

# Visual SLAM for 3D Large-Scale Seabed Acquisition Employing Underwater Vehicles

Joaquim Salvi, Yvan Petillot and Elisabet Batlle

**Abstract**—This paper presents a novel technique to align partial 3D reconstructions of the seabed acquired by a stereo camera mounted on an autonomous underwater vehicle. Vehicle localization and seabed mapping is performed simultaneously by means of an Extended Kalman Filter. Passive landmarks are detected on the images and characterized considering 2D and 3D features. Landmarks are re-observed while the robot is navigating and data association becomes easier but robust. Once the survey is completed, vehicle trajectory is smoothed by a Rauch-Tung-Striebel filter obtaining an even better alignment of the 3D views and yet a large-scale acquisition of the seabed.

## I. INTRODUCTION

The strong attenuation and scattering of light underwater has limited the use of optical systems and acoustic sensors have been chosen. However, video cameras are ubiquitous, cheaper and video images are preferred for scientific exploration and offshore man-made structures inspection, bringing about researching in optical underwater imaging.

Nowadays, cameras can be mounted on underwater vehicles acquiring high resolution video images of the seabed at short altitudes. Relevant results have been obtained aligning hundreds of images using the so-called photo-mosaicing technique on flat terrains [1]. However, applications of interest for scientific and offshore community concerns structures with major 3D component. Examples are benthic habitats such as coral reefs, hydrothermal vent fields, ancient and modern shipwrecks and archaeological settlements and man-made underwater structures in need of regular inspection. Mosaics performed in areas with significant 3D structure suffer from misalignments and image artefacts that deteriorate the mapping (parallax) [1].

Besides, Simultaneous Localization and Mapping (SLAM) has an active community in land robots. Excellent research is done in indoor using laser scanners [2], sonar [3] and video [4]; and outdoor using laser scanners [5] and video appearance based models [6]. Most algorithms assume dense features in the environment and good data association. In the case of video-based SLAM, most approaches use features in the 2D video streams to perform the data association and the recent development using appearance require prior learning of the environment. 3D structure is normally not used. In

This work was supported by EC Project MRTN-CT-2006-036186, Spanish Ministry of Education and Science Project DPI2007-66796-C03-02 and Spanish visiting fellowship PR2007-0186.

J. Salvi and E. Batlle are with the Computer Vision and Robotics Group at University of Girona, E-17071 Girona (Spain). J. Salvi is currently a visiting professor at the Ocean Systems Lab at Heriot-Watt University [qsalvi@eia.udg.edu](mailto:qsalvi@eia.udg.edu); [bbatlle@eia.udg.edu](mailto:bbatlle@eia.udg.edu)

Y. Petillot is with the Ocean Systems Lab at Heriot-Watt University, EH144AS Edinburgh (UK) [Y.R.Petillot@hw.ac.uk](mailto:Y.R.Petillot@hw.ac.uk)

an underwater scenario, features can be sparse, appearance alone is normally not discriminant and data association is difficult as images are corrupted by a more significant level of noise and distortion. Unsurprisingly, very few papers have tackled SLAM in underwater. The ones that tried, they have always focused on acoustic data [7] [8] [9].

The key to a successful visual SLAM based system underwater must lie in the selection of very robust features so that data association is possible even under different view points and illumination patterns. The second important factor to take into account is the likely sparseness of the feature maps, due to the environment and the necessary selection of robust features.

This paper proposes a solution to recover the local 3D map of the seabed structure by using a stereo camera mounted on an underwater vehicle. Landmarks are detected in the local 3D structure and characterized considering 2D and 3D features. This is a novel approach compared to existing techniques. An Extended Kalman Filter (EKF) SLAM framework is proposed to filter the navigation data given by the Doppler Velocity Log (DVL) of the vehicle and remove the drift thanks to the re-observation of landmarks. Finally, the output of the Kalman filter is filtered using a Rauch-Tung-Striebel (RTS) smoother obtaining a better alignment of the sequence of local 3D maps and yet delivering a large-scale 3D acquisition of the seabed.

