

# REAL-TIME VISION BASED AUV NAVIGATION SYSTEM USING A COMPLEMENTARY SENSOR SUITE

**Jonathan Horgan & Daniel Toal**

*Mobile and Marine Robotics Research Centre  
Department of Electronic and Computer Engineering  
University of Limerick, Ireland  
Email: {Jonathan.Horgan,Daniel.Toal}@ul.ie*

**Pere Ridao & Rafael Garcia**

*Computer Vision and Robotics Group  
Institute of Informatics and Applications  
University of Girona, Spain  
Email: {pere, rafa}@eia.udg.es*

**Abstract:** This paper proposes a real-time navigation system for an AUV that takes advantage of the complementary performance of a sensor suite including a DVL, a compass, a depth sensor and altimeter sensors with a feature based motion estimator using vision. To allow for real-time performance of the vision based motion estimator a simple but fast correlation algorithm is used for feature matching. The compass and the depth sensors are used to bound the drift of the heading and depth estimations respectively. The altimeter is required in order to translate the feature displacements measured from the images into the metric displacements of the robot. While the robot must rely on DVL navigation above a certain altitude where vision is useless, DVL measurements can be complemented with higher frequency accurate motion estimates from the vision system when navigating close to the seafloor. *Copyright © 2007 IFAC*

**Keywords:** marine robotics, robot vision, Kalman filtering, robot navigation

## 1. INTRODUCTION

Underwater navigation is an extensive area of research and poses a number of difficult challenges. Accurate navigation systems are vital for all underwater missions for correct registration between sensor and navigation data and also for control and final recovery of the vehicle. Most of the state of the art navigation systems (Kinsey, Eustice et al. 2006) are based on the use of velocity measurements from a Doppler velocity log (DVL) sensor conveniently fused with accurate heading measurements gathered with a gyrocompass. To bound the drift inherent in the system, position fixes from an acoustic transponder network (LBL, USBL, GIB) are commonly used. However, this option raises the mission cost as transponders require deployment prior to the mission (Long Base Line) or a mother ship is necessary (Ultra Short Base Line). This solution also limits the area in which the vehicle can accurately navigate to within the bounds of the transponder network.

Over recent years, computer vision has been the subject of increased interest as a result of improving hardware processing capabilities and the need for more flexible, lightweight and accurate sensor solutions (Horgan and Toal 2006). Many researchers

have explored the possibility of using computer vision as a primary source for AUV navigation. Techniques for implementing computer vision in order to track cables on the seabed for inspection and maintenance purposes have been researched (Balasuriya and Ura 2002; Ortiz, Simo et al. 2002). Station keeping, the process of maintaining a vehicle's pose, is another application that has taken advantage of vision's inherent accuracy and high update rate (Negahdaripour, Xu et al. 1999). Motion estimation from vision is of particular interest for the development of intervention class vehicle navigation (Caccia 2006). Wreckage visualization and biological and geological surveying are examples of many applications that use image mosaicking techniques to acquire a human interpretable view of the ocean floor but it has also been proven as an appropriate means for near seabed vehicle navigation (Garcia, Cufi et al. 2006); (Negahdaripour and Xu 2002).

In this paper we propose a navigation system based on a complementary sensor suite. The navigation is solved by means of an extended Kalman filter (EKF) using a constant velocity model and updated with velocity, heading and depth measurements. When the robot navigates above a predefined boundary altitude, the bottom tracking velocity readings coming from the DVL are used to update the filter. When the robot navigates close to the seafloor, DVL readings are frequently lost, so velocity

measurements are taken from a motion estimator based on image mosaicking techniques instead. If both measurements are available, the DVL and the vision-based velocity estimates are fused using the EKF. In all cases, yaw and depth measurements coming from the compass and the depth cell are used to limit the drift in the corresponding variables. Moreover, the altitude readings are used to convert pixel measurements into metric displacements needed to compute the velocity.

The paper is organized as follows. First, a brief overview of the mosaicking system structure and functionality is presented. The implementation of the extended Kalman filter is then described, detailing the model used. The experimental results gained from the test tank at the University of Girona are reported and discussed. Finally, the conclusions and future work are presented.

