

Probabilistic Sonar Scan Matching for an AUV

Emili Hernández, Pere Ridao, David Ribas and Angelos Mallios

Abstract—This paper proposes MSISpIC, a probabilistic sonar scan matching algorithm for the localization of an Autonomous Underwater Vehicle (AUV). The technique uses range scans gathered with a Mechanical Scanning Imaging Sonar (MSIS), the robot displacement estimated through dead-reckoning using a Doppler Velocity Log (DVL) and a Motion Reference Unit (MRU). The proposed method is an extension of the pIC algorithm. An Extended Kalman Filter (EKF) is used to estimate the robot-path during the scan in order to reference all the range and bearing measurements as well as their uncertainty to a scan fixed frame before registering. The major contribution consists of experimentally proving that probabilistic sonar scan matching techniques have the potential to improve the DVL-based navigation. The algorithm has been tested on an AUV guided along a 600m path within an abandoned marina underwater environment with satisfactory results.

I. INTRODUCTION

During a long term mission with an autonomous robot it is necessary to keep the track of the vehicle's position. Scan matching is a technique that can be used to estimate the vehicle displacement using successive range scans. Many applications in robotics like mapping, localization or tracking use this technique to estimate the robot's relative displacement [1], [2]. Scan Matching estimates the robot relative displacement between two configurations, by maximizing the overlap between the range scans normally gathered with a laser or a sonar sensor. Moreover, the scans can then be used to set up a local map of the robot's surrounding environment to be used for reactive/deliberative obstacle avoidance [3].

The existing scan matching techniques can be divided into groups depending on if they use high-level entities like lines or planes or otherwise they rely on the raw scan. On one hand, it is possible to assume the existence of polygonal structures in structured environments as is supposed in [4] [5], or even in some underwater applications [6]. However, extracting simple polygons for representing the environment is not always possible, particularly in unstructured scenarios which are the most common in underwater robotics. On the other hand, there is a second type of algorithms that work with raw data from the scanner to solve the matching. Usually, these techniques are based on a two step iterative process which is repeated till convergence. The sensor displacements are computed by approximating the solution to the best overlap between two scans by looking for the closest point for each single data of the scan. After that,

This research was sponsored by the Spanish government (DPI2008-06545-C03-03) and FREESUBNET (MRTN-CT-2006-036186).

The authors are with the Department of Computer Engineering, University of Girona, 17071 Girona, Spain {emilihb, pere, dribas, amallios}@eia.udg.edu

a minimization process to estimate the solution is done. The process is repeated until convergence. The most popular technique is the Iterative Closest Point (ICP) algorithm [7] which has been modified in different ways [8]. Most of the versions of the ICP algorithm use the Euclidian distance to estimate the correspondences between scans. However, this distance does not take into account that small rotations of the sensor mean large displacements as the distance is increased. To overcome this limitation several approaches have been done. The Iterative Dual Correspondence (IDC) algorithm [9] computes two different minimizations, one for the translation and another for the rotation. The Metric-based Iterative Closest Point (MbICP) algorithm [10] establishes a new distance concept which captures the sensor displacement and rotation at the same time. However, none of these algorithms take into account the sensor nor the displacement uncertainties which are very important, specially when sonar sensors are used. The probabilistic Iterative Correspondence (pIC) method [11] explicitly deals with those uncertainties to decide which points in the reference scan are statistically compatible with a certain point of the new scan. Moreover, the minimization process is carried out over the Mahalanobis distance. Although in this case the algorithm was applied to laser scans, its probabilistic formulation is able to improve the robustness and accuracy. On the other hand, in [12] another probabilistic approach is proposed in order to deal with the noisy and sparse measurements obtained from conventional time-of-fly sonar sensors of a mobile robot. This method deals with the sparsity of readings by grouping sonar measurements along short trajectories. It uses probabilistic models of ultrasonic and odometry sensors as well as a method to propagate the error through them in order to estimate a group of scan positions together with their uncertainty.

