Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Towards a Real-Time Vision-Based Navigation System for a Small-Class UUV

R. Garcia, T. Nicosevici, P. Ridao and D. Ribas
Computer Vision and Robotics Group
Institute of Informatics and Applications
University of Girona, E.P.S.
17003 Girona, Spain
e-mail: {rafa,tudor,pere,dribas}@eia.udg.es

Abstract – This paper deals with the problem of navigation for an Unmanned Underwater Vehicle (UUV) through image mosaicking. It represents a first step towards a real-time vision-based navigation system for a small-class low-cost UUV. We propose a navigation system composed by: (*i*) an image mosaicking module which provides velocity estimates; and (*ii*) an Extended Kalman Filter based on the hydrodynamic equation of motion, previously identified for this particular UUV. The obtained system is able to estimate the position and velocity of the robot. Moreover, it is able to deal with visual occlusions that usually appear when the sea bottom does not have enough visual features to solve the correspondence problem in a certain area of the trajectory.

I. INTRODUCTION

When an autonomous vehicle has to carry out a mission, one of the most important aspects is its localization within the mission area. This is a difficult task when the mission takes place in an unstructured location, especially in underwater environments where there is high clutter in the regions of interest and visibility is limited. Nevertheless, for a reliable long-term Unmanned Underwater Vehicle (UUV) mission, precise position and attitude measurements are required. The ideal solution is to use absolute measurements. This is easy for the heading since gyro-compasses are inexpensive and easy to use on-board. On the other hand, problems arise when measuring the position of the vehicle. In the case of underwater vehicles, GPS systems cannot be used while the vehicle is submerged. A possible solution to this problem consists in using a transponder network [1], but this option raises the mission cost, since transponders have to be deployed prior to the mission (Long Base Line) or a mother ship is necessary (Ultra Short Base Line). A low-cost effective way to position the UUV during a mission consists in using dead reckoning techniques. This implies the use of "distance-traveled sensors", which provide readings relative to previous positions of the vehicle. For low-speed robots, where acceleration is very small, Inertial Navigation Systems (INS) are not suitable and Doppler Velocity Log (DVL) systems are usually preferred. Although DVL is an interesting option, its large size and high cost make it unsuitable for the class of UUV we are dealing with (see section III). One of the possible dead reckoning approaches is based on visual mosaics [6]. This technique consists of the use of a down-looking camera carried by the vehicle, which takes images of the sea bottom. The images are analyzed by a system that estimates the motion of the vehicle.

As a dead reckoning system, the typical drawback of visual mosaic is that small errors accumulate at each time instant resulting in increased position drift over long periods. Moreover, when the UUV passes over "blind" regions (*e.g.* dark regions, sand banks, etc.) the mosaicking system does not have enough information to estimate the motion of the vehicle. In the following sections we will refer to this problem as an occlusion.

We conceive the navigation system for a small-class UUV as a system composed by an image mosaicking module, a low-cost INS (for attitude estimation) and a sonar altimeter (for altitude estimation). All these sensors are integrated by means of an Extended Kalman Filter (EKF). This paper presents the first step towards this integrated navigation system. Our proposal consists of using a vision-based sensor (mosaicking) together with the hydrodynamic model of the UUV and integrated through an EKF. The filter provides position and velocity estimates reducing the noise, having also the role of estimating the motion of the vehicle along "blind" regions (occlusions).

The paper is organized as follows. First, a brief overview of the mosaicking system structure and functionality is presented. Next, URIS UUV is described, detailing its model and the implementation of an Extended Kalman filter. Then, some experiments are reported, illustrating the obtained results. Finally, the conclusions are presented, along with further work.

II. THE MOSAICKING SYSTEM

The creation of the mosaic is accomplished in the following stages: First, a detector of *interest points* is used to select the most reliable features of the image and the correspondences of these features are matched in the next image of the sequence. Next, 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

0-7803-7860-1/03/$17.00 © 2003 IEEE

planar transformation matrix relating the coordinates of both images is computed. Finally, the registered images are merged onto a composite mosaic image [6].

