Project SUMARE

Deliverable D3.3b : Segmentation of sonar measures

Coordinated by: Maria João Rendas
Date: August 2003
Participant partners: I3S (*M.-J. Rendas, C. Barat*)

Table of Contents

Introduction 2

Overall goal..2

Sonar and platform...3

Software architecture 4

Contour detection 7

Learning the classes models 7

Classification/Mixture model 9

Experimental Results..11

Feature extraction 11

Segmentation 12

Mixture Coefficient...15

Conclusion 17

Bibliography 18

Chapter

1

Introduction

This deliverable reports on the work conducted in SUMARE on contour detection for tracking of natural boundaries using a profiler sonar.

The deliverable is organized in 5 separate chapters:

1. *Introduction*. This chapter presents the goals of the study presented in this document.

2. *Architecture and sensor*: This chapter presents the sensor used and the platform on which it is installed.

3. *Contour detection* : This chapter presents the method used to segment the sonar returns.

4. *Result* : This chapter presents experimental results of the contour detection algorithm presented in the previous chapter.

5. *Conclusions*. This chapter summarizes the main results, and lists issues that still require further stuy.

Overall goal

To track the boundaries between distinct benthic regions, we need first to detect these boundaries. In this deliverable a novel sonar classification algorithm is presented, which uses the signature of the ocean floor in the incoming profilers to discriminate between distinct sea-bed types. The boundaries correspond to regions of transition between two distinct seabed regions, which are not known a priori and must be learned during observation.

Chapter
2

This chapter presents first the profiler sonar used in these studies, and the platform on which it is mounted. It then describes the software architecture of the module that implements the algorithms proposed.

Sonar and platform

The sensor used to scan the sea bed is a dual frequency Tritech Seaking profiler Sonar, mounted in a tilt platform located in front of the Phantom ROV, see Figures 1 and . During all experiments, the following configuration has been used :

- the sonar is oriented towards the sea bottom, scanning the angular sector between +30° and –30°,
- 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.

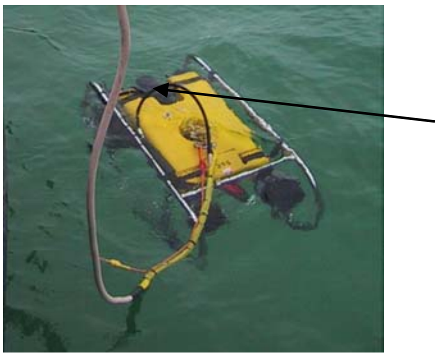

Figure 1. The Phantom being launched at sea.

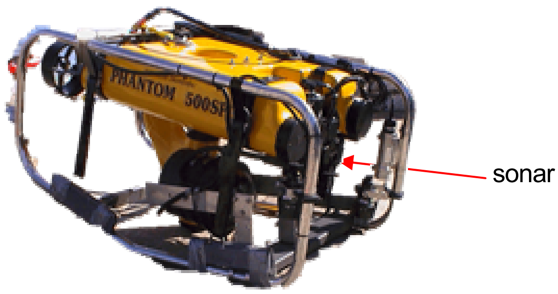

Figure 2. The sonar profiler on the Phantom.

The Phantom is a Remotely Operated Vehicle (ROV) that is remotely controlled from on-shore computers through an umbilical cable of 120 meters long. This cable brings up sensor signals, and enables direct control of the robot's motors and sensor actuators (tilt platform, camera pan&tilt, focus, etc.). It is driven by three propellers, two for control of the horizontal motion of the vehicle (forward/reverse, turning) and another that enables control of its vertical displacements.

Besides the sonar sensor used in the studies presented herein, the Phantom is equipped with other sensors, for navigation and observation purposes:

- A three axis compass;
- A gyroscope, measuring rate of gyration in the horizontal plane (yaw rate);
- A pressure gauge, measuring depth below sea level;
- One incremental encoder per axis, measuring the rotation speed of the propellers;
- An altimeter, measuring altitude above the sea bottom;
- A video camera, with built-in pan and tilt capabilities.

