Camera and Sonar data fusion

Lubin Kerhuel

DEA report
& Final engineering school internship report

March-July 2004

Supervised by Maria João Rendas & Christian Barat (I3S - SAM Project)

ESIEE Tutor : Céline Igier

DEA "SIgnal et teleCOMunication numérique"
École Doctorale STIC
École Supèrieur d'Ingénieur en Électronique et Électrotechnique Paris

I3S - CNRS - Université de Nice Sophia Antipolis
Les Algorithmes, Bât. Euclide B, 2000 Route des Lucioles
BP 121, F-06903 Sophia Antipolis Cedex France

Contents

List of Figures

acronyms

MDL Minimum Description Length

PL Probability Law

iid Independent and Identically Distributed variable

ROV Remotely Operated Vehicle

DCT Direct Cosine Transform

SVD Singular Value Decomposition

PCA Principal Component Analysis

AGC Automatic Gain Control

SAM Systemes Autonome Mobile

Introduction

The goal of the Systemes Autonome Mobile (SAM) project is to give a robot the ability to explore unknown regions without getting lost. The robot must be able to determinate its position without using external reference marks (like acoustic beacons or artificial visual landmarks) or global positioning system data (like GPS or Galileo). For that, it must build a map of the region where it evolves, using the information it gathers about its environment through its perception sensors. The project SAM considers more specifically the problem of autonomous navigation of underwater robots, and conducts regularly sea experiments in the bay of Villefranche using the underwater Remotely Operated Vehicle (ROV) Phantom.

The ROV Phantom operated by I3S is equipped with a set of sensors for navigation and perception, that includes a video camera and a profiler sonar. At present, two separate algorithms have been developed allowing the ROV to track the boundary between two distinct sea-bed regions, for example, between sand and algae. One of the algorithms bases robot guidance on visual (image) information while the other uses the acoustic (sonar) data.

Boundary tracking using camera information is perturbed by sunlight refractions in shallow water, that create unstable artificial bright regions in the acquired images. Boundary tracking using the sonar profiler data has the advantage of not being perturbed by sunlight and also that it is robust with respect to water turbidity, in situations where the visual sensor is not able to accurately locate the regions boundary. However, the sonar only yields, at each sampling instant a narrow view of the sea bottom (a point) and the spatial registration of the received data is sensitive to the errors in the determination of the robot's altitude above sea bottom.

The information extracted from the sonar and the camera are perturbed by different phenomena. Moreover, these two sensors are complementary, in the sense that the sonar data is restricted to a single point of the sea bottom, wile the camera senses a window of positive area but the acoustic sensor allows the determination of the 3D geometry of the bottom, while the camera only provides a planar projection that does not directly yields distance measures. The aim of the stage is to fuse the information extracted from both the sonar and the camera to design a boundary tracking system more robust and with better performance.

Section 1 describes a recursive image segmentation algorithm. This algorithm has the particularity of determining, using the Minimum Description Length (MDL) principle, the number of classes present in the image.

The sonar retrieves about 40 echo per seconds, while the camera acquisition rate is 24 images per second. The information quantity given by the camera is higher than information given by the sonar in the sense that more information can be extracted from the camera images. It is relevant to extract as much information as we can from the sonar to get an equilibrated system of sensors. The algorithm that uses sonar information to guide the robot that is presently implemented in the ROV Phantom, classifies the sonar profiles in one of two classes based only on the energy of the received profiles. In the section 2, we present a classification criteria that considers more general information than profile energy, and compare the results obtained with the performance of the previous algorithm.

To be able to combine sonar and image data, it is necessary to be able to establish the correspondence between sonar impact points and image pixels. For this, and since both sensors data are defined with respect to their own coordinate system (the sensors are *self-centric*) we have to determine the position and orientation of the sonar's frame with respect to the camera's own coordinate system. The video camera of the Phantom is mounted on a tilting mechanism attached to the crash-frame of the robot, and is susceptible to move between experiments. For this reason, estimation of the relative position and orientation of the sonar head and of the video camera should be done at the beginning of each mission. In section we present a method that allows to corresgister the sonar impact points in the video images, under a set of simplifying assumptions.

The current version of the report does not assess the target goal of actually fusing sonar and video data for boundary tracking. This will be done until the end of the intership, and a consolidated report will be provided, containing the results that will be produced. Two approach are envisaged: we can separately classify the sonar returns and the video frames (using the methods presented here), and fuse in a second step the results of the individual classifications, or we can, alternatively, perform classification of "augmented data vectors" composed of the video and acoustic data. The first approach seems to be more robust with respect to the asynchronous sampling rates of the sonar and the video camera, and requires that indexes qualifying the reliability of both the camera and sonar classifications be found, indicating the degree of confidence associated to each classification result. The second approach assumes that we can simultaneously acquire one video image and one sonar profile, but can be directly handled with the segmentation approaches studied in the frst part.

1 Image segmentation

In the context of the work described in this report, the aim of image segmentation is to recognize different objects in a video frame, to subsequently build a map to be used for navigation purposes. This section describes the *segmentation* (or *clustering*) process. An exemaple of final result of one image clustering is presented in figure 8.

1.1 Clustering

The methods studied consider that the image is partitioned in a set of small windows defined by a rectangular grid. A statistical approach is used to classify each window: the probability distribution of the pixels' grey level inside each window is compared to the distribution law of a set of reference classes, which are automatically determined from the image. Creation of the references classes is discussed in section 1.1.1. Each image window is then associated to the class for which the Kullback-divergence (defined in equation (1.1) and that measures similarity between probability laws) between the window's grey level distribution and the distribution of the reference class is minimum.

The algorithm comprises two separate steps. In the first step, it builds the references classes (their probability laws) along with the associated partition of the original image. A very fine partition of the image can be obtained if we let the number of reference classes be large. In the second step, the algorithm deletes similar reference classes. We discuss in section 1.1.2 how similarity is detected and the merged to form new classes.

Refer to the diagram of page 9 describing the for a general view of its structure.

1.1.1 Creating the reference classes

The Kullback-Leibler divergence [1] is used as a "distance" that measures the similarity between the grey level distributions of two windows. Its definition is:

$$D(\nu_1|\nu_2) = E\left[\nu_1 \log \frac{\nu_1}{\nu_2}\right] \tag{1.1}$$

Some properties of this divergence are presented in the Annex A p.(37)).

The function creating the references classes recursively splits the image windows' grey level distribution into two new distributions. Let P_0 be a structure containing the grey level probability law of each window of the image to be segmented. The function splits the grey level distribution P_0 into two new distributions P_1 and P_2. Then, the MDL principle is used (see (A.9) p.(39)) to decide whether the division of P_0 into P_1 and P_2 is relevant. If the classification of P_1 and P_2 is relevant, the algorithm tries to split each one recursively: P_0 is replaced successively by P_1 and P_2 in the algorithm. If the classification of P_1 and P_2 was not relevant, the MDL principle indicates that the distribution laws of the windows that lead to the creation of P_1 and P_2 are not significantly different from P_0. In this case, the recursion stops and returns the reference class found as the average of the histograms composing P_0. See [2] for more information on the MDL principle.

Lets have a closer look on the *splitting* process. The aim of the splitting process is to produce the two distribution laws ν_1 and ν_2 that characterize the two most different region of the image. Finding these two distribution laws is equivalent to find a partition into two parts of the image windows. ν_1 and ν_2 are then calculated as the average of the distribution laws of the windows of each part found. The two reference classes distributions ν_1 and ν_2 are determinated iteratively. Initially, ν_1 and ν_2, are created as follow: a random ϵ is drawn satisfying the following constraints:

- $\sum_i \epsilon_i = 0$
- $\epsilon_i \leq \mu_{0_i} \frac{1-\alpha}{\alpha}$ $\alpha \in [0...1]$ This condition is to guarantee $\mu_{2_i} > 0$ $\forall i$

We assume that

$$\mu_0 = \alpha\mu_1 + (1-\alpha)\mu_2 \ \alpha \in [0...1]$$

$$\begin{aligned} \mu_1 &= \mu_0 + \epsilon \\ \mu_2 &= \frac{1}{1-\alpha}(\mu_0 - \alpha\mu_1) \end{aligned} \tag{1.2}$$

where μ_0 is the average grey level distribution of all the windows to be classified. In the first iteration, where we do not know ν_2, the parameter α is chosen as the solution to the following minimization problem:

$$\alpha = \arg\min_{\alpha} \left(\sum_{P_1(\alpha)} DivergenceKullback(\nu_1, \mu_0) + \sum_{P_2(\alpha)} DivergenceKullback(\nu_2, \mu_0) \right) \tag{1.3}$$

where ν_1 and ν_2 are the distributions of the pixel intensities of the subsets P_1 and P_2, respectively. A numerical approach is used to solve eq(1.3) by evaluating a discrete set $\alpha_1, \alpha_2, ..., \alpha_n$. A splitting is realized for ν_2 corresponding to several values of α. This procedure is computationally heavy because we must compute the partitions $P_1(\alpha)$ and $P_2(\alpha)$ for all α tested.

Using the approach described above we determine ν_1, ν_2 and α. The image is then partitioned using the minimum Kullback dvergence. For each window, two Kullback divergences are computed. The first one is the Kullback divergence between the window's grey level probability law and ν_1, and the other using ν_2 instead of ν_1. The window is associated to $\nu_{i\in[1\ 2]}$ for which the divergence is minimum. Once all windows have been classified, ν_1 is updated as the mixture of the grey level distributions of the window associated to the first class. Then, α should be determined like in the previous step. However, a faster method is used to approximate its value. We use the equation (A.11) where ν_k is the grey level distribution of the whole image. The ν_2 used to approximate α is the average of the grey level distribution of each window associated to its class. Once α is determined, ν_2 is recomputed using equation (1.2) to ensure that μ_0 is a linear combination of ν_1 and ν_2. All theses steps are iterated until no changes at the partitions P_1 and P_2 are observed.

$$\alpha = \frac{D(\nu_k|\mu_2) - D(\nu_k|\mu_1) + D(\mu_2|\mu_1)}{D(\mu_1|\mu_2) + D(\mu_2|\mu_1)} \tag{A.11}$$

Alg. 1 Image Classification

```
First call : Recursion($P_0$)
Require: All Image windows histograms and their respective x,y position
Ensure: Classification into distinct classes according to the MDL principle
function Recursion($P_0$)
$\mu_0 \Leftarrow$ average of all histograms composing $P_0$
$\epsilon \Leftarrow$ constrained aleatory vector
$\nu_1 \Leftarrow \mu_0 + \epsilon$

$Tmp \Leftarrow \infty$
for some $\alpha' \in ]0\ 1[$ do
  $\nu_2' \Leftarrow \frac{1}{1-\alpha'}(\mu_0 - \alpha'\nu_1)$
  $P_1 \& P_2 \Leftarrow$ Classification : minimum distance from respectively $\nu_1$ and $\nu_2$
  if $Tmp > \left(\sum_{P_1} DivKullback(\nu_1, \mu_0) + \sum_{P_2} DivKullback(\nu_2, \mu_0)\right)$ then
    $Tmp \Leftarrow \left(\sum_{P_1} DivKullback(\nu_1, \mu_0) + \sum_{P_2} DivKullback(\nu_2, \mu_0)\right)$
    $\nu_2 \Leftarrow \nu_2'$
  end if
end for

repeat
  $P_1 \& P_2 \Leftarrow$ Classification : minimum distance from respectively $\nu_1$ and $\nu_2$
  $\nu_1 \Leftarrow$ histograms bins mean of $P_1$
  $\nu_2 \Leftarrow$ histograms bins mean of $P_2$
  $\alpha \Leftarrow \frac{D(\mu_0|\nu_2) - D(\mu_0|\nu_1) + D(\nu_2|\nu_1)}{D(\nu_1|\nu_2) + D(\nu_2|\nu_1)}$
  $\nu_2 \Leftarrow \frac{1}{1-\alpha}(\mu_0 - \alpha\nu_1)$
until No classification modification

if $P_1$ not homogeneous to $P_2$ then
  Recursion($P_1$)
  Recursion($P_2$)
else
  Store the type $\mu_0$ as the average of $P_1$ merged to $P_2$
end if

END Recursion



repeat
  Reclassify with all histograms found
  Suppress Classes that do not comply with MDL principle
until No changes
```

1.1.2 Adapt and suppress references classes

The algorithm of section 1.1.1 associates one class to each image window. Let N be the number of different classes found in the whole image. These N references classes are calculated as the average grey level distributions of the windows associated to each of the N classes. Due to the recursive structure of the algorithm described in section 1.1.1, the classification found may not be optimal because each sub-classification is independent from the others . Each class is split independently based on the associated partition. In the second step of the algorithm, the entire image is reclassified using all reference classes $\nu_{[1 \dots N]}$. The reclassification is performed as follow: each window of the grid is associated to the ν_n for which the Kullback-divergence is minimum. Then, the ν_n are re-estimated as the average of their respective class. The reclassification process stops when no changes occur.