## II. EKF-BASED SLAM

The vehicle is equipped with an ExplorerDVL of Teledyne RD Instruments that measures DVL frame  $\{D\}$  absolute orientation  ${}^E\Theta = {}^E R_D$  (roll, pitch and yaw) wrt (with respect to) Earth  $\{E\}$  and linear velocity  ${}^D\dot{x}$  wrt DVL frame  $\{D\}$ . Note that DVL position is not measured and should be obtained by integrating the velocity. Besides, the vehicle is equipped with a stereo-vision system that measures landmark position  ${}^L p_i$  wrt the left camera frame  $\{L\}$ , as shown in Fig. 1. Right camera frame  $\{R\}$  wrt left camera frame  ${}^L T_R$  and left camera frame wrt DVL frame  ${}^D T_L$  are fixed and determined by calibration.

A Kalman filter is composed of three steps: Prediction, Observation and Update. We have added a fourth step to incorporate new landmarks to the state of the filter. The filter is fed by the motion measurements (orientation and velocity) provided by the DVL and landmark re-observation determined by the stereo-vision system, as shown in Fig. 2 and explained in the following sections.

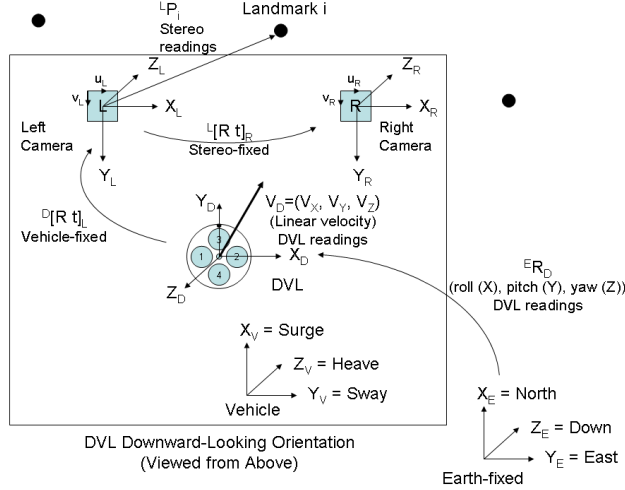


Fig. 1. Layout of the underwater vehicle showing the stereo-vision system and the Doppler Velocity Log and the coordinate systems involved measuring vehicle orientation and linear velocity and potential landmarks.

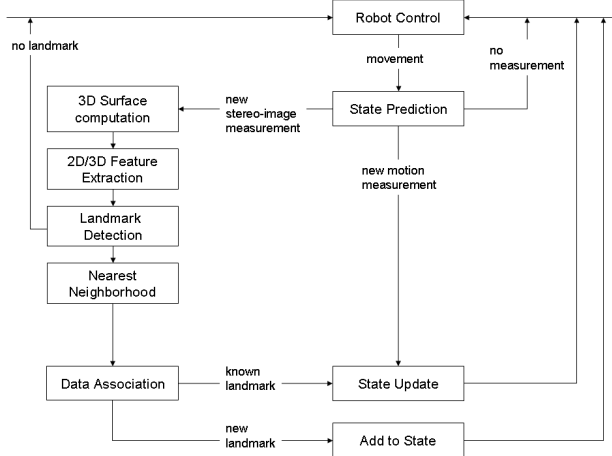


Fig. 2. Flow diagram of the SLAM module.

### A. Process Model

The state of the system consists initially of the orientation  ${}^E\Theta$ , position  ${}^E x$  and linear velocity  ${}^E \dot{x}$  of the vehicle wrt Earth  $\{E\}$ . Note that vehicle position is unknown and hence can only be measured incrementally. Consequently, we assume for simplicity that the Earth frame is located at the DVL initial position and oriented according to DVL measures. Angular velocity was not considered since it is not measured and would increase filter complexity without improving accuracy. Note that orientation is measured in absolute values and hence the DVL is not introducing drift. Once a landmark is observed, the state is augmented with the position  ${}^E p_i$  of the new landmark wrt Earth. Landmarks are kept during the whole mission of the vehicle. Hence, the state of the system at instant  $k$  is defined by the following equation,

$$x(k) = [{}^E\Theta, {}^E x, {}^E \dot{x}, p_1, \dots, p_n]. \quad (1)$$

### B. Prediction Model

Assuming that the state at instant  $k$  is known, the prediction of the next state is modelled by

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) \quad (2)$$

where  $F(k)$  is the state matrix. The orientation and velocity of the vehicle and position of landmarks are assumed constant. The position of the vehicle follows a standard linear model. The predicted covariance matrix for state  $(k+1)$  that it is computed as follows:

$$P(k+1|k) = F(k)P(k|k)F^T(k) + Q \quad (3)$$

where  $Q$  is the process noise matrix. It consists of a diagonal of 0 except in the terms of orientation, position and velocity of the vehicle, where the variances of the corresponding process noises are added. Process noise variances are fixed, determined off-line and define the reaction of the filter to sudden changes of the ground truth orientation/velocity of the vehicle; and the covariance matrix at the initial time stamp  $P(1|1)$  is defined by the variances of the orientation and velocity measuring noise given by the navigation data and the variance of the landmark measurement noise given by the video camera.

