# Stereo EKF Pose-based SLAM for AUVs

Markus Solbach[1], Francisco Bonin-Font[2], Antoni Burguera[2], Gabriel Oliver[2], and Dietrich Paulus[1]

[1]Computational Visualistics Group, University Koblenz-Landau, Koblenz (56070) (Germany)
[2]Systems, Robotics and Vision Group, University of the Balearic Islands (UIB), Palma de Mallorca (07122) (Spain)

## I. Introduction

Nowadays, *Remotely Operated Vehicles* (ROVs) are commonly used in a variety of scientific or industrial applications, such as surveying, sampling, rescue or industrial infrastructure inspection and maintenance. However, *Autonomous Underwater Vehicles* (AUVs) are being progressively introduced to run highly repetitive, long or hazardous missions, reducing notably the operational costs and the complexity of human and material resources.

The localization task becomes a crucial issue in AUVs since significant errors in pose can lead to the programmed mission failure. The motion of an underwater vehicle with 6 *Degrees of Freedom* (DOF) can be estimated, for instance, (a) using inertial sensors, (b) using odometry, computed via cameras or acoustic sensors, or, (c) fusing all these sensorial data in *Extended Kalman Filters* (EKF) or particle filters, to smooth trajectories and errors [7]. However, all these methods are, to a greater or lesser extent, prone to drift, being necessary a periodical adjustment of the vehicle pose to minimize the accumulated error. *Simultaneous Localization And Mapping* (SLAM) [3] techniques constitute the most common and successful approach to perform precise localization by identifying areas of the environment already visited by the robot. Traditionally, SLAM has been developed using range sensors, but cameras outperform range sensors in temporal and spatial resolutions.

Imaging natural sub-aquatic environments has additional difficulties not present in land: the light attenuation, flickering, scattering, the lack of man made structured frameworks, and the subsequent difficulty to register images, that is, to identify the same scene visualized from different viewpoints, maybe under different environmental conditions, with partial or total overlap, and taken at different time instants.

The literature is scarce in efficient visual SLAM solutions especially addressed to underwater robots and tested in field robotic systems. Many of them particularize the approach commonly known as EKF-SLAM [3], correcting the odometry with the results of an image registration process in an EKF context. These systems normally include the vehicle pose and the landmarks in the state vector, correcting continuously the vehicle trajectory and the whole map [10]. However, this approach presents two major problems: (a) the computational cost increases significantly with the number of the detected landmarks, and (b) the linearization errors inherent to the EKF. Eustice *et al* [4] adopted a *Delayed State Filter* (DSF) to alleviate both problems.

EKF-SLAM approaches can be *pose-based*, if each iteration of the filter gives in the state vector a set of successive robot poses with respect to an external fixed global frame, or *trajectory-based*, if the state vector contains the successive robot relative displacements from point to point of the trajectory. The trajectory based approach reduces the EKF linearization errors with respect to pose based approaches but, contrarily to the later, it does not scale well for large environments, since the Jacobian of the observation function is non-zero with respect to all intermediate elements between two poses closing a loop [5]. Although the trajectory-based schema can be adopted to abate EKF linearization errors [2], it is more suitable for low and mid scale missions.

This paper presents a stereo pose-based EKF-SLAM approach, with the next relevant characteristics: a) it is a generic solution for vehicles with up to 6DOF ([$x, y, z$,roll, pitch, yaw]), so especially useful in AUV; it is feed with pure 3D data computed only from stereo vision; all orientations involved in the approach are represented in the quaternion space to avoid filtering errors due to singularities, b) the vector state contains only the set of robot global poses, keeping the sparsity of the covariance matrix at each iteration; the computational resources needed are drastically reduced with respect other EKF approaches that include the landmarks in the state vectors; c) it pioneers the adaptation of the well known Perspective N-Point problem (PNP) [1] to the image registration process underwater, framing it in such a stereo EKF SLAM approach; the algorithm performs robustly two tasks in one shot, firstly, it confirms or it rejects the existence of overlap between two stereo pairs (i.e. if both views represent a loop closing) and, in case, there is a coincidence, it calculates the camera relative transformation, in translation and orientation, between the two poses at which both views were taken; these transformations are later used as the measurements to correct the predictions in the EKF. d) the implementation has been published in a public repository (https://github.com/srv/6dof_stereo_ekf_slam) to facilitate further research and development in this area.