During the reclassification process, some classes may have an empty partition and are thus eliminated. Nevertheless, some similar but not empty classes which do not satisfy the MDL criterion will remain. These classes are then deleted using the following heuristic: the MDL criterion (see (A.9) is computed for all pairs of classes. The two classes for which the MDL criterion is the smallest and smaller than the threshold are merged. The image is then reclassified using algorithm described below. This process is repeated until the worst MDL criterion is compliant. In practice, no merging occurs. All classes satisfy the MDL criterion after the recursive class division. A deeper study should provide a reason for this behaviour. However, the class reduction algorithm can have another aim. It allows to set a max number of classes, N_{max}. If the number of classes in the MDL sense found is superior to N_{max}, the two closest classes are merged until the target number of classes, N_{max}, is reached .

1.2 Results

The results are presented here in the chronological order they have been produced. The first figure (fig.1) shows graphs illustrating the determination of the first two references classes $\nu_{i \in [1\ 2]}$. The white and grey pictures in the middle show the resulting classification. μ_0 always represents the grey level distribution of the partition to which splitting is applied. In the case of (fig.1), the partition is the complete picture (the first recursion of the algorithm). ν_1 and ν_2 are the grey level distributions of the two classes found. The last image on this figure shows the Kullback divergence between the grey level distribution of each window to the two reference distributions (ν_1 and ν_2).

Figure 2 shows the grey level distribution of each of the 8 classes found by the recursive iteration. A typical texture corresponding to each distribution is also created using random generators with probability law equal to the distributions of each class. Figure 3 shows the original picture in the upper left part. The upper right plot represents the different classes found, coded with distinct grey levels. The first class, ν_1, is shown in black and the other one is presented in white. The lower left plot shows a reconstructed image using the final classification and the textures shown in figure 2. The lower right picture shows the region boundaries.

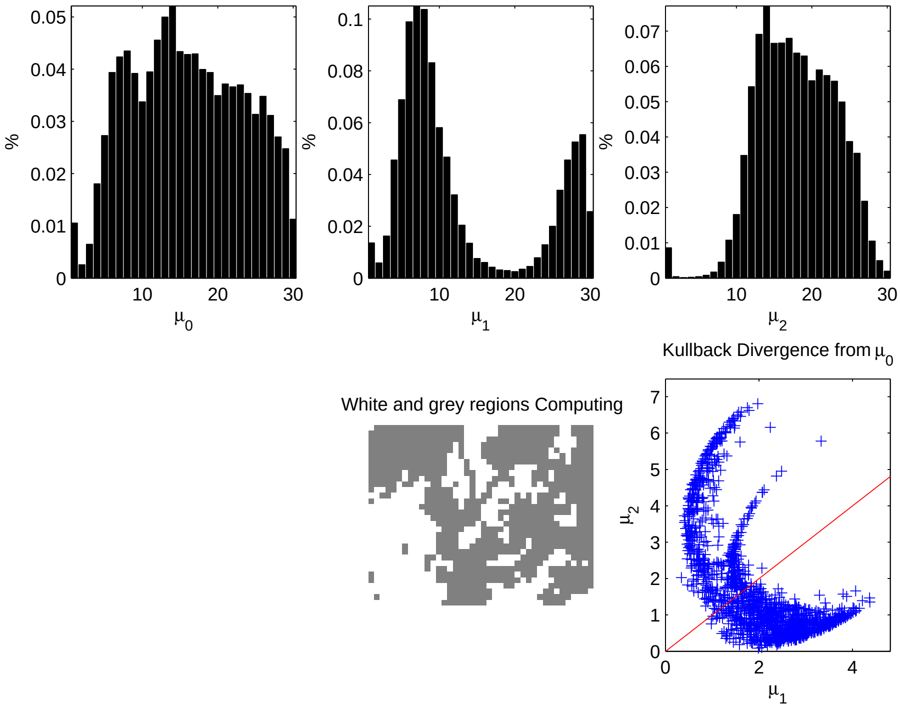

Figure 1: Classificating an underwater picture

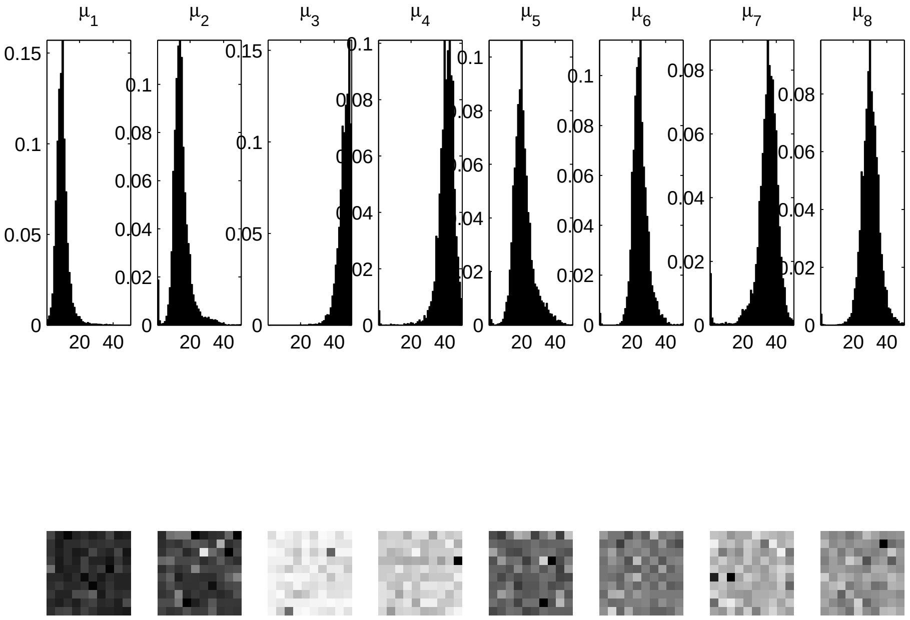

Figure 2: Types of Classification found with 50 bins and 10*10 windows size of an underwater picture

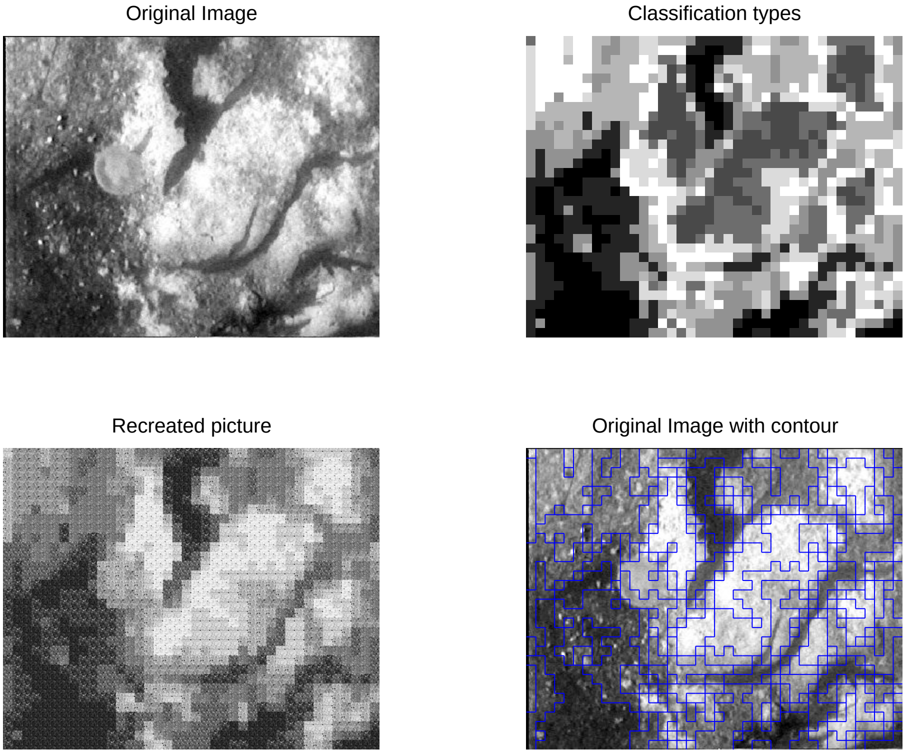

Figure 3: Classification found with 50 bins and 10*10 windows size of an underwater picture

Figures 4 and 5 show the result after some classes have been merged two by two. In this case merging was not based on the MDL principle. Instead, we manually limited the number of classes to three, and merged the closest classes based on the Kullback-divergence.

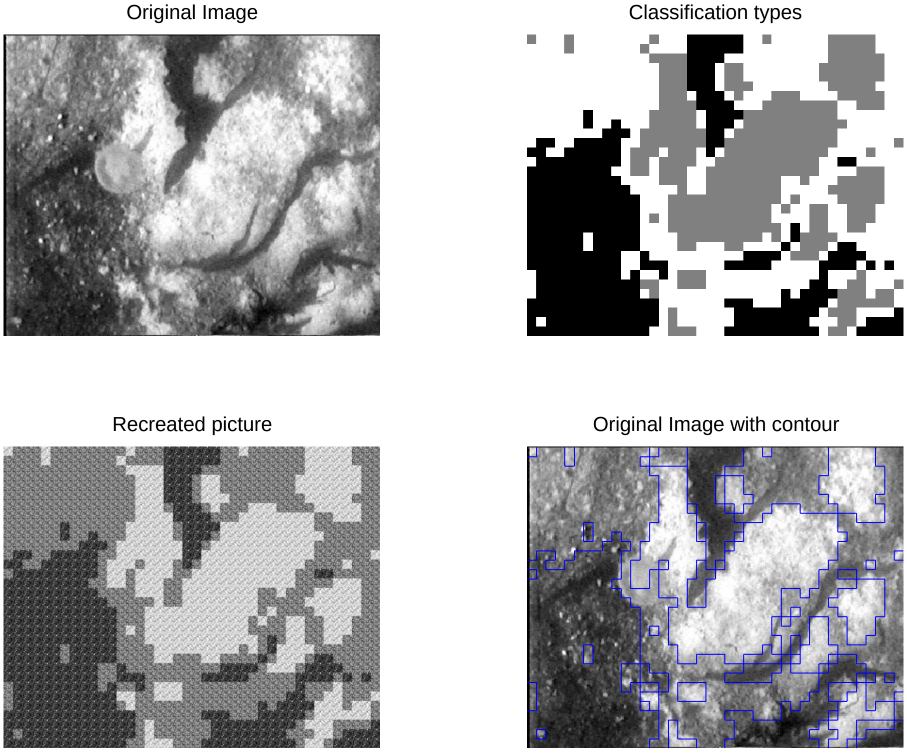

Figure 4: reduced Classification found with 50 bins and 10*10 windows size of an underwater picture

Figures 6 and 7 show the result for a contrasted image, before and after classes merging. We can see on (fig.7) that one class contains at the same time the lightest values of the image (the sun), and the grey corresponding to the upper part of the sky.

Figure 8 shows the result of fitting a spline curve to the boundary of the classes. The number of classes was limited to three. Several splines have been computed. Each one is the contour of one region corresponding to one reference distribution found.