Although a large literature exists reporting successful applications of scan matching to mobile robots, very few attempts have been done to use sonar scan matching in underwater applications. In [13] a non-probabilistic variation of ICP is proposed to achieve on-line performance for registering multiple views captured with a 3D acoustic camera. Silver *et al.* [14], proposed to use a particle filter to deal with the sonar noisy data. Like us, they use a MSIS but in their case only simulated results are reported. A very nice application of sonar scan matching underwater is reported in [15] where the standard ICP is used for registering bathymetric sub-maps gathered with a multibeam sonar profiler. In their work the standard ICP algorithm is used and hence the uncertainty of the scan points is not taken into account during the registering. In this paper we propose the MSISpIC,

an extension of the basic pIC algorithm to deal with data gathered by an AUV using a MSIS. In our approach, the robot moves while scanning the environment. Hence, an EKF using a constant velocity model with acceleration noise, updated with velocity and attitude measurements obtained from a DVL and a MRU respectively, is used to estimate the trajectory followed by the robot along the scan. This trajectory is used to remove the motion induced distortion of the acoustic image as well as to predict the uncertainty of the range scans prior to register them through the standard pIC algorithm. To the best of the authors knowledge, this is the first reported application of a probabilistic sonar-based scan matching technique over a real data dataset gathered with an AUV using a MSIS.

The paper is structured as follows. In section II the pIC algorithm is described. Section III details the process to grab complete scans from the MSIS to be used in our scan matching algorithm which is described in section IV. Section V reports the experimental results before conclusions and future work.

II. PROBABILISTIC ITERATIVE CORRESPONDENCE

As stated before, the pIC algorithm is a statistical extension of the ICP algorithm which is able to deal with sparse sonar data. As detailed in Algorithm 1, the inputs are the reference scan S_{ref} , the new scan S_{new} and the initial relative displacement estimation $\hat{\mathbf{q}}$ between them with its covariance \mathbf{P}_q . The following procedure is iteratively executed until convergence. First, the points of S_{new} are compounded with the robot displacement (\mathbf{q}_k). The result are the points of the new scan referenced to the reference frame (\mathbf{c}_j). Then, for each \mathbf{c}_j , a set (A_j) of all the compatible points in the reference scan (S_{ref}) is established using a compatibility test over the Mahalanobis distance. Next step consists of computing the virtual association point (\mathbf{a}_j) as the expectancy over the random variable defined by the set (A_j). It is worth noting that if $\mathbf{q} \equiv N(\hat{\mathbf{q}}, \mathbf{P}_q)$, $\mathbf{n}_j \equiv N(\hat{\mathbf{n}}_j, \mathbf{P}_{n_j})$ and $\mathbf{r}_i \equiv N(\hat{\mathbf{r}}_i, \mathbf{P}_{r_i})$, it is possible to compute the probability $p(\mathbf{r}_i = \mathbf{c}_j)$ for every element (\mathbf{r}_i) of A_j of being a correct matching for \mathbf{c}_j . In order to do so, let us define a random Gaussian variable (r.g.v.) which describes the error of the $\{\mathbf{r}_i, \mathbf{c}_j\}$ pairing:

$$\mathbf{e}_{ij} = \mathbf{r}_i - \mathbf{q}_k \oplus \mathbf{n}_j ; \mathbf{e}_{ij} \approx N(\hat{\mathbf{r}}_i - \hat{\mathbf{q}}_k \oplus \hat{\mathbf{n}}_j, \mathbf{P}_{e_{ij}})$$

$$\mathbf{P}_{e_{ij}} = \mathbf{P}_{r_i} + \mathbf{J}_q \mathbf{P}_q \mathbf{J}_q^T + \mathbf{J}_n \mathbf{P}_{n_j} \mathbf{J}_n^T$$

with

$$\mathbf{J}_q = \left. \frac{\partial \mathbf{r}_i - \mathbf{q} \oplus \mathbf{n}_j}{\partial \mathbf{q}} \right|_{\hat{\mathbf{q}}}, \mathbf{J}_n = \left. \frac{\partial \mathbf{r}_i - \mathbf{q} \oplus \mathbf{n}_j}{\partial \mathbf{n}_j} \right|_{\hat{\mathbf{n}}_j}$$

then $p(\mathbf{r}_i = \mathbf{c}_j) = p(\mathbf{e}_{ij} = 0)$ where $p(\mathbf{e}_{ij} = 0)$ can be computed as follows:

$$p(\mathbf{e}_{ij} = 0) = \frac{f_{e_{ij}}(\mathbf{r}_i - \mathbf{c}_j)}{\sum_{r_i \in A_j} f_{e_{ij}}(\mathbf{r}_i - \mathbf{c}_j)} \quad (1)$$