## 2. MOSAICKING SYSTEM

The creation of the mosaic is accomplished in the following stages (See Fig. 1). First, a detector of interest points is used to select scene features in the current image that can be reliably detected and matched in the next image of the sequence after the camera location has changed. Image correspondences are then located between incremental images using a correlation-based matching approach. The system identifies the points that describe the dominant motion of the image by means of a robust outlier-detection algorithm. Once the pairs of features describing the dominant motion have been selected, a 2D planar transformation matrix relating the coordinates of both images is computed. Finally, the registered images can be merged onto a composite mosaic image and an estimation of the vehicle linear velocities  $u$  and  $v$  and angular velocity  $r$  can be computed and outputted to the EKF.

### 2.1 Selection of Interest Points

The first step of the mosaicking algorithm is the selection of adequate interest points in the present image to be matched in the next frame. An interest point is a point in an image which has a well-defined position and can be robustly detected. The feature detector implemented searches for small zones presenting high spatial gradient information in more than one direction, as performed by the Harris corner detector (Harris and Stephens 1988). To do this, the image is convolved with two directional high-pass filters (in the x and y directions). The areas with the highest gradient in both directions are selected as interest points.

### 2.2 Interest Point Matching

Once the interest points in the reference image  $I^{(k)}$  have been obtained, the algorithm searches for the candidate matches in the next image  $I^{(k+1)}$ . The

matching process is accomplished in the following way (see Fig. 1): For every interest point  $m_j^{(k)}$  in image  $I^{(k)}$  a correlation is performed by convolving a small search window over image  $I^{(k+1)}$ . This search window centered at  $m_j^{(k+1)}$  in  $I^{(k+1)}$  is translated from its position  $m_j^{(k)}$  in  $I^{(k)}$  taking into consideration the previous homograph calculated from the mosaicking system. This helps reduce the size of the search window and improve system performance. A similarity measurement is computed for each feature by means of a correlation score.

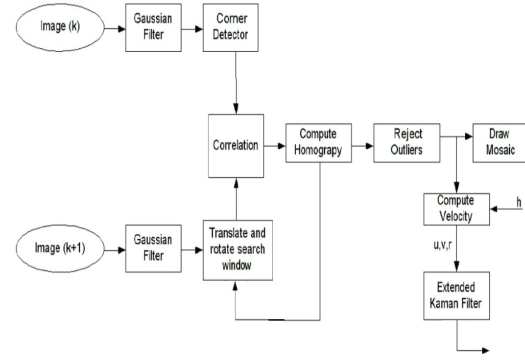


Fig. 1. Mosaicking Process

### 2.3 Estimation of Dominant Motion through Outlier Rejection

After the correspondences have been solved, a set of displacement vectors relating the features of two images of the sequence is obtained. Every vector relates the coordinates of the same feature in both images. Our aim is now to recover the apparent motion of the camera from these features. This can be done by computing a 2D transformation matrix  $\mathbf{H}$  which relates the coordinates of a scene point in frame  $I^{(k+1)}$  with the coordinates of the same point in the previous frame  $I^{(k)}$ ,

$$\text{i.e.} \quad \tilde{\mathbf{m}}^{(k)} = \mathbf{H} \cdot \tilde{\mathbf{m}}^{(k+1)} \quad (1)$$

$$\text{where,} \quad \tilde{\mathbf{m}}^{(k)} = (x_i^{(k)}, y_i^{(k)}, 1)^T \quad (2)$$

$$\text{and} \quad \tilde{\mathbf{m}}^{(k+1)} = (x_i^{(k+1)}, y_i^{(k+1)}, 1)^T \quad (3)$$

denote a correspondence point in two consecutive images; and the symbol  $\sim$  indicates that the points are expressed in homogeneous coordinates. The matrix that performs this transformation is known as “homography”, and can be computed by singular value decomposition (SVD) if 2 or more pairs of matchings are available (Cufi, Garcia et al. 2002). The *similarity* transformation described by matrix  $\mathbf{H}$  has 4 degrees of freedom as the test vehicle has been designed to be passively stable in pitch and roll (its center of gravity is below the center of buoyancy). For this reason, rolling and pitching motion of the vehicle are very small, and therefore better results are obtained with a simple motion model.