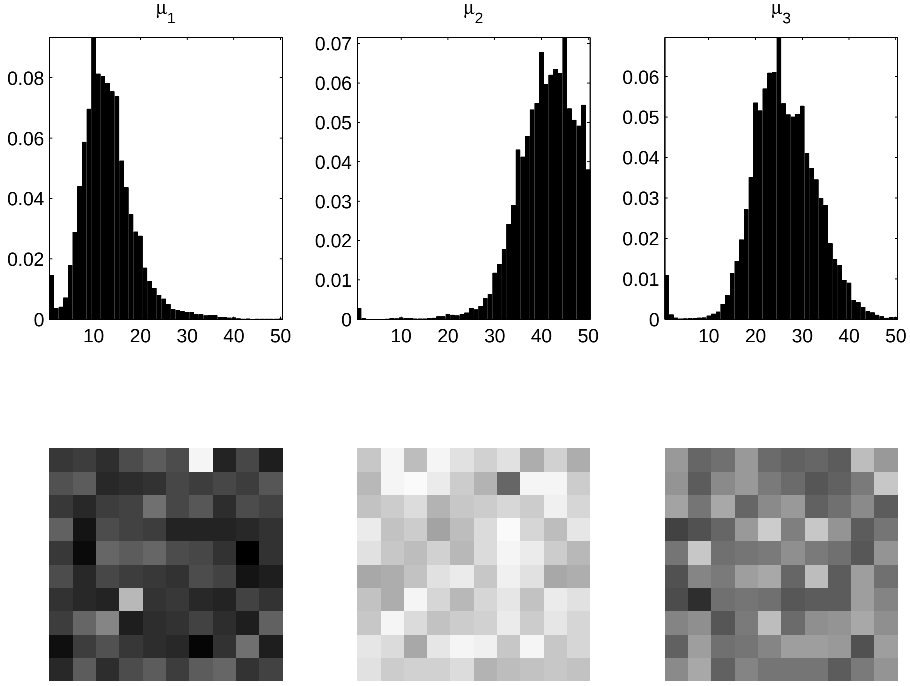

Figure 5: Types of reduced Classification found with 50 bins and 10*10 windows size of an underwater picture

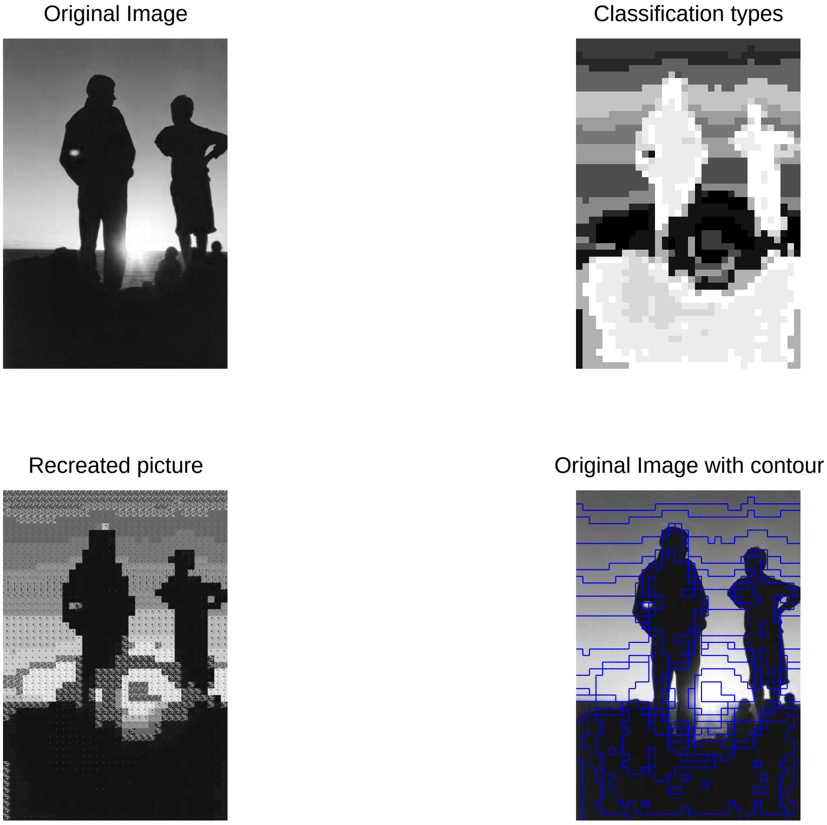

Figure 6: Classification found with 100 bins and 10*10 windows size of a contrasted picture

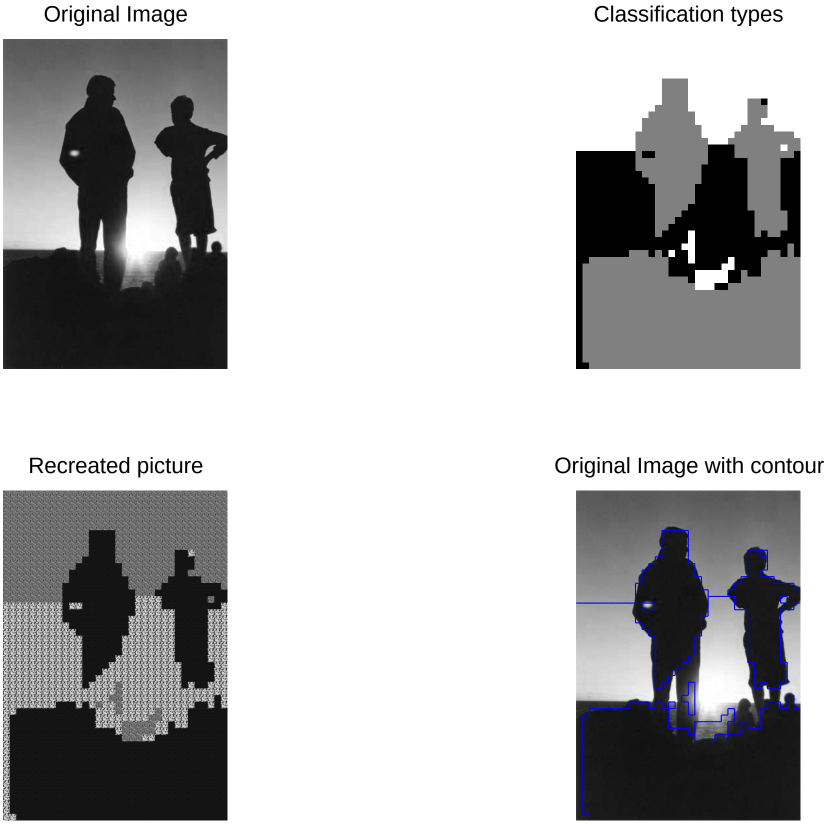

Figure 7: Reduced classificatioin found with 100 bins and 10*10 windows size of a contrasted picture

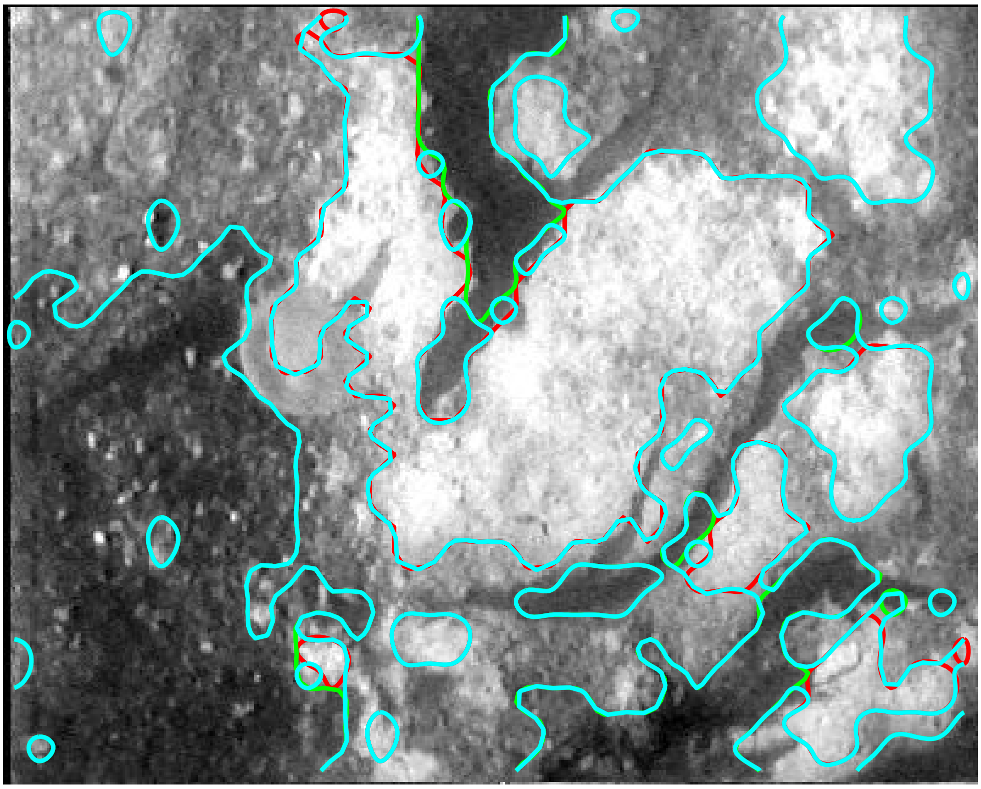

Figure 8: spline on Classification limited to 3 classes

1.3 Results Comments

One interesting result is that the Kullback-divergence between the different classes found with the recursive algorithm comply with the MDL test.

Another result is that the MDL test realized using two windows from both side of a frontiers always says that both windows belong to the same class. This is directly due to the MDL test. From Equation (A.9) taking $N_1 \approx N_2 \approx N$, we obtain equation (1.4). We see that the left term is proportional to $\log(N)$ and right term is proportional to N. Because too few data are presented (small N), the test will always choose H_0. In addition, the precision of the frontiers is not accurate because it is necessarily defined on the grid. Moreover, the real physical frontiers are usually not thin but progressive.

$$(M-1)\log\left(\frac{N^2+1}{2N+1}\right) \lessgtr_{H_0}^{H_1} N\left\{\sum_i \left(\mu_1^i * \log(\frac{2\mu_1^i}{\mu_1^i+\mu_2^i})\right) + \sum_i \left(\mu_2^i * \log(\frac{2\mu_2^i}{\mu_1^i+\mu_2^i})\right)\right\} \quad (1.4)$$

2 Sonar profiles

The image processing described in section 1 allows the ROV to create a description of natural boundaries by fusing local views of the contour. However, several phenomena like refraction of sunlight on the seafloor or turbidity can result in a misinterpretation of perceptual information provided by the camera. A sonar, added to the set of on-board sensor, can provide complement of information. Fusing data from both the sonar and camera should result in a more robust positioning and mapping system. In this section, we describe the processing of the sonar data.

A single profile does not only contains distance information but also information about the seafloor structure expressed by the diffusion of the received energy. Let P_i be the profile corresponding to the emission i. P_i^k is the bin energy received after a lap time proportional to k after the i^{th} emitting. In each profile of the sequence $P_1, P_2, ..., P_N$ we consider only a subset of the entire profile $P_i^{[U]}$ corresponding to the instant of the echo reception. The $[U]$ interval of length 50 is different for each profile. It is chosen so as to have the profile centered [1]. $P_i^{[U]}$ is the vector used during the classification process. $P_i^{[U]}$ will sometimes be noticed as P_i to lighten the notation.

The aim of our study is to distinguish, from theses profiles, the type of the seafloor (e.g. sand, alga,...) that has reflected the acoustic signal. The shape of $P_i^{[U]}$ provides the information that allows to distinguish among different types of the seafloor. Only this energy shape is used in the method developed here to classify the seafloor.

2.1 Data Pre-Processing

A mechanically scanning profiler sonar is used that gather up to forty profiles per second. A sequence of profiles is shown in figure 9 [2]. On the X axis are presented the $P_{i=[1:800]}$ profiles. The y axis is the time diffused energy distribution $\{P_1^{[U]}, P_2^{[U]}, ..., P_{800}^{[U]}\}]$ composing the received profile. The color, from blue to red represent the echo power intensity. Theses profiles have been recorded during a boundary tracking between two regions, respectively composed of sand and algae. During this experiment, the altitude of the ROV was maintained at 1 meter. For each profile, we can distinguish two components. The first correspond to $P_i^{\{1,...35]\}}$ is due to reflections of the transmitted sonar pulse on the ROV crash-frame and contains no information about the environment. The second part of the profile corresponds to the actual seafloor reflections. Only this second part is analyzed, by removing the first 50 values of each profile P_i. $[U]$ is a sub interval contained in this second part.

The profile is normalized : $\forall i \ \sum_{U_{1...n}} P_i^{[U]} = 1$.

2.2 Several discrimination methods to classify profiles

2.2.1 Classification algorithm

The Lloyd algorithm is used to classify the profiles introducing two partition on the image. Lets $[A]$ and $[B]$ be the profile indexes associated to each part $P_{[A]}$ and $P_{[B]}$. The Lloyd algorithm processes iteratively : at the first iteration, two reference vectors R_1^0 and R_2^0 of same length as $P_i^{[U]}$ are created.

[1] The centering method is describe in section 2.3

[2] The document should be color printed to see echo intensity

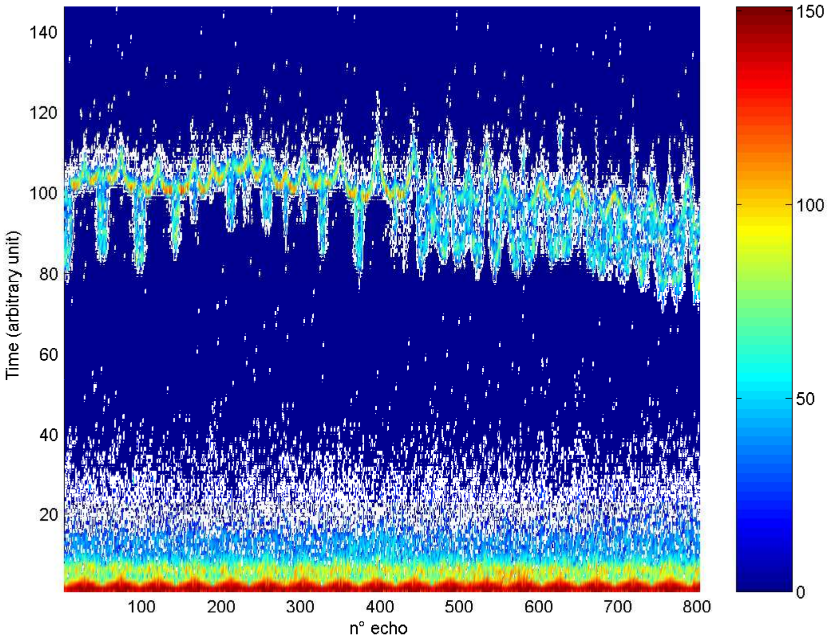

Figure 9: Echos retrived from the scanning profiler sonar

For the following iteration, the profiles are associated to the nearest reference vector, creating two partitions $P^0_{[A]}$ and $P^0_{[B]}$. Then, R_1 and R_2 are updated as the centroid of each part : $R_1^{k+1} = Mean(P^k_{[A]})$ and $R_2^{k+1} = Mean(P^k_{[B]})$. The algorithm iterates until the partition remain unchanged $P^{k+1}_{[A]} = P^k_{[A]}$ and $P^{k+1}_{[B]} = P^k_{[B]}$.