where $f_{e_{ij}}$ is the probability density function of \mathbf{e}_{ij} r.g.v. Once $\hat{\mathbf{a}}_j$ has been computed, a similar procedure can be used to estimate its uncertainty \mathbf{P}_{a_j} , before computing the uncertainty \mathbf{P}_{e_j} of the matching error ($\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j$). Now, it is possible to estimate the displacement $\hat{\mathbf{q}}_{min}$ which minimizes the square error of the *Mahalanobis Distance* between $\hat{\mathbf{a}}_j$ and $\hat{\mathbf{c}}_j$. This is done using Least Squares minimization method. If there is convergence, the function returns, otherwise another iteration is required.

Algorithm 1 The pIC algorithm

```

 $\hat{\mathbf{q}}_{pIC} = pIC(S_{ref}, S_{new}, \hat{\mathbf{q}}, \mathbf{P}_q) \{$ 
 $k = 0; \hat{\mathbf{q}}_k = \hat{\mathbf{q}}$ 
do {
  for( $j = 0; size(S_{new}); j++$ ) {
     $\hat{\mathbf{c}}_j = \hat{\mathbf{q}}_k \oplus \hat{\mathbf{n}}_j$ 
 $A_j = \{\mathbf{r}_i \in S_{ref} / D_M^2(\mathbf{r}_i, \mathbf{c}_j) \leq \chi_{2,\alpha}^2\}$ 
 $\hat{\mathbf{a}}_j = \sum_j \hat{\mathbf{r}}_i p(\mathbf{r}_i = \mathbf{c}_j), \forall \mathbf{r}_i \in A_j$ 
 $\mathbf{P}_{a_j} = \sum_{r_i \in A_j} [(\hat{\mathbf{r}}_i - \hat{\mathbf{a}}_j)(\hat{\mathbf{r}}_i - \hat{\mathbf{a}}_j)^T + \mathbf{P}_{r_i}] p(\mathbf{r}_i = \mathbf{c}_j)$ 
 $\mathbf{P}_{e_j} = \mathbf{P}_{a_j} + \mathbf{J}_q \mathbf{P}_q \mathbf{J}_q^T + \mathbf{J}_n \mathbf{P}_{n_j} \mathbf{J}_n^T$ 
  }
 $\hat{\mathbf{q}}_{min} = arg \min_{\mathbf{q}} \left\{ \sum_j ((\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j)^T \mathbf{P}_{e_j}^{-1} (\hat{\mathbf{a}}_j - \hat{\mathbf{c}}_j)) \right\}$ 
  if(Convergence())
     $\hat{\mathbf{q}}_{pIC} = \hat{\mathbf{q}}_{min}$ 
  else {
     $\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_{min}; k++$ 
  }
}
while(!Convergence() and  $k < maxIterations$ )
}

```

III. SCAN GRABBING USING A MSIS

The pIC algorithm was conceived to accept as input parameters two range scans and optionally a rough displacement estimation between them. Moreover, it uses a laser range finder which gathers scans almost instantaneously. However, for the underwater environment, commercially available scan sensors are based on acoustics. Most of these sensors have a mechanical head that rotates at fixed angular steps. At each step, a beam is emitted and received, measuring ranges to the obstacles found across its trajectory. Hence, getting a complete scan lasts few seconds while the vehicle is moving, generating deformed scans. Therefore, a correction taking into account the robot pose when the beam was grabbed is necessary.

A. Beam segmentation and range detection

The MSIS returns a polar acoustic image composed of beams. Each beam has a particular bearing angle value and a set of intensity measurements. The angle corresponds to the orientation of the sensor head when the beam was emitted. The acoustic linear image corresponding to one beam is returned as an array of acoustic intensities detected at certain distances. The beam is then segmented using a predefined threshold to compute the intensity peaks. Due to

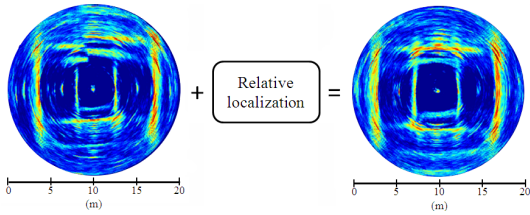


Fig. 1. The distortion produced by the displacement of the robot while acquiring data can be corrected with the relative displacement.

the noisy nature of the acoustic data, a minimum distance between peaks criteria is also applied. Hence, positions finally considered are those corresponding to high intensity values above the threshold with a minimum *distance* between each other.

