)$ and a target image $i_t(\xi)$, where ξ is a pixel position within a frame X: $\xi \in X$. We assume that the relationship between the two frames is described by the convolution

$$i_t(\xi) = \sum_{m=1}^{M} h(\xi_m, \xi) i_r(\xi - \xi_m) + \varepsilon(\xi), \tag{1}$$

where $h(\xi_m, \xi)$ is an unknown kernel at pixel location ξ, ξ_m is a displacement and $\varepsilon(\xi)$ a measurement noise. The displacements ξ_m are limited to a support X_r, $\xi_m \in X_r$, e.g. X_r

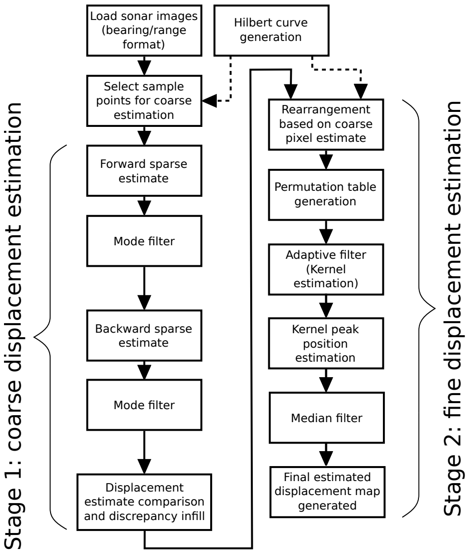


Fig. 4. Block diagram of the registration algorithm that generates the pixel displacement map between a pair of sonar images. The algorithm is split into two stages, there is the coarse estimation which is to a whole pixel precision and the fine estimator that refines the estimate to a sub-pixel precision.

is a rectangular window around the origin. The support size is defined by dynamics of the sonar platform and sonar frame rate. In this section, the displacements ξ_m are assumed to be discrete to the pixel positions, while in Section II-B we will consider the sub-pixel case. This is a simplified signal model that, for example, does not describe the occluded and revealed pixels. At this stage, we assume that these effects are encompassed in the measurement noise. Note that the number M in 1 can be high. In our numerical example, the support X_r is a square window of size 41 × 41 so that $M = 41^2 = 1681$. Ideally, with a pixel-width movement, the kernel $h(\xi_m, \xi)$ would contain a single non-zero coefficient. However, generally the movements are fractions of a pixel width, which require several discrete pixel-wide movements to approximate accurately. Besides, several objects can move independently within the support X_r and so several (L) coefficients $h(\xi_m, \xi)$ will be non-zero. However, typically $L \ll M$; in our numerical example, we use $L = 3 \ll M = 1681$.

At this stage, the task is to identify the kernel $h(\xi_m, \xi)$ at a pixel position ξ. For the identification, the target frame in the vicinity X_t of pixel ξ is treated as a measurement vector (vect) γ:

$$\gamma = \text{vect}\{i_t(\xi - \xi_m),\ \xi_m \in X_t\}. \tag{2}$$

The size (cardinality) $N = |X_t|$ of the set X_t should be related to the search size M and the number of non-zeros L. The number of measurements N should be sufficiently high to allow the kernel identification at low SNRs. However, if the kernel is fast varying over the frame, N should be reduced. For our numerical example, we choose X_t as a 13×13 square aperture, so that $N = 13^2 = 169$. This is large enough to be comparative with the identifiability criteria for random measurement selection, where $N = \mathcal{O}(L \log(M))$ [26], or the more conservative criterion of $N \geq \mathcal{O}(L^2 \log(M/L))$[27]. We use the reference frame in the vicinity of pixel ξ to generate an $N \times M$ dictionary matrix $\mathbf{A}$ with columns

$$\mathbf{A}^{(m)} = \text{vect}\{i_r(\xi - \xi_m),\ \xi \in X_t,\ \xi_m \in X_r\}. \tag{3}$$

The columns of $\mathbf{A}$ are patches of images from the reference frame in the vicinity of the point of interest ξ. Then we can write

$$\mathbf{A}\mathbf{h} = \boldsymbol{\gamma}, \tag{4}$$

where $\mathbf{h} = \text{vect}\{h(\xi_m, \xi),\ \xi_m \in X_r\}$ is an $M \times 1$ vector associated with modeled displacements.

Since the linear system (4) has a sparse solution, the first L coefficients of $\mathbf{h}$ are estimated using the Orthogonal Matching Pursuit (OMP) [28][26]. In the estimate $\hat{h}(\xi_m, \xi)$ only the displacement $d(\xi)$ corresponding to the coefficient with the highest magnitude is kept for our further processing:

$$d(\xi) = \underset{\xi_m \in X_r}{\arg\max}\, |\hat{h}(\xi_m, \xi)|; \tag{5}$$

We call this the coarse DM.

2) Selection of measurement (sample) points: Ideally, the coarse displacement estimation would be made for each pixel position in the frame. However, this would make the problem too complex in practice. Besides, with a measurement window X_t of size N, the number of estimates in a frame can be reduced by a factor comparable to N. In order to ensure a uniform density of sample points over the frame, a selection from a uniform distribution over the frame could be made. However, for a sonar image, portions of the image that give high intensity returns are likely to provide most accurate displacement estimates. Therefore, in our algorithm the reference image intensity is used as a probability density function (PDF) for generating the sample points $\xi \in X_{\text{sample}}$ [29][30]. An example distribution of sample points X_{sample} can be seen in plot 3 of Fig 5, where the PDF is defined from the reference frame (plot 1). In our numerical example, generating the sample points stops when $|X_{\text{sample}}| = 2891 \approx 1/17 |X|$.

3) Mode filter: From the displacement estimates $d(\xi) \in X_{\text{sample}}$, we wish to interpolate displacement values for all pixels in the frame, whilst eliminating outliers. Moreover, we wish to preserve the pixel precision estimates. To that end we can consider the individual