A distance definition must be chosen to define the "nearest" reference vector $R_{\{1,2\}}$ from each profile. Several distance are tested in the following subsection. The performance of the classification using several distance is studied in the following subsection.

2.2.2 Performance indexes

To compare the influence of the distances used for the echo classification, we define three scalar quantity R, V and F that qualify the resulting classification. R is the average of the ratio of the distance between each profile to the two references classes. This metric has the advantage to be independent of the distance scale. However, an infinite result is found if one echo perfectly match the centroid reference. This case is never appeared during this study.

$$R = \frac{1}{N}\sum_{i=1}^{N} \frac{\mathbf{max}\big(\Delta(R_1,P_i),\Delta(R_2,P_i)\big)}{\mathbf{min}\big(\Delta(R_1,P_i),\Delta(R_2,P_i)\big)}$$

The bigger R, the better the classification. The distance function $\Delta(.,.)$ that are used are presented in section 2.2.3 and 2.2.4.

The quantity V is defined as follow : Let Γ_i^W be the bins intra class covariance matrix of the class i. V is defined as the average of $Trace(\Gamma_i^W)$ over all classes.

$$V = \frac{1}{NbrClasses} * \sum_{i=1}^{NbrClasses} Trace(\Gamma_i^W)$$

The lower V , the less diffused are the data of one class, so the better is the classification. A graphic panel is used to present the result. In this panel, (example are figure 11, 12 and 15), the R value is computed using the same data that are presented on the lower left graph. R gives a qualification of the separation of the two clusters of points. The variances of the bins of the two classes found are presented on the two lower right histograms of the panel. V is the average of the two sum from the two variance histogram. See table 1 for panel contents of figure 11, 12 and 15.

The scalar F is computed as follow : Let Γ_i^B be the variance interclass. $diag(\Gamma_i^W)$ and $diag(\Gamma_i^B)$ are two column vectors containing the diagonal values of the intraclass and interclass variance respectively. $diag(\Gamma_i^B)^{-1}$ is defined as a column vector containing the inverse of all its scalar entries.

$$F = diag(\Gamma_i^W)' * (diag(\Gamma_i^B)^{-1})$$

F is the entity that reflects most the separability of the different classes. It has the advantage to be independent from the distance function used. It allow to compare the classification result using several distance functions. This last quantity is known as the Fisher Criterion. See [3] pages 93 for more details.

2.2.3 Comparing echo energy

The energy of the returned signal can be different, depending on the signal's energy that is absorbed by the material, on the directions reflection and on diffraction phenomena. A simple method to segment sonar profiles uses a distance that simply compare the received energy information.

$$D_{E_1} = \sum_i p_i^2 - \sum_i q_i^2 \tag{2.1}$$

where p_i and q_i are the bins of two received profiles. It allows distinction between most of the materials encountered during the experimentations available for this study. However, two different materials may return the same energy but still have a different temporal energy shape. The classification of 1000 profiles $P_{[A,B]}^{[U]}$ is presented on figure 10. The x axis correspond to the profile index and the y axis is the interval $[U]$ composed of 50 bins centered on 25. On the left part of the graph is presented the profiles classified in the first type $[A]$, and on the right part are presented the profiles classified in the second type $[B]$. On the right part of the graph, we can see that some peaked profile and some more diffused profile are mixed. Theses two types of profile correspond to two different kind of seafloor. Several errors were made during the classification process. The following result is obtained:

- $R = 9.9095$
- $V = 0.01886$
- $F = 6.61$

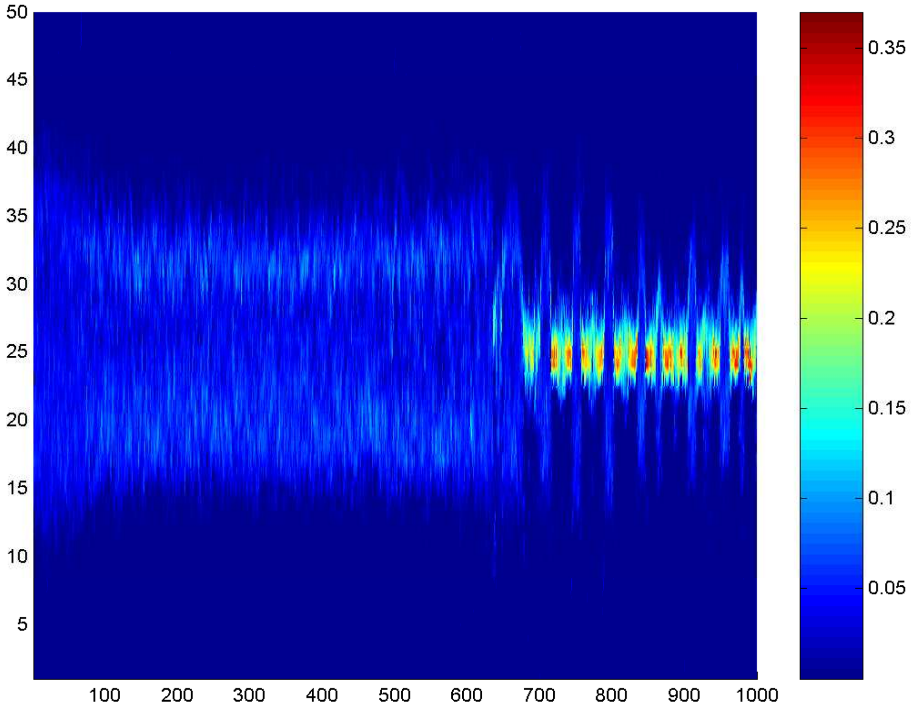

Figure 10: Echo Classification using profile shape

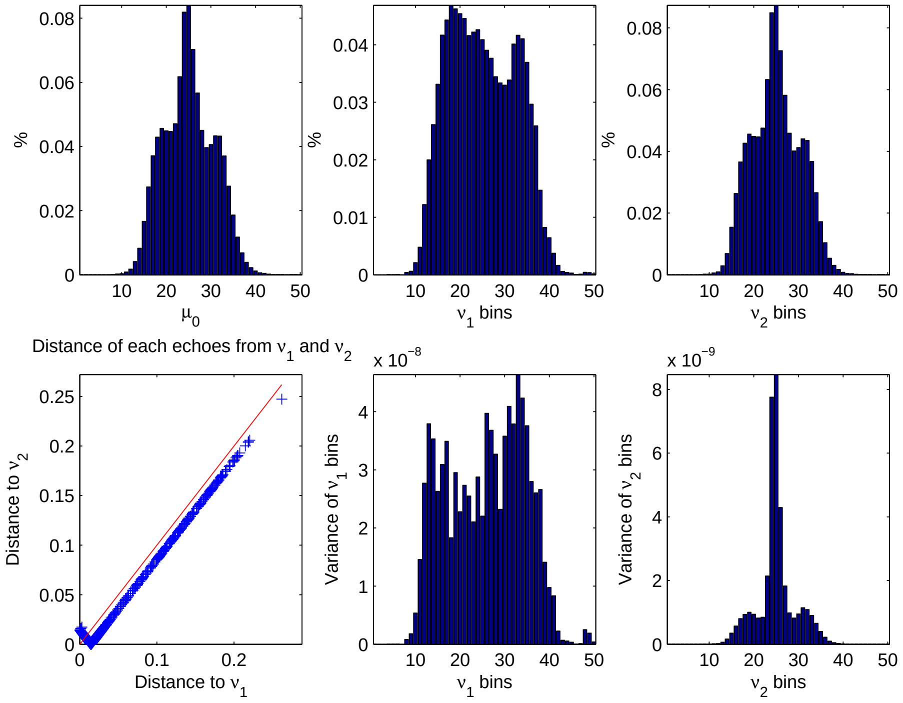

Figure 11: Echo Classification Characteristics using profile Energy

Figure 11 shows the characteristics of the two classes, $[A]$ and $[B]$, that have been found. The upper left plot is the average of all the received profiles. The upper center and right graphs indicate the shape of the two resulting classes corresponding to the centroid of the respective partition. The lower center and right graphs indicate the variance of the energie for each profile bin. The lower left graph represents the distance of each profile to the centroid of the two classes A and B. Table 1 summarize figure organization. It is interesting to note that on figure 11, the data of the lower left graph create a ”U“ with right angle. This is more noticeable when classifying more than 1000 profiles.

Table 1: result panel of figure 11, 12 and 15

Mean of profiles being classified	Mean of profiles belonging to class 1	Mean of profiles belonging to class 2
Distance of each echo to the reference class	Variance of profiles belonging to class 1	Variance of profiles belonging to class 2

Depending on the characteristics of the seafloor, the sonar echo is more or less spread in time. Dense material, such as sand, returns a profile with an accentuated peak. When the echo is reflected by a soft material like algae, the received profile is rather a time-diffused peak.

Considering the echo shape instead of just comparing their energy allows to use more information. We will describe in the next section the different classification methods based on the received energy shape.

2.2.4 Comparing echo shape

Comparison of profile shapes requires that each echo profile be centred. The profiles are considered here to be centred, the centring method being discussed in section 2.3. The distance used is defined as the sum of the Euclidian distance between each bin composing the profiles. Another distance can be defined using the Euclidian distance between the Direct Cosine Transform (DCT) of each echo profile.

$$D = \sum_i (p_i - q_i)^2 \tag{2.2}$$

$$D = \sum_i abs(p_i - q_i) \tag{2.3}$$

$$D = \sum_i abs(DCT_i(p) - DCT_i(q)) \tag{2.4}$$

Other distance definitions have been tried, such as :

$$\sum_i \frac{abs(p_i - q_i}{p_i + q_i}$$

But the distance from equation (2.2) is the only one that separate the two classes creating an “L” on figure 12. The “L” shows that each profile belong without ambiguity to one class because its distance to its reference class very low and its distance to the other reference class is high. Comparatively to figure 11, we notice the lower variance of the profile bins, the peaked profile is sharpen and the distance to each profile type is more contrasted. We conclude that the classification using the profile

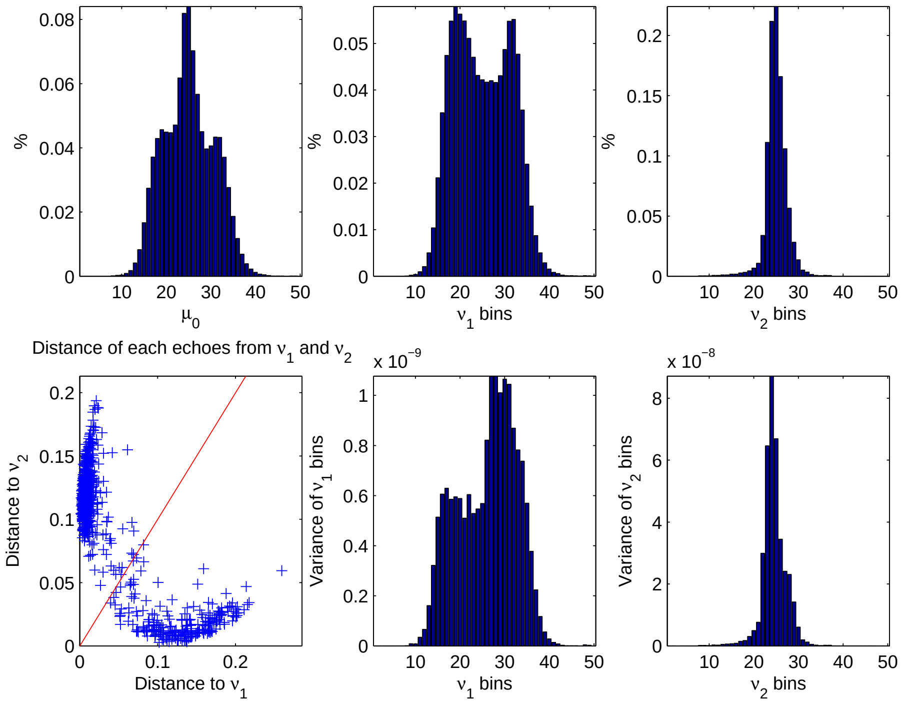

Figure 12: Echo Classification Characteristics using profile shape

shape is better than the one using only the profile energy. On this figure, the distance from equation (2.2) is used. Figure 13 shows the classification result using equation (2.2). The results obtained using