Software architecture

The complete software architecture of the Phantom comprises several independent threads distributed accross several processors, which communicate through messages. We can identify the following main existing threads:

- Data acquisition and conditioning: it receives the raw data from all navigation sensors, formats it and send it to the Data Server thread;
- Data Server: it records all received data, and distributes it to other threads, in response to queries for specific items.
- Sonar Acquistion: it reads the frames sent by the sonar and sends the relevant data to the Data Sever;

- Video Acquisition: it reads the frames sent by the video camera and sends the relevant data to the Data Sever;

- Control thread: it generates the signals that directly act upon the motors and actuators of the robot;

- Guidance threads : generate the reference signals and select the relevant control mode of the control thread to implement the desired behaviour (visual tracking, **sonar tracking**, reach a given position);

- Positioning thread: updates the estimated position of the robot in a coordinate frame associated to the mission.

- Mission control thread: it supervises the execution of all other threads in order to execute the mission defined by the user. Priority is always given to events generated by the User Interface thread.

- User Interface thread: it updates the information passed to the human operator, enabling the generation of abortion or re-direction signals on-line, during execution of the pre-defined mission.

As it is clear from the description above, the studies reported here concern the definition of a new observation behaviour, in which the robot uses the information provided by the sonar sensor to move along a boundary.

The implementation of this new behavior required the definition of a new Guidance thread, consisting of two new software modules. The first one is the **sonar classifier,** whose goal is to associate a label (class) C_k to each newly acquired profile p_k , and the acoustic controller, whose responsibility is to generate appropriate commands r_k that guide the robot along the contour between the distinct classes. In this document only the classifier is presented.

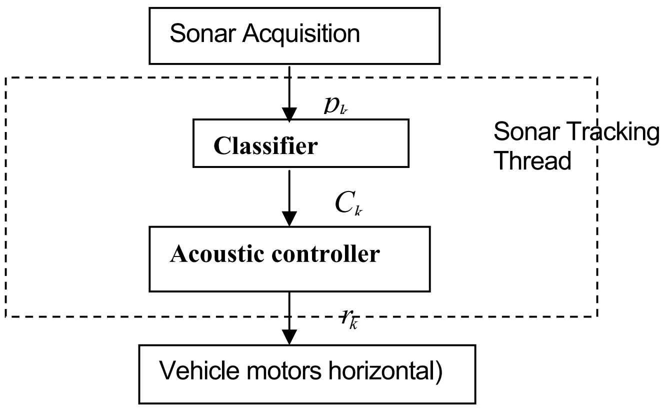

Figure 3.Signal processing and control architecture (Sonar Trackign thread).

In figure 3, the output of the Classifier (the input of the contour tracker) are the labels C_k of the received profiles. Actually, as we will see in the next chapter, instead of using hard classifications as the input of the acoustic controller, we drive it with a continuous signal that indicates *the relative percentage of the classes* (π_k) defining the tracked boundary inside a sliding window extending over the most recently acquired profiles.

Chapter
3

Contour detection

In previous studies, we assessed the problem of boundary detection using visual information [1], which are implemented by the Visual Tracking behaviour mentioned in the previous section. However, use of video data in the ocean can be often compromised by lack of ambient light, or by water turbulence. A more robust alternative is the use of acoustic sensors. Here, we consider the use of sonar information to detect and track boundaries between distinct habitats occupying the sea floor.

The following section presents the method to used to detect and learn the unknown classes and finally to generate the mixture coefficient (soft-classifications) input to the controller module.

Learning the classes models

Our goal is to automatically detect transitions between distinct regions, and at the same time learn the characteristics of the adjacent regions that define the detected boundary. 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 fully automate these steps.