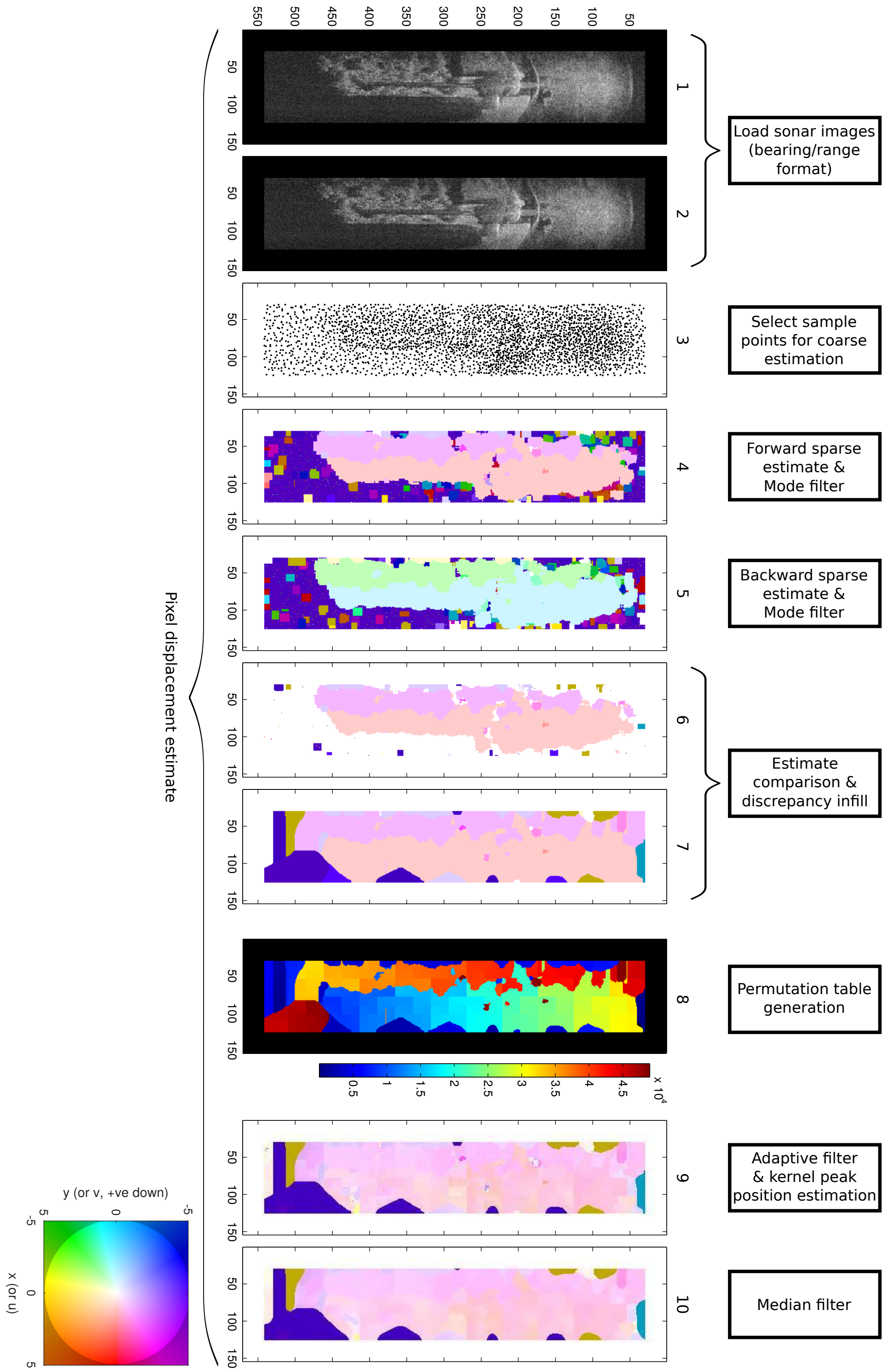


Fig. 5. Example outputs at selected stages of the proposed inter-frame DM estimation algorithm.

motions as a nominal data type, and apply a square aperture mode filter to propagate the displacement values. More formally, the operation is as follows. An interpolated DM is given by

$$\hat{d}(\xi) = \arg\max_{k} g(k), \quad \xi \in X, \tag{6}$$

where $g(k)$ are elements of a vector $\mathbf{g}$ representing a histogram (hist) of displacements measured in the vicinity X_{mode} of pixel ξ:

$$\mathbf{g} = \text{hist}\{d(\xi - \xi_m), \xi_m \in X_{\text{mode}}, (\xi - \xi_m) \in X_{\text{sample}}\}. \tag{7}$$

In our numerical example, X_{mode} is a square window of size 13×13.

4) Comparison of forward and backward coarse DMs: To estimate the movement between two frames, the registration can be performed not only from the reference to the target frame (forward), but also from the target to the reference frame (backward). A comparison between these two estimated displacements allows validation of the quality of the registration because the two displacements should complement, in that one should be the reverse of the other [24]. In addition, the comparison can be used to remove estimates where pixels are revealed or occluded and therefore have no complement. The comparison for every pixel position $\xi \in X$ is made by taking the forward displacement estimate and examining the backward estimate at the location $\xi - \hat{d}(\xi)$. If the magnitude of the sum of these two displacements is less than a threshold (2 pixels in our example) then the forward estimate is retained. Example forward and backward DMs can be seen in plots 4 and 5 of Fig. 5. The accepted displacements from the forward and backward comparison are shown in plot 6 of Fig. 5.

The pixel locations where the displacement estimate is not retained are interpolated, taking the value of the nearest (in terms of pixels) accepted estimate. An example of the interpolated DM $\tilde{d}(\xi)$ is shown in plot 7 of Fig. 5.

B. Fine displacement estimation

At this point in the proposed algorithm, there is a coarse displacement estimate $\tilde{d}(\xi)$ associated with each pixel location in the reference frame. The next stage is to refine the estimate further to a sub-pixel precision.

Adaptive filters have been widely used in signal processing applications for system identification [31], and so are a good candidate technique to apply to the *optical flow* problem [32], [33], [34], [35]. Here an adaptive filter is used to identify the convolution kernel $h(\xi_m, \xi)$, related to the sonar sensor motion, to a sub-pixel precision.

The adaptive filter works most effectively if the change in the convolution kernel from one adaptive iteration to the next is a slow evolution. With this aim, the order that the pixels are presented to the filter is chosen to try to group similar movements. To achieve this, we propose and use the following ordering algorithm producing a permutation table. Firstly, each displacement $\xi_m \in X_r$ represents a bin in the permutation table, i.e. M bins in total. The bins are ordered to provide the slowest evolution between consecutive bins. The algorithm scans the DM $\tilde{d}(\xi)$ along a pseudo-Hilbert space filling curve (SFC) pixel-by-pixel [30]. The use of the SFC itself ensures a slow evolution of the kernel. For every pixel, the pixel position is added into the bin associated with the value $\tilde{d}(\xi)$. After the scan, every bin in the permutation table contains positions of all pixels with the same displacement, thus providing the slowest evolution of the convolution kernel within the bin. The position within a bin is ordered according to appearance along the SFC. In total, I positions, I being the number of pixels in a frame, are recorded in all bins, thus the table describes an $I \times I$ permutation matrix. Thus, when reading from the permutation table from the first to the last bin, and, within a bin from the first to the last pixel, the convolution kernel to be estimated would have a slow evolution. Plot 8 of Fig. 5 shows an example of the rearranged pixels; where the pixel color represents where in the order it is, blue is processed first and red last.

The input to the adaptive filter is a sequence of square pixel apertures representing the regressor vectors from the reference frame, in the order determined by the permutation table. The size M_{fine} of the filter kernel $h_{\text{fine}}(\xi_m, \xi)$ to be estimated, and consequently size of the input vector (*regressor*) should be kept as small as possible to improve the identification accuracy. To guarantee this small size, the *desired signal* of the adaptive filter, which is taken from the target image $i_t(\xi)$, is translated according to the coarse estimate $\tilde{d}(\xi)$, i.e. the desired signal is $i_t(\xi - \tilde{d}(\xi))$. Thus, the fine estimation is performed only in a part of area X_r, in the vicinity of the pixel position $\xi - \tilde{d}(\xi)$, thus allowing a reduction in the filter aperture. The size of the filter is therefore set with reference to the discrepancy allowed in the sum of the coarse forward-backward DM comparison, and in our example is 2 pixels width. To encompass a variation of ±2 pixels width, an aperture of 7×7 pixels is used in our example. This translation of the *desired signal* is illustrated in Fig. 6. The signal model in (1) is now refined to

$$i_t(\xi - \tilde{d}(\xi)) = \sum_{m=1}^{M_{\text{fine}}} h_{\text{fine}}(\xi_m, \xi) i_r(\xi - \xi_m) + \varepsilon(\xi - \tilde{d}(\xi)). \tag{8}$$

For each pixel position ξ, the adaptive filter produces an estimate of the convolution kernel h_{fine}, by minimizing the least squares error [36], [23], [31]. The adaptive filter is applied to the whole frame and a convolution kernel of 7×7 coefficients is produced for each pixel location. The adaptive filter algorithm used is an exponentially weighted recursive least squares (ERLS) [31] with a forgetting factor $\lambda = 0.98$.

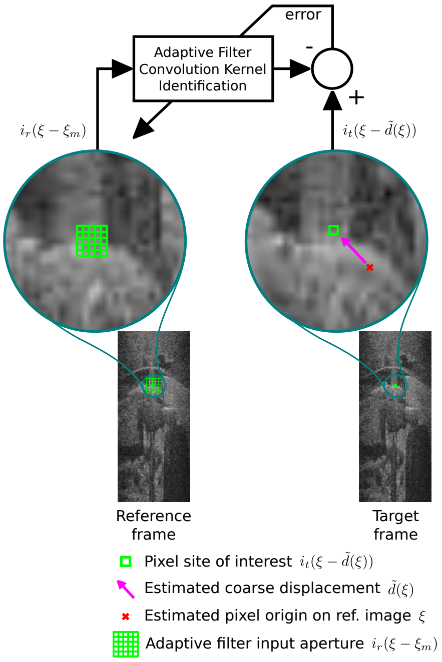


Fig. 6. An illustration of an adaptive filter in identification mode. A convolution kernel is estimated which when convolved with an aperture of pixels from the reference image will produce a single pixel of the target image.

Assuming that in reality there is a continuous 2-D convolution kernel whose samples at the pixel positions have been estimated by the adaptive filter as a vector $\hat{\mathbf{h}}$, we can identify the pixel displacement using interpolation. In our example, we use parabolic interpolation leading to the two-sided technique for estimation of the peak position as described in [37]. An initial fine DM is then created from the sum of the coarse and fine pixel estimates.