Although the correlation system is devoted to the matching procedure, some false matches (known as

outliers) could still appear among the correct correspondences. For this reason, a robust estimation method has to be applied. The Least Median of Squares (LMedS) algorithm is used for finding the matrix  $\mathbf{H}$  that minimizes the median of the squared residuals.

$$M_{err} = \text{med}(d^2(\tilde{\mathbf{m}}_j^{(k)}, \mathbf{H}\tilde{\mathbf{m}}_j^{(k+1)})) + (d^2(\tilde{\mathbf{m}}_j^{(k+1)}, \mathbf{H}^t\tilde{\mathbf{m}}_j^{(k)})) \quad (4)$$

where,  $d^2(\tilde{\mathbf{m}}_j^{(k)}, \mathbf{H}\tilde{\mathbf{m}}_j^{(k+1)})$  is the square Euclidean distance from a point  $\tilde{\mathbf{m}}_j^{(k)}$ , defined on image  $I^{(k)}$ , to the projection on the same image plane of its correspondence  $\tilde{\mathbf{m}}_j^{(k+1)}$ . Hence, the error is defined by the distance of a point to the projection of its correspondence. In our case, with only three motion parameters estimated (x and y displacement and yaw rotation), the number of samples required to have a probability of 0.99 that at least one sample has no outliers is 7 assuming 20% of outliers in the data. Hence, the algorithm can be easily implemented in real-time.

#### 2.4 Mosaic Construction and velocity estimation

The process of mosaic construction selects the initial image of the sequence as a base frame. The mosaic coordinate system is placed at the origin of this reference frame. When image  $I^{(k+1)}$  has to be added to the mosaic, a 2D planar transformation  ${}_k\mathbf{H}_{k+1}$  provides its best fitting with respect to the previous image. The matrix  $\mathbf{H}$ , defined for each image, is also converted into linear vehicle velocity measurements  $u$  and  $v$  and rotation velocity  $r$  to be inputted into the extended Kalman filter. The linear velocity calculations require knowledge of the intrinsic parameters of the camera gained from camera calibration, periodic altimeter readings averaged from the 3 cosine corrected altitude measurements from the DVL and by noting the sampling time between frames. A pixel to metric conversion can then be calculated by applying the geometric law of perspective relation (Cufi, Garcia et al. 2002). Incremental measure  $d$  between two images can be decomposed into  $dx$  and  $dy$ , measured in pixels with respect to the coordinate system of the previous image.

$$D_x = \frac{d_x \cdot Z}{f} \quad D_y = \frac{d_y \cdot Z}{f} \quad (5)$$

where,  $D_x$  and  $D_y$  are the components of the incremental motion from image  $I^{(k)}$  to  $I^{(k+1)}$  in metric,  $Z$  is the camera altitude and  $f$  is the camera focal length. With knowledge of the time between frames  $D_x$  and  $D_y$  can be converted into vehicle velocities. The rotation parameter in each homography is converted into radians/sec to gain the angular velocity  $r$ . The differences in elevation of the floor relief are assumed to be negligible with respect to the navigation altitude of the vehicle.

#### 4. EXTENDED KALMAN FILTER

Due to the non-linear nature of vehicle motion the extended Kalman filter (EKF) has been chosen as an appropriate filter for solving the navigation problem. The EKF is used to fuse sensor information from the mosaicking system, DVL, depth sensor and compass to provide better navigational estimates (see Fig. 2). The following vector has been formulated to represent the estimated state of the vehicle:

$$\hat{\mathbf{x}} = [\hat{x} \ \hat{y} \ \hat{z} \ \hat{\psi} \ \hat{u} \ \hat{v} \ \hat{w} \ \hat{r}]^T \quad (6)$$

The first four elements in equation (6) represent the position and yaw of the vehicle in world coordinate frame and the other four elements represent the linear and angular velocities in the vehicle coordinate frame. This state estimation is associated with a covariance that is defined by the  $\mathbf{P}$  matrix.

