# USING STATISTICAL MIXTURE MODELS FOR TRACKING NATURAL UNDERWATER BOUNDARIES

**Maria João Rendas** **Christian Barat**

**Laboratoire I3S (Informatique, Signaux et Systèmes de Sophia Antipolis), CNRS-UNSA**

2000 route des Lucioles, BP 121, 06903 Sophia Antipolis cedex, FRANCE

rendas@i3s.unice.fr barat@i3s.unice.fr

## Abstract

The paper presents signal processing and control algorithms that enable autonomous tracking of the boundaries between distinct benthic regions by an AUV equipped of a profiler sonar. By exploiting sonar scans of the region below the robot, a classical control loop is closed around the sonar data, using a feedback signal that is robust with respect to classification "noise".

## Introduction

This paper presents work on the definition and implementation of a perception-based tracking behavior for underwater platforms, enabling an autonomous robot to track the boundary between distinct benthic habitats. This kind of autonomous observation behavior can be useful in a variety of applications, such as the observation of the patchy structure of some biological species, mapping of regions of contamination or of concentration of a given product, and avoidance of dangerous operational regions. While several underwater robotics groups have addressed in the past the problem of tracking perceptual features using vision sensors – mostly man-made structures, such as piplelines – we believe that the use of acoustic sensors provides a more robust approach, being able to operate in troubled waters and at larger depths, without the need to resort to artifical lightning.

The ability to track natural features is, moreover, an intrinsic tool in the definition of completely autonomous navigation systems at a larger scale, in the framework of the approach proposed in [1,2]. In this work, we define an integrated perception/guidance approach for large scale navigation of autonomous robots that have no external measures of their geographical position (no GPS, nor transponders located at calibrated positions), completely based on the use of a map that is incrementally learned by the robot during its mission. This approach has been fully validated using a terrestrial mini-robot equipped of short-range distance sensors, operating in a planar environment where several medium-sized objects of arbitrarily curved shape are placed at distances which are very large compared to the range of the robot's sensors. The robot incrementally maps the shape and the relative positions of the objects detected, and uses this map to periodically reset its position error. When executing its mission, the robot alternates between "useful" goal driven periods and periods of environment exploration. Exploration of the environment is triggered when the current representation is not rich enough to guarantee safe progression of the robot. In particular, the risk that the platform gets lost is permanently updated, and the necessary actions are undertaken to maintain its value below a pre-specified threshold. The present work is a contribution to the extension of this approach to underwater environments, providing the guidance mode that enables direct autonomous acquisition of interesting detected features. In this paper, we concentrate on the definition of the perception guidance mode, presenting both the signal processing and control algorithms that support it. Future communications will address the problems of incremental map update and of map use for navigation.

The paper is organised as follows. We first formulate the tracking problem. The next section is dedicated to data processing issues: how the statistical model of received data is learned, and how this model is used to compute a signal that indicates the offset of the robot with respect to the tracked boundary. The following section defines the controller structure. The subsequent section presents data acquired during real tracking experiments conducted at sea, illustrating the achieved level of performance. Finally, the last section summarizes the main results of the paper, and lists some ideas for future work.

## Problem formulation and approach

Our goal is to define signal processing and control algorithms that enable autonomous observation of the boundary between distinct natural benthic regions (either corresponding to distinct species or to different materials) by an autonomous robot. Let $C(\ )$, $?L$ be the 2D curve of length $L$, corresponding to the projection of the tracked boundary in an horizontal plane, parametrised by arc length . This curve is an abstraction of real boundaries between natural habitats, which are, most often, a transition region of varying width, inside which the characteristics of the sea floor, or the relative mixture of distinct species, gradually change from one type to another. A simple mathematical model of these transition regions is the dilation of $C(\ )$ -- the skeleton of the transition region – by circle of varying radius:

$$T ? \quad ?( )B?C( )? , \qquad (1)$$
$$?L$$

where $B(x)$ denotes the unit circle of center *x*, and $?( )$ is the width of the transition region at point $C( )$.

We assume in this paper that the *boundary is uniform*, i.e., that the classes on each side of the transition region *T* do not change during the entire interval *L*. Moreover, we assume that the vertical motion of the paltform is controlled independently of its evolution in the horizontal plane, which is true for the underwater robot used in this study. We thus assess the control problem in 2D (surge speed and yaw)., assuming that the robot is kept at a constant altitude above the (locally flat) sea bottom by an separate control loop which is not described here.

To map the transition region, the platform must be able to use its perception sensors to describe a trajectory that covers *T*, permamently keeping the transition region in the field of view of the sensor, as shown in Figure 1.