Finally, the fine DM $\bar{\mathbf{h}}(\xi)$ is obtained by applying a median filter to all pixels in the initial fine DM. This is to remove isolated extreme results whilst preserving the boundaries between larger regions of motion. In our numerical example, the aperture of the median filter is a square of size 13×13.

An example of the output of the adaptive filter and the median filtered fine DM are shown in plots 9, and 10, respectively, of Fig. 5.

III. ESTIMATION OF THE INTER-FRAME SONAR SENSOR MOTION

The fine displacement estimation has produced the DM $\bar{d}(\xi)$. The purpose of the algorithm described in this section is, based on $\bar{d}(\xi)$, to estimate an inter-frame motion vector $\boldsymbol{\alpha}$. Firstly, we will describe a preprocessing of $\bar{d}(\xi)$ to select an area X_B of the frame that contains most reliable information relevant to the dominant motion of the sensor. We then introduce the motion model and establish the relationship between the motion parameters ($\boldsymbol{\alpha}$) and the DM.

We then formulate the motion estimation as a least squares (LS) minimization problem. This LS problem can be solved using standard optimization techniques, such the interior-point method [38], which would however require a high computational load. A significantly more efficient method for real-time processing is firstly to transform the DM into a small vector of auxiliary statistics and to apply a dichotomous coordinate descent (DCD) search to match the statistics with a set of precomputed modeled 'statistics'.

A. Preprocessing of the fine DM

The purpose of the preprocessing is to remove unreliable parts of the DM to reduce their influence on the final estimate. This is done in two steps. The first step is a thresholding of the reference frame to remove low intensity image parts with weak sonar returns. The threshold is based on computing a histogram of intensity in the reference frame and choosing a predefined quantile, for example 25%. The second step is based on processing the DM only at pixels kept after the first step. In this processing, a histogram for magnitudes of the displacements is computed, to which Tukey's rule [39] is applied to identify outliers. More specifically, if Q_1 and Q_3 are lower and upper quartile, respectively, then an outlier is a displacement outside the range: $[Q_1 - \gamma(Q_3 - Q_1), Q_3 + \gamma(Q_3 - Q_1)]$ for a nonnegative constant γ, e.g. $\gamma = 0.65$ [39]. The outlier pixels are also removed from further processing, thus finally identifying a reliable set X_B on the reference frame. An example X_B is shown (in white) in plot 11 of Fig. 7.

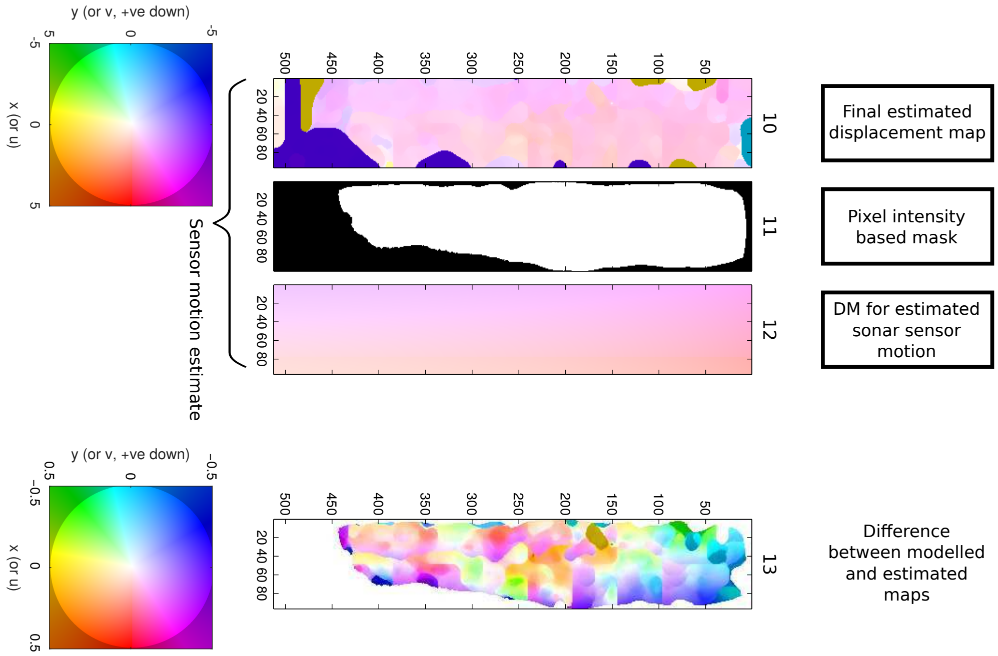


Fig. 7. An example match for the estimated DM and a modeled motion DM.

B. The motion model

There are six possible degrees of motion of the sonar sensor, which are translations in axis x, y, z and rotations around each axis. We assume that the altitude z is predefined and is constant for the duration of the experiment. Also, the assumption is made that the platform roll (rotation around axis x) and pitch (rotation around axis y) are negligible (see Fig.1 for an illustration of the coordinate system). Therefore for estimation we consider only translations Δ_x and Δ_y in x and y and rotation Δ_θ around z: $\boldsymbol{\alpha} = [\Delta_x, \Delta_y, \Delta_\theta]$. These assumptions are specific for our numerical example. However, a more general case can be similarly considered.

The displacement is described by a model

$$d_{\text{model}}(\xi) = [\xi^{(\psi)} - \xi_t^{(\psi)}, \xi^{(r)} - \xi_t^{(r)}], \tag{9}$$

where $\xi = [\xi^{(\psi)}, \xi^{(r)}]$ is a pixel position (beam angle and range, respectively) in the reference frame and $\xi \in X_B$, and $\xi_t = [\xi_t^{(\psi)}, \xi_t^{(r)}]$ is the new pixel position (beam angle and range, respectively) after the modeled motion.

The pixel position is transformed from polar coordinates in the reference frame to Cartesian coordinates $[u_x, u_y]$ on the seabed:

$$\begin{aligned} u_x &= \cos\left(\delta_\psi \xi^{(\psi)}\right) \sqrt{[\delta_r \xi^{(r)}]^2 - z^2}, \\ u_y &= \sin\left(\delta_\psi \xi^{(\psi)}\right) \sqrt{[\delta_r \xi^{(r)}]^2 - z^2}, \end{aligned} \tag{10}$$

where δ_r and δ_ψ are range and angle resolutions defining a single pixel and z is the predefined sensor altitude. The projection onto the seabed is illustrated in Fig. 8. The motion $\boldsymbol{\alpha}$ is transformed into a new position on the seabed:

$$\begin{aligned} v_x &= u_x \cos(\Delta_\theta) + u_y \sin(\Delta_\theta) - \Delta_x, \\ v_y &= -u_x \sin(\Delta_\theta) + u_y \cos(\Delta_\theta) - \Delta_y. \end{aligned} \tag{11}$$

Finally, the polar coordinates in the target frame for the new position on the seabed are given by

$$\begin{aligned} \xi_t^{(\psi)} &= \frac{1}{\delta_\psi} \left(\arctan\left(\frac{v_y}{v_x} \right) \right), \\ \xi_t^{(r)} &= \frac{1}{\delta_r} \left(\sqrt{v_x^2 + v_y^2 + z^2} \right). \end{aligned} \tag{12}$$

C. Estimation of the motion vector $\boldsymbol{\alpha}$

The motion can be estimated by solving the LS optimization problem:

$$\hat{\boldsymbol{\alpha}}_{LS} = \arg\min_{\boldsymbol{\alpha}} J(\boldsymbol{\alpha}), \tag{13}$$

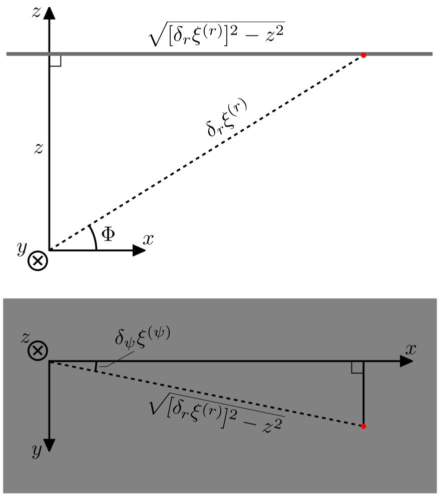


Fig. 8. Position of the point $[u_x, u_y]$ on the plane being illuminated. In our example, this plane is the underside of a ship's hull and therefore the sonar is looking upward.

where the LS cost function is given by

$$J(\boldsymbol{\alpha}) = \sum_{\xi \in X_B} ||\bar{d}(\xi) - d_{\text{model}}(\xi)||^2, \tag{14}$$

and $d_{\text{model}}(\xi)$ depends upon $\boldsymbol{\alpha}$ as described in Section III-B. Obtaining the solution to (13) has high complexity because, for an iterative LS solver, a model DM must be regenerated multiple times using the complicated non-linear transforms (10) to (12) for every pixel $\xi \in X_B$.