A. Selection of Interest Points

The first step of the mosaicking algorithm consists of the selection of adequate interest points in the present image to be matched in the next frame. The selection of robust interest points depends, to a large extent, on the technique used to detect correspondences. Normally, small windows containing high frequencies are quite adequate since they are located at the border of different image textures. For this reason, our interest point detector searches for small zones presenting high spatial gradient information in more than one direction. To do this, the image is convolved with two directional high-pass filters (in the x and y directions). When a feature is selected, the algorithm goes on to search for any other selected features in its neighborhood. If a higher-valued feature exists in this neighborhood, only the best feature is selected as an interest point.

B. Region Matching and Texture Characterization

Finding correspondences between images is not an easy task in computer vision, and even less in underwater imaging. On that account we pay special attention to the matching process, carrying out a two-step approach. First, a block-matching strategy is applied to the gray-level images, selecting a set of candidate matches for a given interest point. Then a texture characterization of the points is used for selecting the best correspondence. For every interest point in the present image I, a correlation function is computed in the next image I'. This is performed by comparing a small $n \times n$ window centered at the interest point $\mathbf{m} = (x, y)$ with all the possible locations of the feature $\mathbf{m}' = (x, y)$ in the next image.

After a process of normalization, the texture vector of the interest point is compared with the textural properties of all the possible matches by means of the weighted Euclidean distance. A texture similarity measure is then obtained for every possible correspondence.

After this process, every candidate match has two measures of similarity: (*i*) a block-matching correlation score; and (*ii*) a texture score produced by feature characterization. By averaging these two values, the best correspondence is selected.

Once this procedure has been accomplished, a unique match is obtained in image I' for every interest point in image I.

C. Estimating the 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 $\mathbf{T}$ which relates the coordinates of a scene point in frame I' with the coordinates of the same point in the previous frame I, *i.e.* $\tilde{\mathbf{m}} = \mathbf{T} \cdot \tilde{\mathbf{m}}'$, where $\tilde{\mathbf{m}} = (x_i, y_i, 1)^T$ and $\tilde{\mathbf{m}}' = (x'_i, y'_i, 1)^T$ denote a correspondence point in two consecutive images; and the symbol ~ indicates that the points are expressed in homogeneous coordinates. The matrix that performs this transformation is known as "homography", and can be computed by SVD if 4 or more pairs of matchings are available [4].

Although an accurate texture analysis is devoted to the matching procedure, some false matches (known as *outliers*) could still appear among the right correspondences. For this reason, a robust estimation method has to be applied. The *Least Median of Squares* (LMedS) algorithm can be used for finding the matrix $\mathbf{T}$ which minimizes the median of the squared residuals [9].

D. Mosaic Construction

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. Then, when image (k+1) has to be added to the mosaic, a 2D planar transformation ${}^{k}\mathbf{T}_{k+1}$ provides its best fitting with respect to the previous image. In order to obtain a global registration from image (k+1) to the mosaic reference frame, the following matrix product has to be performed:

$$ {}^{1}\mathbf{T}_{k+1} = \prod_{i=1..k} {}^{i}\mathbf{T}_{i+1}, \qquad (1) $$

where ${}^{1}\mathbf{T}_{k+1}$ is the homography that produces the co-ordinates of a point in the mosaic image, from the coordinates of the same point in image (k+1).

Once a first estimate of the position of the present image is known in the mosaic frame, this position can be refined by registering the image directly with respect to the mosaic image, reducing accumulation of drift errors [7].

III. URIS UUV

The navigation system proposed in this paper has been designed for URIS, a small-class low-cost UUV developed at the University of Girona. The hull is composed of a stainless steel sphere with a diameter of 350 mm, designed to withstand pressures of 3 atmospheres (30 meters depth). On the outside of the sphere there are two video cameras (forward and down-looking) and 4 thrusters (2 in the X direction and 2 in the Z direction). Due to the stability of the vehicle in pitch and roll, the robot has four degrees of freedom (DOF): *surge*, *sway*, *heave* and *yaw*. Except for the *sway* DOF, the other DOFs can be directly controlled. The robot has an onboard PC-104 computer, running the real-time operating system QNX. In this computer, the low and high-level controllers are executed. An umbilical wire is used for communication, power and video signal transmissions. All the experiments were carried out in a water tank located at our lab, where an artificial environment was setup.