### C. Observation Model

In the observation model we have to deal with the real measurements obtained by the on-board DVL and stereo-vision system at the real pose of the vehicle, together with the predicted measurements performed by the filter at the filter current vehicle state. Stereo camera noise has been experimentally proved to be approximately Gaussian. DVL noise is Gaussian. The difference between both measurements is the innovation vector and it is the basis of minimization. The DVL measurement  $z_m(k+1) = [{}^E\Theta, {}^D \dot{x}]$  is a  $6 \times 1$  vector composed of the measurements of vehicle orientation wrt Earth and the linear velocity wrt DVL frame. We consider that all landmarks are stationary and due to our data association process a single landmark is at the most observed at a given instant of time (see section V-B). Hence, when a landmark is observed the stereo-vision measurement  $z_l(k+1) = {}^D p_i = {}^D T_L {}^L p_i$  is a  $3 \times 1$  vector defined by  $(X, Y, Z)$  position of the observed landmark wrt DVL frame. The measurement vector is finally given by  $z(k+1) = [z_m(k+1) \ z_l(k+1)]$ .

The predicted motion measurement  $\hat{z}(k+1) = [\hat{z}_m(k+1) \ \hat{z}_l(k+1)] = H(\hat{x}(k+1|k))$ , where  $H$  is a non-linear function defined as follows,

$$H = \begin{pmatrix} {}^E \hat{\Theta}(k+1|k) \\ {}^D \hat{R}_E {}^E \hat{x}(k+1|k) \\ {}^D \hat{T}_E {}^E \hat{p}_i(k+1|k) \end{pmatrix} \quad (4)$$

#### D. Process Update

When a new measurement of the vehicle motion is given by the DVL or a landmark is re-observed by the video camera, the innovation vector is computed accordingly:  $v(k+1) = z(k+1) - \hat{z}(k+1)$ ; together with an associated innovation covariance matrix  $S(k+1)$  given by,

$$S(k+1) = \dot{H}(k+1)P(k+1|k)\dot{H}^T(k+1) + R(k+1) \quad (5)$$

where  $\dot{H}(k+1)$  is the jacobian matrix of function  $H(k+1)$  evaluated at  $\hat{x}(k+1|k)$ .

$$\dot{H}(k+1) = \left. \frac{\partial H}{\partial x} \right|_{\hat{x}(k+1|k)} \quad (6)$$

and depends on whether the orientation and velocity and/or any landmark is observed at  $k+1$ ; and  $R(k+1)$  is the measurement noise matrix defined as a diagonal matrix containing the vehicle orientation and velocity measurement noise variances and the landmark position measurement noise variance at time  $k+1$ , respectively.

The estimate of the state vector and its corresponding covariance matrix are then updated according to:

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + W(k+1)v(k+1) \quad (7)$$

$$P(k+1|k+1) = P(k+1|k) - W(k+1)S(k+1)W^T(k+1) \quad (8)$$

where

$$W(k+1) = P(k+1|k)\dot{H}(k+1)S^{-1}(k+1) \quad (9)$$

is known as the optimal Kalman gain at time  $k+1$ .

#### E. Adding New Landmarks

New landmarks are introduced in the filter state just after the process update step, since all vectors and matrices forming the filter have to be updated to use the new landmark in the filtering process. When a new landmark is observed: **a**) the observed position is added to the vector state  $x(k+1|k+1)$ ; **b**) the covariance matrix  $P(k+1|k+1)$  is enlarged by adding the rows and columns corresponding to the new landmark. The vehicle orientation variance at that time together with the landmark measurement noise variance are used to initialize the variance of the landmark in the filter; **c**) the state matrix  $F(k+1)$  is enlarged by adding 1's to the corresponding landmark position; and **d**) the process noise matrix  $Q$  is enlarged adding 0's, since landmarks are stationary.