## II. 3D Transformations

### A. Composition

One of the key targets of this work is modeling, for 6DOF and in the quaternion space, the classical *composition* ($\oplus$) and *inversion* ($\ominus$) transformations, described by Smith *et al* [11] in the context of stochastic mapping, and deriving their Jacobians.

Both operations define a transformation in translation and rotation. The $\oplus$ operation permits accumulating a pose transform $Y$ (translation, $[x^Y, y^Y, z^Y]$ and rotation in roll, pitch and yaw, represented as a quaternion $\hat{q}^Y = [q_w^Y, q_1^Y, q_2^Y, q_3^Y]$) to a current global pose $X$ (position $[x^X, y^X, z^X]$ and its quaternion orientation $\hat{q}^X = [q_w^X, q_1^X, q_2^X, q_3^X]$).

Let us define $X_+$ as the global pose obtained from the composition between $X$ and $Y$. $X_+ = X \oplus Y = [X_+^t, X_+^r]$, where

$$X_+^t = [x^X, y^X, z^X, 1] + A^X \cdot [x^Y, y^Y, z^Y, 1] \quad (1)$$

, being $A^X$ the rotation matrix obtained from $\hat{q}^X$ and, $X_+^r = \hat{q}^X * \hat{q}^Y$, where the operator $*$ denotes the product of quaternions.

The covariance of the composition function $f_\oplus = X \oplus Y$ is:

$$C_+ = J_{1\oplus} \cdot C^X \cdot J_{1\oplus}^T + J_{2\oplus} \cdot C^Y \cdot J_{2\oplus}^T \quad (2)$$

, where $C^X$ and $C^Y$ are the corresponding covariances of $X$ and $Y$, $J_{1\oplus} = \frac{\partial f_\oplus}{\partial X}|_{\hat{X},\hat{Y}}$ and $J_{2\oplus} = \frac{\partial f_\oplus}{\partial Y}|_{\hat{X},\hat{Y}}$, being $\hat{X}$ and $\hat{Y}$ the mean of the $X$ and $Y$ variables.

### B. Inversion

The operation ($\ominus$) returns the *inverse* of a given transformation in position and orientation. Let us denote $X = [t, \hat{q}^X]$, being, $t = (x^X, y^X, z^X)$ and $\hat{q}^X = (q_w^X, q_1^X, q_2^X, q_3^X)$ a global pose with 6DOF. $X$ can also be represented as a matrix,

$$\begin{array}{cccc} \vec{n} & \vec{o} & \vec{a} & \vec{p} \\ \begin{pmatrix} & A & & t \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \quad (3)$$

, where $A$ is the $3 \times 3$ rotation matrix obtained from $\hat{q}^X$.

Let us denote the inverse of $X$ as $f_\ominus = \ominus X = [-\vec{n} \circ \vec{p}, -\vec{o} \circ \vec{p}, -\vec{a} \circ \vec{p}, \hat{q}^{X(-1)}]$, where $\circ$ represents the dot product and $\hat{q}^{X(-1)}$ is the quaternion result of inverting $\hat{q}^X$.

The covariance of $f_\ominus = \ominus X$ is:

$$C_- = J_\ominus \cdot C^X \cdot J_\ominus^T \quad (4)$$

, being $J_\ominus = \frac{\partial f_\ominus}{\partial X}|_{\hat{X}}$.

## III. Image Registration

The image registration process is in charge of verifying if two stereo images close a loop, that is, if they have a certain overlap, although they are taken at different time instants, at different view points, at different height, or even with different environmental conditions.

Algorithm 1 describes the main steps of this process.