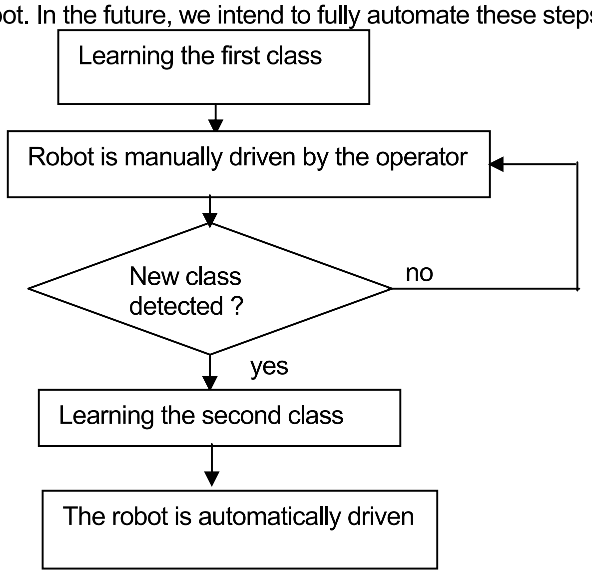

Figure 4. The different steps

Our objective at this learning step is thus to partition the received sonar profiles in two distinct classes, and learn the characteristics of the signals corresponding to each one. Our approach is based on a probabilistic setting, and we will do this characterisation by assigning to each class a *probability distribution of the received signals*.

An unsupervised segmentation algorithm exploits the fact that the sonar profiles corresponding to sea floor regions occupied by distinct species must 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, $p_i = p\left(\{f_l^k\}_{l=1}^L \middle| R_i\right)$, *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 (Ω, A, P) where $\Omega = \{a_1, a_2, \ldots, a_M\}$, is the (finite) realization space, A is a sigma-field of subsets of Ω and P is a probability measure. We denote by lower-case letters x the realizations of X. Consider a sequence $x^{(N)} = \{x_1, x_2, .., x_N\} \in \Omega^n$ of *N* independent realizations of *X*. The **type** of $x^{(N)}$, which we denote by $\nu_{x^{(N)}} : \Omega \mapsto [0,1]$ is the empirical estimate of the probability distribution (*pd*) of *X*, and is given by:

$$\nu_{x^{(N)}}(a_j) = \frac{1}{N}\sum_{i=1}^{N} 1_{a_j}(x_i), j = 1, .., M$$
$$\text{where } 1_{a_j}(x_i) = \begin{cases} 1, & x_i = a_j \\ 0, & x_i \neq a_j \end{cases}. \tag{4}$$

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

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

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

$$\frac{(M-1)}{M}\left[2\log(N+1) - \log(2N+1)\right] \underset{H_1}{\overset{H_0}{\gtrless}} D(\nu_1 \| \hat{\mu}) + D(\nu_2 \| \hat{\mu}) \tag{6}$$

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

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

where ν_1, ν_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, $\nu_k, k > 1$, 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: $\hat{p}_2^N = \nu_{k^*}$.

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 $p\left(\left\{f_1^k\right\}_{l=1}^L \middle| R_i\right)$ 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, ν_k and the classes' representatives, p_1^n and p_2^n:

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

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:

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

where we defined the notation $p^n(\pi)$.

The unknown mixture coefficient π_k indicates *the relative percentage of the two classes* in the observed sequence
Then we have:

$$D\left(\nu_k \middle\| p_1^n\right) = \sum_{j=1}^{M} \nu_k(a_j) * \log\left(\frac{\nu_k(a_j)}{p_1^n(a_j)}\right) = \sum_{j=1}^{M} \nu_k(a_j) * \left(\log\left(\nu_k(a_j)\right) - \log\left(p_1^n(a_j)\right)\right)$$

$$D\left(\nu \middle\| p_2^n\right) = \sum_{j=1}^{M} \nu_k(a_j) * \log\left(\frac{\nu_k(a_j)}{p_2^n(a_j)}\right) = \sum_{j=1}^{M} \nu_k(a_j) * \left(\log\left(\nu_k(a_j)\right) - \log\left(p_2^n(a_j)\right)\right)$$

$$D\left(\nu_k \| p_1^n\right) - D\left(\nu \| p_2^n\right) = \sum_{j=1}^{M} \nu_k(a_j) * \left(\log\left(p_2^n(a_j)\right) - \log\left(p_1^n(a_j)\right)\right)$$