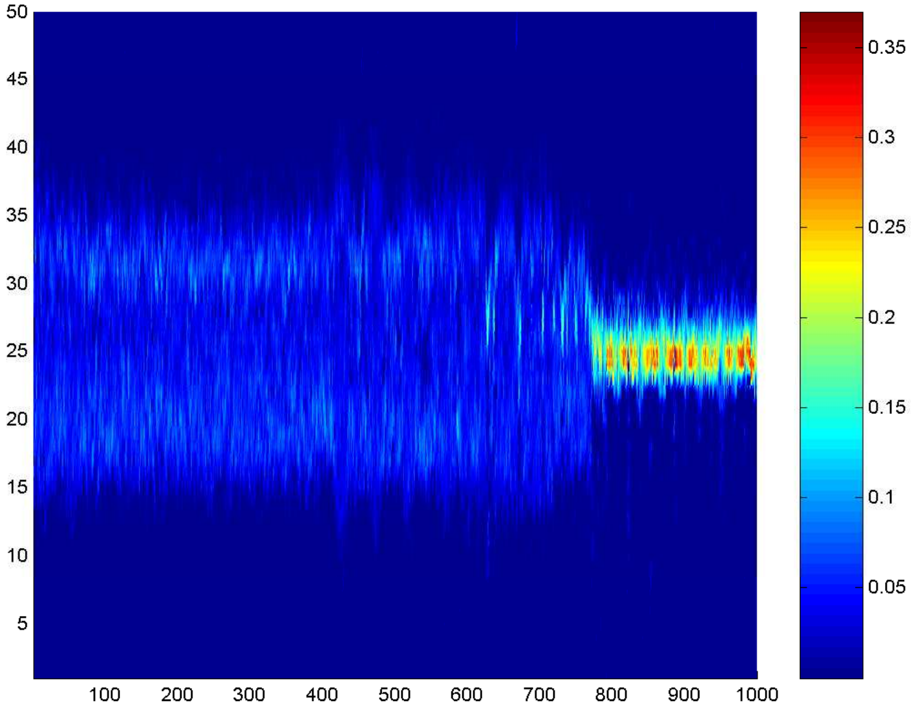

Figure 13: Echo Classification using profile shape

equations (2.1) to (2.4) are presented in table 2. We can see that the distance defined by equation (2.2) yields good results for all three performance indexes R, V and F.

using 1000 profiles	R	V	F
Equation (2.1)	10.6699	0.01881	6.6747
Equation (2.2)	15.0166	0.01497	0.2228
Equation (2.3)	3.4588	0.01577	0.2354
Equation (2.4)	2.8296	0.01433	0.2162

using 5000 profiles	R	V	F
Equation (2.1)	82.23	0.01552	0.3233
Equation (2.2)	11.6409	0.01615	0.3132
Equation (2.3)	3.3053	0.01619	0.3207
Equation (2.4)	2.5324	0.01647	0.3212

Table 2: Classification result using equation (2.1) to (2.4) as a distance

Other parameters analyze like Principal Component Analysis (PCA) should be tried. Read [4] for more details about PCA.

2.3 Profiles shape alignment

The distance defined in section 2.2.4 is very sensitive to a small variation in the alignment of the profile. Three methods to center the profile are presented here.

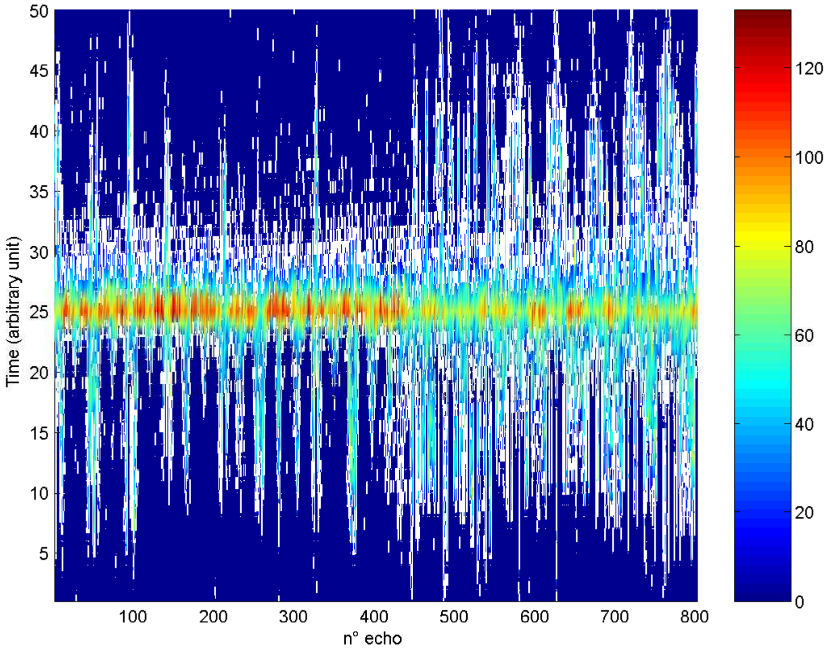

Figure 14: Centred Echos

The first one centers the profiles around their max intensity value. Only fifty bins are kept around the peaks: Twenty-four before the peak and twenty-five after. In figure 14, we can see the result of this method on the profiles presented by figure 9. The figure 15 present 1000 echo shape aligned using the max peak method. We can see on the ν_1 type of theses classified echoes that the variance is bigger on the edge of the echo. This confirms that an alignement error is committed by the max peak alignment technique.

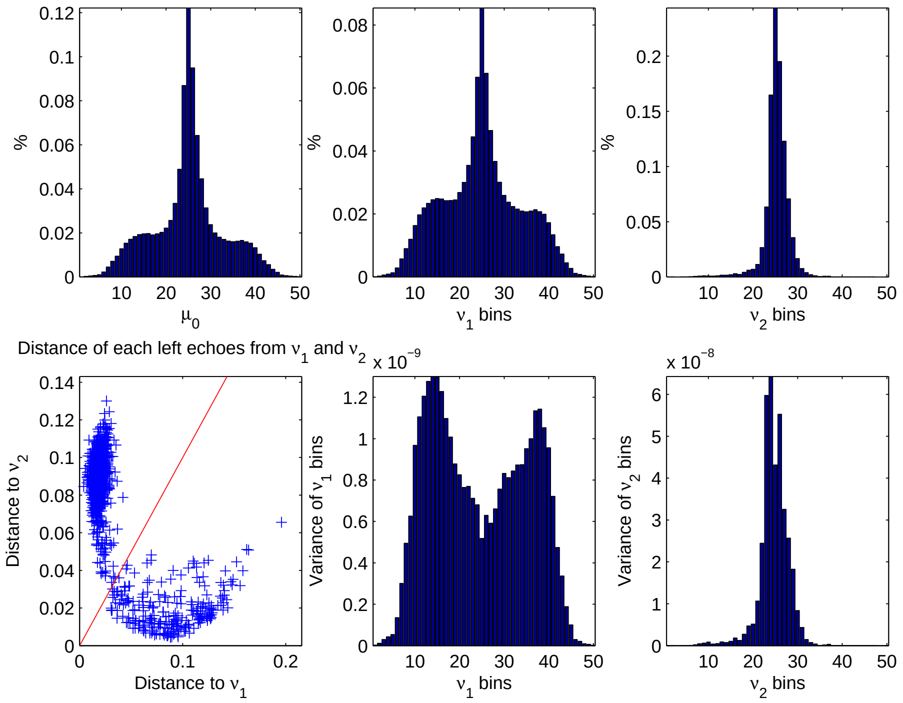

Figure 15: Echo Classification using profile shape and the max peak centering method

An alternative is to compute the gravity center of the shaped signal. Let $P_i^{[U]}$ be a column vector containing the echo profile.

$$idx_{center} = \frac{[1 \ldots N] * P_i^{[U]}}{\frac{(N+1)*N}{2}} \tag{2.5}$$

where each index $[1 \ldots N]$ is weighted by its corresponding bin $P_i^{[U]}$. The sum is divided by the index sum $\frac{(N+1)*N}{2}$.

A third alignment method is a hybrid version of the first two. The echo can be centered on a local max issu of the profile near the index found with equation (2.5). We define k the displacement allowed :

$$idx2_{center} = argmax \left(\max_{i=idx-k}^{idx+k} echo(i) \right) \tag{2.6}$$

Table 3 compare the classification of 1000 sonar profiles by using these three methods:

We see on the Table 3 that the best result is obtained with the gravity center obtained with equation (2.5).

	R	V	F
Max peak	5.2129	0.01953	0.5570
gravity center eq(2.5)	15.0166	0.01497	0.2228
localMax k=3 eq(2.6)	9.02	0.01727	0.3606
localMax k=2 eq(2.6)	9.8864	0.01717	0.3329

Table 3: Classification result with differents centering methods

Additional clipping allows to improve the centering precisio by removing noise from the sonar profiles. The clipping threshold is chosen as a fraction of the profile max value. Values lower than the clipping offset are set to zero. The echo is then centered using the gravity center technique of equation (2.5). The analysis has been done on a set of 5000 sonar profiles. Classification results for several clipping thresholds are shown in Table 4.

Threshold	R	V	F
0 : no threshold	10.9173	0.01793	0.3552
$\frac{1}{8}$	11.5	0.01669	0.3248
$\frac{1}{5}$	11.6409	0.01615	0.3132
$\frac{1}{3}$	11.4787	0.01546	0.3043
$\frac{1}{2}$	10.78	0.01512	0.3110

Table 4: Classification results with different centering methods

The best result is obtained using an offset set to $\frac{1}{3}$ of the max echo value. The offset of $\frac{1}{5}$ and the gravity center centering method were used for the result presented on figure 11 and 12. We can notice that in contrary to the results presented in figure 15, the bin variances are not bigger on the profile border.

2.4 Effect of the sonar angle

The profiles are affected by the sonar steering angle for two reasons. The first one is that the path length of the echo depends on this angle. The longer the path, the more diffused in time is the received signal. The echo shape can also be affected by the reflection angle on the seafloor. Figure 16, shows classification results for three different sample set. the left part is the classification of the profiles taken at large angle on one side ; on the right part is the classification for the profiles taken at a large angle on the opposite side and in the middle is the classification for the profiles corresponding to a small angle i.e. taken under the ROV. Figure 17 presents the different classes found for the three angle. On the left part of the graph is the echo captured on the left side of the ROV. On the middle are the profiles corresponding to the seafloor under the ROV and on the right part are the classified profiles of the right side of the ROV. The colors shows that the profiles taken directly below the ROV contains more energy. However, the general shape of the two classes found for each side of the ROV are similar.

We get the following result:

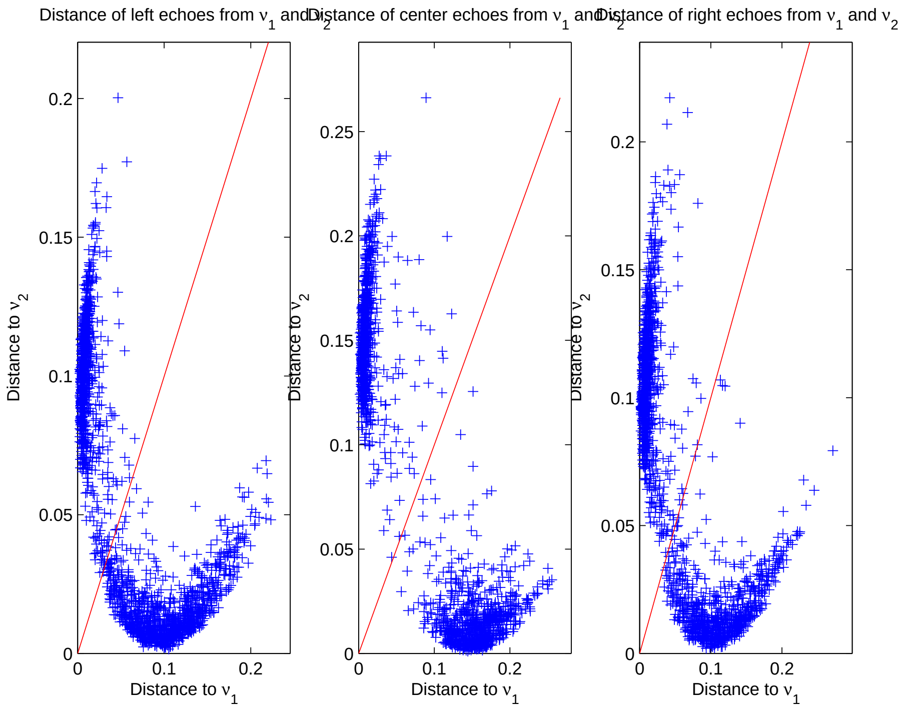

Figure 16: Concatenation of 3 sonar profile shape from 3 distincts sonar steering angle

