

AUV NAVIGATION IN A STRUCTURED ENVIRONMENT USING COMPUTER VISION

M. Carreras, P. Ridao, J. Batlle and X. Cufi¹

*Institute of Informatics and Automation
University of Girona, Spain*

Abstract: This paper presents a vision-based navigation approach for an underwater robot in a structured environment. The system is based on a coded pattern placed on the bottom of a water tank and an onboard down-looking camera. Main features are, absolute and map-based navigation, landmark detection and tracking, and real-time computation (12.5 Hz). The proposed system provides three-dimensional position and orientation of the vehicle along with its velocity. Accuracy of the drift-free estimates is very high, allowing them to be used as feedback measures of a velocity-based low level controller. The paper details the navigation algorithm and shows some results. Copyright © 2003 IFAC

Keywords: Marine systems, Robot navigation, Computer Vision

1. INTRODUCTION

The positioning of an underwater vehicle is a big challenge. Techniques involving inertial navigation systems, acoustic or optical sensors have been developed to estimate the position and orientation of the vehicle. Among these techniques, visual mosaics have greatly advanced during last years offering, besides position, a map of the environment, see (Negahdaripour *et al.*, 1999) and (Gracias and Santos-Victor, 2000). Main advantages of mosaicking with respect inertial and acoustic sensors are smaller cost and smaller sensor size. Another advantage respect to acoustic transponder networks is that the environment does not require any preparation. However, position estimation based on mosaics can only be used when the vehicle is performing tasks near the ocean floor and requires a reasonable visibility in the working area. There are also unresolved problems like motion estimation in presence of shading effects, presence

of "marine snow" or non-uniform illumination. Moreover, as the mosaic evolves, a systematic bias is introduced in the motion estimated by the mosaicking algorithm, producing a drift in the navigation of the robot (Garcia *et al.*, 2002).

Current work on underwater vehicle positioning at the University of Girona concentrates on visual mosaics (Garcia *et al.*, 2001). While a real time application which deals with the mentioned problems is being developed, a simplified positioning system was implemented. The aim of it is to provide an accurate estimation of the position and velocity of URIS Autonomous Underwater Vehicle (AUV) in a water tank, see Figure 1. The utility of this water tank is to experiment in different research areas, like dynamics modelling or control architectures, in which the position and velocity of the vehicle are usually required.

In this paper we present an original vision-based navigation system to estimate the position, orientation and velocity of an underwater robot in a structured environment. Main features of this system are absolute and map-based navigation,

¹ This work was supported by the Spanish commission MCYT (DPI2001-2311-C03-01)

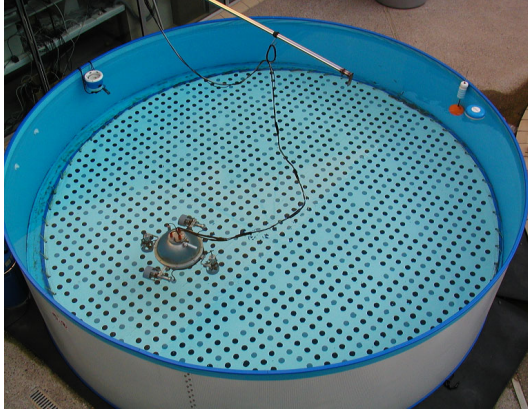


Fig. 1. URIS's experimental environment

landmark detection and tracking, and real-time computation. The components of the system are an onboard down-looking camera and a coded pattern placed on the bottom of the water tank. The algorithm calculates the three-dimensional position and orientation referred to the water tank coordinate system with a high accuracy and drift-free. An estimation of the vehicle's velocities, including *surge*, *sway*, *heave*, *roll*, *pitch* and *yaw*, is also computed. These estimates are used by the velocity-based low level controller of the vehicle.

The structure of this paper is as follows: section 2 describes URIS's underwater vehicle and its experimental setup. In this section emphasis is given to the down-looking camera and to the visually coded pattern, both used by the navigation system. Section 3 details the navigation algorithm explaining the different phases. In section 4, some results which show the accuracy of the system are presented. And finally, conclusions are given in section 5.

