

Image Segmentation by Unsupervised Adaptive Clustering in the Distribution Space for AUV guidance along sea-bed boundaries using Vision

Albert Tenas, Maria-João Rendas and Jean-Pierre Folcher

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S)
2000 Rte des Lucioles, BP 121, 06509 Sophia Antipolis Cedex, France

Abstract

We address the problem of autonomous underwater vehicle guidance along the boundaries of different benthic species using video information. This form of guidance provides a robust navigation behavior enabling observation of the occupancy of the sea bed independently of the presence of external position references (either GPS or installed acoustic beacons). The major innovation of the work presented concerns the image segmentation algorithm. It is an unsupervised algorithm, which identifies clusters in the space of gray level probability distributions of image neighborhoods. The metric used to compare gray level distributions is the Kullback-Leibler directed divergence, which is related to the probability of confusing members of distinct clusters. The algorithm is self-tuned, in the sense that the number of clusters is automatically determined. It works adaptively (frame-to-frame), updating the classes' representations for each new frame, accommodating gradual lighting and texture variations within the same region. The visual controller, a simple integral law with saturation, controls heading rate to minimize the distance between the contour and the image center, while keeping a constant forward speed along the body axis. A separate controller (classic PI) keeps the robot at constant altitude from the sea bottom. The design of these controllers was based on the identified hydrodynamic model of the vehicle. The performance of the algorithm proposed is validated by real experiments conducted with the robot Phantom 500 XTL (Deep Oceans Engineering, USA)¹.

1. Introduction

We present a novel approach to unsupervised image segmentation, based on information theory tools, that enables visual guidance of an underwater robot along the contour of two natural sea-bed regions. Previous work on perceptual guidance of underwater vehicles used the simple geometry of man-made features – mostly linear, e.g. pipelines. For the application that we consider – autonomous mapping of the limits of natural benthic regions – these approaches are not applicable, given the lack of pure geometric features. Only a region-based segmentation approach can detect the desired boundary, enabling the subsequent generation of appropriate control signals.

The paper is organized in the following way. First the contour tracking problem using vision is introduced, and the basic video processing/control architecture presented. Then, we present the image segmentation algorithm, showing examples of its use in a real environment at the bay of Villefranche-sur-mer (France), for detecting the boundary between posidonia and sand. We then present the contour tracker. Finally, we present a results of a real experiment where the robot (Phantom) uses vision to track the boundary of two distinct regions.

2. Control/perception architecture

The overall architecture of the robot Phantom is presented in Figure 1. It comprises a global decisional level, signal acquisition and processing modules, positioning and mapping functions and a control module that can enforce a series of distinct behaviors of the vehicle.

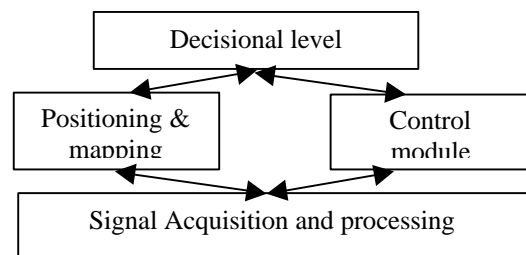


Figure 1: software architecture.

¹ The Phantom 500 XTL manufactured by Deep Oceans Eng, Palo Alto, USA, is used in the projects Narval and SUMARE benefiting from a special education/research arrangement.

The complete mission of the robot is defined as a series of goals to be attained, which are translated, by the real-time planner, into a plan (sequence of tasks) whose execution is under the responsibility of the task execution controller.

The work presented here corresponds to the execution of one of these tasks: track a boundary between two distinct species using visual information. The execution of this task requires the cooperation between two distinct tools: an image segmentation algorithm, which detects the relevant information for guidance and a guidance algorithm that keeps the contour in the visual field of the robot as it progresses its observation.

The basic flow of information is represented in figure 2. Using the video frames the image segmentation algorithm extracts the contours between regions of distinct texture. This is the input information for the tracking controller that specifies the vehicle heading reference that induces the appropriate rotation in the horizontal. At the same time, constant references are kept for the altitude controller, imposing a constant distance from the sea floor during the entire observation, to minimize tracking problems due to varying observation conditions. In the present implementation, the surge speed of the robot is also kept constant during the entire observation. Control of this extra degree of freedom, for instance increasing speed in the quasi-linear parts of the contour, and decreasing it whenever the contour direction is rapidly varying may increase efficiency of this observation mode and will be the subject of future studies.