***Line 1***: finds and matches image features between $S_l$ and $S_r$, applying RANSAC to eliminate outliers, and stores them in $F_l$ and $F_r$. ***Line 2*** finds image features in $I_l$ and stores them in $F_t$. ***Line 3*** performs a feature matching between $F_l$ and $F_t$ refined by RANSAC. If the number of features matched between $F_l$ and $F_t$ is greater than a certain threshold, then there is a loop closing. Otherwise, it returns an error. ***Line 4*** updates the features in $F_r$ that remain as inliers after the matching between $F_l$ and $F_t$, to be in line with the inliers matching between $F_l$ and $F_t$. ***Line 5*** computes the 3D points coordinates, using the stereoscopy principle, corresponding to the remaining inliers in $F_l$ and $F_r$, and stores them in $P_{3D}$. ***Line 6*** solves the Perspective N-Point problem (PNP), returning a pose transformation $[R, t]$ between $S_l$-$S_r$ to $I_l$-$I_r$ that minimizes the error of reprojecting the 3D points stored in $P_{3D}$ onto the 2D features of the image $I_l$. The PNP-problem is widely discussed and can be found in the literature formulated in multiple solutions. This technique is applied in a wide range of applications such as *object recognition* or *structure from motion* [8].

**Algorithm 1:** Image Registration

**input** : Current Stereo Image pair $S_l$ (left frame), $S_r$ (right frame) and Recorded Stereo Image $I = (I_l, I_r)$ candidate to close a loop with $S_l$ and $S_r$
**output**: 3D Transformation $[R, t]$
**begin**
1 $[F_l, F_r] \leftarrow$ `stereoMatching` $(S_l, S_r)$;
2 $F_t \leftarrow$ `findFeature` $(I_i)$;
3 **if** `match` $(F_l, F_t)$ == *true* **then**
4 $[F_l, F_r] \leftarrow$ `updateFeature` $(F_l, F_r)$;
6 $P_{3D} \leftarrow$ `calc3DPoints` $(F_l, F_r)$;
7 $[R, t] \leftarrow$ `solvePnPRansac` $(F_t, P_{3D})$;
*return* $[R, t]$
**else**
8 *return error*;

## IV. Stereo Pose-based EKF-SLAM

The localization module performs a pose-based stereo SLAM approach in an EKF context. The Kalman state vector $\chi$ contains a successive set of robot poses expressed with respect to a global static frame, in the form of $X = [t, q_p]$ (position in 3D and a quaternion representing an orientation in 3 axis). The initial state of $\chi = (0, 0, 0, 1, 0, 0, 0)$ (position= (0,0,0), and an orientation of 0 in all axis). The covariance $C$ of the state vector is initially set to a $7 \times 7$ zero-matrix. The approach has 3 main stages, the Prediction step, the State Augmentation step and the Update step.

During the prediction stage, the vehicle motion is estimated by a stereo visual odometer, in the form of $Y_o = [t, q_o]$ (translation in 3D and a rotation in 3D) with a $7 \times 7$ covariance matrix $C_0$. The predicted pose is $X_p = X \oplus Y_o$ with an associated $7 \times 7$ matrix covariance $C_t^+$ calculated as detailed in section II-A. Then, $\chi$ is augmented with $X_p$, giving rise to the prediction function $f_p(\chi, Y_o) = [\chi, X_p]$. The covariance $C$ of the state vector is also augmented according to: $C^+ = J_c C J_c^T + J_o C_o J_o^T$, being $J_c = \frac{\partial f_p(\chi, Y_o)}{\partial \chi}|_{\hat{\chi}}$ and $J_o = \frac{\partial f_p(\chi, Y_o)}{\partial Y_o}|_{\hat{\chi}}$. After $n$ iterations, the length of the state vector will be $n * 7$.

The update step is in charge of correcting the predicted motion using the loop closings detected between the image grabbed at the current filter iteration and all the images grabbed previously. When the stereo image grabbed at the current state is registered with an image captured and stored during any other previous state of the covered trajectory, the system is providing an additional pose constraint between both camera positions. This constraint can be compared with

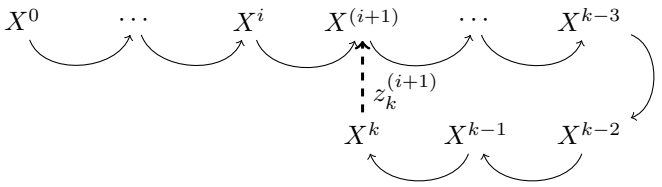


Fig. 1: A loop closing (dashed arrow) in the state vector (black arrows).