We replace $\nu_k = \pi_k p_1^n + \left(1 - \pi_k\right) p_2^n$

$$D\left(\nu_k \| p_2^n\right) - D\left(\nu \| p_1^n\right) = \sum_{j=1}^{M} \left(\pi_k p_1^n(a_j) + \left(1 - \pi_k\right) p_2^n(a_j)\right) * \left(\log\left(p_1^n(a_j)\right) - \log\left(p_2^n(a_j)\right)\right)$$

$$D\left(\nu_k \| p_2^n\right) - D\left(\nu \| p_1^n\right) = \sum_{j=1}^{M} \left(\pi_k p_1^n(a_j) + \left(1 - \pi_k\right) p_2^n(a_j)\right) * \log\left(\frac{p_1^n(a_j)}{p_2^n(a_j)}\right)$$

$$D\left(\nu_k \| p_2^n\right) - D\left(\nu \| p_1^n\right) = \sum_{j=1}^{M} \left[\pi_K p_1^n(a_j) * \log\left(\frac{p_1^n(a_j)}{p_2^n(a_j)}\right) - p_2^n(a_j) * \log\left(\frac{p_2^n(a_j)}{p_1^n(a_j)}\right) + \pi_k p_2^n(a_j) * \log\left(\frac{p_2^n(a_j)}{p_1^n(a_j)}\right)\right]$$

$$D\left(\nu_k \| p_2^n\right) - D\left(\nu \| p_1^n\right) = \pi_K D\left(p_1^n(a_j) \| p_2^n(a_j)\right) - D\left(p_2^n(a_j) \| p_1^n(a_j)\right) + \pi_k D\left(p_2^n(a_j) \| p_1^n(a_j)\right)$$

The parameter can be calculate by the following expression:

$$\pi_k = \frac{D(\nu_k \| p_2^n) + D(p_2^n \| p_1^n) - D(\nu_k \| p_1^n)}{D(p_1^n \| p_2^n) + D(p_2^n \| p_1^n)}, \tag{9}$$

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". The results of tracking using this formula will be developed in Deliverable 4.2 and have been presented in [6].

Another approach have been developed before the utilization of the equation 9 and presented in [5][7]. The maximum likelihood estimate of π_k is obtained by solving the following minimization problem:

$$\hat{\pi}_k = \arg\min_{\pi} D\left(\nu_k \| \nu^n(\pi)\right). \tag{10}$$

We have verified numerically, for a large number of class models, that the criterion $D\left(\nu_k \| \nu^n(\pi)\right)$ presents a dependency on π which is well approximated by a quadratic:

$$D\left(\nu_k \| \nu^n(\pi)\right) \cong c_0^k + c_1^k \pi + c_2^k \pi^2, \tag{11}$$

where the unknown coefficients $\left\{c_i^k\right\}_{i=0}^{2}$ depend on the observed type and on the classes models. If these coefficients were known, $\hat{\pi}_k$ could be determined by the following analytical expression:

$$\hat{\pi}_k = -\frac{c_1^k}{2c_2^k}, \tag{12}$$

In order estimate the coefficients $\left\{c_i^k\right\}_{i=0}^{2}$, we evaluate the criterion $D\left(\nu_k \| \nu^n(\pi)\right)$ for three distinct values of π (0.25, 0.5 and 0.75).

The second approach is an approximation of the value compare to the first one. The two methods give slightly different results. In the future works we will use eq. 9.

Chapter 4

Experimental Results

We present in this section results on real data acquired during real experiments performed at Villefranche-sur-mer (South of France). The regions of the sea bed observed during these experiments 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 this information to automatically guide the robot along them.