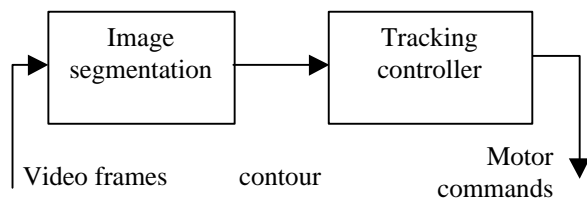


Figure 2: contour tracking.

3. Image Segmentation

We present in this section the image segmentation algorithm that is used in the context of contour tracking. It is an unsupervised algorithm that automatically adjusts to the complexity of the observed scene. The algorithm is based on the analysis of the probability distribution of image intensity (gray level) over small neighborhoods, and uses formal decision theory tools to iteratively learn the distributions of the classes present in the image. We first present some results from type theory on which our algorithm is based. Then, we present the segmentation algorithm, and show some results obtained with real images.

3.1 Type theory

Type theory is a branch of statistics that studies repeated realizations of a basic random variable. Let X be a

discrete random variable (rv) with probability space (Ω, A, P) where $\Omega = \{a_1, a_2, \dots, a_L\}$ is the (finite discrete) 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 $x^{(n)} = \{x_1, x_2, \dots, x_n\} \in \Omega^n$ of n

independent realizations of X . The **type** of $x^{(n)}$, which we denote by $\mathbf{n}_{x^{(n)}} : \Omega \mapsto [0,1]$ is the empirical estimate of the probability distribution (pd) of X , and is given by:

$$v_{x^{(n)}}(a_j) = \frac{1}{n} \sum_{i=1}^n 1_{a_j}(x_i), j=1, \dots, L$$

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

We quote without proof several results related to the type of a sequence that will be used in the sequel. The interested reader is referred to [2] for the demonstrations.

Lemma 1.

The probability of observing a given sequence of n iid² realizations of a rv with pd μ depends only on its type and is given by

$$\Pr\{x^{(n)} | \mathbf{m}\} = e^{-n[H(v_{x^{(n)}}) + D(\mathbf{n}_{x^{(n)}} || \mathbf{m})]}$$

In the above expression, $H(\mathbf{n})$ is the Shannon entropy of the distribution v , and $D(\mathbf{n} || \mathbf{m})$ is the Kullback-Liebler directed divergence (also called relative entropy) between v and μ :

$$H(v) = - \sum_{j=1}^L v(a_j) \ln v(a_j)$$

$$D(\mathbf{n} || \mathbf{m}) = \sum_{j=1}^L v(a_j) \ln \frac{\mathbf{n}(a_j)}{\mathbf{m}(a_j)}$$

Both H and D are always positive. Other properties can be found in [2].

Lemma 2.

Let \mathbf{L}_n be the set of all possible types of a sequence $x^{(n)}$. Then,

$$|\mathbf{L}_n| \leq (n+1)^{|A|-1} \leq (n+1)^{|A|}$$

The proof of this statement can be found in [2].

We can now state the following result

Lemma 3.

Consider that we are given two sequences of length n

² iid stands for independent and identically distributed.

$$x_1^{(n)} = (x_{1_1}, \dots, x_{1_n})$$

$$x_2^{(n)} = (x_{2_1}, \dots, x_{2_n})$$

of iid discrete rv 's taking values in alphabet

$\Omega = \{a_1, \dots, a_L\}$ of size $|\Omega| = L$. The MDL (Minimum Description Length, see [1] for a deep discussion of these approach to model selection) test for choosing between the two hypothesis

$$H_0 : x_1^{(n)} \propto p_{\mathbf{m}}^n \quad x_2^{(n)} \propto p_{\mathbf{m}}^n$$

$$H_1 : x_1^{(n)} \propto p_{\mathbf{m}_1}^n \quad x_2^{(n)} \propto p_{\mathbf{m}_2}^n, \quad \mathbf{m}_1 \neq \mathbf{m}_2$$

where the probability distributions \mathbf{m}, \mathbf{m}_1 and \mathbf{m}_2 are unknown, is given by

$$\frac{(L-1)}{L} (2 \log(n+1) - \log(2n+1)) \stackrel{H_0}{>} \underset{H_1}{D(\mathbf{n}_1 \|\hat{\mathbf{m}}) + D(\mathbf{n}_2 \|\hat{\mathbf{m}})}$$

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

$$\hat{\mathbf{m}} = \frac{1}{2} (\mathbf{n}_1 + \mathbf{n}_2)$$

is the empirical estimate of the distribution law under H_1 , which coincides with the balanced mixture of the two types.

The proof of this result can be found in [3]. It shows that the Kulback-Liebler divergence is the relevant metric to decide whether two sequences are realizations of the same rv or of distinct ones. It also indicates that we must compare the two individual types to their mixture, and not directly to each other, as intuition could suggest. The decision test is always well defined, since the original types are necessarily absolutely continuous with respect to their mixture, even if they may not be mutually absolutely continuous.