D. DM dimension reduction and dichotomous coordinate descent search

To reduce the complexity of the inter-frame motion estimator, the fine DM $\bar{d}(\xi), \xi \in X_B$, is represented by a small number of statistics (20 in our example) as follows. Four histograms are computed from the DM:

$$\mathbf{G}_{\pm}^{(j)} = \text{hist}\{\bar{d}^{(j)}(\xi), \xi \in X_B^{\pm}\}, j = 1, 2, \tag{15}$$

where X_B^+ and X_B^- denote parts of X_B for positive and negative beam angles and $\bar{d}^{(j)}(\xi)$ denotes the j^{th} coordinate of vector position $\bar{d}(\xi)$. For each of the histograms, five percentiles are computed (5th, 25th, 50th, 75th and 95th), thus twenty parameters in total; we denote all of these percentiles as a vector s.

The vector s is then compared with vectors $\mathrm{s}_{\text{model}}(\boldsymbol{\alpha})$ computed on a grid of motions $\boldsymbol{\alpha} \in T$ to find the best match, where T is a grid of sensor motions bounded by the dynamics of the platform. Specifically, the modeled DMs for all possible sensor motions $\boldsymbol{\alpha} \in T$ are pre-computed, and for each of them we store the twelve 'statistics' as was described above.

The match is based on the minimization of the Euclidean distance

$$\hat{\boldsymbol{\alpha}}_s = \arg\min_{\boldsymbol{\alpha}\in T} ||\mathbf{s} - \mathbf{s}_{\text{model}}(\boldsymbol{\alpha})||^2. \quad (16)$$

The solution of this minimization problem can be found using an exhaustive search over the grid. Our comparison of run time in Matlab for this technique and the LS optimization in (13) has shown the complexity ratio to be about 5. In our numerical example, the cardinality of T is $|T| = 97 \times 97 \times 97 \approx 9 \times 10^5$ which results in approximately 18×10^6 addition and multiplication operations to find $\hat{\boldsymbol{\alpha}}_s$. This is still high for real-time processing. To further reduce the complexity, we propose to use the DCD search [23] on the grid T. With the DCD search, the complexity is $20 \log_2 |T| \approx 400$ multiplications and additions which is significantly lower than that of the other two techniques; about 4×10^4 times lower than using the exhaustive search. The last two techniques require a storage space for the $20 \times |T|$ statistics, which is a payment for faster computation. However, this is comparable to the storage for a single frame and therefore is not a significant overhead.

The result of the processing described above when applied to all frames in a sequence of P frames is a sequence of motion vectors $\{\boldsymbol{\alpha}_k\}_{k=1}^{P-1}$. For further processing we also retain the values $l_k = J_{LS}(\boldsymbol{\alpha}_k)$ or $l_k = J_s(\boldsymbol{\alpha}_k)$, depending on the estimation technique used. These will be transformed into a set of weights $\{w_k\}_{k=1}^{P-1}$ that characterize the accuracy of the inter-frame motion estimation. In our example, we use the weights

$$w_k = \max\left\{0, 1 - \frac{\beta l_k}{\max_n l_n}\right\}, \quad (17)$$

where β is a positive constant set to $\beta = 0.9$. An example of a matched modeled DM is shown in plot 12 of Fig. 7. A DM showing the difference between the estimated DM and the matched modeled DM is shown in plot 13 of Fig. 7.

E. Attitude-trajectory estimation

The aim is now to produce an estimate of the sonar sensor attitude and position at any time t within an experiment. More specifically, the incremental movements $\boldsymbol{\alpha}_k = [\Delta_x(t_k), \Delta_y(t_k), \Delta_\theta(t_k)]^T$, where $t_k = kT_{\text{frame}}$ need to be processed to produce positions and attitudes $\mathbf{p}(t) = [x(t), y(t), \theta(t)]$ that are in a coordinate system fixed on the seabed.

Firstly, we represent $\theta(t)$ as a smoothed spline found as a trade off between an error in the fit to the data points and smoothness of the spline. One efficient way is to use P-splines [40]. The spline is given by

$$\theta(t) = \sum_{m=1}^{N_b} c_m B(t - (m-1)\tau), \quad (18)$$

where $B(t)$ is a b-spline [41], N_b the number of basis functions, c_m are basis expansion coefficients, and in our example $\tau = T_{\text{frame}}/4$. The basis functions used in this work are cubic B-splines. The angular velocity $\theta'(t)$ is then given by

$$\theta'(t) = \sum_{m=1}^{N_b} c_m b(t - (m-1)\tau), \tag{19}$$

where $b(t) = B'(t)$. For finding the spline coefficients c_m, a weighted LS optimization problem with a penalty is formulated as follows. For P-splines, the smoothness penalty is efficiently calculated based on the difference in the values of the spline coefficients themselves [40]. Therefore, the cost function takes the form

$$S = \sum_{k=1}^{P-1} w_k \left[\frac{1}{T_{\text{frame}}} \Delta_\theta(t_k) - \theta'(t_k) \right]^2 + \mu \sum_{m=n+1}^{N_b} (\Delta^n c_m)^2, \tag{20}$$

where $\mu > 0$ is a regularization parameter, Δ^n is the n^{th} difference operator. In our example, we use $n = 2$, i.e. $\Delta^2 c_m = c_m - 2c_{m-1} + c_{m-2}$. From [40], minimization of S over the vector of expansion coefficients $\mathbf{c} = [c_1, \ldots, c_{N_b}]^T$ is equivalent to solving the system of equations

$$(\mathbf{B}^T \mathbf{W} \mathbf{B} + \mu \mathbf{D}^T \mathbf{D})\mathbf{c} = \mathbf{B}^T \mathbf{W} \boldsymbol{\Delta}_\theta, \tag{21}$$

where $\mathbf{D}$ is the difference operator in matrix form and $\mathbf{B}$ is a matrix with the basis functions $b(t_k - (m-1)\tau)$ as its $P-1$ length columns, $\mathbf{W}$ is a diagonal matrix where weights w_k are the diagonal elements, and $\boldsymbol{\Delta}_\theta = [\Delta_\theta(t_1), \ldots, \Delta_\theta(t_{P-1})]^T$.

Having obtained the spline $\theta(t)$ we are able now to rotate the incremental motions $\Delta_x(t_k)$, $\Delta_y(t_k)$ onto the seabed coordinate system,

$$\begin{aligned} \Delta_{xx}(t_k) &= \Delta_x(t_k)\cos[\theta(t_k)] + \Delta_y(t_k)\sin[\theta(t_k)], \\ \Delta_{yy}(t_k) &= -\Delta_x(t_k)\sin[\theta(t_k)] + \Delta_y(t_k)\cos[\theta(t_k)], \end{aligned} \tag{22}$$

where $\Delta_{xx}(t_k)$ and $\Delta_{yy}(t_k)$ are the incremental movements aligned to x and y axis of the seabed coordinates, respectively. The same spline smoothing procedure as described above is then applied to increments $\Delta_{xx}(t_k)$ and $\Delta_{yy}(t_k)$ to find the smoothed splines $x(t)$, and $y(t)$, respectively.

The dynamics of the system can be incorporated into the smoothing regularization, for instance the regularizing parameter μ can be increased until the attitude-trajectory conforms to a maximum acceleration which is known in advance from the expected behavior of the system.