B. Trajectory estimation

The pIC algorithm needs a complete scan to be registered with the previous one in order to estimate the robot's displacement. Since MSIS needs a considerable period of time to obtain a complete scan, if the robot does not remain static, the robot's motion induces a distortion in the acoustic image (Fig. 1). To deal with this problem it is necessary to know the robot's pose at the beam reception time. Hence, it is possible to define an initial coordinate system I to reference all the range measurements belonging to the same scan. In our case, this initial frame is fixed at the robot pose where the first beam of the current scan was read.

The localization system used in this work is a slight modification of the navigation system described in [16]. In this system, a Xsense MRU provides heading measurements and a SonTek Argonaut DVL unit which includes 2 inclinometers and a depth sensor is used to estimate the robot's pose during the scan (navigation problem). MSIS beams are read at $30Hz$ while DVL and MRU readings arrive asynchronously at a frequency of $1.5Hz$ and $10Hz$ respectively. An EKF is used to estimate the robot's 6DOF pose whenever a sonar beam is read. DVL and MRU readings are used asynchronously to update the filter. To reduce the noise inherent to the DVL measurements, a simple 6DOF constant velocity kinematics model is used.

The information of the system at step k is stored in the state vector \mathbf{x}_k with estimated mean $\hat{\mathbf{x}}_k$ and covariance \mathbf{P}_k :

$$\hat{\mathbf{x}}_k = [\hat{\eta}^B, \hat{\nu}^R]^T \quad \mathbf{P}_k = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \quad (2)$$

with:

$$\eta^B = [x, y, z, \phi, \theta, \psi]^T; \quad \nu^R = [u, v, w, p, q, r]^T \quad (3)$$

where, as defined in [17], η^B is the position and attitude vector referenced to a base frame B , and ν^R is the linear and angular velocity vector referenced to the robot's coordinate frame R . The coordinate frame B is chosen coincident with I but oriented to the north, hence the compass measurements can be integrated in a straight forward manner.

The vehicle's movement prediction is performed using the 6DOF kinematic model:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}) = \begin{bmatrix} \eta_k^B \\ \nu_k^R \end{bmatrix} = \begin{bmatrix} \eta_{k-1}^B + J(\eta_{k-1}^B) \nu_{k-1}^R T \\ \nu_{k-1}^R \end{bmatrix} \quad (4)$$

$$J(\eta) = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - s\psi c\phi & c\psi \theta c\phi + s\psi s\phi & 0 & 0 & 0 \\ s\psi c\theta & s\phi s\psi s\theta + c\psi c\phi & s\psi \theta c\phi - s\phi c\psi & 0 & 0 & 0 \\ -s\theta & c\theta s\phi & c\theta c\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & s\phi t\theta & c\phi t\theta \\ 0 & 0 & 0 & 0 & c\phi & -s\phi \\ 0 & 0 & 0 & 0 & s\phi/c\theta & c\phi/c\theta \end{bmatrix} \quad (5)$$

Although in this model the velocity is considered to be constant, in order to allow for slight changes, a velocity perturbation modeled as the integral of a stationary white noise v_k is introduced. The covariance matrix \mathbf{Q}_k of this acceleration noise is diagonal and in the order of magnitude of the maximum acceleration increment that the robot may experience over a sample period.

$$\nu_k^R = \hat{\nu}_k^R + v_k T \quad (6)$$

$$E[v_k] = 0; \quad E[v_k v_j^T] = \delta_{kj} \mathbf{Q} \quad (7)$$

Hence, v_k is the acceleration noise which is integrated and added in velocity (6), being nonlinearly propagated to the position. Finally, the model prediction and update is carried out as detailed below:

1) *Prediction*: The estimate of the state is obtained as:

$$\hat{\mathbf{x}}_k = f(\hat{\mathbf{x}}_{k-1}) \quad (8)$$

and its covariance matrix as:

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

where \mathbf{F}_k and \mathbf{G}_k are the Jacobian matrices of partial derivatives of the non-linear model function f with respect to the state \mathbf{x}_k and the noise v_k , respectively.