IV. HYDRODYNAMIC MODEL

As described in the literature [5], the non-linear hydrodynamic equation of motion of an underwater vehicle with 6 DOF, in the body fixed frame, can be conveniently expressed as[1]:

$$\begin{gathered} {}^{B}\tau+\mathbf{G}(\eta)-\mathbf{D}\left({}^{B}\upsilon\right){}^{B}\upsilon+\tau_{p}= \\ \left({}^{B}\mathbf{M}_{RB}+\mathbf{M}_{A}\right)\cdot{}^{B}\dot{\upsilon}+\left({}^{B}\mathbf{C}_{RB}({}^{B}\upsilon)+\mathbf{C}_{A}\left({}^{B}\upsilon\right)\right)\cdot{}^{B}\upsilon \end{gathered} \tag{2}$$

where ${}^{B}\tau$ is a vector which contains forces and moments from the thrusters, $\mathbf{G}$ represents the gravity and buoyancy forces, η includes *Roll*, *Pitch* and *Yaw* angles, $\mathbf{D}$ is the matrix of linear and quadratic damping coefficients, ${}^{B}\dot{\upsilon}$ is the acceleration vector of the vehicle, ${}^{B}\upsilon$ is the velocity vector, τ_p are non-modeled perturbations, ${}^{B}\mathbf{M}_{RB}$ is the inertia matrix, $\mathbf{M}_A$ models the added mass matrix, ${}^{B}\mathbf{C}_{RB}$ is the Coriolis matrix of rigid body and $\mathbf{C}_A$ is the Coriolis matrix of added mass.

Through the manipulation of (2) the robot acceleration can be computed. Velocity can be obtained through integration and the position rate of change can be computed through the following kinematic transformation:

$${}^{E}\dot{\eta}=\mathbf{J}(\eta)\cdot{}^{B}\upsilon \tag{3}$$

For URIS UUV, the model can be simplified since it is stable in pitch and roll, so it can be formulated in 4 degrees of freedom. Section V shows the discrete-time model used for the experiments. The dynamic parameters needed in eq. (12), where previously identified through exhaustive experimentation. Refer to [3] for details about the model identification.

V. EXTENDED KALMAN FILTER

Due to the non-linear behavior of the dynamics model of an UUV, an extended Kalman filter has been considered as a convenient method to formulate the navigation problem [8]:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{x-1},\mathbf{u}_k,0) \tag{4}$$

$$\mathbf{P}_k^- = \mathbf{A}_k\mathbf{P}_{k-1}\mathbf{A}_k^T+\mathbf{W}_k\mathbf{Q}_{k-1}\mathbf{W}_k^T \tag{5}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T+\mathbf{V}_k\mathbf{R}_k\mathbf{V}_k^T)^{-1} \tag{6}$$

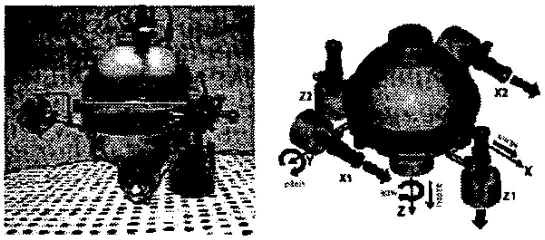

Fig. 1. (left) URIS in the water tank. (right) URIS reference frame.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-,0)) \tag{7}$$

$$\mathbf{P}_k = (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{8}$$

Fig. 2 shows the block diagram of the proposed navigation system. The output of the photo-mosaicking system, the velocity estimation, is used as the measurement update for the EKF. The other input to the filter is the control input *u*, which is the vector of the angular speed of the propellers, from which we calculate the forces and torques affecting the robot. These are used as the input to the model in order to estimate velocity and position.

In the following sections, the main components of the filter are presented.

A. Implementation of the filter

A.1 State vector

To implement the filter the following state variables have been chosen:

$$\mathbf{x}=\begin{pmatrix}x & y & z & \psi & u & v & w & r\end{pmatrix}^T \tag{9}$$

The first 4 components are the 3D position and the heading of the vehicle, while the last 4 components are the linear velocity vector and the angular speed.