### III. RAUCH-TUNG-STRIEBEL SMOOTHER

The Kalman filter uses all measurements up to the last iteration to estimate the state at the last iteration. The Rauch-Tung-Striebel (RTS) smoother uses all the measurements before and after each iteration to estimate the state at each iteration. It is a post-processing filter that works on the stored

outputs of the Kalman filter by re-processing them. The smoother works by combining a forward pass filter with a backward pass filter. It was originally designed to work with fixed size state vectors. However, the stochastic map adds new states to the state vector as it observes new landmarks. The algorithm adapts the RTS fixed-interval smoother to work with the stochastic map by fixing the size of the state vector to the size of the stochastic map on the last iteration. The output of the RTS has been shown to improve the accuracy of the stochastic map solution as well as providing smoother trajectories [8].

So, once the Kalman filter has finished, we fix  $k$  to the instant of time  $n-1$  and we go backwards till we reach instant of time 1. The predicted smoother state is computed

$$\hat{\tilde{x}}(k+1|k) = F(k)\hat{x}(k|k) \quad (10)$$

and the predicted covariance matrix

$$\hat{\tilde{P}}(k+1|k) = F(k)P(k|k)F^T(k) + Q. \quad (11)$$

Then, the smoother gain matrix  $J(k)$  is computed

$$J(k) = P(k|k)F^T(k)\hat{\tilde{P}}^{-1}(k+1|k) \quad (12)$$

and, hence, the filtered state is given by,

$$\tilde{x}(k|k) = \hat{x}(k|k) + J(k)(\tilde{x}(k+1|k+1) - \hat{\tilde{x}}(k+1|k)) \quad (13)$$

$$\tilde{P}(k|k) = P(k|k) + J(k)\left(\tilde{P}(k+1|k+1) - \hat{\tilde{P}}(k+1|k)\right)J^T(k). \quad (14)$$

We initialize the smoother so that  $\tilde{x}(n|n) = \hat{x}(n|n)$  and  $\tilde{P}(n|n) = P(n|n)$ .

### IV. LOCAL 3D SURFACE ACQUISITION

The problem addressed here is to recover 3D structure from a video stereo pair mounted on an underwater vehicle with changing illumination and an unknown surface structure. We have decided to use a wide-baseline stereo approach as depicted in Fig. 3.

First, a Homomorphic filter is used to normalize the brightness across the image and compensate for non uniform lighting patterns. This is followed by a Contrast-Limited Adaptive Histogram Equalization (CLAHE) to enhance the contrast of images. CLAHE operates on small data regions of the image. A further bilinear interpolation is performed to remove artificially induced boundaries between regions. Finally, an Adaptive Noise-Removal Filtering is carried out to remove the noise produced by the equalization especially in those areas with small variance (constant brightness). The resulting images are brighter, better contrasted and normalized. This facilitates the comparison of two images acquired at different times and viewpoints, enabling the matching of image features. Applying this process the number of features detected in the image is multiplied by ten times and features are spread throughout the whole image, which is quite satisfactory.

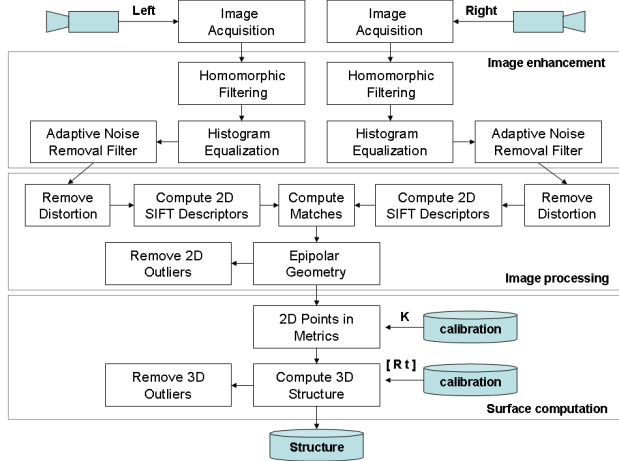


Fig. 3. Flow diagram detailing Image Enhancement, Image Processing and Surface Computation modules and their corresponding tasks to compute surface structure from raw images.

In order to get the metrics from two stereo images, both cameras need to be calibrated obtaining the intrinsic matrices of both cameras  $K_L$  and  $K_R$  and the relative transformation  ${}^R T_L = [{}^R R_L \quad {}^R t_L]$ , where  ${}^R R_L$  is the rotation of camera  $L$  wrt camera  $R$  and  ${}^R t_L$  is the position of camera  $L$  wrt camera  $R$ . At this point, intrinsic matrices  $K_L$  and  $K_R$  are used to rectify both images removing lens distortion.