2) *Update using DVL measurements*: The model prediction is updated by the standard Kalman filter equations each time a new DVL measurement arrives:

$$\mathbf{z}_{DVL,k} = [u_b, v_b, w_b, u_w, v_w, w_w, \phi_i, \theta_i, \psi_c, z_{depth}]^T \quad (10)$$

where subindex b stands for bottom tracking velocity, w for through water velocity, i for inclinometers and c represents the compass. The measurement model is:

$$\mathbf{z}_{DVL,k} = \mathbf{H}_{DVL,k} \mathbf{x}_{k|k-1} + w_k \quad (11)$$

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

where w_k (measurement noise) is a gaussian zero-mean white noise:

$$E[w_k] = 0; \quad E[w_k w_j^T] = \delta_{kj} \mathbf{R}_{DVL,k} \quad (13)$$

Since the DVL sensor provides a status measurement for bottom tracking and water velocity, depending on the quality of the measurements, different versions of the \mathbf{H} matrix are used to fuse one (removing row 2), the other (removing row 1), or both readings (using the full matrix).

Algorithm 2 Scan grabbing

```
[Snew, q̂new, Pqnew] = ScanGrabbing() {
  ResetDeadReckoningXYZ()
  [x̂BI, PIB] = GetDeadReckoning()
  for all beams {
    beam = GetBeam()
    beam = Segment(beam)
    [ρ̂, Pρ] = LocalMaximaFinder(beam)
    [x̂RB, PBR] = GetDeadReckoning()
    //ρ̂ and Pρ from the local frame I
    n̂ = x̂BI ⊕ x̂RB ⊕ xSR ⊕ P2C(ρ̂)
    Pn = JI1⊕ PIB JI1⊕T + JI2⊕ [JB1⊕ PBR JB1⊕T +
    JB2⊕ JR⊕ JS⊕ Pρ JS⊕T JR⊕T JB2⊕T] JI2⊕T
    Snew = Snew ∪ {[n̂, Pn]}
  }
  q̂new = x̂RB; Pqnew = PBR
}
```

two consecutive scans (S_{new} and S_{ref}) and its relative displacement which coincides with the position occupied by the robot at the end of the first scan ($\hat{\mathbf{q}}_{ref}$). The output is an improved estimation of the robot displacement ($\hat{\mathbf{q}}_{new}$). The iterative compounding of the relative displacement allows to track the global robot position.

Algorithm 3 MSISpIC

```
MSISpIC() {
  [Sref, q̂ref, Pqref] = ScanGrabbing()
  q̂global = 0
  while(true) {
    [Snew, q̂new, Pqnew] = ScanGrabbing()
    q̂pIC = pIC(Sref, Snew, q̂ref, Pqref)
    q̂global = q̂global ⊕ q̂pIC
    Sref = Snew; q̂ref = q̂new
  }
}
```

V. EXPERIMENTAL RESULTS

The method described in this paper has been used with a dataset obtained in an abandoned marina located in Sant Pere Pescador, on the Catalan coast [18]. The experiment was carried out using ICTINEU^{AUV} [6] traveling along a 600m path. Because the Differential Global Positioning System (DGPS) signal is not available when submerged, a surface buoy equipped with a DPGS receiver was attached to the robot in order to gather the ground truth trajectory. The MSIS was configured to scan the whole 360° sector and it was set to fire up to a 50m range with a 0.1m resolution and a 1.8° angular step. Dead-reckoning was computed using the velocity reading coming from the DVL and the heading data obtained from the MRU sensor, both merged using the described EKF. Standard deviation for the MSIS sensor was set as it is specified by the manufacturer, 0.1m in range and 1.8° in angular measurements. The whole dataset

was acquired in 53min and the off-line execution of the algorithm implemented in MATLAB took around 16min in a CoreDuo@1,83GHz laptop.

Fig. 3.a shows the trajectory of the DGPS and the raw map plotted on an orthophotomap used as a ground truth. Fig. 3.b and 3.c show the trajectories and raw maps estimated using the dead-reckoning method and the MSISpIC algorithm respectively. It can be clearly appreciated that the dead-reckoning estimated trajectory suffers from an important drift which is considerably reduced when the MSISpIC algorithm is used. Most of the error of the MSISpIC estimated trajectory appears when the robot is traversing an area where the scan only observes one or two walls parallel to the robot path, being able to correct the lateral displacement but still drifting in the forward direction. It is worth noting that, even in the presence of structures in all the directions, scan matching algorithms are expected to drift due to its iterative formulation. Note that we are performing scan to scan registration and not scan to map registration.