Finally, we quote also without proof another result that will be used to formulate our algorithm.

Lemma 4.

$$\Pr \{ D(\mathbf{n}_{x^{(n)}} \|\mathbf{m}) > d \mid \mathbf{m} \} \leq e^{-nd}.$$

This Lemma states that the probability of observing a type at a given distance (in the sense of the Kullback-Liebler divergence) from the true distribution decays exponentially.

The proof of this Lemma can be found in [4].

3.2 Segmentation Algorithm

We can now present our unsupervised segmentation algorithm, which is an reformulation of existing clustering algorithms, working in type space. More precisely, we present a variation of the Lloyd algorithm

[5] extensively used in the context of image compression (vector quantization). Although profoundly inspired from this algorithm, our method presents an important difference: contrary to standard K-means clustering techniques, our algorithm does not require the specification of the number of clusters to be defined, and automatically adjusts it to the complexity of the observed image.

We point out that, as said before, our algorithm differentiates the different image regions by comparing local estimates of the intensity level. Thus, it will not be able to distinguish regions that may have distinct texture (granularity) if they result in images with the same gray level distributions. Although this possibility exists, the extensive tests that we conducted on real underwater images show that the algorithm is able to separate the different sea-bed habitats present in all practical cases found until now.

Let I be a $N_l \times N_c$ image, and $I(i, j)$ be the intensity of pixel (i, j) . Define a regular grid of size $n_c \times n_l$

over the original image where $n_c = \frac{N_c}{W_c}, n_l = \frac{N_l}{W_l}$, so

that each point of the grid represents a window of size $W_c \times W_l$ pixels of the original image. Associate to each point of this grid $(i, j) \in \{1, \dots, n_c\} \times \{1, \dots, n_l\}$ the type $\mathbf{n}_{i,j}$ of the intensity values observed in the corresponding window.

The goal of our algorithm is to partition the complete set of types $\{\mathbf{n}_{i,j}\}_{i=1:n_c, j=1:n_l}$ into a set of *homogenous* classes

$C_k, k = 1 \dots, K$, such that the types associated to the same class are plausible realizations of the same rv .

As in the Lloyds algorithm, we start by considering a single class. The best estimate for the underlying probability law is the balanced mixture of all observed types:

$$\mathbf{m}^0 = \frac{1}{n_c n_l} \sum_{i=1}^{n_c} \sum_{j=1}^{n_l} v_{i,j}$$

We then test the initial cluster (formed actually by all the types) for homogeneity, using Lemma 4, by checking the exponentiality of the distribution of the observed $D(\mathbf{n}_{i,j} \|\mathbf{m}^0)$. If this test is passed, it means that the image contains a single region (is homogenous). Otherwise, as in the Lloyds algorithm, we create, from the original class representative (\mathbf{m}^0) two new class

representatives, that will be iteratively modified by the algorithm. Our *splitting* process is somewhat more complicated than the original random process used in the Lloyd's algorithm, which is well suited to work in vector spaces (while in our case, the space of probability laws has, instead, a projective structure).

In the splitting step, we search for two new probability laws such that the representative of decompose the original probability law as a mixture:

$$\mathbf{m}^0 = \mathbf{a}\mathbf{m}^1 + (1 - \mathbf{a})\mathbf{m}^2$$

Randomly generate one of the terms of the mixture as a perturbation of \mathbf{m}^0 :

$$\mathbf{m}^1 = \mathbf{m}^0 + \mathbf{e}$$

where \mathbf{e} must necessarily be larger than $-\mathbf{m}^0$, and its elements sum to zero. Once \mathbf{m}^1 is defined, the previous equation completely defines \mathbf{m}^2 if the scalar \mathbf{a} is given. At this point, we use our mixture model constraint and solve the following optimization problem:

Determination of α

Let $\mathbf{m}_a^2 = \frac{1}{1 - \mathbf{a}}(\mathbf{m}^0 - \mathbf{a}\mathbf{m}^1)$ be the *pd* determined by

a given \mathbf{a} and \mathbf{m}^1 , and let $N_1(\mathbf{a}), N_2(\mathbf{a})$ be the sizes of the elements of the partition of the set of types associated to \mathbf{m}^0 determined by the following association algorithm:

$$\begin{aligned} v_{i,j} &\leftrightarrow \mathbf{m}^1 \\ D(\mathbf{n}_{i,j} \parallel \mathbf{m}^1) &< D(\mathbf{n}_{i,j} \parallel \mathbf{m}_a^2) \\ &> \\ v_{i,j} &\leftrightarrow \mathbf{m}_a^2 \end{aligned}$$