$$\mathbf{P} = E([\mathbf{x}_k - \hat{\mathbf{x}}_k][\mathbf{x}_k - \hat{\mathbf{x}}_k]^T | \mathbf{Z}) \quad (7)$$

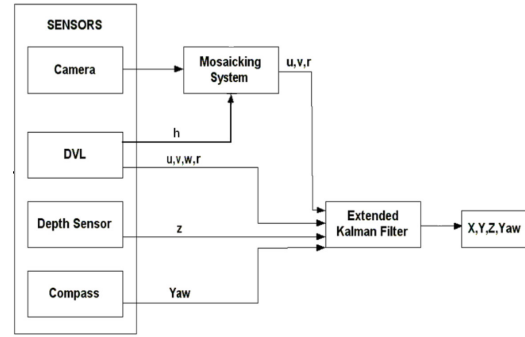


Fig. 2. System structure

#### 4.1 Initialization

To initialize the state vector the first valid measurement of the compass is taken as the vehicle yaw ( $\psi$ ). The appropriate value of uncertainty for the yaw is also inserted into the  $\mathbf{P}$  matrix. Without loss of generality, the other values in the state vector can be initialized to 0 thus assuming the vehicle is starting from a static position at the origin of the world coordinate frame.



Fig. 3. ICTINEU<sup>AUV</sup> testing vehicle

#### 4.2 Prediction

A four-degree of freedom constant velocity kinematics model is employed, as the vehicle is passively stable in pitch and roll. The following equation describes the state prediction model:

$$\hat{\mathbf{x}}_{k+1}^- = \begin{bmatrix} x \\ y \\ z \\ \psi \\ u \\ v \\ w \\ r \end{bmatrix}_{(k+1)} = \begin{bmatrix} x_k + u_k t \cos(\psi_k) - v_k t \sin(\psi_k) \\ y_k + u_k t \sin(\psi_k) + v_k t \cos(\psi_k) \\ z_k + w_k t \\ \psi_k + r_k t \\ u_k \\ v_k \\ w_k \\ r_k \end{bmatrix} \quad (8)$$

The unmodeled perturbations are represented by a zero mean white Gaussian noise in acceleration with covariance  $\mathbf{Q}$  propagating to the velocity and position through integration. The uncertainty in this prediction is updated after each measurement using the common EKF equations:

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T \quad (9)$$

where,  $\mathbf{W}$  is the Jacobian matrix of the partial derivatives of the model with respect to process noise and  $\mathbf{A}$  is the Jacobian matrix of the partial derivatives of the model with respect to state vector.

#### 4.3 DVL, compass and depth sensor update

The update step can merge information from multiple sensors depending on which is outputting the latest information. The DVL provides two types of velocity measurements, water track velocity and bottom track velocity. In the absence of currents, as in the experimental setup used for this work, the through water velocity can be considered as another robot velocity measurement, with less certainty than the bottom track velocity. For both measurements, the DVL provides a status parameter indicating the quality of the data, which allows us to neglect inaccurate readings. A pressure sensor is also used to acquire depth measurements while a compass provides measurements of the yaw. The generic measurement vector for all these sensors is as follows:

$$\mathbf{z}_D = [u_w \ v_w \ w_w \ u_b \ v_b \ w_b \ z \ \psi]^T \quad (10)$$

where, subscript  $w$  and  $b$  represent water and bottom tracking velocity respectively. As all these measurements are direct observations of the state vector a linear measurement model is implemented as follows:

$$\hat{\mathbf{z}}_D = \mathbf{H}_D \hat{\mathbf{x}}_{k+1}^- \quad (11)$$

$$\mathbf{H}_D = \begin{bmatrix} 0_{3 \times 4} & \mathbf{I}_{3 \times 3} & 0_{3 \times 1} \\ 0_{3 \times 4} & \mathbf{I}_{3 \times 3} & 0_{3 \times 1} \\ 0_{1 \times 2} & 1 & 0 & 0_{1 \times 4} \\ 0_{1 \times 2} & 0 & 1 & 0_{1 \times 4} \end{bmatrix} \quad (12)$$