2. URIS'S EXPERIMENTAL SETUP

In order to experiment with URIS underwater vehicle, a water tank is used, see Figure 1. The shape of the tank is a cylinder with 4.5 meters in diameter and 1.2 meters in height. This environment allows the perfect movement of the vehicle in the horizontal plane and a restricted vertical movement of only 30 centimeters. The navigation system is compounded by a coded pattern which covers the whole bottom of the tank and a down-looking camera attached on URIS. Next subsections describe URIS, the model of the camera and the coded pattern.

2.1 URIS's Autonomous Underwater Vehicle

The robot for which has been designed this navigation system is URIS, see Figure 2. This vehicle was developed at the University of Girona with

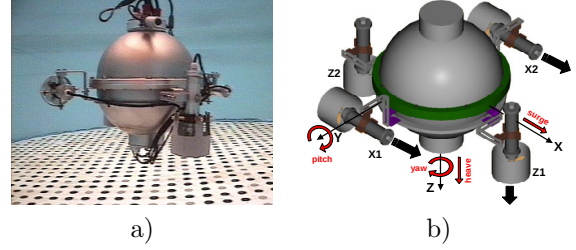


Fig. 2. URIS's AUV, a) picture b) schema

the aim of building a small-sized AUV. The hull is composed of a stainless steel sphere with a diameter of 350mm, 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 X direction and 2 in 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 others DOFs can be directly controlled.

The robot has an onboard PC-104 computer, running the real-time operative 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. The navigation system is currently being executed on an external computer. A new onboard computer for video processing purposes will be incorporated in the near future.

2.2 Down-Looking Camera Model

The camera used by the positioning system is an analog B/W camera. It provides a large field of view (about 57° in width by 43° in height underwater). The camera model that has been used is the Faugeras-Toscani algorithm (Faugeras and Toscani, 1986) in which only a first order radial distortion has been considered. This model is based on the projective geometry and relates a three-dimensional position in the space with a two-dimensional position in the image. These are the equations of the model:

$$\frac{{}^C X}{{}^C Z} = \frac{(x_p - u_0)(1 + k_1 r^2)}{f k_u} \quad (1)$$

$$\frac{{}^C Y}{{}^C Z} = \frac{(y_p - v_0)(1 + k_1 r^2)}{f k_v} \quad (2)$$

$$r = \sqrt{\left(\frac{x_p - u_0}{k_u}\right)^2 + \left(\frac{y_p - v_0}{k_v}\right)^2} \quad (3)$$

where, $({}^C X, {}^C Y, {}^C Z)$ are the coordinates of a point in the space respect the camera coordinate system $\{C\}$ and (x_p, y_p) are the coordinates, measured in pixels, of this point projected in the image

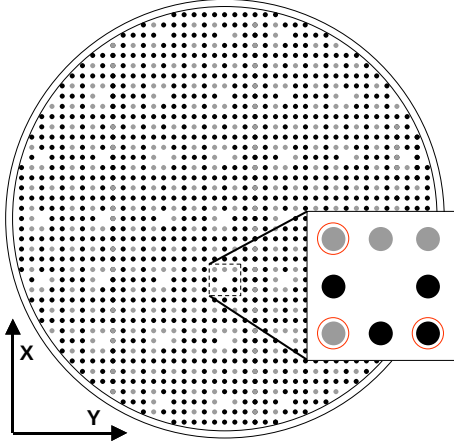


Fig. 3. Visually Coded pattern. The absence of a dot identifies a global mark. The dots marked here with a circle are used to find the orientation of the pattern

plane. And, as intrinsic parameters of the camera: (u_0, v_0) are the coordinates of the center of the image, (k_u, k_v) are the scaling factors, f is the focal distance, k_1 is the first order term of the radial distortion and r is the distance, in length units, between the projection of the point and the center of the image.

The calibration of the intrinsic parameters of the camera was done off-line using several representative images and applying an optimization algorithm, which by iteration, estimated the optimal parameters.