We "split" \mathbf{m}^0 into $\mathbf{m}^1, \mathbf{m}_a^{2*}$, where

$$\mathbf{a}^* = \arg \min_{\mathbf{a}} D(N_1(\mathbf{a})\mathbf{m}^1 + N_2(\mathbf{a})\mathbf{m}_a^{2*} \parallel \mathbf{m}^0)$$

Once the splitting step is done, we re-assign each observed type to the existing class representatives by a criterion of minimal Kullback-Liebler relative entropy to the class representative, and new class representative are computed as the balanced mixture of the types associated to that class. These two steps are iterated until no type changes class.

At the end of this iteration, the final classes are again, tested for homogeneity, by checking the exponentiality of the observed directed divergences to the class representatives. Merge of distinct classes (redundant classes can appear from division of impure original classes) is also tested by using Lemma 3.

3.3 Contour filtering

The segmentation algorithm presented below produces an original segmentation of the image, whose contours have a granularity dependent on the chosen grid size. We defined a post-processing algorithm that adjusts these approximate contours to the boundaries of the image regions. In this subsection we briefly present this work.

Consider a set of (control) points on the boundary detected by the segmentation algorithm. We compute, in a first step, a smooth curve by making a spline fitting to the control points. We then gradually deform this smooth curve by applying to it artificial forces that are generated as follows.

For each contour line, consider three image strips of constant width (equal to the size of the original grid used by the segmentation algorithm): one centered at the smoothed control points, one on its left and another on its right. Define a set of equidistant points $\{p_n\}$ located

along the smoothed contour. Let $\mathbf{m}^l, \mathbf{m}^r$ be the representative *pd* of the classes associated to the regions on the left and right of the contour, respectively. Define

$$\mathbf{n}^r = 0.75\mathbf{m}^r + 0.25\mathbf{m}^l$$

$$\mathbf{n}^c = 0.5\mathbf{m}^r + 0.5\mathbf{m}^l$$

$$\mathbf{n}^l = 0.25\mathbf{m}^r + 0.75\mathbf{m}^l$$

Let \mathbf{n}_n be the type of the set of pixels inside a window centered at p_n and lying on the central strip. The adjusting algorithm iteratively displaces these points along the normal to the contour at that point, \vec{n}_n , according to the following rule:

$$p_n^{k+1} = p_n + \mathbf{b}\vec{n}_n$$

where

$$\mathbf{b} = \begin{cases} +\mathbf{d}, & \text{if } D(\mathbf{n}_n \parallel \mathbf{n}^r) < D(\mathbf{n}_n \parallel \mathbf{n}^l), D(\mathbf{n}_n \parallel \mathbf{n}^c) \\ 0, & \text{if } D(\mathbf{n}_n \parallel \mathbf{n}^c) < D(\mathbf{n}_n \parallel \mathbf{n}^l), D(\mathbf{n}_n \parallel \mathbf{n}^r) \\ -\mathbf{d}, & \text{if } D(\mathbf{n}_n \parallel \mathbf{n}^l) < D(\mathbf{n}_n \parallel \mathbf{n}^r), D(\mathbf{n}_n \parallel \mathbf{n}^c) \end{cases} \quad \mathbf{A}$$

An optimal version of this procedure would solve for the problem of estimating the degree of mixing of the central strip (which ideally would be an equal mixture of the two neighboring classes), implying the solution of a minimization problem. The approximate procedure described above yields satisfactory results with less computations.

We note that this procedure is reminiscent of the deformable templates (snakes) approach. In our case, no internal forces are required, the continuity of the resulting

contour being guaranteed by the comparison with the same classes' representatives (determined by the segmentation algorithm).

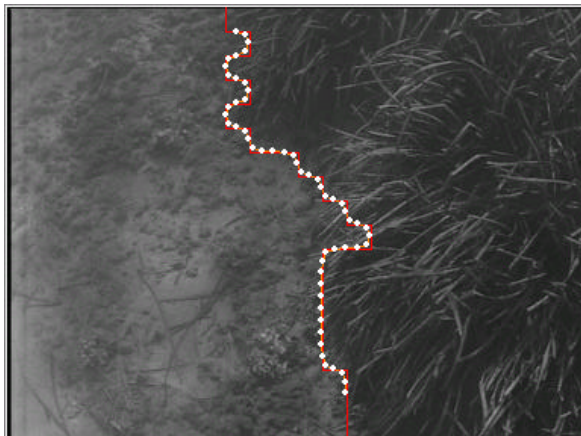
3.4 Adaptive segmentation

Actual use of the segmented images for contour tracking imposes stringent constraints on the complexity image processing step. We defined an adaptive version of the previous algorithm, which, starting from an initial full-complexity step (the algorithm described in section 3.2), adaptively updates the number of classes and their representatives, accommodating slow variations of the classes' characteristics. Instead of starting from scratch at each new image frame, the algorithm uses the prototypes obtained in the previous frame, assigns the observed types to one of them (according to minimal Kullback divergence) and updates them. After convergence, the classes are tested for homogeneity, as explained in section 3.2, and new classes are created if required. These steps are repeated until the number of classes remains constant.