Feature extraction

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*, simply the energy of the profile inside a fixed length window centered on the detected maximum (shown in blue in Figure 5). For the configuration presented, each profile consists of 150 samples. The first part of the profile (20 samples) corresponds to returns from the ROV crash frame and is discarded (green window in Figure 5).

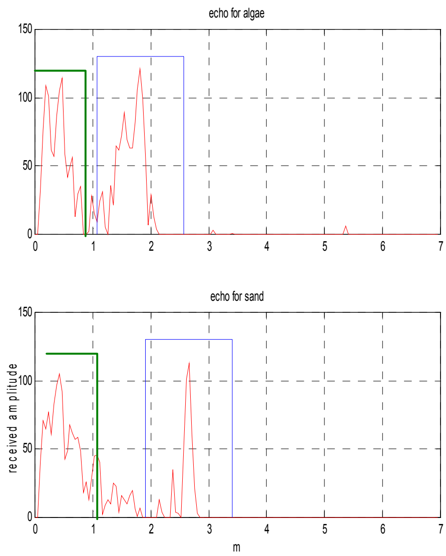

Figure 5: Computation of energy for each profile.

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).

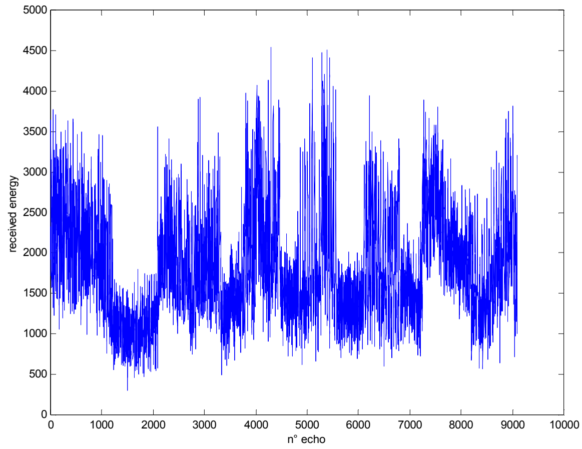

Figure 6: Evolution of received energy during one tracking experiment.

Segmentation

To show the classification results the Phantom has been manually driven (using the joy-stick controls) along a trajectory that crosses several times the boundary of one *Posidonia* patch, while sonar scans and video images (at a rate of 2 images/sec) were simultaneously recorded. At the same time the two distributions characterizing the two classes are automatically learned using the method presented in a previous section of the paper. Figure 7 shows the learned distributions.

A mosaic of the acquired video frames has been created off-line, using a correlation method, and used as a ground truth against which the results of the sonar data processing is compared, see Figure 8.

The sonar scans are processed using the method presented in chapter **3.** Sonar scans too close to the sea bottom (profile maximum occurs at a distance less than 0.4m) are not processed, since they correspond to returns from the ROV crash frame. They are represented in black in the figures.

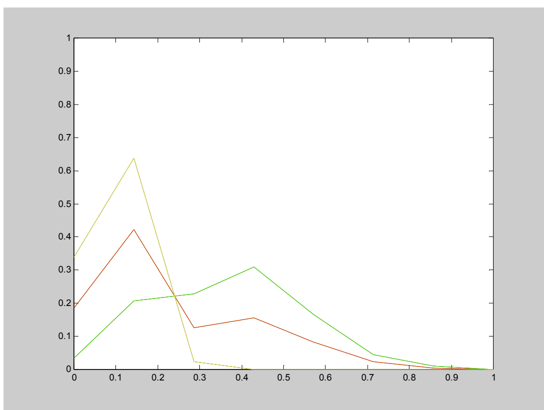

Figure7.Estimated histograms for the 2 classes (yellow = sand and green = posidonia) and the mixture (red).**Estimated histograms for the 2 classes (yellow = sand and green = posidonia) and the mixture (red).**

The classification results are estimated by comparing the ground true represented by the mosaic of images to the classified scans.