2.3 Visually Coded Pattern

The main goal of the pattern is to provide a set of known global positions to estimate, by solving the projective geometry, the position and orientation of the underwater robot. The pattern is based on grey level colors and only round shapes appear on it to simplify the landmark detection, see Figure 3,a). Each one of these rounds or dots will become a global position used in the position estimation. Only three colors appear on the pattern, white as background, and grey or black in the dots. Again, the reduction of the color space was done to simplify the dots detection and to improve the robustness. The dots have been distributed among the pattern following the X and Y directions, see Figure 3. These two directions are called the *main lines of the pattern*.

The pattern contains some global marks, which encode a unique global position. These marks are recognized by the absence of one dot surrounded by 8 dots, see Figure 3. From the 8 dots that surround the missing one, 3 are used to find the orientation of the pattern and 5 to encode the global position. The 3 dots which mark the orien-

tation, appear in all the global marks in the same position and with the same colors. The detailed view seen in Figure 3 shows with a circle these 3 dots. The global position is encoded in the binary color (grey or black) of the 5 remainder dots. The maximum number of positions is 32. These global marks have been uniformly distributed on the pattern. A total number of 37 global marks have been used, repeating 5 codes in opposite positions on the pattern. The zones of the pattern that do not contain a global mark, have been fulfilled with alternately black and grey dots, which helps the tracking algorithm, as will be explained in Section 3.3.

In order to choose the distance between two neighbor dots several aspects were taken into account. A short distance represents a higher number of appearing dots in the image, and therefore, a more accurate estimation of the vehicle's position. On the other hand, if a lot of dots appear in the image and the vehicle moves fast, dot tracking can be very hard or impractical. A long distance between two neighbor dots produces the contrary effects. Therefore, an intermediate distance was chosen for this particular application. The aspects which influenced the decision were the velocities and oscillations of the vehicle, the camera field of view and the range of depths in which the vehicle can navigate. The final distance between each two neighbor dots was 10 cm.

3. NAVIGATION PROCEDURE

The vision-based navigation algorithm was designed to work at 12.5 frames per second, half of the video frequency. Each iteration requires a set of sequential tasks starting from image acquisition to velocity estimation. Next subsections describe the phases that constitute the whole procedure.

3.1 Pattern Detection

The first phase of the navigation algorithm consists in detecting the dots of the pattern. To accomplish this phase a binarization is first applied to the acquired image, see Figure 4a and 4b. Due to the non-uniform sensitivity of the camera in its field of view, a correction of the pixel grey level values is performed before binarization. This correction is based on the illumination-reflectance model (Gonzalez and Woods, 1992) and provides a robust binarization of the pattern also under non-uniform lighting conditions.

Once the image is binarized, the algorithm finds the objects and checks the area and shape of them, dismissing the ones that do not match the characteristics of a dot object. Finally, for each

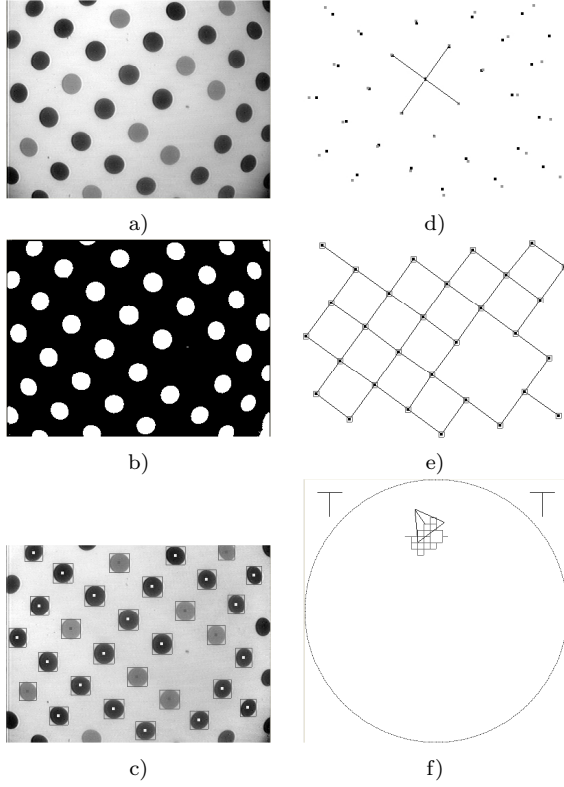


Fig. 4. Phases of the navigation system: a) acquired image, b) binarization, c) detection of the dots, d) main lines of the pattern, e) dots neighborhood, f) estimated position and orientation

detected dot, the algorithm classifies its grey level labelling them in three groups: *grey*, *black* or *unknown*. In case the label is unknown, the dot will be partially used in next phases, as Section 3.3 details. Figure 4c shows the original image with some marks on the detected dots.