Then, we use the Scale Invariant Feature Transform (SIFT) proposed by Lowe [10] to extract distinctive image features. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. This is ideal for wide-base line stereo matching.

Once the matches between both images are obtained by SIFT, we compute the Fundamental matrix to remove false matches not detected by SIFT. Note that it is preferable to be strict at this point removing some correct matches instead of allowing false matches to proceed deteriorating the 3D reconstruction. Since the whole system is calibrated, the Fundamental matrix is computed so that  $m_R^T F m_L = 0$  where:  $F = K_R^{-T} {}^R R_L T K_L^{-1}$ ;  $m_R$  and  $m_L$  are the 2D points of the form  $(x, y, 1)^T$  in pixels, respectively; and  $T$  is the skew matrix of the translation vector  ${}^R t_L$ . Then, we remove matches that do not lay on their corresponding epipolar lines.

Furthermore, we compute the disparity between the remaining 2D points and we remove those whose disparity is larger than  $3\sigma$ , where  $\sigma$  is the square root of the standard deviation of the disparity distribution. This process permits the removal of remaining outliers, since usually outliers suffer large disparity discrepancies.

Once the set of correct matches has been obtained, the 3D structure can be extracted by using a linear triangulation. So, first we transform the pixels to metric coordinates  $\hat{m}_L = K_L^{-T} m_L$  and  $\hat{m}_R = K_R^{-T} m_R$  and then, we compute matrix  $A_i$  for every pair  $i$  of points as follows,

$$A_i = \begin{pmatrix} 0 & -1 & \hat{y}_{Li} & 0 \\ -1 & 0 & \hat{x}_{Li} & 0 \\ (-R_2 + \hat{y}_{Ri} R_3) - t_y + \hat{y}_{Ri} t_z \\ (-R_1 + \hat{x}_{Ri} R_3) - t_x + \hat{x}_{Ri} t_z \end{pmatrix} \quad (15)$$

where  ${}^R R_L = (R_1 \ R_2 \ R_3)^T$  and  ${}^R t_L = (t_x, t_y, t_z)^T$ . Finally, the singular value decomposition of matrix  $A_i$  is computed so that  $A_i = U_i D_i V_i^T$ . The 3D point  $M_i$  corresponds to the fourth column of  $V_i$  before normalization [11].  $M_i$  are measured wrt camera  $L$ .

Finally, we remove isolated 3D points as the ones that have less than 2 neighbours in a certain range distance. Isolated 3D points are not desirable since they introduce large residues in the re-observations of landmarks. The whole process permits the acquisition of a local 3D surface of the imaged seabed measured wrt the current vehicle position.

## V. DATA REPRESENTATION

Let  $X(k)$  be the position of the vehicle at time  $k$  in its six degrees of freedom. Assuming a rigid body motion for the vehicle, the position of the vehicle wrt a fix reference is a the combination of a rotation  $R(k)$  and a translation  $t(k)$ . A partial reconstruction  $S(k)$  of the surface can be associated to each vehicle position  $X(k)$ . If a partial reconstruction is not possible at this time (bad visibility, lack of structure in image), a void surface is stored. The 3D large scale  $S$  can be computed as the union of the partial reconstructions in a global reference frame as:  $S = \bigcup [R(k) \ t(k)] S(k)$ .

### A. Landmark Characterization

A landmark is represented by the cloud of 3D points and their corresponding 2D SIFT descriptors in camera  $L$ . Once the landmark is stored, we also compute landmark position as the gravity centre of the cloud of 3D points. A partial reconstruction  $S(k)$  is selected as a landmark only if the number of 2D points is significative and well spread in the image. This criterion avoids the detection of landmarks in poor textured images. Note that the amount of features per landmark is important in data association. Finally, a new landmark can only be detected if it is at a certain distance of already stored landmarks ensuring that at maximum one landmark is detected per image, keeping the algorithm simple but yet reliable.