This is the generic form of the  $\mathbf{H}_D$  matrix, however, depending on the measurements available and the status parameter the  $\mathbf{H}_D$  matrix can change accordingly by adding or removing rows. Matrix  $\mathbf{R}_D$  is the covariance associated with the measurement and as such is also subject to change depending on the available measurements.

$$\mathbf{R}_D = \text{diag}(\sigma_{u_w}^2 \ \sigma_{v_w}^2 \ \sigma_{w_w}^2 \ \sigma_{u_b}^2 \ \sigma_{v_b}^2 \ \sigma_{w_b}^2 \ \sigma_z^2 \ \sigma_\psi^2) \quad (13)$$

The typical EKF equations are used to update the state estimate. Equation (13) computes the Kalman gain and equation (14) uses this gain, the new measurements and the previous state estimate to update the state. The uncertainty in this estimate is updated using equation (15).

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (14)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (15)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (16)$$

#### 4.4 Mosaic measurement update

For each new frame considered by the mosaicking system a new measurement for the velocities  $u_c$ ,  $v_c$  and  $r_c$  in the camera coordinate frame are provided. The following equations are the measurement vector and measurement model respectively:

$$\mathbf{z}_M = [u_c \ v_c \ r_c]^T \quad (17)$$

$$\hat{\mathbf{z}}_M = \mathbf{H}_M \hat{\mathbf{x}}_{k+1}^- \quad (18)$$

For ICTINEU<sup>AUV</sup>, while the robot frame is located at the center of the DVL position, the camera is not centered with respect to the robot  $z$ -axis. Hence, if the robot rotates around while maintaining its position, the image mosaicking system would detect a linear motion due to the camera displacement. Hence, the camera position ( $X_c$ ,  $Y_c$ ) has to be used within the  $\mathbf{H}_M$  matrix to account for this offset.

$$\mathbf{H}_M = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & -Y_c \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & X_c \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (19)$$

$$\mathbf{R}_M = \text{diag}(\sigma_{u_c}^2 \ \sigma_{v_c}^2 \ \sigma_{r_c}^2) \quad (20)$$

The EKF equations (14, 15 and 16) are again used to perform the update.

## 5. EXPERIMENTAL RESULTS

A data set was collected using the ICTINEU<sup>AUV</sup> (see Fig. 3) in the test tank at the University of Girona (see left of Fig. 5.). A printed poster of a seabed environment was placed at the bottom of the tank in an attempt to simulate real seabed conditions (see Fig. 5 left). All sensor data collected was processed off line.

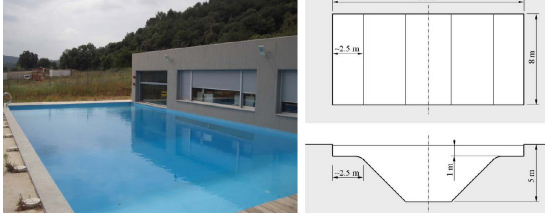


Fig. 4. Test tank facility at the University of Girona

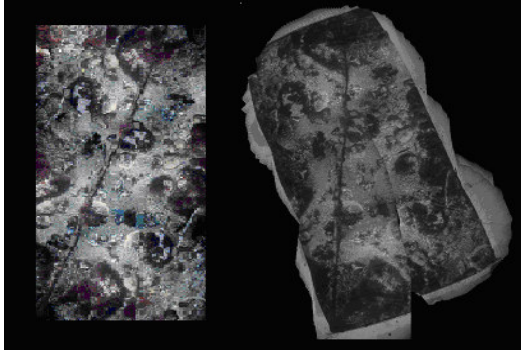


Fig. 5. *Left* –Printed poster of seabed. *Right* – Image mosaic of the poster in the test tank constructed using 1080 images and rendered from the last to the first image

During the experiment the robot was manually piloted to complete a loop trajectory over an approximate area of  $25\text{m}^2$ . During the 3mins 42 seconds needed to follow the trajectory, images were grabbed at a rate of approximately 10Hz and synchronized with compass, depth sensor, altitude and DVL measurements. A subset of these images were used to create the mosaic and estimate velocities, in this case one in every two (5Hz) .