3.2 Dots Neighborhood

The next phase in the navigation system consists in finding the neighborhood relation among the detected dots. The goal is to know which dot is next to which one. This will allow the calculation of the global position of all of them, starting from the position of only one. Next phase will consider how to find this initial position.

The first step, in this phase, is to compensate the radial distortion that affects the position of the detected dots in the image plane. In Figure 4d, the dots before distortion compensation are marked in black and, after the compensation, in grey. The new position of the dots in the image is based on the ideal projective geometry. This means that lines in the real world appear as lines in the image. Using this property, and also by looking at relative distances and angles, the main lines of the pattern are found. Figure 4d shows the detected main lines

of the pattern. To detect the main lines, at least 6 dots must appear in the image.

Next step consists in finding the neighborhood of each dot. The algorithm starts from a central dot, and goes over the others according to the direction of the main lines. To assign the neighborhood of all the dots, a recursive algorithm was developed which also uses distances and angles between dots. After assigning all the dots, a network joining all neighbor dots can be drawn, see Figure 4e.

3.3 Dots Global Position

Two methodologies are used to identify the global position of the detected dots. The first one is used when a global mark is detected, what means that, a missing dot surrounded by 8 dots appears on the network and, any of them has the *unknown* color label, see Figure 4e. In this case, the algorithm checks the three orientation dots to find how the pattern is oriented. From the four possible orientations, only one matches the three colors. After that, the algorithm checks the five dots which encode a memorized global position. Then, starting from the global mark, the system calculates the position of all the detected dots using the dot neighborhood.

The second methodology is used when any global mark appears on the image, or when there are dots of the global mark which have the color label *unknown*. It consists on *tracking* the dots from one image to the next one. The dots that appear in the same zone in two consecutive images are considered to be the same, and therefore, the global position of the dot is transferred. The high speed of the navigation system, compared with the slow dynamics of the underwater vehicle, assures the tracking performance. The algorithm distinguishes between grey and black dots, improving the robustness on the tracking. Also, because different dots are tracked at the same time, the transferred positions of these dots are compared, using the dot neighborhood, and therefore, mistakes are prevented.

3.4 Position and orientation estimation

Having the global positions of all the detected dots, the navigation of the robot can be carried out. Equation 4 contains the homogeneous matrix which relates the position of one point (X_i, Y_i, Z_i) respect the camera coordinate system $\{C\}$, with the position of the same point respect to the water tank coordinate system $\{T\}$. The parameters of this matrix are the position $({}^T X_C, {}^T Y_C, {}^T Z_C)$ and orientation $(r_{11}, \dots, r_{33})$ of the camera respect $\{T\}$.

The nine parameters of the orientation depend only on the values of *roll*, *pitch* and *yaw* angles.

$$\begin{pmatrix} {}^T X_i \\ {}^T Y_i \\ {}^T Z_i \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & {}^T X_C \\ r_{21} & r_{22} & r_{23} & {}^T Y_C \\ r_{31} & r_{32} & r_{33} & {}^T Z_C \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} {}^C X_i \\ {}^C Y_i \\ {}^C Z_i \\ 1 \end{pmatrix} \quad (4)$$