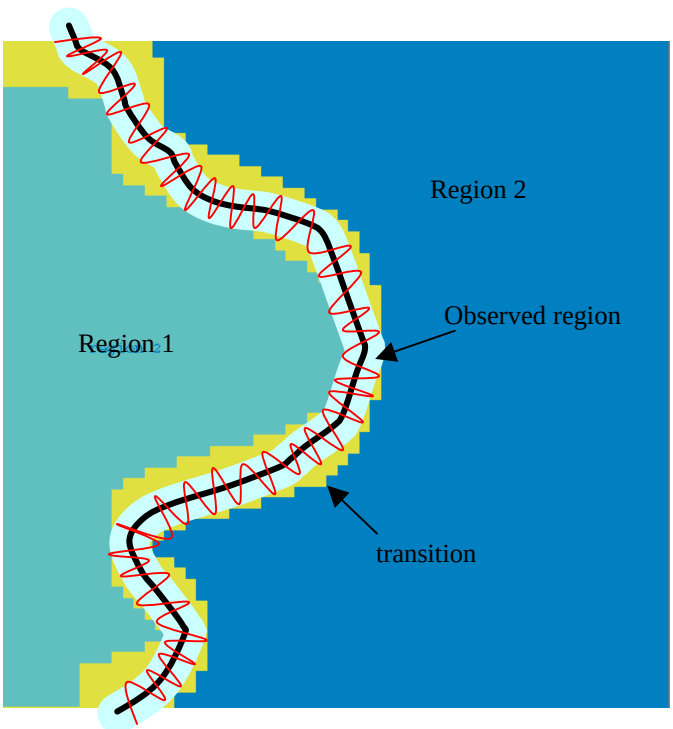


**Figure 1: tracking a "wide" boundary.**

When tracking a contour using vision, the observed region has a geometry as shown in Figure 1 (light blue region). However, with a mechanically scanning sonar, as we consider in this study, the observed region is a one-dimensional curve, as the red line in Figure 2.

The observations inside *T* may not be characteristic of neither of the adjacent regions, and are in most cases difficult to model. To be able to guarantee that the desired boundary is tracked, the observed region must be wider than the transition region, extending inside both region 1 and region 2 (this is not the case in Figure 1, where the transition region is shown in yellow, for simplicty of drawing). With a single point acoustic beam solidary with the robotic platform, this would require an rapidly oscillating trajectory of the platform. We exploit the ability of the sonar to scan the regions on each side of the robot, to transfer par of these oscillations to the sensor, resulting in a smoother platform motion during observation.

The tracking approach presented in the sections below is completely unsupervised, in the sense that the characteristics of the sonar returns from the two adjacent regions (in the form of a non-parametric probability distribution $p_i$ for each region $R_i$ ) are learned from data, and not inferred from tailored learning sets prepared ahead of mission time:

$$\begin{aligned} &\text{Region 1 ?} \quad p_1 \\ &\text{Region 2 ?} \quad p_2 \end{aligned} \qquad (2)$$

For control purposes, the tracking objective is then formulated as keeping the empirical distribution of the data acquired inside a sliding window, $?_k$ extending over the most recent measures as a balanced mixture of these two distributions:

$$?_k ?0.5p_1 ?0.5p_2 . \qquad (3)$$

As we detailed below, the size of the sliding window over which the empirical distribution is computed must be matched to the scanning period of the sonar head. In the next section we present the algorithms used to learn the probability laws $p_1$ and $p_2$, and to estimate the coeficient of the mixture model which is fitted to the measurements. The subsequent section presents how this coefficient is used to generate the control inputs to the robot actuators.

## Data processing

### *A Learning the classes models*

Our goal is to automatically detect transition regions, and at the same time learn the characteristics of the regions that define it. In the current version of our system, the robot is manually guided during this learning phase, by making the robot start on top of one region, and manually driving it to the other side of the contour. However, the transition from one class to the other is automatically detected by the algorithm: no indication is given by the operator of when this transition occurs. When the two classes have been detected and learned, tracking of the boundary is manually triggered by the operator, by pushing a button in the control interface window of the robot. In the future, we intend to full automate these steps.

Our objective at this learning step is thus to partition the received sonar profiles in two distinct classes, and learn their probability distributions.

The unsupervised segmentation algorithm exploits the fact that the sonar profiles corresponding to sea floor regions occupied by distinct species have *distinct shapes*. Each complete profile $p_k$ is first reduced to a small set of features $\{f_l^k\}_{l=1}^{L}$. For the experiments presented in this paper, $L=1$. As we mention ed before, the segmentation algorithm is based on a probabilistic framework, and associates to each individual class $R_i$ a probability distribution of the extracted features, $p_i = p(f^k \mid R_i)$, $i=1,2$. These probability distributions are initially unknown, and are learned dynamically by the algorithm described below.

We introduce first some nomenclature and notation. Let $X$ be a discrete random variable (*rv*) with probability space $(\Omega, A, P)$ where $\Omega = \{a_1, a_2, \ldots, a_M\}$, is the (finite) realization space, $A$ is a sigma-field of subsets of $\Omega$ and $P$ is a probability measure. We denote by lower-case letters $x$ the realizations of $X$. Consider a sequence $x^{(N)} = \{x_1, x_2, \ldots, x_N\} \in \Omega^n$ of $N$ independent realizations of $X$. The **type** of $x^{(N)}$, which we denote by $\nu_{x^{(N)}} : \Omega \to [0,1]$ is the empirical estimate of the probability distribution (*pd*) of $X$, and is given by:

$$\nu_{x^{(N)}}(a_j) = \frac{1}{N}\sum_{i=1}^{N} 1_{a_j}(x_i), \; j=1,\ldots,M \tag{4}$$

where $1_{a_j}(x_i) = \begin{cases} 1, & x_i = a_j \\ 0, & x_i \neq a_j \end{cases}$.