### B. Data Association

Each time a new partial reconstruction is obtained, we first check if there are any landmarks in the vicinity. Vicinity is determined as a function of the camera field of view (range and aperture), the navigation data uncertainty and the covariance matrix of the Kalman filter that determines the uncertainty of vehicle position and of every landmark position. For every detected landmark, we match the SIFT descriptors of the current 3D local reconstruction to those of the detected landmark obtaining a number of matches. Then, we compute the Fundamental matrix ( $F$ ) to remove false matches not detected by SIFT. Note that in this case we need to use a F estimator since although the relative transformation

between both images is given by the Kalman filter, it is very imprecisely known to be used in such computation. We have used as  $F$  estimator the technique of Least Median of Squares (LMedS) based on Singular Value Decomposition and point data normalization, which has been proved to perform well compared to other  $F$  estimators [12].

Then, we remove false matches and, finally, we keep as a potential re-observation the landmark in the vicinity that maximizes the number of inliers. For every  $2D$  matches, its corresponding  $3D$  point is known. So, we now have two clouds of  $3D$  points and we can compute the transformation between the two clouds. First, the landmark points are transformed to the vehicle current frame so that now both clouds are in the same reference. Then, the relative transformation  $[R \ t]$  between both clouds of points is computed using the method proposed by Mian [13], which proceeds as follows.

Consider  $M$  and  $S$  the two clouds of  $3D$  points in  $3 \times n$  matrix form, consider  $\hat{m}$  and  $\hat{s}$  their corresponding gravity centers and  $n$  the number of  $3D$  points in any of both clouds ( $n_1 = n_2$ ). Compute the matrix  $\hat{M}$  and  $\hat{S}$  with zero translation subtracting  $\hat{m}$  and  $\hat{s}$  to every point, respectively. Compute  $K = \hat{S}\hat{M}^T/n$  and perform a singular value decomposition to obtain  $K = UAV^T$ . Then, compute the rotation matrix  $R_1 = VU^T$ . Finally, the desired rotation matrix  $R$  is  $R = R_1$  if  $\det(R_1) > 0$  and

$$R = V \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(VU^T) \end{pmatrix} U^T \quad (16)$$

if  $\det(R_1) < 0$ ; and the translation vector  $t$  is  $t = \hat{m} - R\hat{s}$ . The re-observed landmark position  ${}^D L_i$  in the current vehicle (DVL) frame is  ${}^D L_i = R \ {}^D L_s + t$ , where  ${}^D L_s$  is the stored landmark gravity center in the current vehicle frame.

## VI. SIMULATION RESULTS

In the experiment we have simulated a virtual  $3D$  scenario of an underwater environment composed by a  $3D$  surface which can be either introduced by an user or imported. The user can select a real underwater (or aerial) image which is stuck on the  $3D$  surface conforming a virtual  $3D$  scene but yet with a real texture. Note that the texture is deformed according to the shape of the surface. Then the user is asked to introduce the trajectory of the vehicle in 6 degrees of freedom. The algorithm interpolates the introduced trajectory generating the navigation data that is measured by the Doppler Velocity Log. The vehicle is equipped with two virtual stereo cameras. Virtual images are rendered at every vehicle position by means of ray tracing simulating image acquisition.

At every instant of time the Doppler Velocity Log is measuring vehicle orientation wrt Earth and vehicle linear velocity wrt the vehicle frame. Orientation measurement is absolute so a zero-mean Gaussian noise ( $\sigma = 0.01rad.$ ) has been added. Velocity may suffer error propagation and hence a biased Gaussian noise ( $\mu = 0.05m/s, \sigma = 0.08m/s$ ) has been considered. The experiment will show how the SLAM approach is able to readjust vehicle trajectory even in the

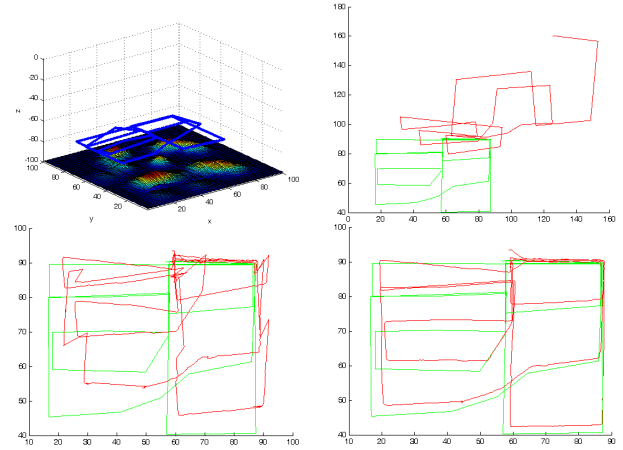


Fig. 4. SLAM Results (from left to right and up to left). Fig.a: Ground truth  $3D$  surface and vehicle trajectory in 6-DOF; Fig.b: the unfiltered trajectory performed by the vehicle (red) compared to ground truth (green) without using SLAM; Fig.c: the filtered trajectory (red) compared to ground truth (green) obtained by the EKF-SLAM algorithm (trajectory jumps are not due to filter inconsistency but to the fact that we are detecting few landmarks to simplify data association); and Fig.d: the smoothed trajectory (red) compared to ground truth (green) obtained by the RTS.