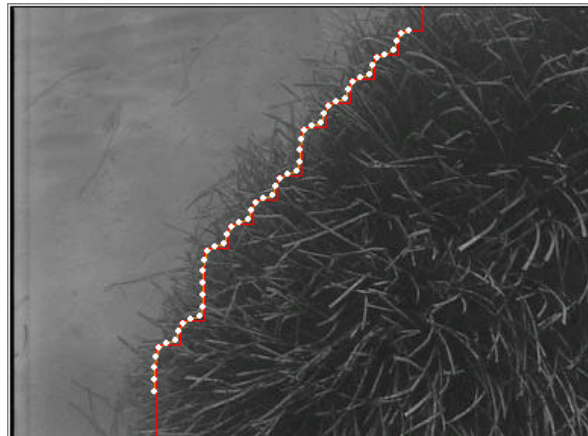
3.5 Results

We present in this section results obtained with the algorithm described in the previous subsection on real underwater images taken with the ROV Phantom,³ equipped of a video camera pointing at the sea bottom.

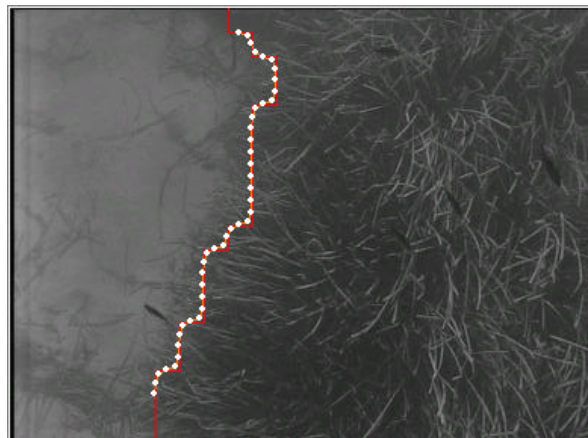
The test set contains images from Villefranche-sur-mer, in the south of France where only two classes are present (sand and posidonia). The following images show the results obtained: the first frame (where the initial representation of the two classes present is learned) and frames numbers



frame # 000.



frame # 050.



frame # 200.

4. Contour Tracker

The main objectives of the contour tracker are:

- (i) to maintain the vehicle at a constant distance from the bottom (altitude stabilization),
- (ii) to track the sea-bed boundary detected by the image segmentation algorithm.

We consider that the vehicle body motion is controlled with the usual decoupled structure:

- altitude closed loop controller,
- heading closed loop controller,
- surge speed open loop controller.

Note that this control structure is justified for the Phantom 500. This vehicle is equipped only with two horizontal thrusters and a vertical thruster : the number of actuators is smaller than the number of degrees of freedom. Pitch, roll and sway dynamics are not controlled but assumed to be intrinsically stable. The three thrusters are speed controlled. The 3 decoupled design models for the body motion are reasonably simple and capture the main dynamical features of the vehicle dynamics. The basic control structure (inner body motion control loops

³ Phantom is a product of Deep Ocean Engineering, USA, and has been available to the team working in projects Narval and Sumare thanks to special educational arrangement.

and the feed forward surge controller) in loop with the plant is depicted in Fig.4. Input signals u_r , ψ_r , h_r represent the surge reference, the heading reference and the altitude reference, respectively. For a reasonable set

of dynamics, we assume that the measured signals ψ , h track the reference signals ψ_r , h_r .