The absolute error of dead-reckoning and MSISpIC trajectories with respect to the DPGS ground truth are depicted in Fig. 3.d. The results show an important improvement of the MSISpIC respect to dead-reckoning trajectory, which after traveling the 600m path differs from DGPS around 40m while the MSISpIC maximum difference is set at 12m.

Fig. 3.e and 3.f show an occupancy grid map built using the dead-reckoning and MSISpIC estimated trajectories respectively plotted on the orthophotomap. Both maps are built using a standard occupancy grid mapping algorithm [19] with simple range finder model [20] to project the sensor data into the map. The cell resolution of the map is 0.4m. The probabilities for the sensor model were set up experimentally to be 0.3, 0.5 and 0.9 for the free, the unknown and the occupied cells respectively. The color scale of the maps is blue for the free space, transparent green for the unknown area and from yellow to red for the occupied cells.

Using the dead-reckoning trajectory, most of the map (Fig. 3.e) allows to distinguish the shape of the marina except at last part (after the corridor) where the points that should compound the left wall are spread away. The map generated with the scan matching trajectory (Fig. 3.f) shows a substantial improvement with respect to the one generated using dead-reckoning as the walls are clearly less sparse.

VI. CONCLUSIONS

This paper proposes a variation of the pIC algorithm called MSISpIC which is able to perform underwater scan matching using a MSIS. To deal with the motion induced distortion of the acoustic image, an EKF is used to estimate the robot motion during the scan. The filter uses a constant velocity model with acceleration noise for motion prediction and velocity (DVL) and attitude measurements (MRU) for updating the state. Through the compounding of the relative robot position within the scan, with the range and bearing measurements of the beams gathered with the sonar, the acoustic image gets undistorted. Assuming Gaussian noise, the algorithm is able to predict the uncertainty of the sonar measurements

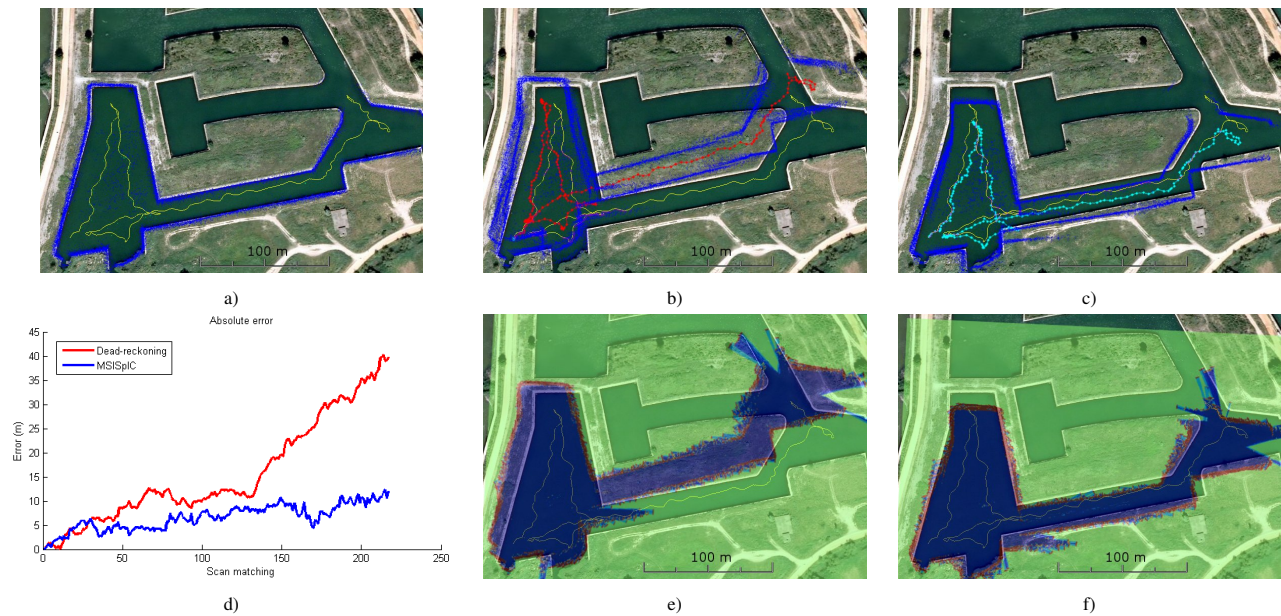


Fig. 3. Results: a) b) and c) Raw map and trajectories comparison between DGPS (a), dead-reckoning (b) and the MSISpIC output (c) over the orthophotomap. d) Dead-reckoning and MSISpIC absolute error respect DGPS. e) and f) Maps generated using dead-reckoning and MSISpIC trajectories respectively.