Consider that we are given two sequences of length $N$: $x_1^{(N)} = \{x_{2_1}, \ldots, x_{2_N}\}$ and $x_2^{(N)} = \{x_{1_1}, \ldots, x_{1_N}\}$. Then, the MDL (Minimum Description Length, see [4]) test for choosing between the two following composite hypotheses:

$$\begin{aligned} H_0&: \; x_1^{(N)} \sim p_0^N \; x_2^{(N)} \sim p_0^N \\ H_1&: \; x_1^{(N)} \sim p_1^N \; x_2^{(N)} \sim p_2^N, \; p_1^N \neq p_2^N \end{aligned} \tag{5}$$

where the probability laws $p_0^N$, $p_1^N$ and $p_2^N$ are unknown, i.e., for deciding whether the two sequences were generated by the *same* probability law or if they are samples from *distinct* distributions, is

$$\frac{(M-1)}{M}(2\log(N-1)+\log(2N-1)) \underset{H_1}{\overset{H_0}{\gtrless}} D(\nu_1 \| \nu) + D(\nu_2 \| \nu) \tag{6}$$

In the previous expression, $\nu$ is the balanced mixture of the types of the two observed sequences, and $D(\cdot\|\cdot)$ is the Kullback-Leibler divergence between probability laws:

$$\nu = \frac{1}{2}(\nu_1 + \nu_2), \text{ and } D(\nu \| \nu) = \sum_{j=1}^{M} v(a_j) \ln \frac{\nu(a_j)}{\nu(a_j)}, \tag{7}$$

where $v_1, v_2$ are the types of the sequences $x_1^{(N)}, x_2^{(N)}$, respectively.

Eqs. (6) and (7) show that under the hypothesis that the individual samples (in our case, the set of features extracted from each profile) are statistically independent, the types of the observed sequences are sufficient statistics for the decision problem formulated above.

To learn the classes models, we initialize the probability law of the first class with the type of the sequence of the first $N$ measures: $\hat{p}_1^N = \nu_1$. We then use test (6) to decide if the types of the subsequently observed sequences, $\nu_k, k \geq 1$, correspond to the same distribution ($\hat{p}_1^N$). This test is repeated until hypothesis $H_1$ is accepted, for a given $k = k^*$. The learning phase is then stopped and we set the estimate of the second class probability law equal to the corresponding type: $\hat{p}_2^N = \nu_{k^*}$.

### *B Classification/Mixture model*

Consider a set of $N$ consecutive profiles acquired by the robot, $p^k=\{p_k, p_{k-1}, \ldots p_{k-N+1}\}$ and denote by $f^k$ the corresponding set of $(N \times L)$ extracted features. We assume in this step that the probability laws $p(f^k \mid R_i)$ associated to each class have been learned in a previous step using the method outlined above.

If during the acquisition of all these $N$ profiles the robot observed the same sea-bed type, then, according to our hypothesis, the type of the sequence $f^k$ should be close (in the Kullback-Leibler "metric") to the corresponding probability law, and the optimal test to decide which class has been observed would choose the class $m^*$ that minimizes the Kullback-Leibler divergence between the type of the observed sequence, $\nu_k$ and the classes' representatives, $p_1^n$ and $p_2^n$:

$$m_k^* = \arg\min_{m=1,2} D(\nu_k \| p_m^n). \tag{8}$$

However, the assumption that the observed class is constant during the set of $N$ consecutive profiles is not realistic, and the test above too simplistic. In general, the $N$ successive sonar beams will hit sea-bed regions occupied by distinct habitats, such that a more realistic model for the type of the sequence of length $N$ observed at time $k$ is a *mixture* of the two basic distributions corresponding to each of the two classes present:

$$\nu_k = \alpha_k p_1^n + (1-\alpha_k) p_2^n \quad = \quad p^n(\alpha_k), \quad \alpha_k \in [0,1],$$

where we defined the notation $p^n ? ?$. The unknown mixture coefficient $? _k$ indicates *the relative percentage of the two classes* in the observed sequence. The mixture coefficient can be estimated using the following criteria:

$$? _k ? \frac{D(? _k \| p_2^n) ? D(p_2^n \| p_1^n) ? D(? _k \| p_1^n)}{D(p_1^n \| p_2^n) ? D(p_2^n \| p_1^n)} \quad (9)$$

if the minimum is inside the interval [0,1], and on one of its extrema (0 or 1) otherwise, indicating in this last two cases a "pure type".

## Tracker

As we mentioned in a previous section, our control objective is to keep a balanced mixing of the two classes returns inside a sliding window extending over the most recently received profiles. Figure 2 illustrates the rationale behind our approach.

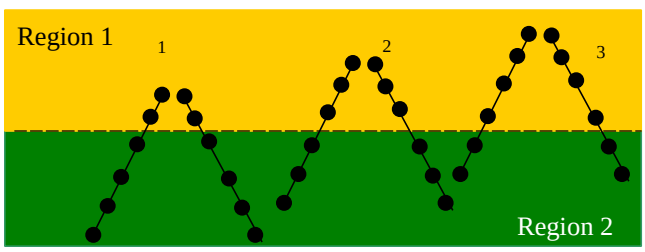


**Figure 2: three distinct relative positions of the robot with respect to the contour. Dots indicate the points of impact of the sonar beams on the sea floor**

In case 1 we obtain a majority of hits from *Posodonia* (class 2, green area) leading to an estimated $? _k$ smaller than 0.5. In situation 2 the robot is well aligned with the contour, receiving the same number of returns from the two regions, corresponding to $? _k$ =0.5. Finallly, for case 3 a majority of returns from sand (class 1) are received, implying that $? _k ? 0.5$. The mixture coefficient computed over a window of length equal to the scanning period of the sonar gives thus an indication of whether the robot is on the right ($? _k$ < 0.5) or on the left ($? _k$ > 0.5) of the contour, and can thus be used to steer the robot.