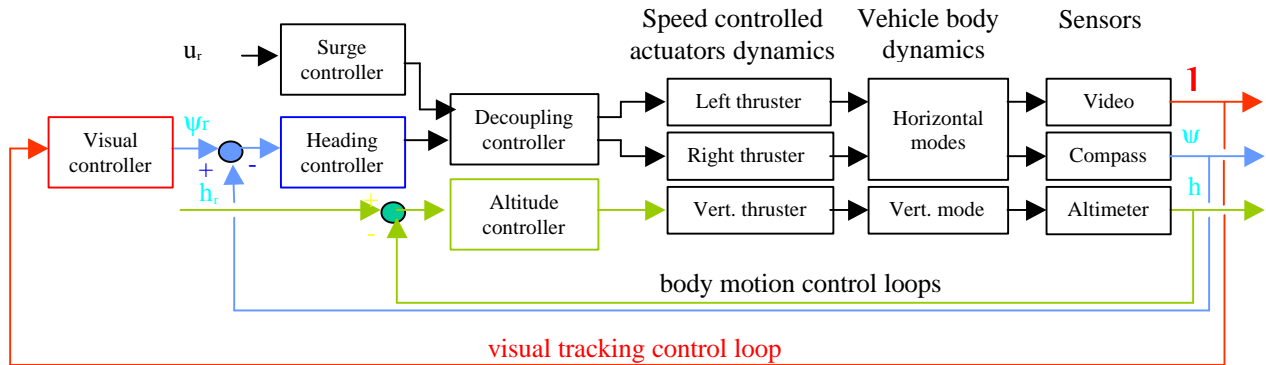


Fig.4 : The vehicle in closed loop with the basic body motion controllers and the visual controller.

When the basic control loops are active, the control objective (i) is ensured by imposing a fixed altitude reference signal h_{ref} (1 m for instance). The control problem reduces to steer the vehicle in the horizontal plane. The approach retained in this paper consists in fixing a constant surge for the vehicle (constant reference surge signal u_r , 1m/s) and to minimize a distance in the image. The trajectory of the vehicle in the global frame and the sea-bed contour are represented in Fig 5.

The basic control structure imposes a heading reference input (the visual controller output). According to the physical interpretation of the visual control

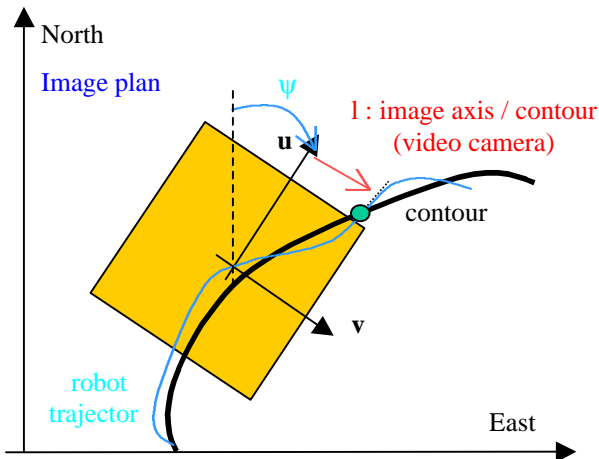


Figure 5: tracker control variables.

The proposed approach to impose control objective (ii) consists in minimizing the distance l between the contour and the axis u of the camera. This distance can be interpreted as the error of the visual controller. A physical interpretation of the control strategy is to steer on the left the vehicle when signal l is positive. Naturally, the control signal for this loop is the vehicle yaw.

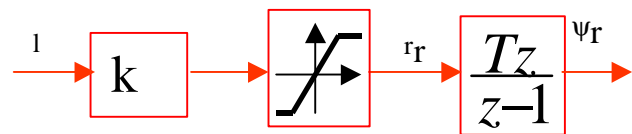


Figure 6: visual controller.

strategy, and to adjust the input of the heading loop (expressed in the global reference frame) to the visual loop (which process in the body frame), the last block of the visual controller is an integrator, see Fig. 6.

The intermediate bloc is a saturation which limits the slope of the heading reference ψ_r . The first bloc is a gain, k . Proper tuning of this parameter allows the achievement of satisfactory tracking performance adjusting the robot dynamics to the variation of the contour curvature.

The following figure shows the input and output signals of the visual controller, l and ψ_r during a real experiment at sea (Villefranche-sur-mer). Note that the sign of derivative of the controller output (ψ_r) induces a decrease of the visual error l .