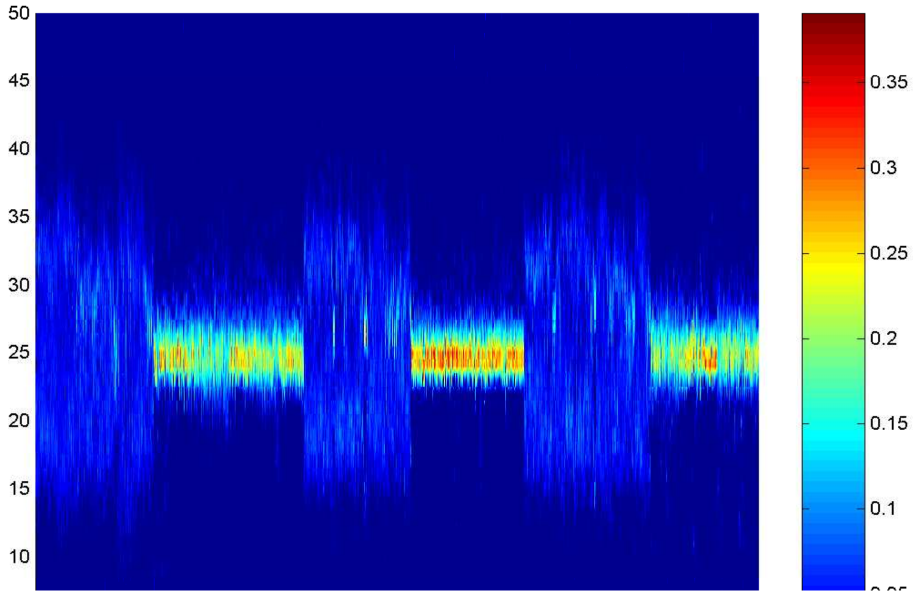

Figure 17: 3 sonar profile shape from 3 distincts sonar steering angle

	R	V	F
Left side	10.6171	0.0143	0.3346
Center	17.1750	0.0153	0.2231
Right side	10.5537	0.0162	0.3402

The ratio R is better for the region under the ROV. The variance is small for the three regions. The echo types classification found is similar for all the three regions. We conclude that we do not need to use the angle information to classify the profiles.

2.5 Simulating echo sonar

To fuse the data from the camera and from the sonar, we create a simulation programme. This simulation has to simulate the ROV comportment and the perception it has of the environment. The camera and the sonar data are simulated. In this section, we describe how the simulated sonar profile are generated.

From the classification results, we extract parameters to simulate a sonar profiles. Each class is treated separately. To generate a profile with characteristics similar of the profiles contained in one class, the mean of each bin of the profiles belonging to this class is computed. The mean of the first bins of each echo is computed. The operation is repeated for the $m = 50$ bins of the centred echo. The variance of each bin is also computed in the same way. Then, the correlation between bins of the same echo must also be taken into account. If we do not use the correlation information betweens bins of the same echo, the randomly generated echo will have the same statistical characteristic of order 2 for each bin but will not look like a real profile due to the absence of the correlation between it bins.

A Singular Value Decomposition (SVD) is used in order to keep the three following characteristics :

- Mean of each bin
- Variance of each bin over several profile realizations
- Co-Variance of bins of the same profile

Let T_1^{m*n} be a matrix whose columns are the profile classified as type 1. The SVD decomposition used gives :

$$T_1^{m*n} = U^{m,m} * S^{m*n} * V^{n*n}$$

This decomposition can be explained in geometrical term: Let T be a transformation matrix. The SVD decompose this transformation into a rotation matrix (V) in the departure space, a diagonal scaling matrix (S) and a rotation matrix (U) in the arrival space. The diagonal matrix (S) contains the singular values, That we consider to be stored on decreasing order. U is an orthogonal basis of the arrival space. The first column of U generates most of the profile, as it is associated to the strongest singular value of T_1^{m*n}. More information about geometric linear algebra can be found in [5].

We define the matrix A^{m*n} as :

$$A^{m*n} = S^{m*n} * V^{n*n}$$

The T^{m*n} matrix is considered as gathering several realizations of a random column vector. Its mean and unbiased covariance matrix are computed as:

$$CoVar(A)^{m,m} = \frac{1}{n-1} * \left(\sum i = 1^n (A^i - \hat{A})^H * (A^i - \hat{A})\right)$$

Let ϵ^m be a random gaussian vector of variance $\sigma^2 = 1$ centred on 0. A vector with the same characteristics as A is created by using:

$$A_{created} = \sqrt{CoVar(A)^{m,m}} * \epsilon^m + \hat{A}^m$$

The profiles are then generated using:

$$Echo_{created} = U * \left(\sqrt{CoVar(A)^{m,m}} * \epsilon^m + \hat{A}^m\right)$$

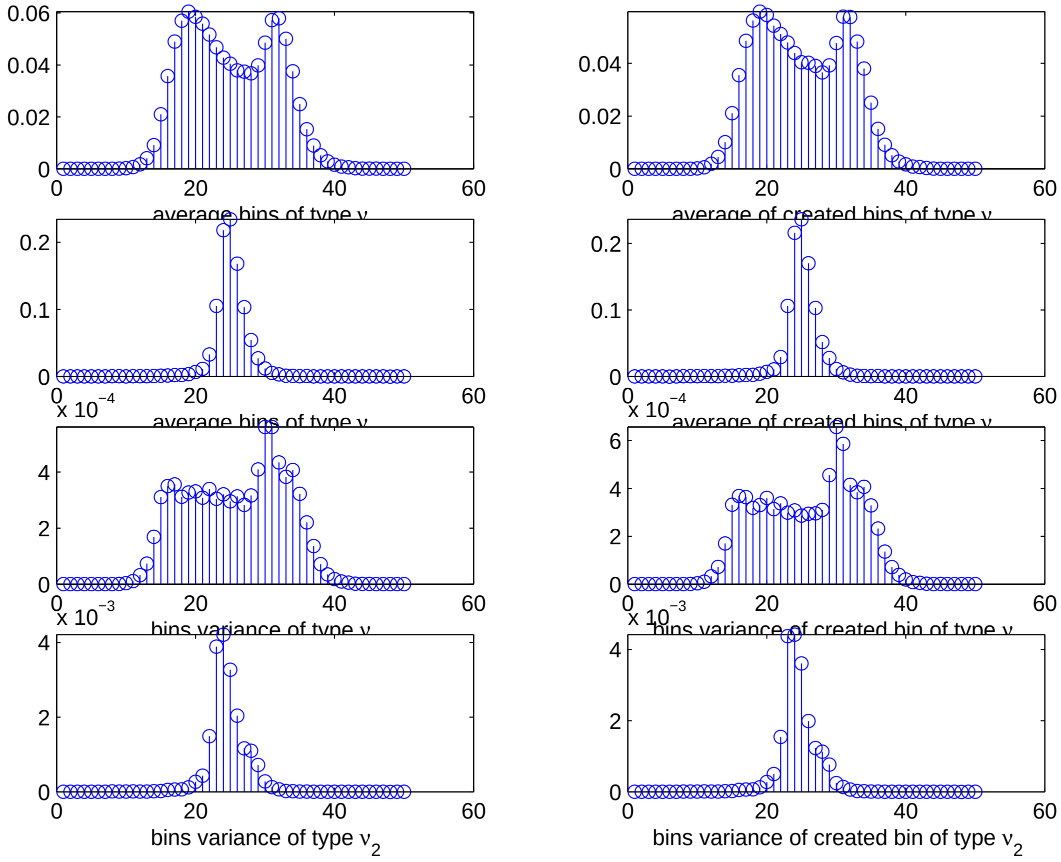

Figure 18: Characteristics of Original and recreated echoes

Typical results characteristics can be see on figure 18. The left part of the figure panel shows the mean and variance of source profiles. The right part shows the mean and variance of the created profiles. We can notice on theses figures that the variance of the bins profiles is not more important when the variation of the profile shape is important. This is a proof that the alignment method described in section 2.3 is relevant. The observation of several profiles generated on figure 19 show that they look like the real one. On this figure, the first and third column show some original profiles. The second and fourth column are generated profiles.

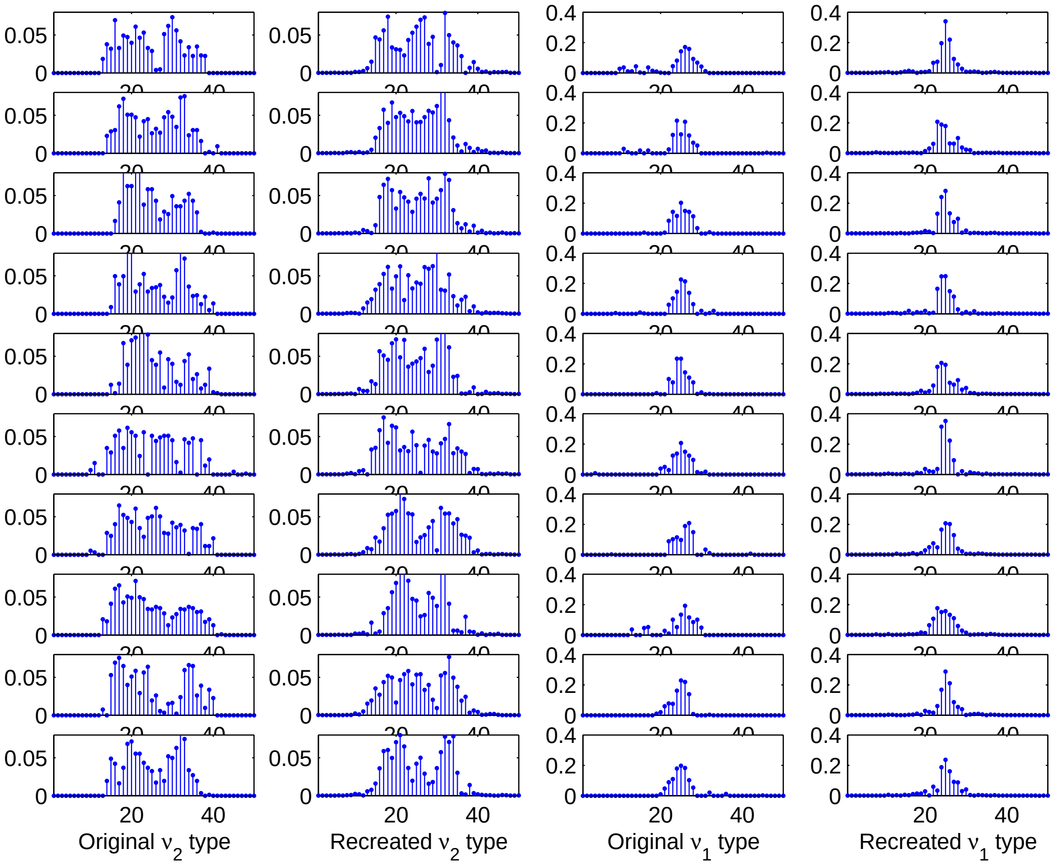

Figure 19: Original and recreated profiles

3 Geometrical matching Sonar - Camera

The sonar and the camera provide complementary information about the environment. Since the data acquired by each sensor is referred to its own coordinate system (the sensors are self-centric), in order to fuse the information generated by them, it is necessary to know the relative position of their coordinate frames. More precisely, we must be able to determine the image-coordinates $P_c(u, v)$ corresponding to the location of any given sonar impact $P_s(\rho, \alpha, \gamma)$. In this section we describe a method to recover the $P_c(u, v)$ coordinate corresponding to the $P_s(\rho, \alpha, \gamma)$ impacts coordinate. In the first subsection 3.1, we defined the relation between P_c and P_s. In theses relation appears the geometrical parameters describing the relative sonar and camera position. Once theses geometrical parameters solved, it is possible to compute any P_c corresponding to the P_s coordinate. In section 3.2, we describe the method used to obtained several P_c coordinate corresponding to their P_s coordinate, without using any parameters, allowing to solve the problem previously described in section 3.1.