Our algorithms exploit the ability of the sonar head to mechanically steer the sonar beam to produce a smooth control signal. We motivate our approach below. Consider that the sonar is made to periodically scan an angular sector $[? ? ?, ? ?]$ with angular resolution $? ? ? ? ? / N_s$, in the vertical plane that passes through the center of sonar reference frame ( and orthogonal to the robot direction of motion). Consider also that in the processing algorithm presented in the previous section we use a sequence length which is equal to the period of the sonar scanning angle, i.e. $N$=2 $N_s$ . In this circumnstances, and in the ideal situation when the robot is perfectly aligned with the boundary (the center of the sonar reference frame is at the vertical of the contour) $? _k$ will be constant and equal to 0.5. Departures form this value indicate an offset of the robot with respect to the boundary. We propose to use the "error" signal

$$? _k ? ? _k ? 0.5 \quad ? \quad ? 0.5, 0,5 ?,$$

as a substitute for the continuous error distance that drives the iso-depth line tracker in [5].

As it is fully motivated in [5], the natural control parameter to track an horizontal line is the curvature $?$ of the robot trajectory at each point. The output of our acoustic controller is thus used to generate the desired curvature:

$$? _k ? K_p ? _k ? K_d ? _k ? ? _{k ? 1} ?, \quad (10)$$

where the signs of the proportional and derivative gains $K_p$ and $K_d$ depend on whether class 1 is on the left or on the rigth side of the boundary during tracking.

Note that basing the tracking task on this error signal, which is based on the set of $2 N_s$ most recently received sonar profiles, improves the robustness of the algorithm with respect to outliers in the acquired signals. This controller is completely driven by the perception-derived error signal $? _k$, involving that no reconstruction of the exact trajectory of the robot, or of the localisation of the impacted points, and is thus independent of the assumed dynamic and kinematic platform models, increasing its robustness with respect to modeling errors.

A simple implementation of this controller would consider constant surge velocity $u_k ? u_0$, and directly control the trajectory curvature by imposing and instantaneous yaw rate equal to

$$r_k ? u_0 ? _k . \quad (11)$$

However, the actual yaw rate of the platform is physically limited to a maximum absolute value, i.e. $|r_k| ? r_{\max}$. Curvature values larger than $r_{\max} / u_0$ can be obtained by switching to lower surge speeds, i.e., using $u_k ? u_0$. Our controller simultaneously acts on both surge speed and yaw rate, by using a pre-specified swithing curve $U(r)$. At each iteration, the imposed heading rate is determined by solving the equation $? _k ? \frac{r_k}{U ? r_k ?}$ with respect to $r_k$ , and the value

$u_k ? U(r_k)$ is input to the surge controller of the vehicle. The experiments shown below do not exploit this possibility, using a constant surge speed.

## Experimental results

We present in this section sea-trials results that validate the algorithms presented in the two previous sections. We first describe the underwater robot used in these experiments, and the sonar that is used to track the boundary. The last subsection presents results under closed-loop control.

The regions of the sea bed where these experiments were conducted – Villefranche-sur-mer, south of France) are, in its vast majority, either sandy, or occupied by patches of *Posidonia.* Our goal is to detect the boundaries of these patches (which can be neatly observed in the images simultaneously acquired by the Phantom video camera), and use the sonar information to automatically guide the robot along the limits of the *Posidonia* patches.

*A Platform and sensor*

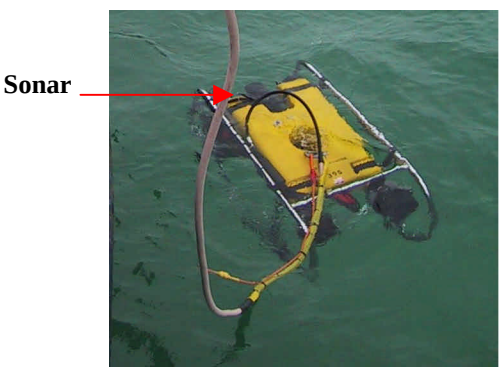


**Figure 3: the Phantom being launched at sea.**

The underwater platform used in this study is the ROV Phantom,[1] shown in Figures 3 and 4. This robot is equipped of three thrusters, two allowing control in the horizontal plane (forward/reverse, turning) and another controlling the motion in the vertical plane (up/down motions), and of the following navigation and perception sensors: a magnetic compass, a rate gyro, a pressure (depth) gauge, an altimeter, a profiler sonar mounted on a tilt platform and a video camera. Moreover, each axis is equipped of sensors allowing the measurement of the rotation speed of the corresponding motor axis. The vehicle is linked to a dry-end operational station through an umbilical cable of about 120 meters, which allows remote automatic control of the robot, by on-shore PCs

.

[1] Phantom is a Remotely Operated Vehicle produced by Deep Ocean Engineering, USA, which has been made available for research in underwater robotics at I3S through a special educational arrangement.

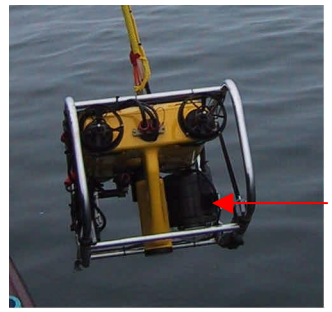


**Figure 4: Backward view of the robot, showing the camera position.**

The sensor used to scan the sea bed is a dual frequency Tritech Seaking profiler Sonar, mounted in a tilt platform in fromt of the Phantom, see Figure 3. During these experiments, the following configuration has been used :

?? the sonar is oriented towards the sea bottom, scanning the angular sector between +20° and –20°,

?? mechanical step size: 0.9°,

?? depth (range) resolution: 0.04 m,

?? frequency of the emitted signal: 1.2 MHz,

?? beamwidth: 1.4° (conical).