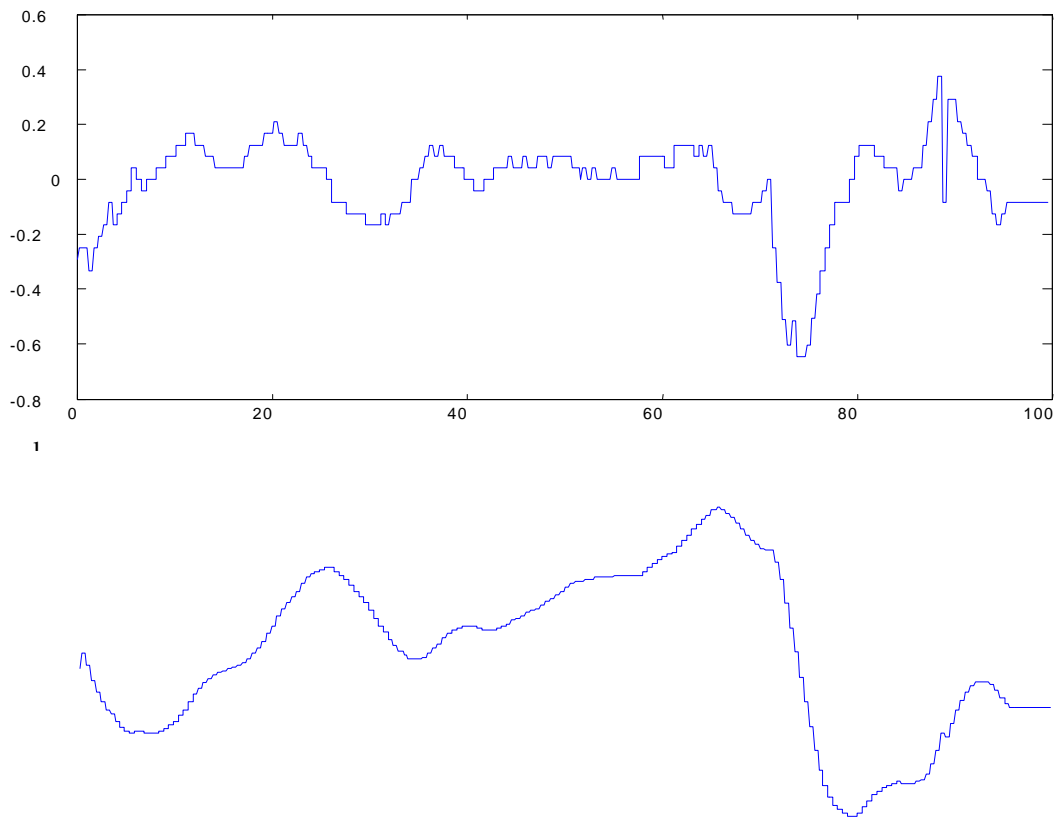
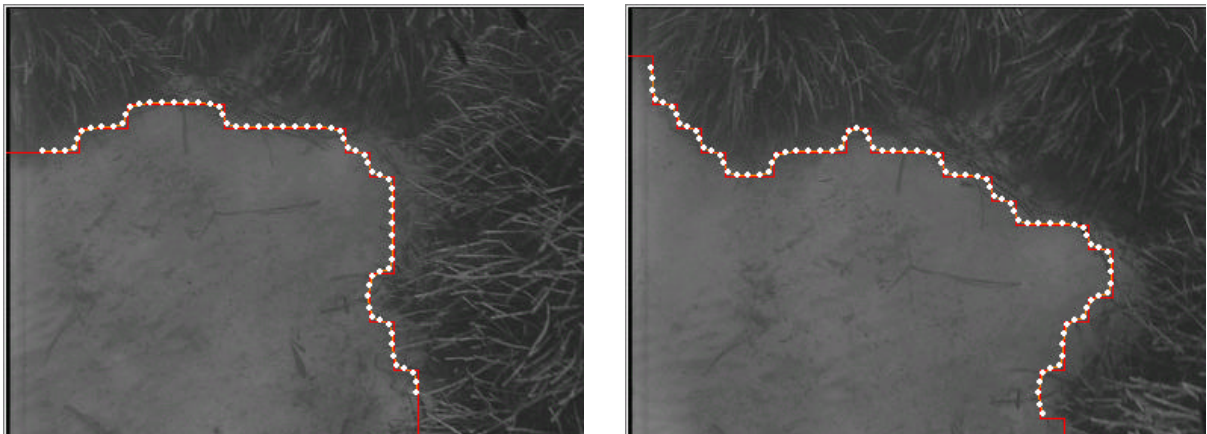
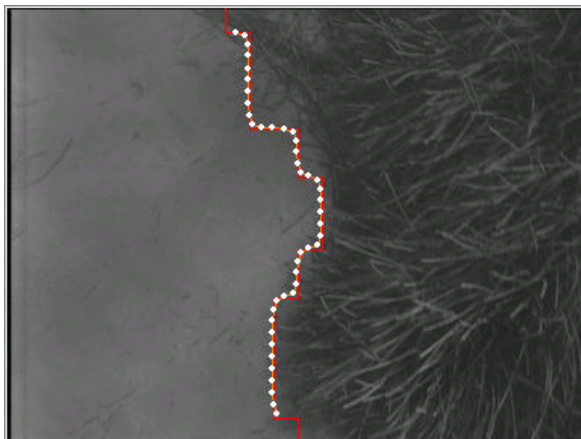
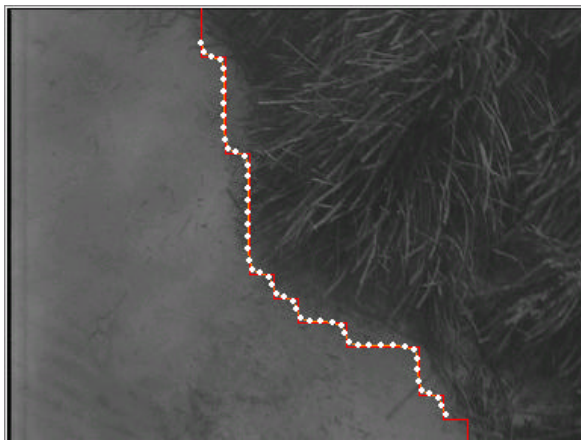
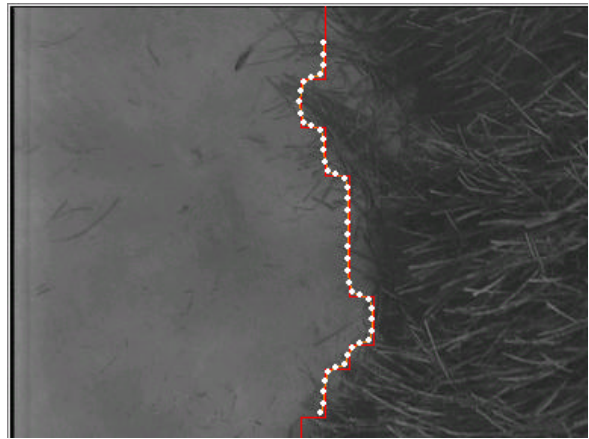
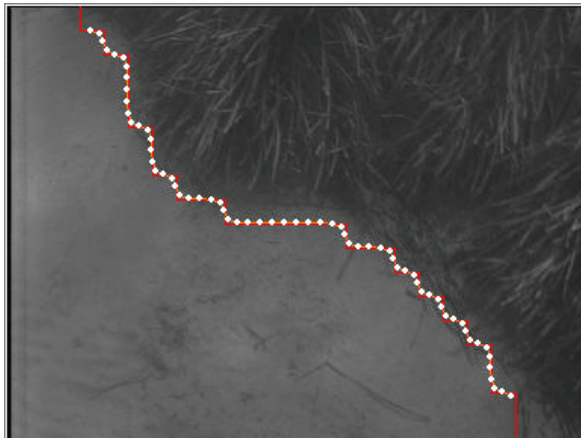