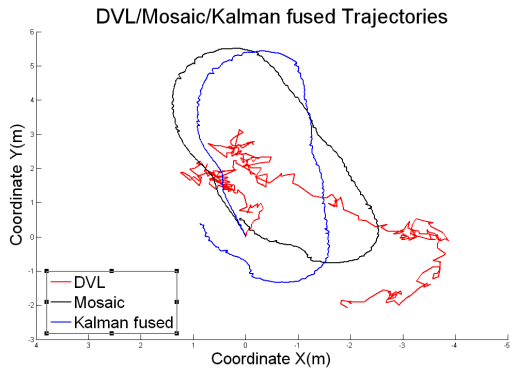


Fig. 6. Trajectory estimated using only mosaic measurements, only the DVL, and fusing all the sensors through an EKF.

Fig. 5 (right) shows a rendering of the image mosaic based on the raw mosaic motion estimations (without the Kalman Filter) illustrating its accuracy. The black trajectory in fig.6 was estimated using only the mosaic measurements. It can be appreciated that the trajectory loop is totally closed. By carefully inspecting fig.6 it is clear that after the last turn before closing the loop (bottom right corner), and mainly due to the drift accumulated in the mosaic heading, the trajectory gets closer to the initial track than it should. For this reason several objects appear duplicated and the bottom boundary becomes discontinuous.

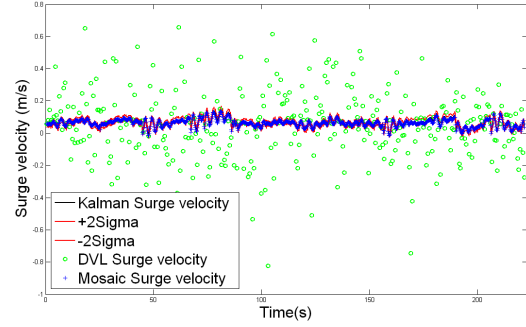


Fig. 7. Surge velocity estimation illustrating the  $2\sigma$  bound and the measurements from both the DVL and the mosaicking system

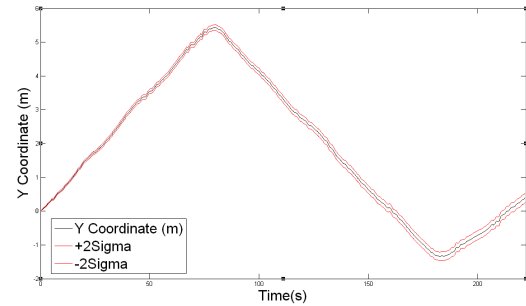
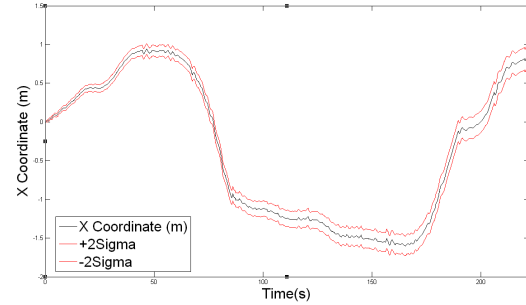


Fig. 8. X and Y Position estimates detailing the  $2\sigma$  bound.

Fig.6 also shows dead reckoning estimation using only the DVL measurements and in this case the trajectory is very inaccurate. This is mainly due to the poor acoustic conditions of the small test tank as well as the slow motion of the robot. This can be appreciated in fig.7, which shows the difference in the noise of the DVL velocity measurements and the mosaic velocity measurements. For this reason, the filter was tuned to put more trust in the mosaic

measurements and the model rather than in the DVL velocity measurements. On the other hand, yaw measurements of the compass provide absolute yaw fixes that bound the drift in the heading. While DVL measurements are of little help (in water tank conditions), depth and yaw give absolute measures of the vector state, and thus improve the estimated trajectory. As a result, the trajectory estimated by the EKF is better. Unfortunately, at this point, our software does not allow us to render the mosaic using the trajectory estimated by the EKF. Hence, the only way to validate the trajectory (fig.6 and 8) is to manually check that the misalignment observed in the rendered mosaic is in agreement with the distance between the initial and the last tracks, where the mosaic detected a false intersection.