?? Maximal range 7m.

For this configuration, each profile consists of 150 samples. Input dimension is reduced to $L = 1$, by simply computing the energy of the profile inside a fixed length window centered in the detected maximum (shown in blue in Figure 5).

$$E_n ? \frac{1}{32} \mathop{?}_{i? j?16}^{j?15} x_i^2 . \qquad (12)$$

The first part of the profile (20 samples) corresponds to returns from the ROV crash frame and is discarded (green window in Figure 5).

Figure 6 presents the evolution of this feature during one complete experiment. We can notice the oscilaltion between regions of larger values (*Posidonia*) that alternate with periods of smaller received energy (corresponding to the observation of sand).

*B Tracking experiments*

Segmentation

During the learning phase the Phantom has been manually driven (using the joy-stick controls) first over the *Posidonia* area and then over the sand area. At the same time the two distributions characterising the two classes are automatically learned using the method presented in a previous section of the paper. Figure 7 shows the learned distributions.

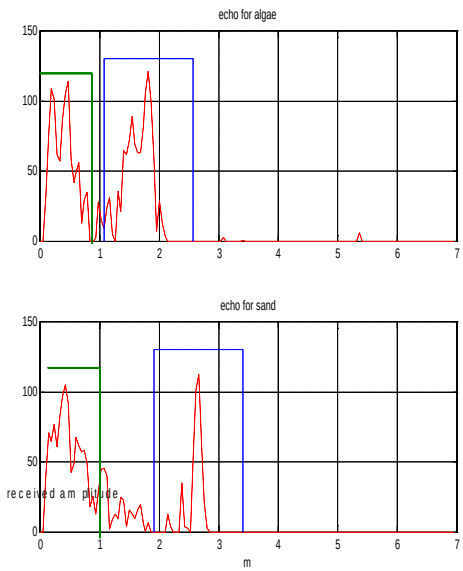


**Figure 5: computation of energy for each ptofile.**

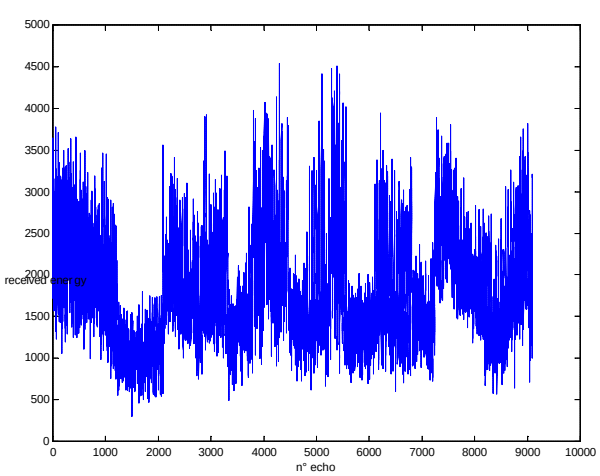


**Figure 6: evolution of received energy during one tracking experiment.**

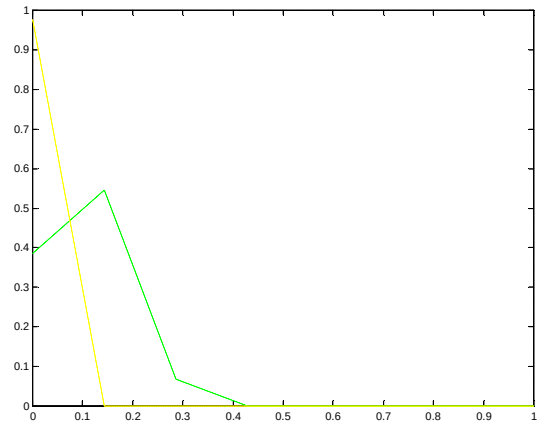

**Figure 7: probabiliuty distributions estimated using 44 profiles (yellow: sand; green: *Posidonia*).**

Contour Tracking

To present the contour tracking results a mosaic of the video frames acquired during the experiment (at 2 images/sec) has been created off-line using a correlation method. This mosaic is used as a ground truth against which the estimation of the mixture coefficient is compared.

We co-registered the mixture coefficient onto the video mosaic, using the following color code: white indicates pure *Posidonia* ($?_k ?1$), dark blue indicates pure *sand* ($?_k ?0$), while intermediate values are coded by varying intensities of blue. The result is shown in Figure 8, for an observation window of length equal to twice the length of the sonar scanning period (in the configuration of this experiments this leads to *N=44*), as explained before.

Figure 8 displays, on the left, the evolution of the mixture signal $?_k$ during one tracking experience at Villefranche-sur-mer (tracking of the boundary of a *Posidonia* patch, dark patches on the left of the mosaic). This signal is indpendently plotted in Figure 9 along time. As we can see, the mixture coefficient oscillates between the extreme values of 0 and 1, indicating that the robot's path is oscillating between sand and *Posidonia* regions. The right plot in this Figure shows the trace of the sonar beams on the sea-floor. Reconstruction of this trace uses the noisy information provided by the robot's compass, and is also prone to errors in the mosaic reconstruction. It should thus be taken as an indication of the overall agreement between the acoustic processing and visual information acquired by the robot's video camera.