TABLE1. CLASSIFICATION RESULTS

	class sand	class posidonia
class sand	97.2 %	5.8 %
class posidonia	2.8 %	94.2 %

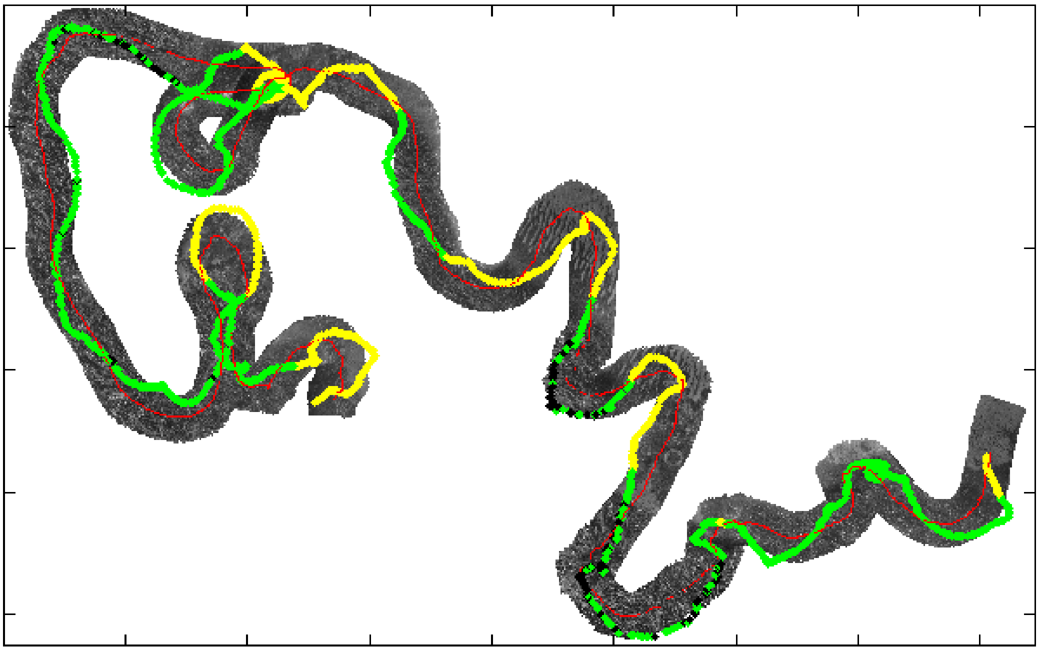

Figure 8.Classification of sonar measures in 2 classes using the segmentation method. In yellow the classification into class sand, in green into the class posidonia

The two following figures show details of the figure 8, enabling a better appreciation of the correspondance between the sea-bed shown in the image and the class detected from the acoustic signal.