For each dot i , the position $({}^T X_i, {}^T Y_i, {}^T Z_i)$ is known, as well as the ratios:

$$\frac{{}^C X_i}{{}^C Z_i} \text{ and } \frac{{}^C Y_i}{{}^C Z_i} \quad (5)$$

which are extracted from Equations 1 and 2. These ratios can be applied to Equation 4 eliminating ${}^C X_i$ and ${}^C Y_i$. Also, ${}^C Z_i$ can be eliminated by using the next equation:

$$({}^T X_i - {}^T X_j)^2 + ({}^T Y_i - {}^T Y_j)^2 + ({}^T Z_i - {}^T Z_j)^2 = ({}^C X_i - {}^C X_j)^2 + ({}^C Y_i - {}^C Y_j)^2 + ({}^C Z_i - {}^C Z_j)^2 \quad (6)$$

in which the distance between two dots, i and j , calculated respect $\{T\}$ is equal to the distance respect $\{C\}$. Using Equation 6 together with 4 and 5 for dots i and j , an equation with only the camera position and orientation is obtained. And repeating this operation for each couple of dots, a set of equations is obtained from which an estimation of the position and orientation can be performed. In particular, a two-phase algorithm has been applied. In the first phase, ${}^T Z_C$, *roll* and *pitch* are estimated using the non-linear fitting method proposed by Levenberg-Marquardt. In the second phase, ${}^T X_C$, ${}^T Y_C$ and *yaw* are estimated using a linear least square technique. Finally, the position and orientation calculated for the camera are recalculated for the vehicle. Figure 4f shows the vehicle position in the water tank marked with a triangle. Also the detected dots are marked on the pattern.

3.5 Filtering

Main sources of error that affect the system are the imperfections of the pattern, the simplification on the camera model, the intrinsic parameters of the camera, the accuracy in detecting the centers of the dots and, the error of least-square and Levenberg-Marquardt algorithms on its estimations. These errors cause small oscillations on the vehicle position and orientation even when the vehicle is not moving. To eliminate these oscillations, a first order Savitzky-Golay (Savitzky and Golay, 1964) filter was used. Figure 5 shows the estimated three-dimensional position and orientation with and without filtering. Finally, the velocity of the robot respect the onboard coordinate system

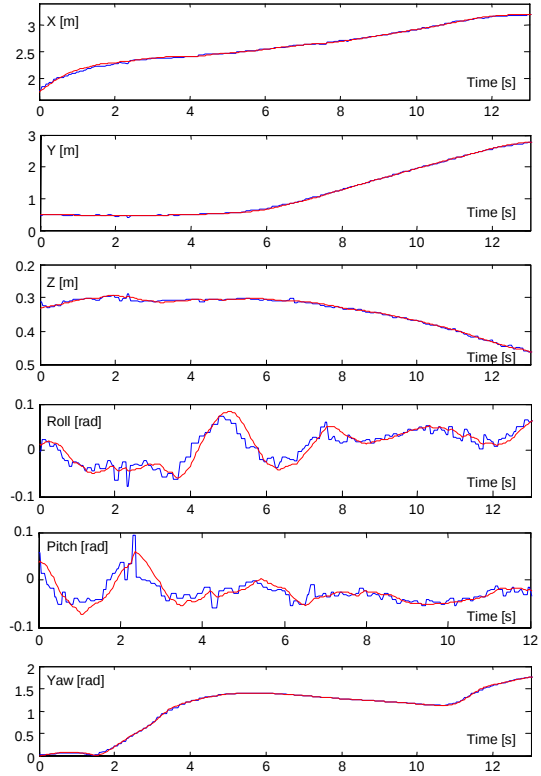


Fig. 5. Position and orientation before and after filtering

is also estimated applying a first order Savitzky-Golay filter with a first order derivative included on it. Refer to section 4 to show results about the estimated velocities.

4. RESULTS

The vision based navigation system, which has been presented in this paper, offers a very accurate estimation of the position, orientation and velocity². An example of a trajectory measured by the navigation system can be seen in Figure 6.

After studying the nature of the source of errors (refer to Section 3.5), it has been assumed that the navigation system behaves as an aleatory process in which the mean of the estimates coincides with the real position of the robot. It is important to note that the system estimates the position knowing the global position of the dots seen by the camera. In normal conditions, the tracking of dots and the detection of global marks never fails, what means that there is not drift in the estimates. By normal conditions we mean, when the water and bottom of the pool are clean, and there is indirect light of the Sun. After measuring the estimations in different locations and having the robot completely static, the standard deviation was calculated obtaining these values: 0.006[m] in

² Some videos showing the performance of the system can be seen at: <http://eia.udg.es/~marcc/research>