F. Pitch and altitude estimation

The motion estimator can be further extended by incorporating the pitch and altitude estimation. These estimates are then used to select a subset of motion models providing the best fit to the inter-frame DM. Assume that the sonar is looking at a flat plane and that the vertical beam-width Φ_w is known. Fig. 9 shows that the illuminated area on the seabed plane is dependent on the altitude and the pitch angle of the sonar sensor. The pitch angle Φ is estimated as

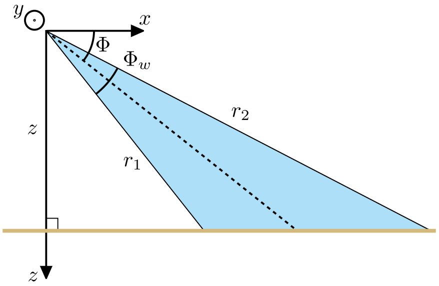


Fig. 9. The illuminated area from the vertical beam. From this area an estimate of the altitude (z) and pitch angle (Φ) of the sensor is made.

$$\Phi = \frac{\pi}{2} - \arctan\left(\frac{\cos(\Phi_w) - \frac{\xi^{(r_1)}}{\xi^{(r_2)}}}{\sin(\Phi_w)}\right) - \frac{\Phi_w}{2}, \tag{23}$$

where $\xi^{(r_1)}$ and $\xi^{(r_2)}$ are the ranges to the first and last points of the illuminated area, respectively, estimated from the sonar image. The altitude z is estimated as

$$z = \xi^{(r_1)} \cos\left(\frac{\pi}{2} - \Phi - \frac{\Phi_w}{2}\right). \tag{24}$$

IV. EXAMPLE DATASETS

The first example dataset is from the inspection of a ship's hull. The data, provided by Bluefin Robotic Ltd. [25], was recorded using a Bluefin Robotics Hovering Autonomous Underwater Vehicle (HAUV) [25] in a manner illustrated in Fig. 2 and described in [4]. The second considered dataset is where pitch and altitude are changing over a sequence of sonar frames. This example dataset was obtained by a surveying robot moving forward over a debris field (lumps of concrete and re-enforcing bars). The sonar rapidly sweeps back and forth in the yaw angle.

A. Results: Ship's hull dataset

Two mosaics have been produced, each from a separate track along the ship's hull. Fig. 11a shows a mosaic of 560 frames along the ship's hull moving predominantly in a positive

y direction. Fig. 14a shows a mosaic of 550 frames moving predominantly in a negative y direction. The components of the attitude-trajectories are plotted in Fig. 12 and Fig. 15. For comparison, a single frame from the dataset is shown in Fig. 10. Using the sensor's estimated attitude-trajectory, the pixel location from each frame is projected onto the seabed with reference to a fixed Cartesian coordinate system. The pixel values for the seabed mosaic image are then interpolated from the frame pixels. For clarity, only the center 24 beams are plotted, apart from the first and last frames where all of the beams are plotted. Where the frames overlap the pixel intensities are averaged over the number of frames to produce the final mosaic. The resolution of the final seabed image is 512 pixels per metre. In both mosaics, the estimated sonar sensor trajectory is shown as a red line and the sensor attitude, for every 30th frame, is shown as a blue arrow. The sonar sensor trajectory is estimated from consecutive frames with all of the frames in the dataset considered. Pitch and altitude estimation is not used for this dataset.

To asses the quality of the registration, the variation of intensity $V_p(\xi)$ at each pixel ξ is compared to the mosaic $\mu(\xi)$ (intensity mean) computed over $N(\xi)$ contributing frames:

$$V_p(\xi) = \frac{1}{N(\xi)} \sum_{n=1}^{N(\xi)} \frac{[i_t^n(\xi) - \mu(\xi)]^2}{\mu(\xi)^2}, \tag{25}$$

where $\xi \in X_m$ and X_m is the mosaic image, $i_t^n(\xi)$ is the nth registered frame in the sequence of $N(\xi)$ sonar frames contributing into the pixel ξ.

Fig. 11b and Fig. 14b show the intensity variation $V_p(\xi)$. The error can be seen to build up where many images are superimposed at one end of each track. In addition, errors from the movement of the acoustic shadow from the keel and the propeller can be seen in the maps.

The straightness of the cooling pipe is considered as a "ground truth" to measure the accuracy of the attitude-trajectory estimation. A Canny edge detector [42] is used to extract the edge of the uppermost cooling pipe in both the mosaics. The extracted lines are shown in blue in Fig. 13 and Fig. 16, respectively. The cooling pipe is supposed to be straight. It is approximated by first order polynomials shown in green in Fig. 13 and Fig. 16, respectively. The standard deviations in these two tracks are shown in Table I. This demonstrates a high accuracy of the attitude-trajectory estimation.

Fig. 17 shows an image created by overlaying the two mosaics. The scale for the cooling pipes and keel can be seen to be consistent. There is a greater difference at the stern of the ship where the motion model is less accurate since the shape of the ship's hull changes rapidly. This violates the assumption of constant altitude and pitch within each track and between tracks, resulting in a higher error.

TABLE I

THE STANDARD DEVIATION FROM AN IDEAL STRAIGHT LINE FOR THE TWO SONAR TRACKS.

Track	Std. dev. from straight line [cm]
1	2.9
2	1.2

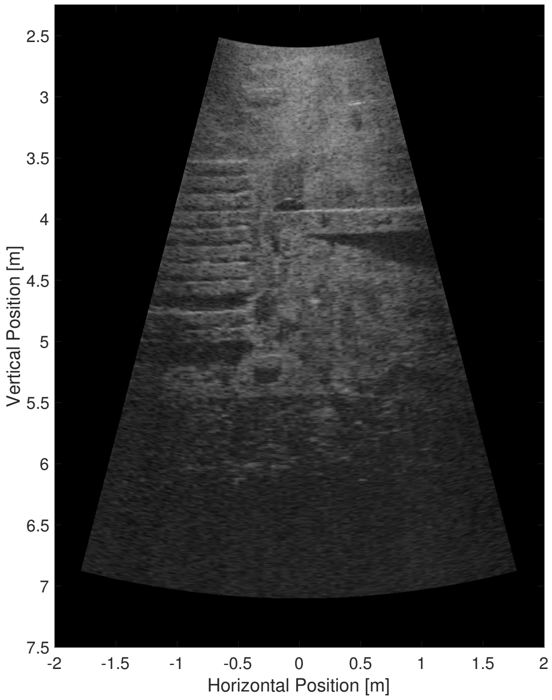
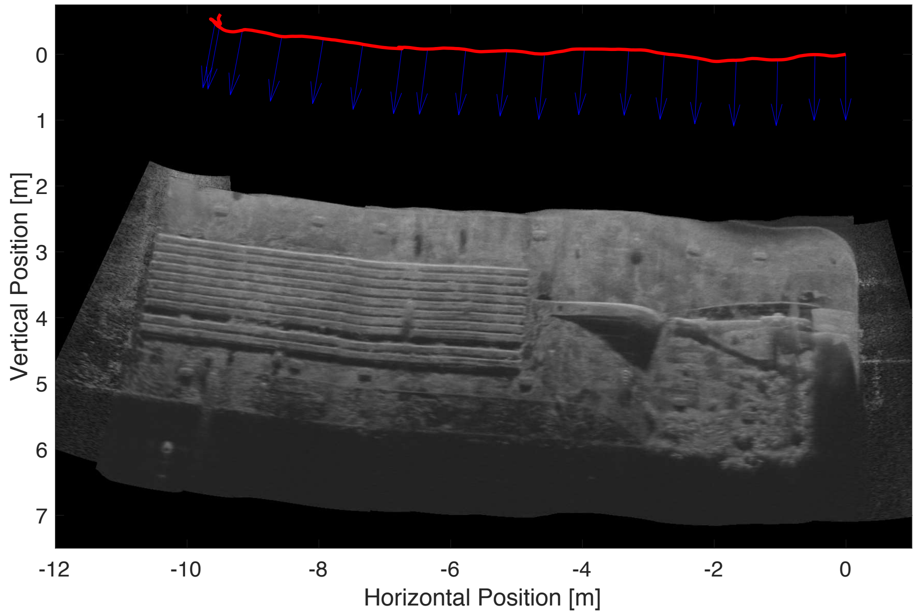
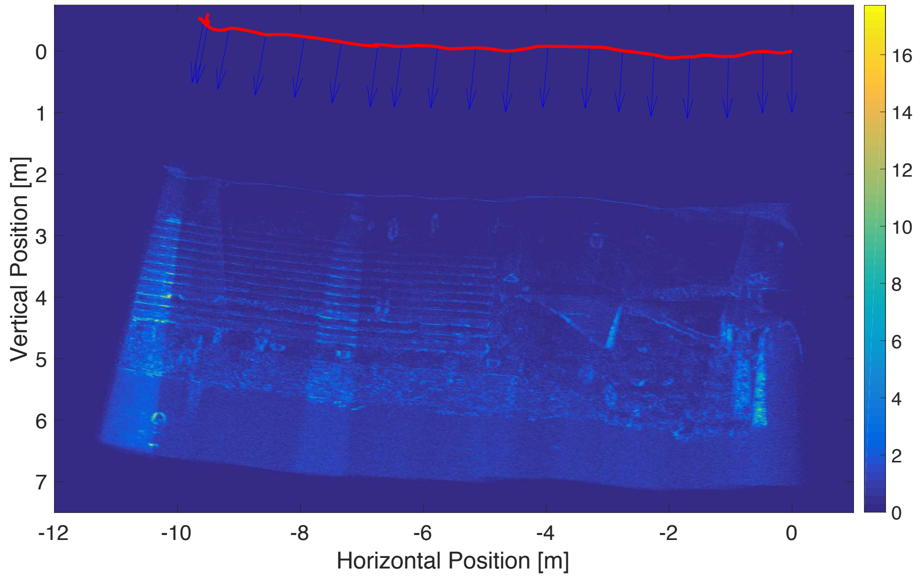


Fig. 10. A single frame from the ships hull dataset.