A.2 Discrete time model

Adapting eqs. (2) and (3), and simplifying the model to 4 degrees of freedom, the discrete time model shown in eq. (10) is obtained.

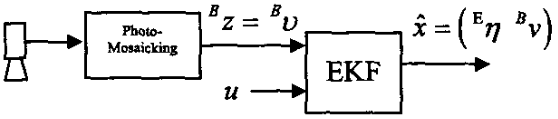
Photo-Mosaicking
${}^{B}z = {}^{B}\upsilon$
u
EKF
$\hat{x}=\left({}^{E}\eta \;\; {}^{B}v\right)$

Fig. 2. Block diagram of the proposed navigation system.

[1] All the equations shown heareafter follow the standard nomenclature proposed in [5].

$$\begin{pmatrix} x \\ y \\ z \\ \psi \\ u \\ v \\ w \\ r \end{pmatrix}_{k+1} = \begin{pmatrix} x+(\cos(\psi)\cdot u-\sin(\psi)\cdot v)\cdot dt \\ y+(\sin(\psi)\cdot u+\cos(\psi)\cdot v)\cdot dt \\ z+w\cdot dt \\ \psi+r\cdot dt \\ u+\left(\dfrac{\mathrm{X}+(m-y_{\dot v})\cdot v\cdot r-(x_u+x_{u|u|}\cdot|u|)\cdot u+\delta_u}{(m-x_{\dot u})}\right)\cdot dt \\ v+\left(\dfrac{\mathrm{Y}+(m-x_{\dot u})\cdot u\cdot r-(y_v+y_{v|v|}\cdot|v|)\cdot v+\delta_v}{(m-y_{\dot v})}\right)\cdot dt \\ w+\left(\dfrac{\mathrm{Z}-z_w\cdot u+\delta_w}{(m-z_{\dot w})}\right)\cdot dt \\ r+\left(\dfrac{\mathrm{N}-(n_r+n_{r|r|}\cdot|r|)\cdot r-((m-y_{\dot v})-(m-y_{\dot u}))\cdot v\cdot u+\delta_r}{(m-n_{\dot r})}\right)\cdot dt \end{pmatrix}_k + \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \end{pmatrix}_k \tag{10}$$

A.3 Matrices of the filter

To complete the filter is also necessary to determine the following matrices:

- **A**: Jacobian matrix of partial derivatives of the model with respect to the state vector.
- **W**: Jacobian matrix of partial derivatives of the model with respect to the process noise vector.
- **H**: Jacobian matrix of partial derivatives of the sensor model respect to the state vector.
- **V**: Jacobian matrix of partial derivatives of the sensor model with respect to the measurement noise vector.
- **Q**: covariance matrix of process noise.
- **R**: covariance matrix of measurement noise.

The set of previously described matrices is not reproduced here due to space limitations. However, the peculiar behavior of matrix **H** is detailed below.

$$\mathbf{H}=diag\begin{pmatrix}0 & 0 & 0 & 0 & 1 & 1 & 0 & 1\end{pmatrix} \qquad \mathbf{H}=0_{8x8} \tag{11}$$

Matrix **H** has two different forms, as can be seen in (11). The first value is used to merge the velocity measures from the mosaic based sensor with the estimate of the model. Note that the mosaic provides information about velocities in the (*X*-*Y*) plane and angular speed in the yaw DOF. The second configuration of **H** isolates the model estimation, or what is the same, no measurement update is taken into account to correct the prediction of the model. It could be used to disconnect a damaged sensor, or as in the experiments described in the following section, to deal with an occlusion.