3.1 Problem description

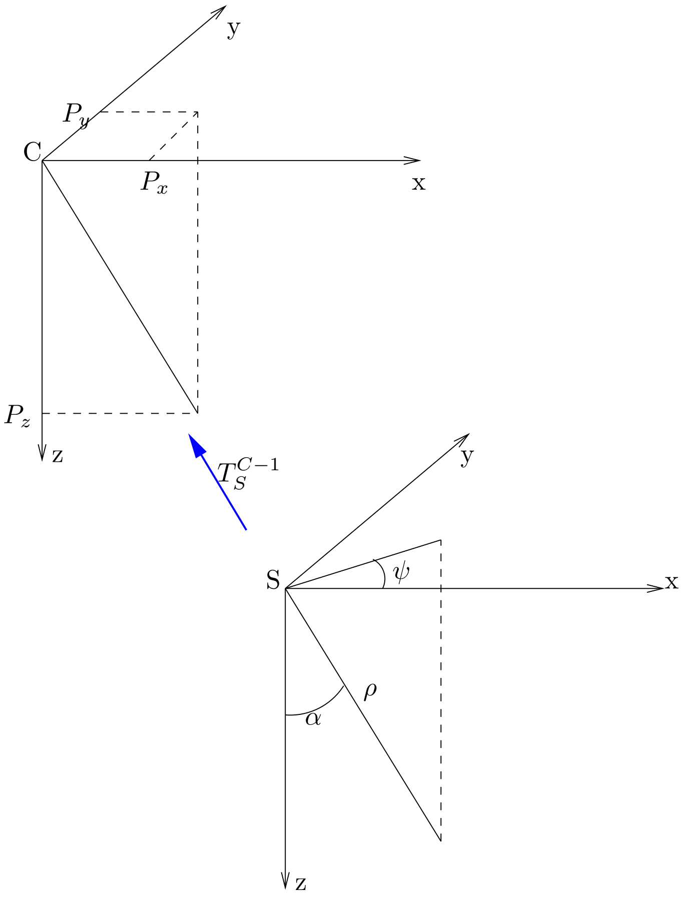

Figure 20: S and C coordinate system

Let us define the following two coordinate systems:

- S : The Sonar coordinate system,

- C : The Camera coordinate system,

and let T_S^C be the transformation matrix applying points defined in the sonar coordinate system (S) into the camera coordinate system (C). The operator T_S^C must compensate the translation relating the origin of the two coordinate systesm, and an eventual 3D rotation ROT due to misalignment of their axis: refer to figure 20

$$ROT = R_z(\psi) * R_y(\theta) * R_x(\phi)$$

where $R_x(\theta)$ denotes the operator of rotation of angle θ around axis x:

$$ROT = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

which finally leads to the following generic expression for the rotation operator:

$$ROT = \begin{bmatrix} \cos(\psi)\cos(\theta) & -\sin(\psi)\cos(\phi) + \cos(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & -\cos(\psi)\sin(\phi) + \sin(\psi)\sin(\theta)\cos(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}$$

In the expressions above ψ, θ and ϕ are the yaw, pitch and roll angles, respectively. The rigid motion T_S^C is also dependent on the translation vector T between the origin of the two coordinate frames:

$$T = \begin{bmatrix} Tx \\ Ty \\ Tz \end{bmatrix}$$

The mapping T_S^C between the two coordinate systems is simply written then as:

$$T_S^C = \begin{pmatrix} \begin{bmatrix} \\ ROT^{3*3} \\ \\ \end{bmatrix} & \begin{bmatrix} Tx \\ Ty \\ Tz \end{bmatrix} \\ 0 \quad 0 \quad 0 & 1 \end{pmatrix}$$

Denote by P_s the coordinates, in the sonar coordinate frame S, of a sonar impact point P. In spherical coordinates, we have :

$$P_s^{spherical} = \begin{bmatrix} \rho \\ \alpha \\ \Gamma \end{bmatrix},$$

where ρ is the distance from the sonar head to the point of impact; α is the sonar steering angle, and γ is the the sonar yaw angle. Refer to figure 20

$$P_s = \begin{bmatrix} \rho \sin(\alpha)\cos(\Gamma) \\ \rho \sin(\alpha)\sin(\Gamma) \\ \rho \cos(\alpha) \\ 1 \end{bmatrix}$$

The coordinates of P in the frame C, P_c, are obtained by applying the rigid motion operator to P_s :

$$P_c = T_S^C \ . \ P_s$$

We can finally obtain the image coordinates of point P, (u, v), by projecting P_c in the image plane, using the projection matrix A:

$$A \quad = \quad \begin{bmatrix} f & 0 & c_x & 0 \\ 0 & f & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

where f is the focal distance of the camera and (c_x, c_y) is the camera optic center.

We can finally write the complete transformation from sonar coordinates P_s to image coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \\ 0 \end{bmatrix} \sim A * \underbrace{T_S^C * P_s}_{C \text{ 3D coordinate}} \tag{3.1}$$

where symbol $\sim$ means "is proportional to":

$$\begin{bmatrix} u \\ v \\ 1 \\ 0 \end{bmatrix} = \lambda * A * T_S^C * P_s \tag{3.2}$$

and the proportionality constant is given by λ which include the $\frac{1}{z}$ magnifying factor :

$$\lambda = -\left(-\cos(\theta)\sin(\phi)\,\rho\,\sin(\alpha)\sin(\Gamma) - \cos(\theta)\cos(\phi)\,\rho\,\cos(\alpha) - T_z + \sin(\theta)\,\rho\,\sin(\alpha)\cos(\Gamma)\right)^{-1}.$$

The previous expressions show that the mapping T_S^C depends on the 9 parameters defining the rotation operator, the translation and the intrinsic camera parameters. We supposed the camera parameters already identified : f, c_x and c_y are known.

Expanding the definitions of the operators in the previous expressions, and retaining only the lines corresponding to the image coordinates (u, v), we obtain:

$$\begin{aligned} u \;=\; & -\lambda\ (-\rho\,\sin(\alpha)\cos(\Gamma)\,f\ \cos(\psi)\cos(\theta) + \rho\,\sin(\alpha)\cos(\Gamma)\,c_x\ \sin(\theta) \\ & +\rho\,\sin(\alpha)\sin(\Gamma)\,f\ \sin(\psi)\cos(\phi) - \rho\,\sin(\alpha)\sin(\Gamma)\,f\ \cos(\psi)\sin(\theta)\sin(\phi) \\ & -\rho\,\sin(\alpha)\sin(\Gamma)\,c_x\ \cos(\theta)\sin(\phi) - \rho\,\cos(\alpha)\,f\ \sin(\psi)\sin(\phi) \\ & -\rho\,\cos(\alpha)\,f\ \cos(\psi)\sin(\theta)\cos(\phi) - \rho\,\cos(\alpha)\,c_x\ \cos(\theta)\cos(\phi) - f\ Tx - c_x\ T_z\ ) \end{aligned}$$

$$\begin{aligned} v \;=\; & \lambda\ (\rho\,\sin(\alpha)\cos(\Gamma)\,f\ \sin(\psi)\cos(\theta) - \rho\,\sin(\alpha)\cos(\Gamma)\,c_y\ \sin(\theta) \\ & +\rho\,\sin(\alpha)\sin(\Gamma)\,f\ \cos(\psi)\cos(\phi) + \rho\,\sin(\alpha)\sin(\Gamma)\,f\ \sin(\psi)\sin(\theta)\sin(\phi) \\ & +\rho\,\sin(\alpha)\sin(\Gamma)\,c_y\ \cos(\theta)\sin(\phi) - \rho\,\cos(\alpha)\,f\ \cos(\psi)\sin(\phi) \\ & +\rho\,\cos(\alpha)\,f\ \sin(\psi)\sin(\theta)\cos(\phi) + \rho\,\cos(\alpha)\,c_y\ \cos(\theta)\cos(\phi) + f\ T_y + c_y\ T_z\ ) \end{aligned}$$

(u, v) are obtained using the method described in section 3.2. For each correspondence between the sonar coordinates $P_s = (\rho, \alpha, \gamma)$ of an impact point and its image coordinates (u, v): $P_s \leftrightarrow (u, v)$ we can write the two equations above. We see thus that to determine the six unknown parameters (the rotation angles (ψ, ϕ, θ) and the components of the translation vector, (T_x, T_y, T_z)) we need at least three pairs of sonar and image coordinate points, to be able to establish six equations.

These three points are not arbitrary, and must satisfy the following constraint:

- The three points must be not aligned in the sonar axis system

3.2 positioning sonar impacts on image camera

Let Θ be the vector of unknown parameters relating the sonar and image frames:

$$\Theta = [\psi\ \phi\ \theta\ T_x\ T_y\ T_z]\,,$$

Equation (3.2) allows us to compute the (u, v) image coordinates of any sonar impact $P_s = [\rho, \alpha, \gamma]$:

$$(u, v) = T_S^C\ (P_s, \Theta)$$

Our goal in the sonar-image callibration problem treated here is to learn the transformation $T_S^C(\dot{,}\Theta)$, so that we can relate the information acquired by the two sensors.

One way of solving this problem is to explicitly estimate all six components of Θ and use our knowledge of eq. (3.2). This requires, as we said before, the identification of the image coordinates for at least 3 distinct and non-collinear sonar impact points. One possible approach to this problem is for instance to acquire both sonar and video during a robot path such that a curvilinear boundary between two distinct materials is observed by both sensors, and establish the correspondence between noticeable points in the boundary "images" as perceived by the sonar and the video camera (for instance, points of largest local curvature or inflection points). This would, in general, enable estimation of the entire parameter Θ, and thus enable us from then on to fuse the data acquired by both sensors in any arbitrary configuration (altitude from bottom, scanning angle, distance, orientation).

The approach described in the previous paragraph is very sensitive to detailed knowledge of the robot's path during boundary observation, and is also noise sensitive with respect to sonar noise (errors in the determination of the distances ρ) and to errors in the identification of the image coordinate of the sonar impacts. We use a distinct approach here, and instead of trying to establish the correspondence between sonar impacts and image coordinates using macroscopic contour "images" perceived by both sensors, we present a method that enables the determination of the coordinates (u, v) for three distinct points in the sonar configuration space (ρ, α, γ) that does not rely in the reconstruction of the robot's path during signal acquisition, and that can be performed in real environments.

We assume the following are satisfied:

- The sea bottom observed by the sonar is locally flat;
- There are two distinct regions in the observed sea-bottom, C_1 and C_2, separated by a contour $\mathcal{C}$, which can be distinguished both visually and in the sonar data, i.e., we are able to find the contour $\mathcal{C}$ in each video frame, and we can identify the region where which each sonar profile was reflected;
- The robot's altitude above the sea floor h, is held constant during the entire experiment;
- The sonar steering angle fixed (here, to $\alpha = 0$)
- the video images and sonar data are perfectly synchronized, i.e., we assume that an image is acquired at the moment the sonar profile is received.

Let $M(t_0)$ be a zero matrix filled with the same dimension as the video images, and $(\mathbf{1})$ $(\mathbf{2})$ be a matrix of same size with all entries equal to respectively (1) (2). Denote by $p(t_n)$ and $I(t_n)$, the sonar profiles and images acquired at distinct instants $t_n, n = 1, 2, \ldots$. Let $\hat{I}(t_n)$ be the segmented region, that indicates the labels of the regions C_1 and C_2 in $I(t_n)$ (each pixel in $\hat{I}(t_n)$ takes only the values "1" or "2"), and $\hat{p}(t_n) \in \{1, 2\}$ be the classification of the sonar profile $p(t_n)$.

At each sampling instant t_n, the matrix M is updated according to the following rule:

$$M(t_n) = M(t_{n-1}) + \Delta_I \left((\hat{p}(t_n) - 1) \left[\hat{I}(t_n) - \mathbf{1} \right] + (2 - \hat{p}(t_n)) \left[\mathbf{2} - \hat{\mathbf{I}}(\mathbf{t_n}) \right] \right)$$

where Δ_I is a small increment. According to the previous equation, the matrix M is incremented at each sampling instant for the regions where there is agreement between the sonar and image classifications. Assuming that the errors affecting the determination of the contour are unbiased, and that its location in the image plane varies significantly during observation (the robot must for instance oscillate in one side and the other of the contour), after several update steps the matrix M will exhibit a sharp peak in the location corresponding to the sonar configuration of the experiment. Figure 21 illustrates the principle of our method. The three diagrams are respectively the three first sonar and camera synchronized capture. At the first capture, the impact sonar $E_2 : \hat{p}(t_1) = 2$ takes place in the region 2. Δ_I is added to the corresponding image classification region that appearing in white on the diagram. In the second and third capture, the sonar impact E_1 takes place in the region $1 : \hat{p}(t_1) = 1$. Δ_I is added to the corresponding image classification region. The highest values of the matrix M appears in white ,in the diagrams, and correspond to the region where the probability for the sonar impact to be inside is the highest.