(a) A mosaic of 560 frames showing a track along the ship's hull (track motion is in positive y direction).



(b) The variation $V_p(\xi)$ of the pixel intensity for the mosaic.

Fig. 11. The mosaic created from the first track along the ship's hull and the intensity variation in the sonar frames at each pixel. The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

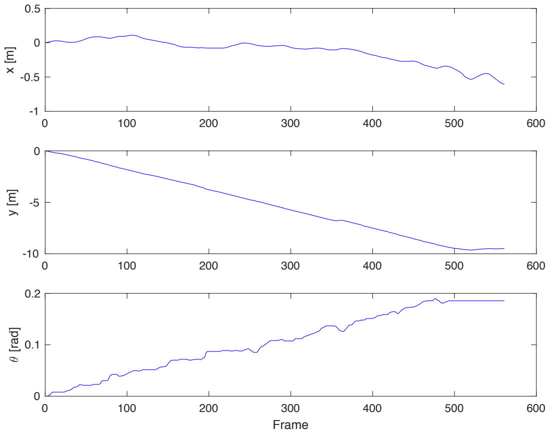


Fig. 12. The smoothed attitude-trajectory of 560 frames in the first track along the ship's hull.

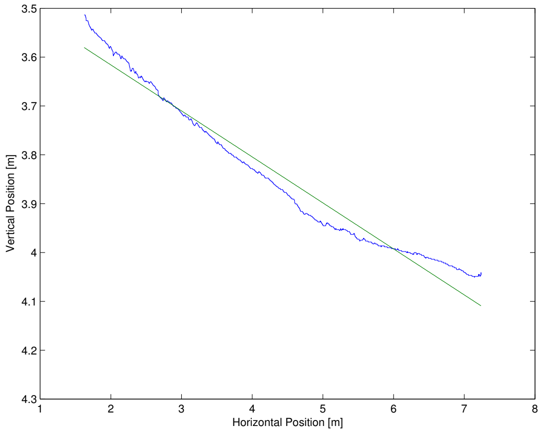
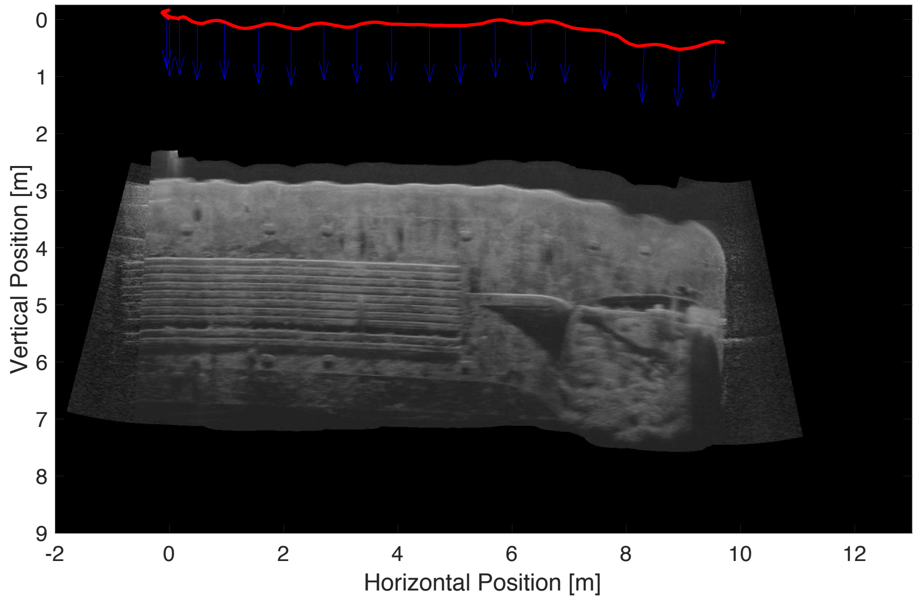
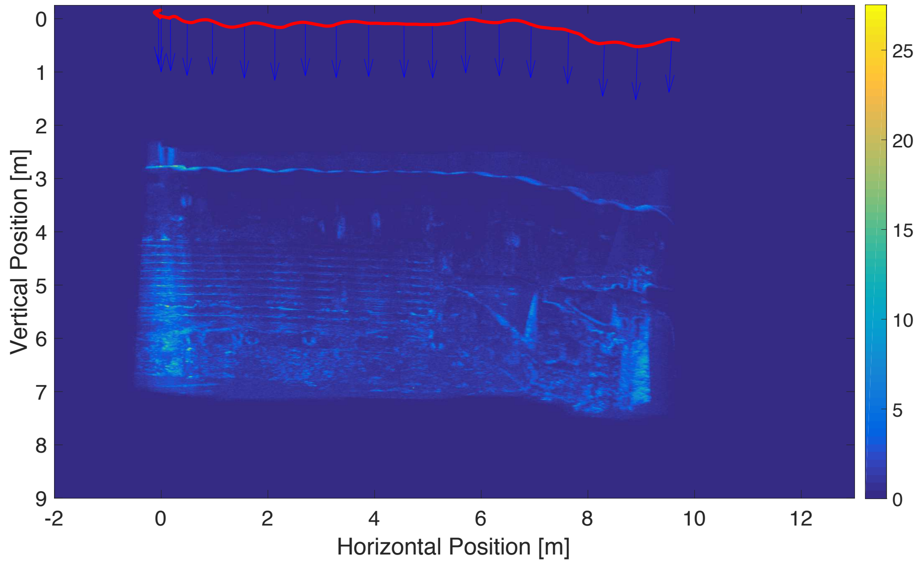


Fig. 13. The shape of the top cooling pipe which is measured from the image (blue) is supposed to be a straight line (first order polynomial). The green line shows the total least squares fit of the first order polynomial to the blue line. The standard deviation is 2.9 cm.



(a) A mosaic of 550 frames showing a second track along the ship's hull (track motion is in negative y direction).



(b) The variation $V_p(\xi)$ of the pixel intensity for the mosaic.

Fig. 14. The mosaic created from the second track along the ship's hull and the intensity variation in the sonar frames at each pixel. The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

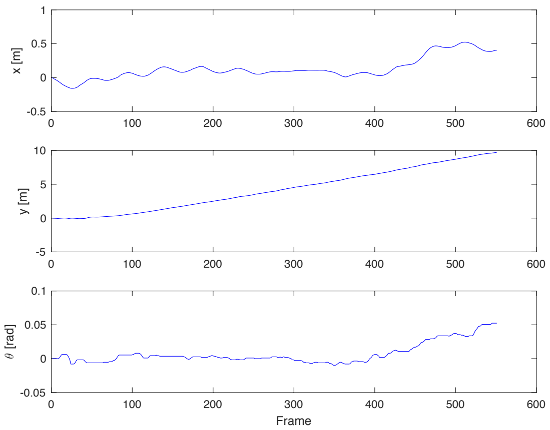


Fig. 15. The smoothed attitude-trajectory of 550 frames in the first track along the ship's hull.