presence of large bias. Besides, the stereo head is capturing two images and the algorithm explained in section IV and depicted in Fig. 3 is executed, obtaining a  $3D$  local map of the imaged scene wrt vehicle frame. Eventually, local maps are considered landmarks and data association as explained in section V-B is carried out. Landmarks are introduced in the EKF wrt Earth and used to filter the trajectory of the vehicle and consequently re-aligning the local maps. Once the whole mission is accomplished the trajectory of the vehicle is smoothed using RTS obtaining an even better alignment of the local maps.

The ground truth  $3D$  surface and vehicle trajectory is depicted in Fig. 4a. The trajectory is composed of 1398 positions. Fig. 4 compares ground truth trajectory (in green) to the unfiltered trajectory (without using SLAM); to the filtered trajectory obtained by SLAM; and, finally, to the smoothed trajectory obtained by RTS.

In order to assert filter consistency, we can check the innovation sequences against the innovation covariance estimates. Fig. 5a shows how the covariance of a given landmark is reduced every time any landmark is re-observed by the vehicle which means that the covariance matrix is fully correlated as desired. Fig. 5b shows the discrepancy of the estimated trajectory to the ground truth and the different landmarks that have been re-visited during the journey.

Fig. 6 shows the interpolated and resampled surface obtained by the EKF-SLAM algorithm and by the post-processing of the RTS smoother, demonstrating qualitatively and quantitatively that our approach obtains an accurate alignment of the  $3D$  surfaces even in the presence of large noises and biases.

Finally, Table I shows the computing time spent in the computation of every task module of the SLAM algorithm

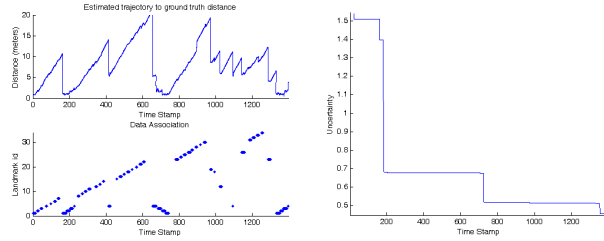


Fig. 5. SLAM Results (from left to right). Fig.a: Discrepancy of the estimated vehicle trajectory with respect to ground truth. The figure shows how the discrepancy is reduced while any landmark is re-observed during the journey; and Fig.b: Covariance estimates of landmark 2. The figure depicts how the covariance is reduced every time any landmark is re-observed.

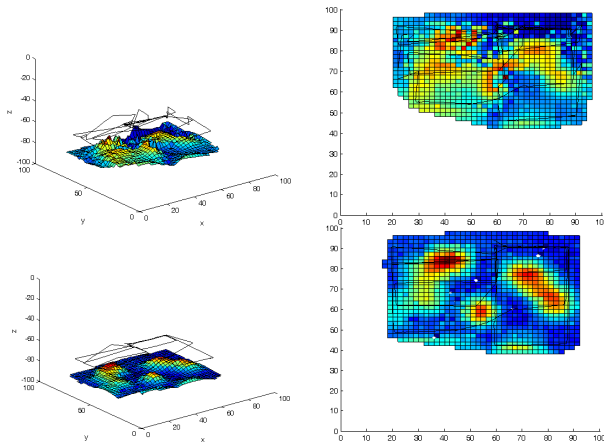


Fig. 6. Alignment of the 1398 local maps corresponding to 93,275 3D points (from left to right and top to bottom): The surface aligned by the EKF-SLAM: 3D view (Fig.a) and Top view (Fig.b); The discrepancy to ground truth shown in Fig. 4 is  $\mu = 4.28m$ . and  $\sigma = 2.80m$ . The surface aligned by the RTS-smoother: 3D view (Fig.c) and Top view (Fig.d). The discrepancy to ground truth shown in Fig. 4 is  $\mu = 0.84m$ . and  $\sigma = 0.78m$ .

shown in Fig. 2. Results have been obtained in a Pentium M 1.20GHz with 1GB of RAM, executing Matlab 7.0.4(R14) under Windows XP. Table I shows for every task the time required to complete a loop and the total time the computer spent in that task during the whole mission (1398 loops). The hardest task concerns the synthetization of the two virtual images of the stereo pair basically cause of the ray tracing of the  $80 \times 80$  pixels of every image. The computation of the local 3D is quite computing expensive due to the number of iterations concerning the computation of the Fundamental matrix and the many checks performed to remove outliers, but still is quite computing efficient. The rest of tasks require far less computing time.