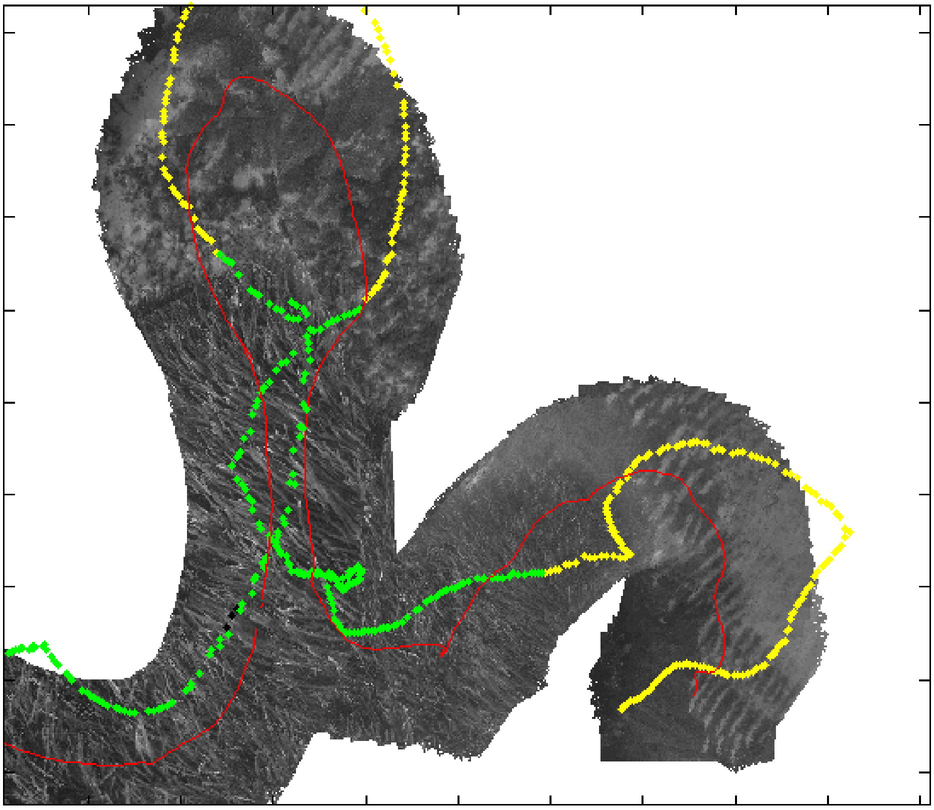

Figure 9. Zoom on the mosaic fo Figure 7.

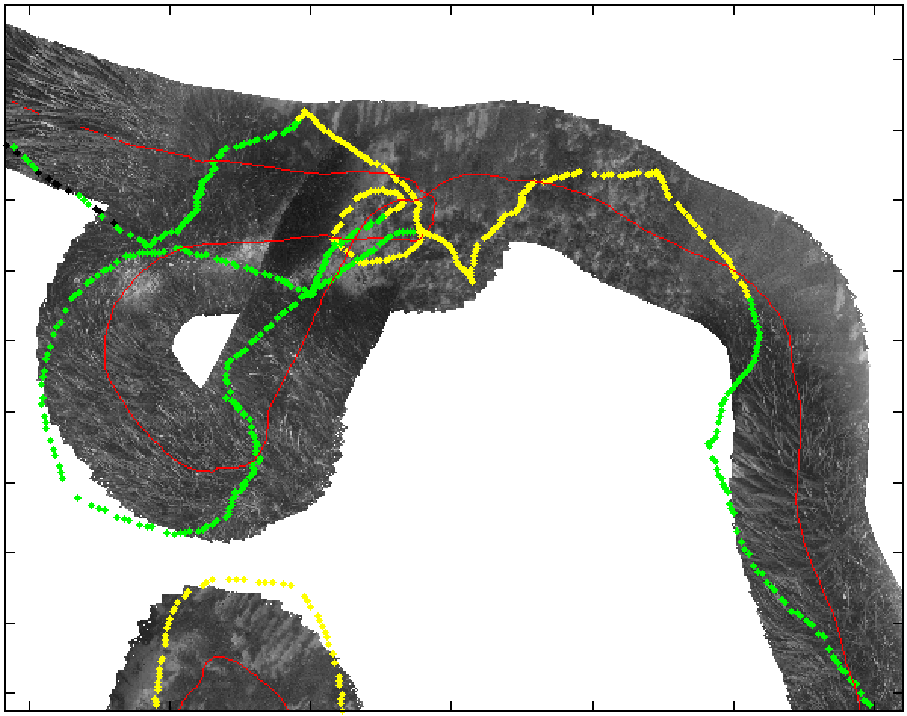

Figure 10. Zoom on the mosaic of Figure 7.

Mixture Coefficient

As we said before, the acoustic controller is not directly driven by hard classifications of the sonar returns, butrather by an estimate of the *mixture coefficient* associated to the most recently received profiles. Figure 11 demonstrates the correct estimation of the mixture coefficient using the eq. 9, for an observation window of length equal to twice the length of the sonar scans (in the configuration of this experiments this leads to $N=134$).