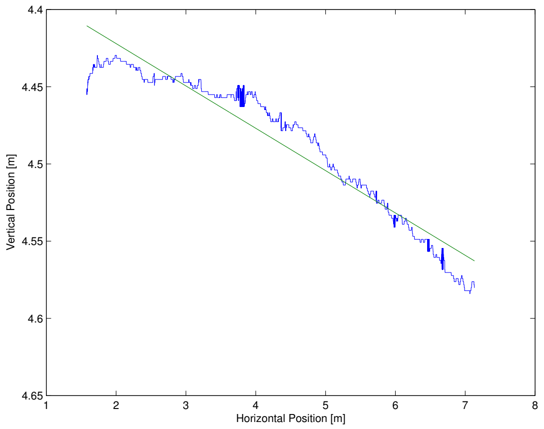


Fig. 16. The shape of the top cooling pipe which is measured from the image (blue) is supposed to be a straight line (first order polynomial). The green line shows the total least squares fit of the first order polynomial to the blue line. The standard deviation is 1.2 cm.

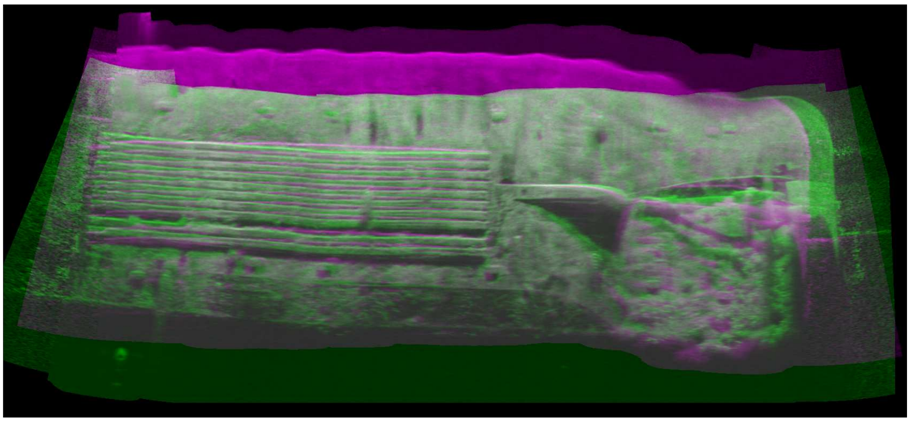


Fig. 17. An image created by overlaying the mosaics from the two tracks along the ship's hull.

B. Results: Debris dataset

Using the estimated attitude-trajectory for the sonar sensor including pitch and altitude estimation, a mosaic for the dataset has been produced. Fig. 18 shows the sonar view of the seabed. Fig. 19 shows a mosaic and attitude-trajectory estimate for 486 frames over the debris field. Fig. 20 shows the five components of the attitude-trajectory in this scenario. These results demonstrate that the proposed method is also applicable to scenarios with varying pitch and altitude.

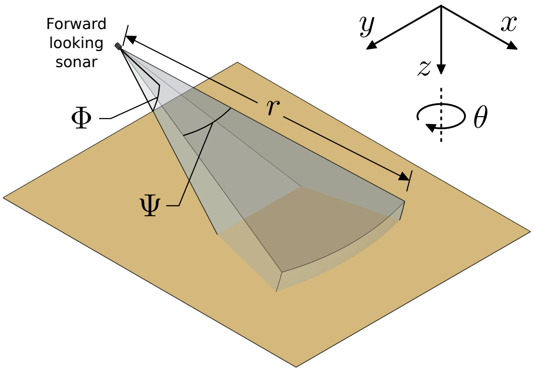


Fig. 18. Position of the FLS field of view and the seabed. r is the range, Ψ is the field of view, Φ the elevation and θ is the rotation around the z axis.

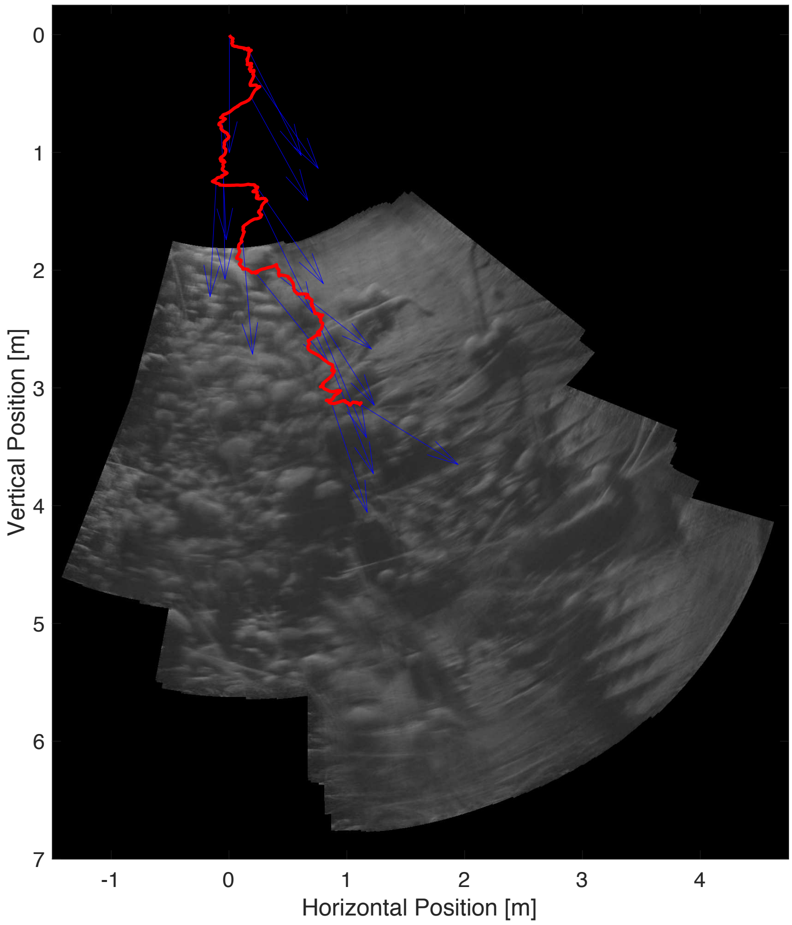


Fig. 19. A mosaic of 486 frames showing a debris field. The sonar sensor trajectory is shown in red and the attitude at every 30th frame is shown as a blue arrow.

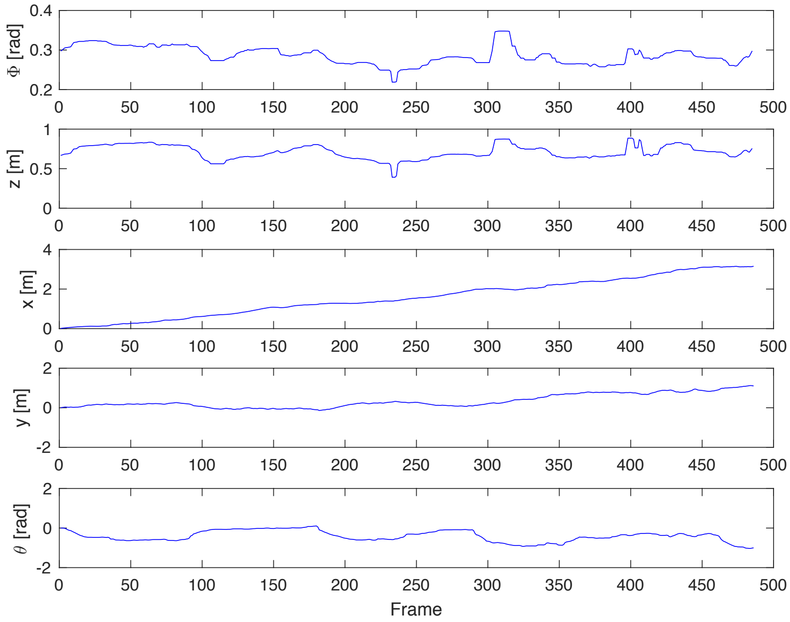


Fig. 20. The smoothed attitude-trajectory of 486 frames over the debris field.

V. CONCLUSION

In this paper, we have proposed a method for estimating the attitude-trajectory of a FLS sonar sensor by analyzing the pixel displacement between sonar frames. The proposed method initially estimates the displacement map (DM) that describes the motion of individual pixels between frames. By comparing the inter-frame DM with those generated analytically from modeled sonar sensor movements, a change in attitude and position for the sensor is estimated. The complexity of the DM comparison is reduced by representing each of the DMs with a set of summary statistics, and the search complexity is further reduced by means of a dichotomous search. Integrating the motion, and smoothing the result with reference to the physical system dynamics, an attitude-trajectory for the sonar sensor is estimated. Using an estimated DM, the sonar sensor movement has be estimated without the explicit identification of geometrical features. The use of smoothing splines (P-splines) is proposed in the generation of the attitude-trajectory. This allows incorporation of both knowledge about the motion bounds based on the vehicle dynamics, and the quality of the motion estimate.