the transformation between both positions giving by a pure composition of the corresponding poses stored in the filter state. Figure 1 illustrates the idea. $X^0$, $X^1$, $\cdots$, $X^k$ represent the successive absolute poses of the vehicle along its trajectory stored in the state vector. After $k$ iterations, the image grabbed at iteration $i+1$ is registered with the current image at $X^k$, so both close a loop. The result of this image registration process is $z_k^{(i+1)}$, a relative transformation from $X^{(i+1)}$ to $X^k$ which depends only on the image registration process. The observation function for one loop closing is defined as $h^k = \ominus X^k \oplus X^{(i+1)}$, which is the relative transformation between both registered states according to the successive filter estimates. The Kalman innovation for one loop closing is defined as $\Upsilon_k = h^k - z_k^{(i+1)}$. The observation vector $h$, the measurements vector and the innovation vector $\Upsilon$ will have as many rows as loop closings are found with the current image. The observation matrix $H = \frac{\partial h}{\partial \chi^+}\Big|_{\hat{\chi}^+}$ will have as many rows as loop closings, as many columns as elements in the state vector, and all positions not corresponding to those states involved in each loop closing will be 0:

$$H = \begin{bmatrix} \mathbf{0} & \frac{\partial h^1}{\partial X^i} & \mathbf{0} & \dots & \mathbf{0} & \frac{\partial h^1}{\partial X^k} \\ \dots & & & & & \\ \mathbf{0} & \mathbf{0} & \frac{\partial h^n}{\partial X^j} & \dots & \mathbf{0} & \frac{\partial h^n}{\partial X^k} \end{bmatrix} \tag{5}$$

where $n$ is the number of loop closings registered with the current image and $X^i, X^j$ represent two of those $n$ registered states.

Due to the nature of the quaternions (completely different quaternions can represent the same orientation and vice-versa), the pure subtraction that defines the innovation might not reflect correctly how is, or how the innovation should be when two orientations are very close. For this reason, our approach calculates the innovation subtracting the translation vector of $h^k$ and $z_k^{(i+1)}$ and subtracting the modules of the corresponding quaternions to get the difference of orientations: $|q_z i| - |q_h i|$, where $q_z i$ represents the quaternion corresponding to the orientation of the $i_{th}$-measurement and $q_h i$ represents the quaternion of the corresponding $i_{th}$ observation.

From now on, by applying the Kalman equations, one can obtain an updated state vector $\chi^+$ and its updated covariance.

$$S = H \cdot C^+ \cdot H^T + R, \tag{6a}$$
$$K = (C^+ \cdot H^T)/S, \tag{6b}$$
$$\chi^+ = \chi + K \cdot \Upsilon, \tag{6c}$$
$$C_u = (1 - K \cdot H) \cdot C^+, \tag{6d}$$

| *Noise Level* | *1* | *2* | *3* | *4* | *5* | *6* |
|---|---|---|---|---|---|---|
| *Noise Covariance* | *0* | *3e-9* | *9e-9* | *3e-8* | *5e-7* | *3e-6* |
| *Odom. error ∅* | 0.038 | 0.417 | 0.494 | 0.806 | 2.614 | 6.898 |
| *EKF error ∅* | 0.027 | 0.282 | 0.285 | 0.309 | 0.590 | 0.953 |
| *Improv. (%)* | 28.9 | 32.3 | 42.3 | 61.6 | 77.4 | 86.1 |

TABLE I: Odometry and EKF-SLAM trajectory mean errors (∅). Error units are meters per traveled meter.

where $R$ is the measurements covariance matrix and $C_u$ represents the updated state vector covariance ($C$).

## V. Experimental Results

Experiments were conducted with the Fugu-C platform, a low-cost mini-AUV developed at the University of the Balearic Islands. The sensor suit for this vehicle includes two stereo rigs, one looking forward and another one looking downwards, a MEMS Inertial Measurement Unit and a pressure sensor. Fugu-C works with ROS [9] as middleware, and thanks to the ROS-bag technology, missions were recorded on-line and reproduced offline with exactly the same conditions as the original mission. A stereo visual odometer based on LibViso2 [6] was used to compute the first estimates of the robot displacement. Visual odometry data was provided at 10Hz and all routes were traveled at a constant depth. The first experiments with the robot were conducted in a water tank 7 meters long, 4 meters wide and 1.5 meters depth, whose bottom was covered with a printed digital image of a real seabed. The trajectory ground truth was computed by registering each image captured online with the whole printed digital image, which was previously known.