Inspection of this Figure shows that the robot is turning left when it detects predominantly sandy bottoms, and to the right when the received profiles are identified as coming from mostly *Posidonia* covered areas.

Figure 10 shows the imposed heading rate, output by the controller presented in the previous section. As we can see,

the controller imposes an instantaneus curvature that is approximayely proportional to the estimated mixture coefficient.

As we can see, the robot suceeds in maintaing the boundary of the patch in the field of view of the camera, and the path of the platform during the experiment captures the macroscopic geometry of the patch boundary.

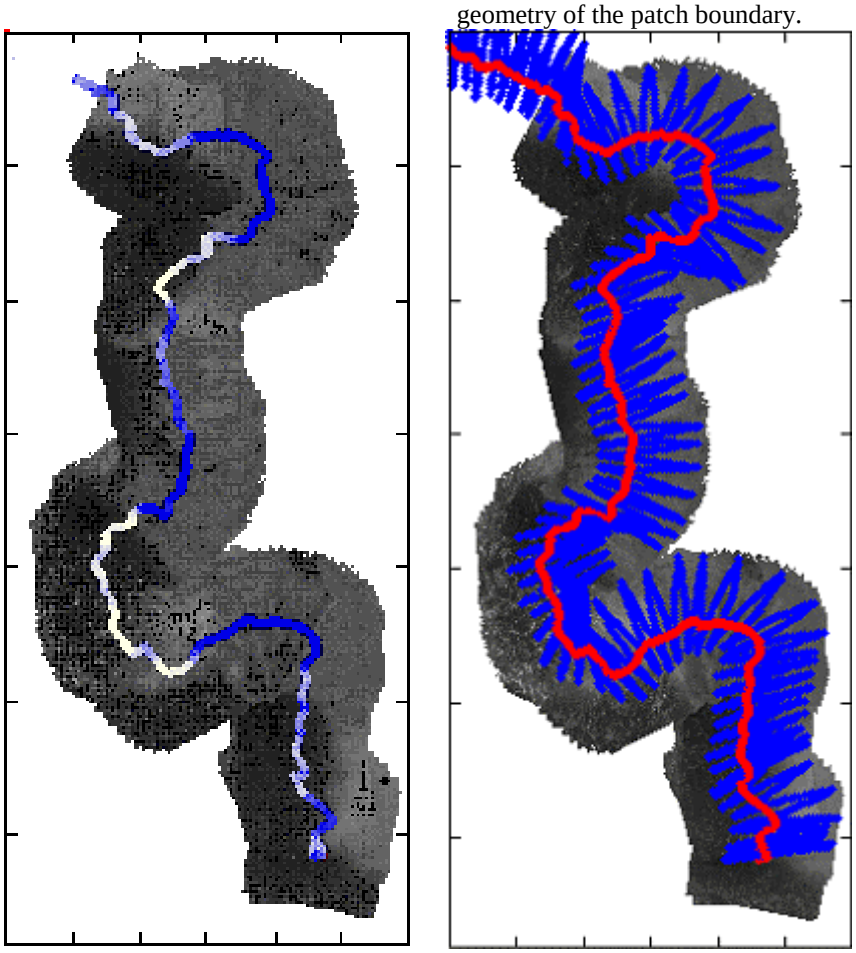

**Figure 5: Experiment 1. Left: estimated mixture coefficient ; right: trace of the sonar beams on the sea floor.**

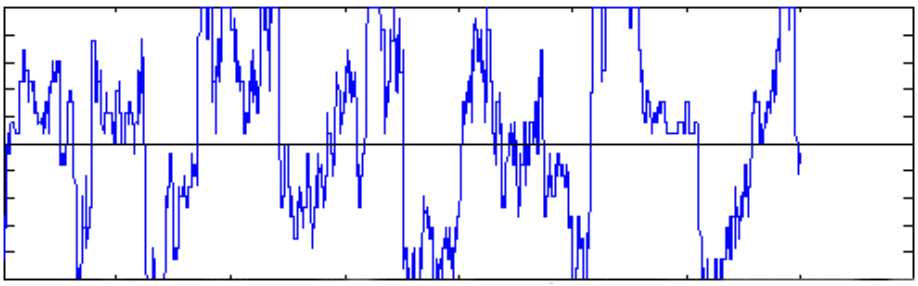

**Figure 6: evolution of the estimated mixture coefficient during experiment of Figure 8.**

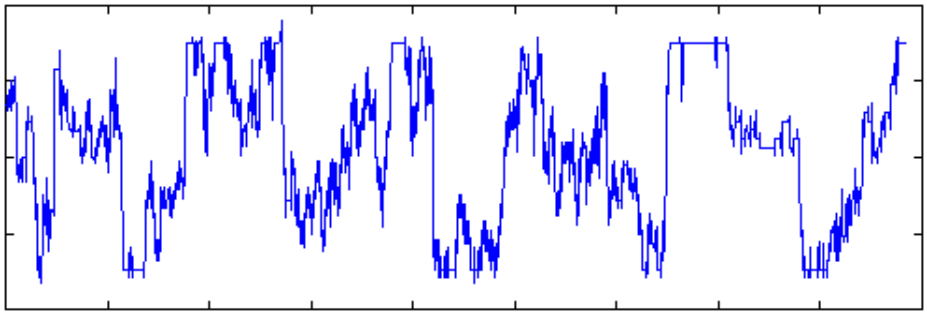

**Figure 7: imposed heading rate during tracking of Figure 8.**

The next three Figures (11—13) correspond to another tracking experience for a different *Posidonia* patch. As in the previous case, the robot keeps locked on the patch boundary, in spite of its rapidly varying curvature. The small scale loops in the robot trajectory are due to errors in the mosaic construction. The mosaic underlying Figure 11 suffers from several registration errors, which lead to artifacts in the reconstruction of the vehicle's trajectory (small loop-like portionsof the registered trajectory). The quality of tracking is better appreciated by examining Figure 10, which shows that the mixture coefficient is iscillating around the target value of 0.5, corresponding to a good alignement of the sonar head with the tracked contour.