Covariance matrices **Q** and **R** describe the process and measurement noises. Both are considered diagonal to simplify the tuning of the filter. The values in the diagonal are the noise variances affecting their corresponding state variables.

$$\mathbf{Q}=diag\begin{pmatrix}\sigma_X^2 & \sigma_Y^2 & \sigma_Z^2 & \sigma_\Psi^2 & \sigma_u^2 & \sigma_v^2 & \sigma_w^2 & \sigma_r^2\end{pmatrix} \tag{12}$$

$$\mathbf{R}=diag\begin{pmatrix}\sigma_{z_1}^2 & \sigma_{z_2}^2 & \sigma_{z_3}^2 & \sigma_{z_4}^2\end{pmatrix} \tag{13}$$

VI. EXPERIMENTAL RESULTS

In this section we present a representative navigation experiment carried out in a water tank located at our lab. An artificial bottom was set up on the bottom of the water tank for this experiment. The robot was teleoperated following an approximately circular trajectory. The duration of the experiment was 82 seconds. During the experiment, the thruster set-points, as well as the images acquired by the camera (2,041 images), were saved together with their time stamps. Afterwards, using this sampled data, two different experiments were carried out offline. The first one consisted in estimating the trajectory of the vehicle using only the information provided by the image sequence. In the second one, an occlusion was simulated discarding a sequence of images during a certain period of time, and the trajectory was estimated by taking into account all the available data (images and navigation data).

A. Position and velocity estimation

In the first set of experiments the main goal was to tune the EKF (*i.e.* adjust the values in the diagonal of covariance matrices **Q** and **R**). The images acquired by the mosaicking system were processed to estimate the trajectory followed by the vehicle. The filter was tuned by finding a set of parameters which provide a response similar to the one estimated by the mosaicking system but filtering the noise, therefore, obtaining a smooth velocity estimate. To find the best set of variance values it was necessary to perform multiple tests and compare its responses.

As shown in Fig. 3, good velocity estimations were obtained with the EKF in the *surge*, *sway* and *yaw* DOF. We can observe that the filter is able to reduce the noise, while providing a non-delayed estimation. There is, in general, a good agreement between the model and the measurements, except for *sway* velocity. Note that the prediction of the model moves towards the positive side while the mosaic data oscillates around zero. For movements like the one performed in this experiment, where the vehicle describes a circular trajectory, we can expect a displacement in the sway DOF due to the centrifuge force, so we can consider that the model response is reasonable. In the data from the mosaic we could not see this behavior; instead we observed an oscillation in the *sway* velocity. The response of the resulting mosaic is probably caused by a small oscillation in the *roll* DOF. A "similarity" motion model (with 4 DOF: translation in *X* and *Y*, rotation and scaling) was assumed for building the mosaic, considering *roll* to be zero. Hence this small movement in *roll* would be interpreted as oscillatory *sway* displacements.