The example shown in this section corresponds to a sweeping task performed in the tank. In order to assess the performance of the SLAM approach with different levels of error and drift in the visual odometry, the results of the stereo odometer were corrupted with different levels of additive zero mean Gaussian noise. In total six noise levels were tested 20 times to obtain significant statistical results. The noise covariance ranges from $[\Sigma_x, \Sigma_y, \Sigma_z, \Sigma_{qw}, \Sigma_{q1}, \Sigma_{q2}, \Sigma_{q3}] = [0, 0, 0, 0, 0, 0, 0]$ (noise level 1) to $[\Sigma_x, \Sigma_y, \Sigma_z, \Sigma_{qw}, \Sigma_{q1}, \Sigma_{q2}, \Sigma_{q3}] = [3e-6, 3e-6, 3e-6, 3e-6, 3e-6, 3e-6, 3e-6]$ (noise level 6). In order to have a quantitative measure of the SLAM quality, the trajectory error was defined as the difference between the ground truth and the corresponding estimate given by the odometry and by the EKF, divided by the length of the trajectory. Calculated like this, the obtained error units are meters per traveled meter. This technique permits the direct comparison of results obtained in different experiments.

Table I shows how the presented EKF-SLAM approach improves the odometric estimates since the mean of the trajectory error with respect to the ground truth are always clearly smaller. In the first column, where, in fact, no noise is used, the improvement is $28.9\%$, from an odometric mean error of $0.038m$ down to a SLAM mean error of $0.027m$. When the noise level added to the odometry increases, the correction given by the EKF-SLAM is more evidently reflected in the percentage of improvement. For a noise level of 4, the odometry mean error is $0.806m$ while the EKF mean error is $0.309m$, an improvement of $61.6\%$.

Even with the highest noise level that causes an odometry

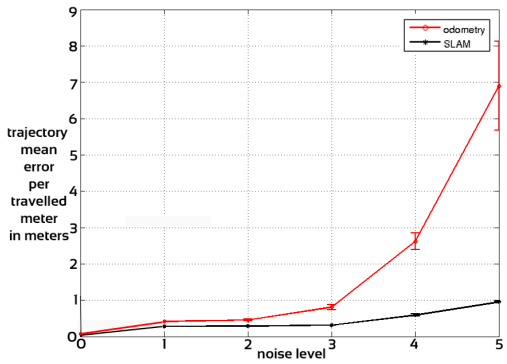


Fig. 2: Evolution of the odometry mean error and the EKF-SLAM mean error, for the different levels of corruptive noise.

trajectory mean error of $6.898m$, the EKF-SLAM is able to improve the estimates a $86.1\%$. Figure 2 shows how the trajectory mean error raises very fast up to $7m$ as the noise level added to the odometry increases, whilst the level of the trajectory mean error of the EKF-SLAM estimates is bounded between $0m$ and $1m$. The vertical error bars correspond to $0.1\sigma$ to provide a clearer representation. $y-$axis shows the error per travelled meter in meters and the $x-$axis represents the different noise levels corrupting the odometry.

Figure 3 shows the trajectory of the aforementioned sweeping task, according to the odometry estimates (in black) corrupted with different levels of Gaussian noise, the ground truth (in blue) and the EKF estimates (in red). All units are expressed in meters. The plot corresponding to the noise level 6 shows clearly how the EKF-SLAM approach is able to correct the odometry trajectory which is clearly drifted, setting it close to the ground truth.

Figure 4 shows the trajectory according to the three different estimates, being the odometry corrupted with a noise level 3, and eight loop closings. Each loop closing is shown as an edge in magenta linking the two images involved. Although in this trajectory there are more than 30 loop closings, only 8 have been plotted just to present the figure with enough clarity.