We remark that the natural contours tracked in these two experiments have a rapidly varying curvature, switching from positive to negative curvature almost instantaneously at some points.

## Conclusion and future work

We presented algorithms for contour tracking using data provided by a profiler sonar. The main innovation of the approach is the proposal of basing tracking control on the estimation of the parameter of a mixture model that is fit to the most recently received sonar returns. A convenient error signal is obtained by centering the estimated mixture coefficient, which drives a simple discrete classical proportional-derivative controller. Results of this data processing algorithm on real data collected at sea for the boundary between two natural sea bed types demonstrate the adequacy of the approach

The simple propotional-derivative controller proposed here is motivated by the study presented in [5], which considers tracking with a continuous point-sensor, whose output indicates, locally, the distance of the platform with respect to the tracked contour. In fact, the sonar data considered in the study reported here can be considered as a quantized (binary) sensor, which provides, at each instant, only an indication of the detected class, i.e., of whether the platform is on the right or on the left of the contour. The mixture coefficient that we propose here tries to allieviate the problems associated with this quantification, but introduce memory in the observation operator, which tend to create oscillating modes. Alternative controller stuctures are under study, which directly assess the discrete nature of the snsor-derived information.

An alternative tracking approach would simultaneously control the platform motion (robot surge speed and yaw rate) as well as the sensor (the scanned angle). Consideration of this possibilty is currently under study.

In this paper we consider tracking using a single sonar beam, which periodically scans the region below the robot during observation. Use of side scan sonars, which simulaneously scan along several directions on the plane orthognal to robot direction of motion should dramatically improve the quality of tracking.

The data processing algorithms presented here all rely on non-parametric estimation of probability distributions from data. In the experiments presented here we consider the simple data type (or empirical distribution), given by equation (4). This estimator is appropriate for discrete data. However, the energy feature which is associated to each profile is continuous, requiring pre-specification of the admissible parameter range and the quantization beams fo rhistogram computation. We are currently studying other non-parametric density estimators, which automatically adjust to the data characteristics.