Figure 6: distance to contour (top) and robot heading reference (bottom) versus time.

5. Results from real experiments

The next images show frames recorded during a contour tracking experiment at Villefranche-sur-mer, where the robot follows a curved part of the boundary between posidonia and sand.





5. Conclusions

In this paper we presented a novel algorithm for image segmentation based on computation of local estimates of the distribution of the intensity level of the image. The algorithm is based on information theory concepts, and automatically adjusts to the complexity of the observed image estimating the number of classes present and their characteristics. We also presented a companion contour adjusting algorithm, which enables more precise estimation of the regions' boundaries. The detected contour is the input to a contour tracking controller, which issues the heading commands that keep the observed contour in the visual field of the robot. Results of the image segmentation on real images acquired in two distinct environments, and of contour tracking in a real underwater environment are presented, showing the adequacy of the proposed approach.

Acknowledgements

This work has been partially funded by the European Union through research contracts NARVAL (LTR-20185, Navigation of Autonomous Robots via Active Environment Perception) and SUMARE (IST-1999-10836, Survey of Marine Resources).

The authors acknowledge the collaboration of Emmanuel Gallo and Olivier Parra on the development of the image processing software that is used in this study.

Bibliography

- [1] Jorma Rissanen, *Stochastic Complexity in Statistical Inquiry*, World Scientific, Series in Computer Science—Vol. 15, 1989.
- [2] Thomas Cover and Joy A. Thomas, *Information Theory*, John Wiley & Sons, Wiley Series in Telecommunications, 1991.
- [3] Albert Tenas, *Unsupervised Segmentation of Images Based on the Kullback Distance*, Dissertation, Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis, UPC Barcelona, June 2001.
- [4] Amir Dembo and Ofer Zeitouni, *Large Deviations Techniques and Applications*, Jones and Bartlett Publishers, Inc., 1992.
- [5] S. P. Lloyd. Least squares quantization in PCM. IEEE Trans. Information Theory, 28: 129-137, 1982.