with respect to a frame located at the position occupied by the robot at the beginning of the scan, before applying the standard pIC algorithm. The proposed method has been tested with a dataset acquired during a survey mission in an abandoned marina located in the Girona coast. The results show substantial improvements in trajectory correction and map reconstruction.

VII. FUTURE WORK

Once it has been experimentally proven that sonar scan matching has the potential to improve the DVL-based navigation, we are currently working on the cloning of the robot pose within the EKF in order to: 1) smooth the trajectory during the scan and 2) use the result of the scan matching as a constrain between the boundary poses of the scan in order to correct the full scan trajectory. Next step will consist of gathering a new dataset in an unstructured environment to check how the algorithm performs in such situation.

REFERENCES

- [1] C. WangThorpe and A. Suppe, "Ladar-based detection and tracking of moving objects from a ground vehicle at high speeds," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pp. 416–421, 2003.
- [2] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pp. 206–211 vol.1, 27–31 Oct. 2003.
- [3] J. Mínguez, "Integration of planning and reactive obstacle avoidance in autonomous sensor-based navigation," *Intelligent Robots and Systems, 2005. (IROS 2005). IEEE/RSJ International Conference on*, 2005.
- [4] A. Grossmann and M. Poli, "Robust mobile robot localization from sparse and noisy proximity readings," 1999.
- [5] J. Castellanos, J. Tardós, and J. Neira, "Constraint-based mobile robot localization," *Advanced Robotics and Intelligent Systems, IEE*, 1996.
- [6] D. Ribas, N. Palomer, P. Ridao, M. Carreras, and E. Hernandez, "Ictineu AUV wins the first SAUC-E competition," in *IEEE International Conference on Robotics and Automation*, Italy, 2007.
- [7] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, 1992.
- [8] S. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pp. 145–152, 2001.
- [9] F. Lu and E. Miliotis, "Robot pose estimation in unknown environments by matching 2d range scans," in *CVPR94*, 1994, pp. 935–938.
- [10] J. Mínguez, F. Lamiroux, and L. Montesano, "Metric-based scan matching algorithms for mobile robot displacement estimation," *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 3557–3563, Apr. 2005.
- [11] L. Montesano, J. Mínguez, and L. Montano, "Probabilistic scan matching for motion estimation in unstructured environments," *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug. 2005.
- [12] A. Burguera, Y. Gonzalez, and G. Oliver, "Probabilistic sonar scan matching for robust localization," *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3154–3160, apr 2007.
- [13] U. Castellani, A. Fusiello, V. Murino, L. Papaleo, E. Puppo, S. Repetto, and M. Pittore, "Efficient on-line mosaicing from 3d acoustical images," *OCEANS '04. MTS/IEEE TECHNO-OCEAN '04*, 2004.
- [14] D. Silver, D. Bradley, and S. Thayer, "Scan matching for flooded subterranean voids," in *IEEE conference on Robotics Automation and Mechatronics (RAM)*, vol. 1, December 2004, pp. 422 – 427.
- [15] C. Roman and H. Singh, "Improved vehicle based multibeam bathymetry using sub-maps and slam," *Intelligent Robots and Systems, 2005. (IROS 2005). IEEE/RSJ International Conference on*, 2005.
- [16] D. Ribas, J. Neira, P. Ridao, and J. Tardós, "Auv localization in structured underwater environments using an a priori map," in *7th IFAC Conference on Manoeuvring and Control of Marine Crafts*, Lisbon, Portugal, september 2006.
- [17] T. Fossen, *Guidance and Control of Ocean Vehicles*, J. Wiley and S. Ltd., Eds., 1994.
- [18] D. Ribas, P. Ridao, J. Tardós, and J. Neira, "Underwater SLAM in man made structured environments," *Journal of Field Robotics*, vol. Accepted for publication, 2008.
- [19] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun 1989.
- [20] S. Thrun, W. Burgard, and F. Dieter, *Probabilistic Robotics*, 2005.