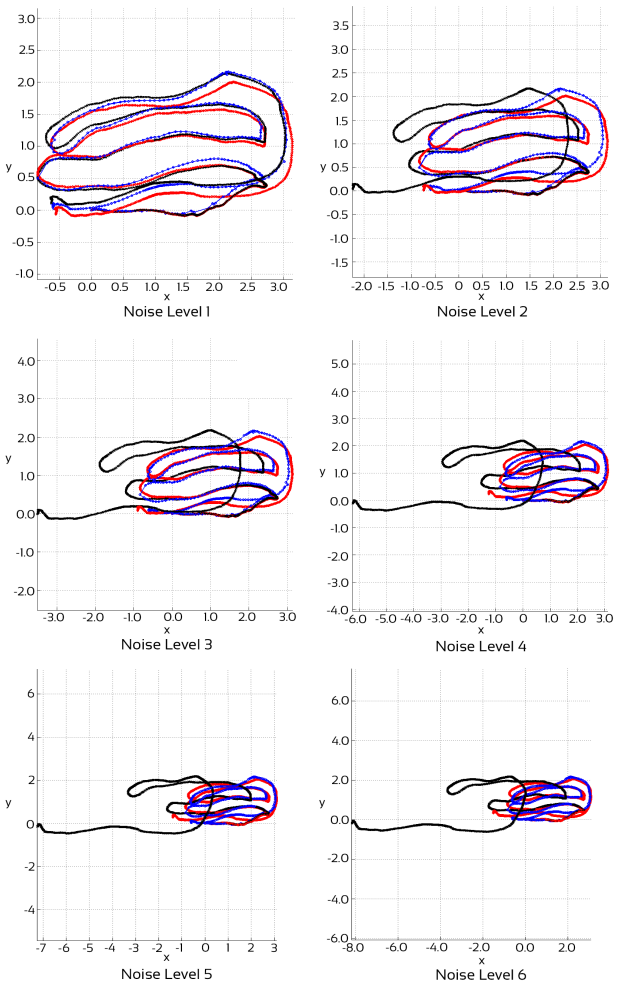


Fig. 3: The sweeping trajectory according to the three different estimates with different levels of corruptive noise.

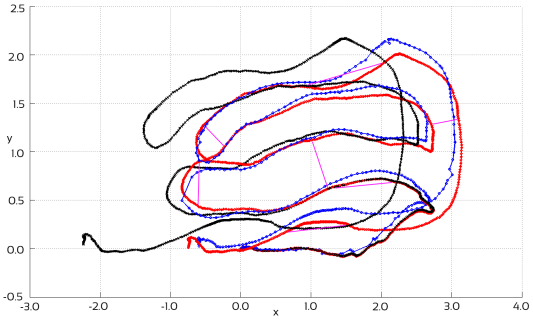

Fig. 4: The trajectory according to the three different estimates plus 8 loop closings (in magenta).

## REFERENCES


[1] M. Bujnak, S. Kukelova, and T. Pajdla, "New Efficient Solution to the Absolute Pose Problem for Camera with Unknown Focal Length and Radial Distortion," *Lecture Notes in Computer Science*, vol. 6492, pp. 11–24, 2011.

[2] A. Burguera, Y. González, and G. Oliver, "Underwater slam with robocentric trajectory using a mechanically scanned imaging sonar," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, CA, October 2011.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping (SLAM): part I," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, June 2006.

[4] R. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation for autonomous underwater vehicles," *IEEE Journal of Oceanic Engineering*, vol. 33, no. 2, pp. 103–122, April 2008.

[5] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters for view-based slam."," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, December 2006.

[6] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[7] M. Hildebrandt and F. Kirchner, "Imu-aided stereo visual odometry for ground-tracking auv applications," in *Proceedings of Oceans*, Sydney, Australia, May 2010.

[8] C. Mei, "Robust and accurate pose estimation for vision-based localisation," *IEEE International Conference on Intelligent Robots and Systems*, pp. 3165–3170, 2012.

[9] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[10] R. Schattschneider, G. Maurino, and W. Wang, "Towards stereo vision slam based pose estimation for ship hull inspection," in *Proceedings of Oceans*, Waikoloa, Hawaii, June 2011, pp. 1–8.

[11] R. Smith, P. Cheeseman, and M. Self, "A stochastic map for uncertain spatial relationships." in *Proceedings of International Symposium on Robotic Research, MIT Press*, 1987, pp. 467–474.