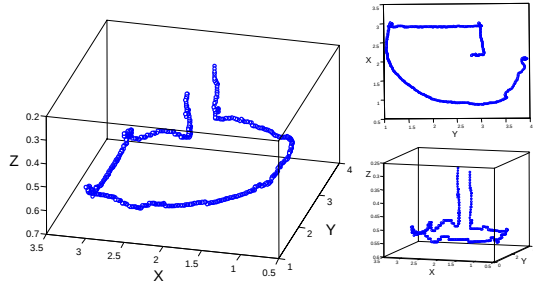


Fig. 6. Three-dimensional trajectory measured by the navigation system. Three views are shown

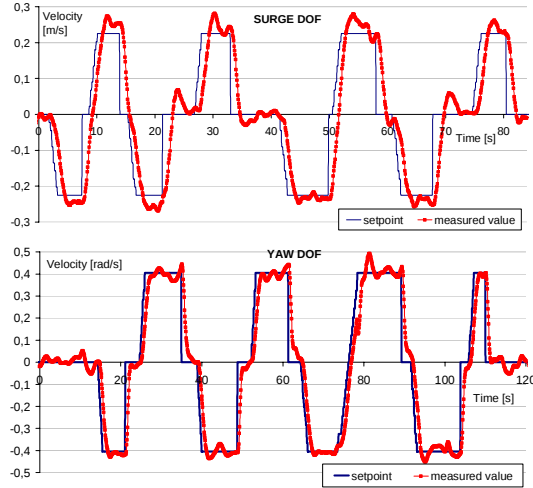


Fig. 7. Performance of the surge and yaw velocity-based controllers

X and Y , $0.003[m]$ in Z , $0.2[^\circ]$ in $roll$, $0.5[^\circ]$ in $pitch$ and $0.2[^\circ]$ in yaw .

The accuracy on the velocity estimations is also very high. These measurements are used by the low level controller of the vehicle which controls the *surge*, *heave* and *yaw* velocities. In Figure 7 the performance of the *surge* and *yaw* controllers is shown.

The only drawback of the system is the pattern detection when direct light of the Sun causes shadows to appear in the image. In this case, the algorithm fails in detecting the dots. Any software improvement to have a robust system in front of shadows would increase the computational time, and the frequency of the algorithm would be too slow. However, the algorithm is able to detect these kind of situations, and the vehicle is stopped.

5. CONCLUSIONS

This paper has presented a vision-based navigation system for an underwater robot in a structured environment. The paper has detailed the experimental set-up, as well as, the different phases of the algorithm. Main feature of the system is its high-accuracy drift-free estimations. The system is fully integrated on the vehicle's controller, giving

new measures 12.5 times per second. Due to the high accuracy of the system, other measures like the heading from a compass sensor, or the depth from a pressure sensor, are not needed. In addition, the navigation system can also be used to evaluate the performance of the video mosaicking system, designed to work in unstructured environments.

REFERENCES

- Faugeras, O. D. and G. Toscani (1986). The calibration problem for stereo. In: *Proc. of the IEEE Computer Vision and Pattern Recognition*. pp. 15–20.
- Garcia, R., J. Batlle, X. Cufi and J. Amat (2001). Positioning an underwater vehicle through image mosaicking. In: *IEEE International Conference on Robotics and Automation*. Rep.of Korea. pp. 2779–2784.
- Garcia, R., J. Puig, P. Ridao and X. Cufi (2002). Augmented state kalman filtering for auv navigation. In: *IEEE International Conference on Robotics and Automation*. Washington. pp. 4010–4015.
- Gonzalez, R.C. and R.E. Woods (1992). *Digital Image Processing*. Addison-Wesley. Reading, MA.
- Gracias, N. and J. Santos-Victor (2000). Underwater video mosaics as visual navigation maps. *Computer Vision and Image Understanding* **79**(1), 66–91.
- Negahdaripour, S., X. Xu and L. Jin (1999). Direct estimation of motion from sea floor images for automatic station-keeping of submersible platforms. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **24**(3), 370–382.
- Savitzky, A. and M.J.E. Golay (1964). *Analytical Chemistry*. pp. 1627–1639. Vol. 36.