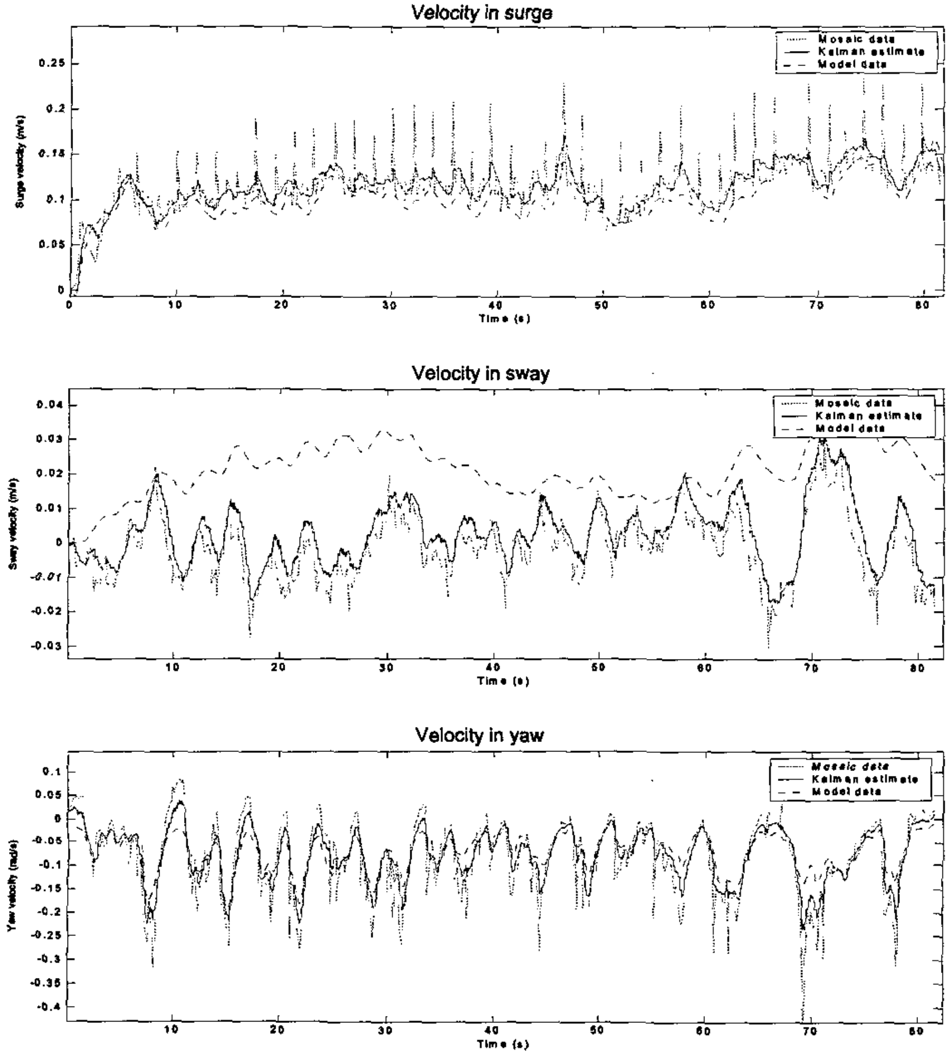

Fig. 3. Velocities obtained from derivation of the position measured with the mosaic, in dashed line, in comparison with the Kalman filtered ones, in normal line. The model predicted velocity is illustrated as a dotted one.

B. Occlusion test

The second experiment was performed in order to test the capability of the system to navigate in the presence of occlusions. In such cases, no measurement update can be provided by the mosaicking system, so the only available information to estimate the trajectory of the vehicle is the data provided by the model.

We wanted to test whether the dynamic model of the vehicle can be used to obtain navigation estimates when no other sensor data is available (occlusion situation). The EKF provides a natural way to deal with occlusions, since matrix **H** can be switched when no visual information is available. Fig. 6 shows the results of the experiment including one occlusion, delimited by the two stars. The blind navigation took place for 19 seconds (480 frames of the mosaicking system). During this period of time, the filter followed the prediction of the model without taking into account any visual information. As expected, a certain deviation takes place due to the absence measurements. However, it is smaller than the one that would be obtained if a constant velocity model was to be used.

VII. CONCLUSIONS

In this paper we have presented a navigation system based on the integration of a photo-mosaicking sensor and the hydrodynamic model of small-class low-cost UUV. An

EKF has been used for the integration and it is expected that it will allow a simple integration of new sensors (INS, sonar altimeter, etc.). The capability of the system to provide undelayed velocity estimates has been shown. Moreover, the proposed system has proved to provide good position and velocity estimates along "blind" regions.

VIII. FURTHER WORK

One of the problems in the assessment of the proposed navigation system is the need of a reference system to compare with. In this sense, in [2] we developed a high-accuracy navigation system based in a coded pattern located in the bottom of the pool.

In the future we plan to modify this system by placing the pattern on the wall of the pool. In this way, the accuracy of the system proposed in this paper could be evaluated.

IX. REFERENCES

[1] D.R. Blidberg and J.C. Jalbert, "Mission & System Sensors," *Underwater Robotic Vehicles – Design and Control*, pp. 185–221, 1995.

[2] M. Carreras, P. Ridao, R. Garcia and T. Nicosevici, "Vision-based localization of an underwater robot in a structured environment," *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.

[3] M. Carreras, A. Tiano, A. El-Fakdi, A. Zirilli and P. Ridao, "On the identification of non linear models of unmanned underwater vehicles", *1st IFAC Workshop on Guidance and Control of Underwater Vehicles*, Wales, UK, 2003.