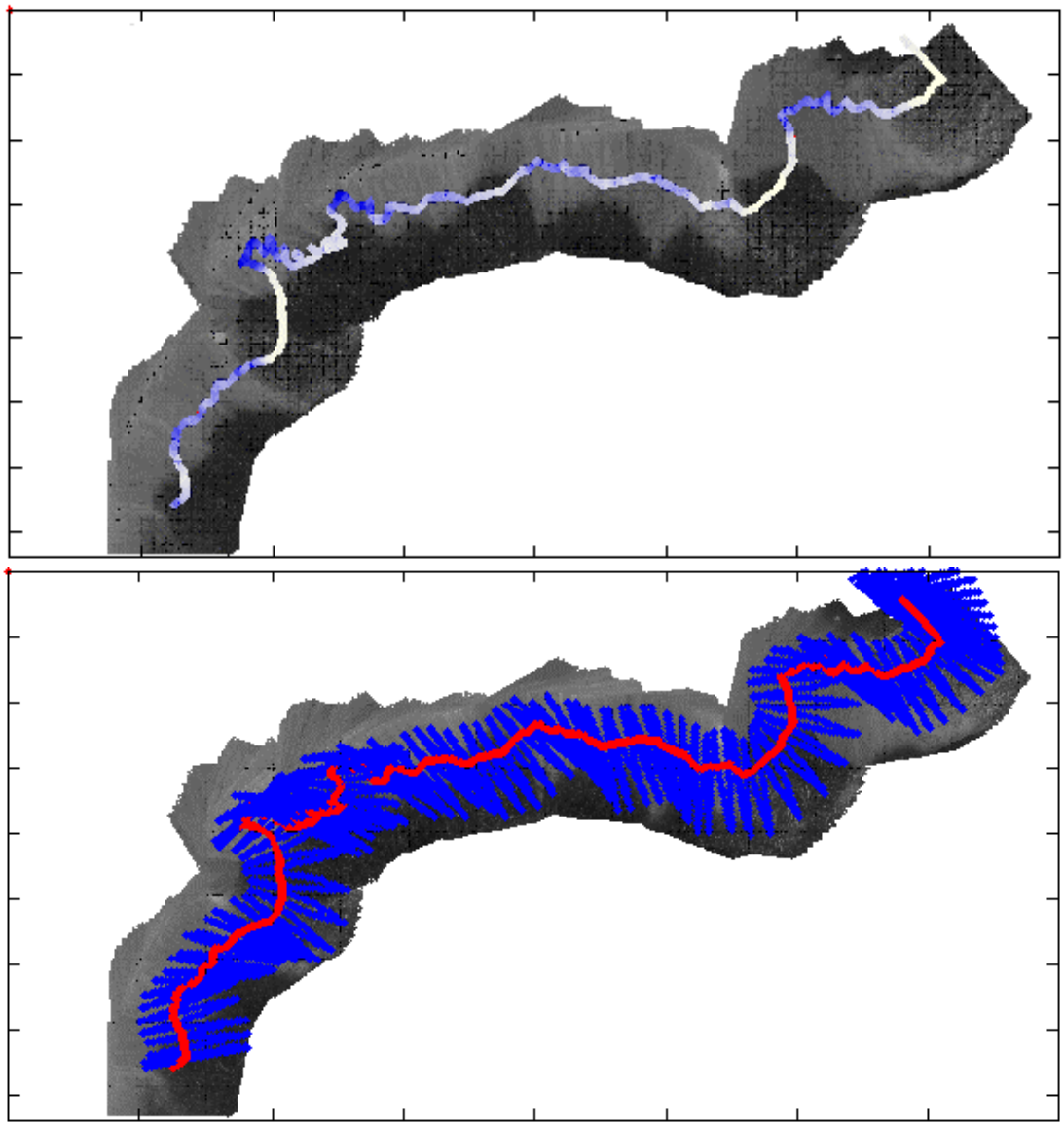

**Figure 8: Experiment 2. Top: estimated mixture coefficient; bottom: trace of the sonar beams on the sea floor.**

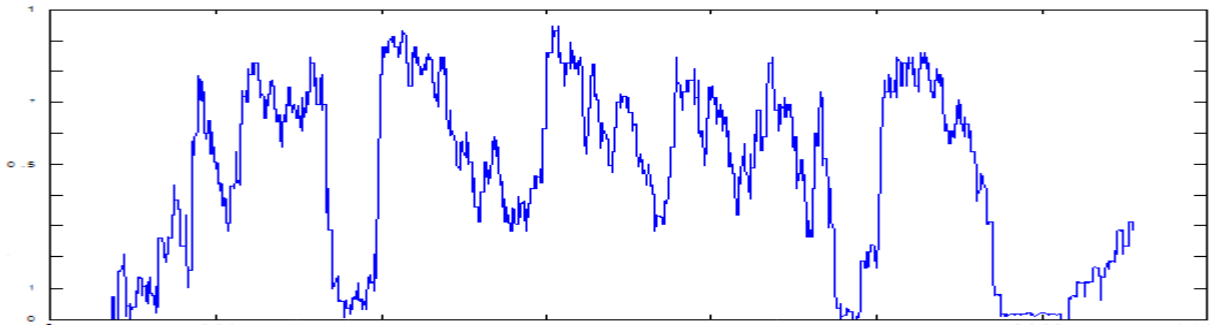

**Figure 9: evolution of the estimated mixture coefficient during experiment 2.**

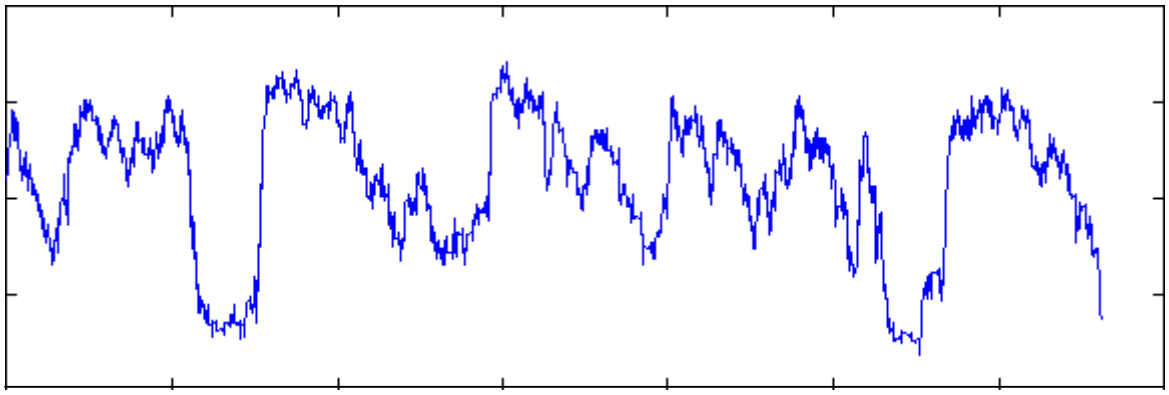

**Figure 10: input to the heading rate controller during experiment 2.**

.

## Acknowledgments

This work has been partially funded by the European Union through the IST program, project SUMARE (*Survey of Marine Resources*), contract IST-1999-10836. See **http:/www.mumm.ac.be/SUMARE** for more details.

## References

[1]. Maria-João Rendas, José Santos-Victor *et al*, *Learning Safe Navigation in Uncertain Environments*, Int. J. of Intelligent Control and Systems, Vol 2, No. 2, pp 253-285, 1998.

[2]. Maria-João Rendas and Stefan Rolfes, *Learning Safe Navigation in Uncertain Environments*, Robotics and Autonomous Systems 31, pp 43-52, 2000.

[3]. Albert Tenas, Maria-João Rendas, Jean-Pierre Folcher, *Image Segmentation by Unsupervised Adaptive Clustering in the Distribution Space for AUV guidance along sea-bed boundaries using Vision*, Proc. OCEANS 2001, Honolulu, Hawaii, USA, November 2001.

[4]. Jorma Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Series in Computer Science—Vol. 15, 1989.

[5]. Maria-João Rendas, Isabel Lourtie, Georges Pichot, *Adaptive Sampling for sand bank mapping using an autonomous underwater vehicle equipped of an altimeter*, ISESS 2003, Vienna, Austria, May 2003.

Figure 4: Estimated percentage of *Posidonia* along a trajectory.