## 6. CONCLUSIONS

This paper presents preliminary work towards a real-time navigation system by integrating an image mosaicking motion estimator with navigation sensors commonly available in today's UUVs. To allow for real-time motion estimation, the feature matching process implemented on successive images is performed through a simple but fast correlation algorithm. Even though this feature correlator is known to present poor performance during fast turns, in the water tank conditions where the navigation system was tested, the accuracy of the velocity estimates from the image mosaicking system significantly outperformed the DVL measurements. On the other hand, absolute compass and depth readings bounded the drift in the heading and depth that would be obtained otherwise. While not possible to demonstrate in lab conditions, we believe that an interesting characteristic of the proposed navigation system comes from the complementary attributes of the DVL and the vision system. While the DVL is able to provide velocity measurements with a moderate update rate at altitudes where vision is useless, vision allows for high update rates, very close to the bottom, with significant accuracy.

## 7. FUTURE WORK

Future work will improve the feature correlator by taking advantage of the heading estimations available from EKF. Another interesting improvement consists of exploiting the fact that interest points are re-observed several times before they disappear from the camera view. Hence, it is possible to include them in the vector state while they are visible. Since their position is correlated with the robot position, it is expected that every observation of the interest points will improve our estimate of the robot position.

As implemented, the ability to accurately convert from pixel to metric measurements for vehicle motion estimation is dependant on reliable altitude measurements. Altimeter sonar's generally have a minimum blanking range of between 0.5 and 1 meter

whereas in the case of computer vision, accuracy improves with decreasing range. A potential solution to extend the applicability of the vision based navigation system in near-seabed/intervention operations and overcome the limitations of current altitude sensors blanking range, is to employ a newly developed 2MHz short range wide angle ultrasonic transceiver with a minimum range of less than 0.02 meters (Nolan and Toal 2006) and this will be investigated.

## REFERENCES

- Balasuriya, A. and T. Ura (2002). "Vision-based underwater cable detection and following using AUVs". *Oceans '02 MTS/IEEE*.
- Caccia, M. (2006). "Laser-Triangulation Optical-Correlation Sensor for ROV Slow Motion Estimation." *IEEE Journal of Oceanic Engineering* 31(3): 711-727.
- Cufi, X., R. Garcia, et al. (2002). "An approach to vision-based station keeping for an unmanned underwater vehicle". *IEEE/RSJ International Conference on Intelligent Robots and System*, 2002.
- Garcia, R., X. Cufi, et al. (2006). "Constructing Photo-mosaics to Assist UUV Navigation and Station-keeping (Chapter 9)". *Robotics and Automation in the Maritime Industries*: 195-234.
- Harris, C. and M. Stephens (1988). "A combined corner and edge detector". *4th Alvey Vision Conference*.
- Horgan, J. and D. Toal (2006). "Vision Systems in the Control of Autonomous Underwater Vehicles". *IFAC Conference of Manoeuvring and Control of Marine Craft (MCMC 06)*.
- Kinsey, J. C., R. M. Eustice, et al. (2006). "Survey of underwater vehicle navigation: Recent advances and new challenges". *IFAC Conference of Manoeuvring and Control of Marine Craft (MCMC 06)*.
- Negahdaripour, S. and X. Xu (2002). "Mosaic-based positioning and improved motion-estimation methods for automatic navigation of submersible vehicles." *IEEE Journal of Oceanic Engineering*, 27(1): 79-99.
- Negahdaripour, S., X. Xu, et al. (1999). "Direct estimation of motion from sea floor images for automatic station-keeping of submersible platforms." *IEEE Journal of Oceanic Engineering* 24(3): 370-382.
- Nolan, S. and D. Toal (2006). "An Ultrasonic Sensor for Close Range Collision Avoidance on AUVs". *IFAC Conference of Manoeuvring and Control of Marine Craft (MCMC 06)*.
- Ortiz, A., M. Simo, et al. (2002). "A vision system for an underwater cable tracker." *International Journal of Machine Vision and Applications* 13(3): 129-140.