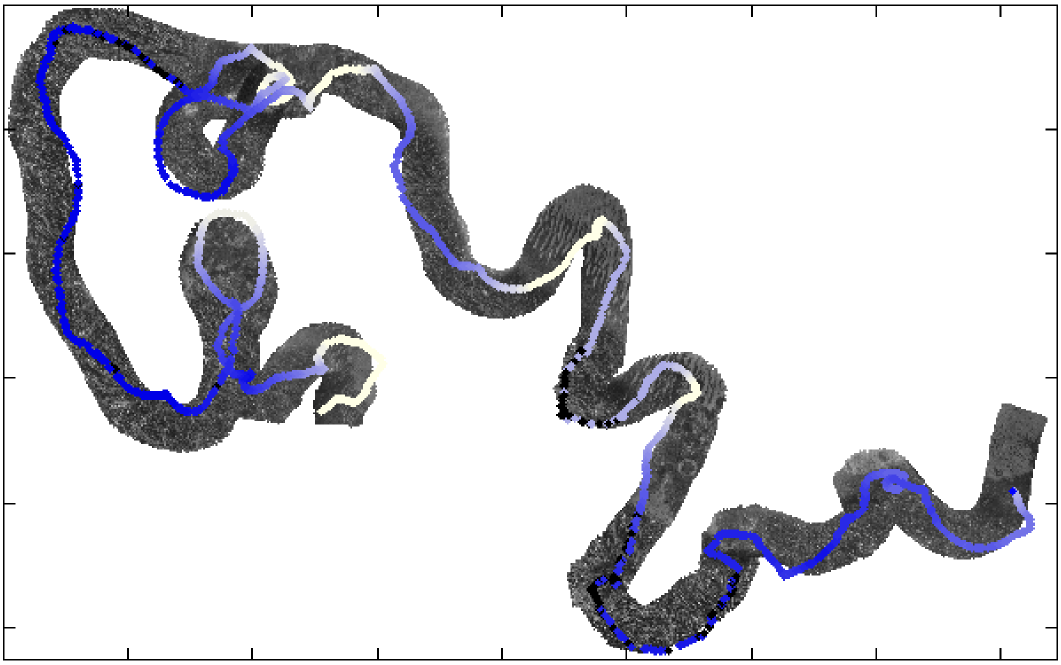

Figure 11. Estimated mixture coefficient .

Chapter

6

Conclusion

In this deliverable we present work done in the Sumare project concerning the definition of an acoustic–based contour tracking behaviour for one of the underwater platforms operated in the project: the ROV Phantom. Results of real at-sea experiments are presented, illustrating the performance of the contour detection for a specific case of a transition between Posidonia and sand.

Several future directions for improving performance are currently under study. 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 for histogram computation. We are currently studying other non-parametric density estimators, which automatically adjust to the data characteristics.

Bibliography

[1]. 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.
[2]. Christian Barat and Maria-João Rendas, Classification of sonar measures using optimized wavelets, Proc. of OCEANS'02, Biloxi, Mississipi, USA, October 2002.
[3]. 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.
[4]. Jorma Rissannen, Stochastic Complexity in Statistical Inquiry, World Scientific, Series in Computer Science—Vol. 15, 1989.
[5]. Christian Barat and Maria-João Rendas, Tracking benthic boundaries using a profiler sonar: a mixture model approach, Proc. of OCEANS'03, San Diego, California, October 2003.
[6]. Maria-João Rendas and Christian Barat , Using statistical mixture models for tracking natural underwater boundaries, Proc. Unmanned Untethered Submersible Technology, Durham, New Hampshire, August 2003.
[7]. Christian Barat and Maria-João Rendas, Benthic boundaries tracking using a profiler sonar, Proc. of IROS'03, Las Vegas, Nevada, September 2003.