[4] X. Cufi, R. Garcia and P. Ridao, "An Approach to Vision-Based Station Keeping for an Unmanned Underwater Vehicle," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 1, pp. 799-804, Lausanne, Switzerland, 2002.

[5] T.I. Fossen, *Guidance and Control of Ocean Vehicles*, John Wiley and Sons, New York, USA, 1994.

[6] R. Garcia, J. Batlle, X. Cufi and J. Amat, "Positioning an Underwater Vehicle through Image Mosaicking," *in Proc. IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, vol. 3, pp. 2779–2784, 2001.

[7] R. Garcia, X. Cufi and V. Ila, "Recovering Camera Motion in a Sequence of Underwater Images through Mosaicking, *Iberian Conference on Pattern Recognition and Image Analysis*, , pp. 255–262, 2003.

[8] R. Grober and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley and Sons. New York, USA, 1983.

[9] P. Rousseeuw and A. Leroy, *Robust Regression and Outlier Detection*, John Wiley & Sons, New York, 1987.

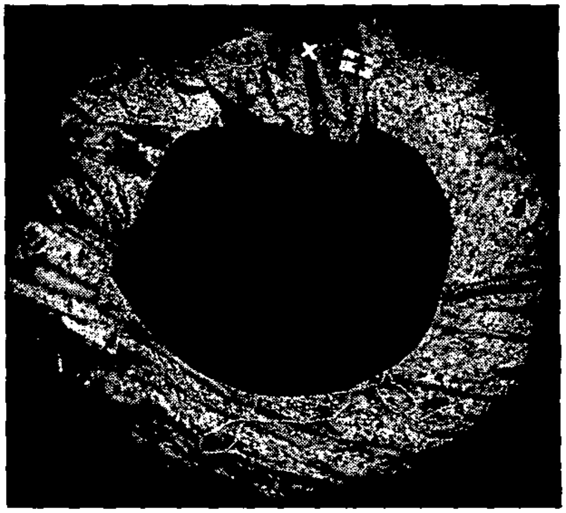

Fig. 5. Resulting photo-mosaic obtained directly from the mosaicking system after processing a sequence of 2041 images.

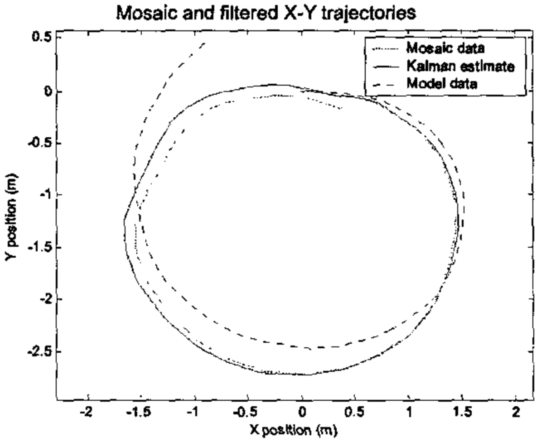

Fig. 4. Trajectory obtained from the mosaic, (dashed line), in comparison with the Kalman filtered one, (solid line. The model-predicted trajectory is the dotted line.

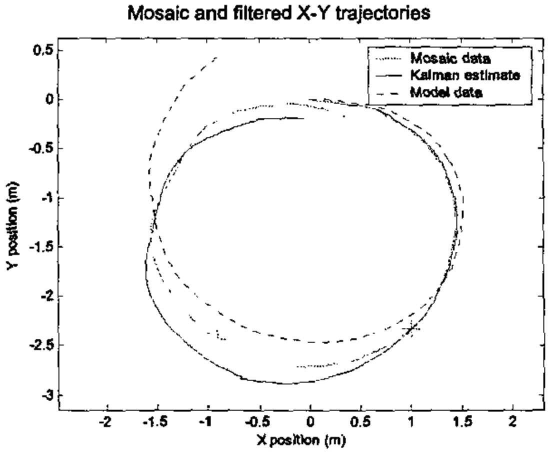

Fig. 6. Trajectory obtained from the mosaic, (dashed line), in comparison with the Kalman filtered one, (solid line). The model predicted trajectory is the dotted one. The stars mark the occlusion zone.