To assess the accuracy of the results, from three datasets mosaics have been produced. Two are from different tracks of a ship's hull inspection. These two mosaics have well defined linear features that have a variation from a "best fit" straight line of approximately 0.5% of the line length. The third dataset is from the forward motion of a sonar over a debris field. The sonar moves back a forth in yaw. This dataset shows that the attitude-trajectory estimate can be extended to accommodate a pitch estimate.

ACKNOWLEDGMENT

The authors would like to thank Prof Yvan Petillot, School of EPS, Heriot-Watt University and Dr. Natàlia Hurtós, Girona Underwater Vision and Robotics, University of Girona, for providing the ship's hull dataset. Also, Dr. Luis A. Conti, School of Sciences and Humanities and Centre for Marine Biodiversity at the University of Sao Paulo and Acquest Subaquatic Geology and Geophysics [43] for providing the Debris dataset.

The authors would also like to thank the reviewers for their valuable comments which have helped to improve the paper.

REFERENCES

[1] A. Alcocer, P. Oliveira, and A. Pascoal, "Underwater acoustic positioning systems based on buoys with GPS," in *Proceedings of the Eighth European Conference on Underwater Acoustics*, vol. 8, 2006, pp. 1–8.

[2] A. Mallios, P. Ridao, D. Ribas, M. Carreras, and R. Camilli, "Toward autonomous exploration in confined underwater environments," *Journal of Field Robotics*, vol. 7, no. 33, pp. 994–1012, 2016.

[3] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. Leonard, “Imaging sonar-aided navigation for autonomous underwater harbor surveillance,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4396–4403.

[4] J. Vaganay, M. Elkins, S. Willcox, F. Hover, R. Damus, S. Desset, J. Morash, and V. Polidoro, “Ship hull inspection by hull-relative navigation and control,” in *MTS/IEEE OCEANS.*, 2005, pp. 761–766.

[5] G. T. Donovan, “Position error correction for an autonomous underwater vehicle inertial navigation system (ins) using a particle filter,” *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 431–445, 2012.

[6] J. N. Bakambu, C. Langley, G. Pushpanathan, W. J. MacLean, R. Mukherji, and E. Dupuis, “Field trial results of planetary rover visual motion estimation in mars analogue terrain,” *Journal of Field Robotics*, vol. 29, no. 3, pp. 413–425, 2012.

[7] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip *et al.*, “Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments,” *IEEE Robotics & Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.

[8] A. Elibol, N. Gracias, and R. Garcia, “Augmented state–extended kalman filter combined framework for topology estimation in large-area underwater mapping,” *Journal of Field Robotics*, vol. 27, no. 5, pp. 656–674, 2010.

[9] R. Garcia and N. Gracias, “Detection of interest points in turbid underwater images,” in *IEEE OCEANS-Spain*, 2011, pp. 1–9.

[10] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi, “Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments,” *Journal of Field Robotics*, vol. 32, no. 1, pp. 123–151, 2015.

[11] M. D. Aykin and S. Negahdaripour, “On feature matching and image registration for two-dimensional forward-scan sonar imaging,” *Journal of Field Robotics*, vol. 30, no. 4, pp. 602–623, 2013.

[12] (Accessed 2nd, April, 2016) Sound Metrics Corp. website. [Online]. Available: http://www.soundmetrics.com/

[13] (Accessed 1st March, 2017) Sony IMX219PQ. [Online]. Available: http://www.sony-semicon.co.jp/products_en/new_pro/april_2014/imx219_e.html

[14] D. Fortun, P. Bouthemy, and C. Kervrann, “Optical flow modeling and computation: a survey,” *Computer Vision and Image Understanding*, vol. 134, pp. 1–21, 2015.

[15] N. Hurtós, S. Nagappa, X. Cufí, Y. Petillot, and J. Salvi, “Evaluation of registration methods on two-dimensional forward-looking sonar imagery,” in *MTS/IEEE OCEANS-Bergen*, 2013, pp. 1–8.

[16] S. Negahdaripour, “On 3-D motion estimation from feature tracks in 2-D FS sonar video,” *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 1016–1030, 2013.

[17] B. K. Horn and B. G. Schunck, “Determining optical flow: a retrospective,” *Artificial Intelligence*, vol. 59, no. 1, pp. 81–87, 1993.

[18] D. M. Lane, M. J. Chantler, and D. Dai, “Robust tracking of multiple objects in sector-scan sonar image sequences using optical flow motion estimation,” *IEEE Journal of Oceanic Engineering*, vol. 23, no. 1, pp. 31–46, 1998.

[19] A. Spears, A. M. Howard, M. West, and T. Collins, “Determining underwater vehicle movement from sonar data in relatively featureless seafloor tracking missions,” in *Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2014, pp. 909–916.

[20] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “From contours to regions: An empirical evaluation,” in *IEEE Conference on Computer Vision and Pattern Recognition.*, 2009, pp. 2294–2301.

[21] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[22] Y. V. Zakharov and V. Nascimento, “Orthogonal matching pursuit with dcd iterations,” *Electronics Letters*, vol. 49, no. 4, p. 1, 2013.

[23] Y. V. Zakharov, G. P. White, and J. Liu, “Low-complexity RLS algorithms using dichotomous coordinate descent iterations,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3150–3161, 2008.

[24] C. Chailloux, J.-M. Le Caillec, D. Gueriot, and B. Zerr, “Intensity-based block matching algorithm for mosaicing sonar images,” *IEEE Journal of Oceanic Engineering*, vol. 36, no. 4, pp. 627–645, 2011.

[25] (2016, Accessed 13th, December, 2016) Bluefin Robotics Corp. website. [Online]. Available: http://http://www.bluefinrobotics.com//

[26] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[27] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[28] Y. C. Pati, R. Rezaiifar, and P. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *IEEE Conference Record of The Twenty-Seventh Asilomar Conference on Signals, Systems and Computers*, 1993, pp. 40–44.

[29] C. Schretter and H. Niederreiter, “A direct inversion method for non-uniform quasi-random point sequences,” *Monte Carlo Methods and Applications*, vol. 19, no. 1, pp. 1–9, 2013.

[30] C.-C. Wu and Y.-I. Chang, “Approximately even partition algorithm for coding the Hilbert curve of arbitrary-sized image,” *IET Image Processing*, vol. 6, no. 6, pp. 746–755, 2012.

[31] S. Haykin, *Adaptive Filter Theory (2nd Ed.)*. Prentice-Hall, Inc., 1991.

[32] M. R. Lynch and P. Rayner, “A new approach to image registration utilising multidimensional LMS adaptive filters,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 1988. ICASSP-88.*, 1988, pp. 920–923.

[33] P. Elad and A. Feuer, “Recursive optical flow estimation-adaptive filtering approach,” in *IEEE Nineteenth Convention of Electrical and Electronics Engineers in Israel*, 1996, pp. 387–390.

[34] G. Caner, A. M. Tekalp, G. Sharma, and W. Heinzelman, “Local image registration by adaptive filtering,” *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 3053–3065, 2006.

[35] B. Henson and Y. Zakharov, “Local optical-flow estimation for forward looking imaging sonar data,” in *MTS/IEEE OCEANS-Monterey*, 2016, pp. 1–8.

[36] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[37] J. Ren, J. Jiang, and T. Vlachos, “High-accuracy sub-pixel motion estimation from noisy images in Fourier domain,” *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1379–1384, 2010.

[38] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.

[39] J. W. Tukey, *Exploratory data analysis*. Reading, Mass., 1977.

[40] P. H. Eilers and B. D. Marx, “Flexible smoothing with B-splines and penalties,” *Statistical science*, vol. 11, no. 2, pp. 89–102, 1996.

[41] M. Unser, “Splines: A perfect fit for signal and image processing,” *IEEE Signal processing magazine*, vol. 16, no. 6, pp. 22–38, 1999.

[42] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[43] (2017, Accessed 8th, May, 2017) Acquest Subaquatic Geology and Geophysics website. [Online]. Available: http://www.acquest.com.br