## VII. CONCLUSIONS

This paper has presented an approach to perform the 3D reconstruction of the seabed from the alignment of hundreds of partial reconstructions thanks to EKF-SLAM and benefiting from the navigation data of the underwater vehicle and the re-observation of landmarks by using a unique stereo camera. RTS smoothing is convenient as a post-processing

TABLE I  
COMPUTING TIME

Task	$\hat{t}$ (seconds)	$\sum t$ (seconds)
State prediction	0.02577	36.01
Motion measurement	0.00056	0.78
Synthetize images	3.83609	5,359.02
Compute Local 3D	1.21995	1,704.27
Landmark matching	0.05076	70.91
Compute Kalman Gain	0.11390	159.12
Filter update	0.00070	0.98
Add landmark	0.00003	0.05
Total time	5.2478	7,331.17

step to filter backwards the trajectory computed by EKF-SLAM obtaining a better estimation of the vehicle trajectory and consequently an even better alignment of the seabed. To the best of our knowledge, this paper is the first that proposes SLAM + RTS to deal with the 3D reconstruction of the seabed by just using video cameras.

Although results have been obtained in a virtual scenario, computational cost shows that a local map is computed in few seconds and, hence, it is readily applicable to land and air robotics. However, we should move to a Compressed EKF and/or hierarchical SLAM to keep computing time bounded if the number of landmarks increase drastically. Besides, Kalman smoothing could be implemented fix-lag on-line if the 3D map is required while the vehicle is navigating.

## REFERENCES

- [1] H. Singh, J. Howland, O. Pizarro. Advances in large-area photomosaicking underwater. *IEEE Journal of Oceanic Engineering*, 29(3):872–886, 2004.
- [2] C. Estrada, J. Neira, J.D. Tardos. Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments. *IEEE Trans. on Robotics*, 21(4):588–596, 2005.
- [3] J. J. Leonard, P. M. Newman, R. J. Rikoski, J. Neira, J.D. Tardos. Towards robust data association and feature modeling for concurrent mapping and localization. *Int. Symp. on Robotics Research*, 2001.
- [4] A. J. Davison, D. Murray. Simultaneous Localisation and Map-Building Using Active Vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):865–880, 2002.
- [5] J. Folkesson, H.I. Christensen. Closing the Loop With Graphical SLAM. *IEEE Trans. on Robotics*, 23(4):731–741, 2007.
- [6] K. Ho, P. Newman. Detecting Loop Closure with Scene Sequences. *Journal of Computer Vision* 74(3):261–286, 2007.
- [7] I. Mahon, S. Williams. SLAM using Natural Features in an Underwater Environment. *Int. Conf. on Control, Automation, Robotics and Vision*, pages 2076–2081, 2004.
- [8] I. Tena-Ruiz, S. Raucourt, Y. Petillot, D.M. Lane. Concurrent Mapping and Localization Using Sidescan Sonar. *IEEE Journal of Oceanic Engineering*, 29(2):442–456, 2004.
- [9] S.B. Williams, P. Newman, G. Dissanayake, H. Durrant-Whyte. Autonomous Underwater Simultaneous Localisation and Map Building. *Int. Conf. on Robotics and Automation*, pages 1793–1798, 2000.
- [10] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *Journal of Computer Vision*, 2(60):91–110, 2004.
- [11] Y. Ma, S. Soatto, J. Kosecka, S. Sastry. An Invitation to 3-D Vision: From Images to Geometric Models *Springer-Verlag*, 2003.
- [12] X. Armangué, J. Salvi. Overall view regarding fundamental matrix estimation. *Image and Vision Computing*, 21:205–220, 2003.
- [13] A.S. Mian, M. Bennamoun, R.A. Owens. A Novel Representation and Feature Matching Algorithm for Automatic Pairwise Registration of Range Images. *Journal of Computer Vision* 66(1):19–40, 2006.