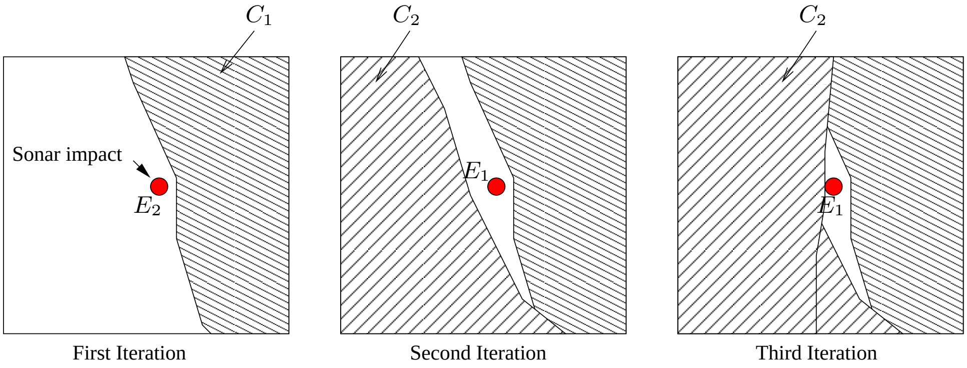

Figure 21: Determining the (u, v) coordinate of one impact sonar on the video image

On figure 22 we present the simulation set-up used to test the method described above. In this simulation, we try to isolate the position of three sonar impact corresponding to three distinct α. The computation are the same since for each distinct α, a distinct M_α is created. The upper right image shows the simulated environment, where a path of dark material can be seen at the centre. Also shown in this Figure are the successive sonar impacts represented with red crosses. The yellow square represent the successive camera frame on the seafloor. The last camera is presented in the upper right diagram. In the centre line we display in the left the received profile, and in the right the corresponding segmented image. The three diagrams in the bottom show the values of matrix M matrix for three distinct scanning angles α. In these three plots, corresponding for three distincts values of α, the true impact sonar place is represented with a red cross.

Using the method described above, we can determine the (u, v) coordinates for several steering angles α_i and for several altitudes h_i allowing to solve the equations of the previous section to estimate the parameters Θ.

Figure 23 presents the final result, after about 1500 steps, obtained for three distincts α.

These three 3D points are situated on a line. Theses points are collinear in the sonar coordinate system. In the camera 3D coordinate, theses points will also be collinear. Using theses three points to determine the $T_S^C(\Theta)$ matrix parameters do not allow to recover the rotational parameter around this axis. As we said in section 3.1, it is necessary to sample points which are not collinear to be able to recover the six parameters of $T_S^C(\Theta)$

A drawback of this method is the limit on the resolution attainable on the identification of the image coordinates (u, v), since their determination is based on the classified images. However, by recovering the parameters of the T_S^C matrix using several corresponding points, it is possible to increase this resolution. Let $\{P_S^i\}$ be a set of points in the sonar frame and $\{P_C^i\}$ be the corresponding points in the camera frame. Knowledge of the geometric place of the impact sonar add an information. If the distances between the $\{P_S^i\}$ points are not a multiple of the dimension of one image grid image, the precision obtained while resolving equation 3.2 is higher.

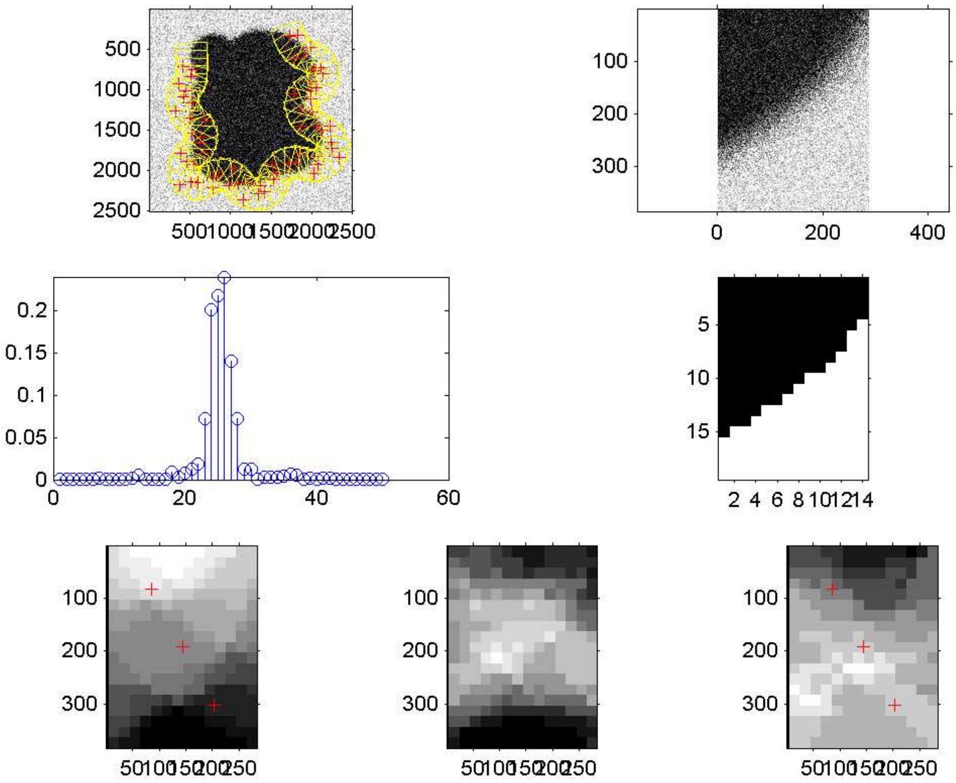

Figure 22: Determination of sonar impact position in video image

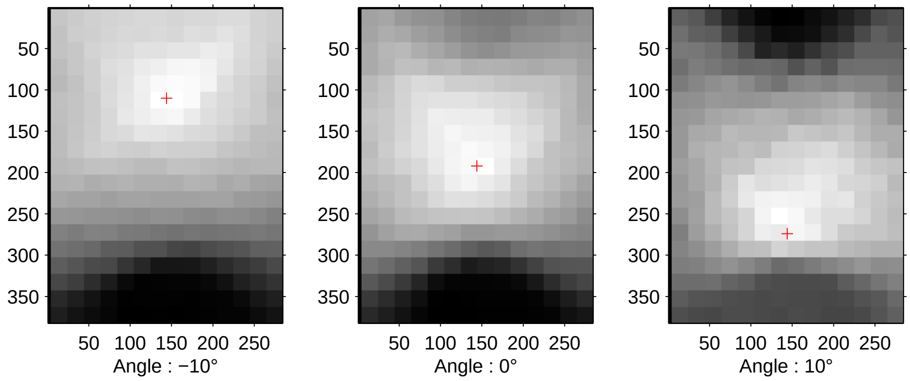

Figure 23: Determination of sonar impact position in video image

A Definition & Equations

Let $\mu(a_i) = P(x = a_i)$ designe the Probability Law (PL) of the random variable x taking values into $A^K = \{a_0, a_1, ..., a_{K-1}\}$.

Definition 1. Entropy

$$\begin{aligned} H(\nu) &= \sum_{a_i} \mu(a_i) \log \frac{1}{\mu(a_i)} \\ H(\nu) &= E\left[\mu \log \frac{1}{\mu}\right] \end{aligned} \tag{A.1}$$

The entropy of a random variable is a function of its PL. It will be denoted by the notation in (A.1)b

Let ν_1 and ν_2 be the PL of two random variables. The Kullback-Leibler divergence, also called relative entropy, is defined as follow:

Definition 2. Kullback-Leibler divergence

$$D(\nu_1|\nu_2) = E\left[\nu_1 \log \frac{\nu_1}{\nu_2}\right]$$

Remark 1. This function is not symmetric :

$$D(\nu_1|\nu_2) \neq D(\nu_2|\nu_1)$$

Let $z^N = \{z_0, ..., z_{N-1}\}$ be a sequence of N Independent and Identically Distributed variable (iid) realization of the random variable Z. $z^N \subset \left(A^K\right)^N$

$$\begin{aligned} P(Z^N = z^N) &= \prod_{n=0}^{N-1} P(Z = z_n) \\ &= \prod_n \mu(z_n) \\ &= \prod_{i=0}^{k-1} \mu(z_k)^{k_i} \\ &= \exp\left\{\log \prod_{i=0}^{k-1} \mu(z_k)^{k_i}\right\} \\ &= \exp\left\{N \sum_i \frac{k_i}{N}\left(\log \frac{\mu(z_k)}{k_i/N} + \log \frac{k_i}{N}\right)\right\} \\ &= \exp\Big\{-N\big(H(\nu) + D(\nu|\mu)\big)\Big\} \end{aligned} \tag{A.2}$$

k_i is the number of occurence of z_k in Z^N.

Assume that we have two sets of data $x^{(1)}$ and $x^{(2)}$ of respective size N_1 and N_2. We want to know whether the two sets of data come from one homogeneous image region or if they come from an image containing two distinct regions. Assume H_0 is the hypothesis that only one type is present in the observed data and H_1 the hypothesis that the observation is a mixture of two types. μ_1 and μ_2 are the PL of the two data set. We don't have the real PL of the data so μ_1 and μ_2 are estimated from the data. We define $\hat{\mu}$ as the weighted average of $\hat{\mu}_1$ and $\hat{\mu}_2$.

$$\hat{\mu} = \frac{N_1 * \hat{\mu}_1 + N_2 * \hat{\mu}_2}{N_1 + N_2}$$

The two set of data are independent, we have:

$$P(x^{(1)}, x^{(2)}|H_0) \;=\; P(x^{(1)}|H_0)P(x^{(2)}|H_0) \tag{A.3}$$

from (A.2) and (A.3) we get :

$$\begin{aligned} P(x^{(1)}, x^{(2)}|H_0) \;&=\; \exp\big\{ - N_1\,(H(\nu_1) + D(\nu_1|\hat{\mu}))\,\big\}\exp\big\{ - N_2\,(H(\nu_2) + D(\nu_2|\hat{\mu}))\,\big\} \\ \log\big(P(x^{(1)}, x^{(2)}|H_0)\big) \;&=\; -N_1\big(H(\nu_1) + D(\nu_1|\hat{\mu})\big) - N_2\big(H(\nu_2) + D(\nu_2|\hat{\mu})\big) \end{aligned} \tag{A.4}$$

for hypothesis H_1, we get :

$$\log\big(P(x^{(1)}, x^{(2)}|H_1)\big) \;=\; -N_1\big(H(\nu_1) + \underbrace{D(\nu_1|\hat{\mu}_1)}_{=0}\big) - N_2\big(H(\nu_2) + \underbrace{D(\nu_2|\hat{\mu}_2)}_{=0}\big) \tag{A.5}$$

Where ν_1 and ν_2 are the estimation of the PL $\hat{\mu}_1$ and $\hat{\mu}_2$ computed from the data $x^{(1)}$ and $x^{(2)}$. The μ_n, $n = 1, 2$, laws are also estimated from data. It explains why $D(\nu_n|\hat{\mu}_n) = 0$.

The decision between the hypothesis H_0 and H_1 can be written

$$P(x^{(1)}, x^{(2)}|H_0) \lesseqqgtr^{H_1}_{H_0} P(x^{(1)}, x^{(2)}|H_1)$$

from (A.4) and (A.5), we obtain

$$0 \lesseqqgtr^{H_1}_{H_0} N_1\big(D(\nu_1|\hat{\mu})\big) + N_2\big(D(\nu_2|\hat{\mu})\big) \tag{A.6}$$

This test will choose H_1 when the PL function of $x^{(1)}$ and $x^{(2)}$ will be different. In practice, equation (A.5) will always decide hypothesis H_1. This test is not relevant. The MDL is used. In this test, the maximum length of the data description is computed in cases H_1 and H_2 thanks to the information theory (see [1]). The Code length find penalize the previous maximum likelihood test. From the philosopher William of Occams: “The simplest explanation is probably the right explanation”.

Let M be the number of values taken by the quantified sample. M is the number of bins composing each grey level distribution histogram. A grey level distribution can be written as:

$$p = \Big[\underbrace{p_1, ..., p_{M-1}}_{\text{Libre}}, p_M\Big]$$

where $\sum_i M_i = 1$ so $M-1$ data are sufficient to recover the complete histogram.

$$p_i = \frac{n_i}{N}$$

with

$$n_i \in \underbrace{[0, ..., N]}_{N+1 \text{ values}}$$

The total number of different sequences of $M-1$ events taken in an ensemble of cardinality $N+1$ is :

$$(N+1)^{M-1}$$

The code length equal to its entropy is then:

$$L(H_0) \; = \; (M-1)\log(N_1+N_2+1) \tag{A.7}$$

and for hypothesis H_1 :

$$L(H_1) \; = \; (M-1)\left[\log(N_1+1)+\log(N_2+1)\right] \tag{A.8}$$

The MDL test is then:

$$P(x^{(1)}, x^{(2)}|H_0) - L(H_0) \lessgtr_{H_0}^{H_1} P(x^{(1)}, x^{(2)}|H_1) - L(H_1)$$

from equations (A.6), (A.7) and (A.8), we get :

$$\begin{aligned}(M-1)\big(\log(N_1+1)+\log(N_2+1)-log(N_1+N_2+1)\big) \lessgtr_{H_0}^{H_1} \; & N_1 * D(\nu_1|\hat{\mu}) \\ & +N_2 * D(\nu_2|\hat{\mu})\end{aligned} \tag{A.9}$$

see [2] for a geometric explanation of MDL principle.

Using the same set Z^k of data, we suppose that Z^k is composed of two types of sample. Therefore, the estimate ν_k of its PL will be asymptotically a mixture of the PL of both types respectively μ_1 and μ_2. We define the mixture coefficient α according to :

$$\nu_k = \alpha\mu_1 + (1-\alpha)\mu_2 \tag{A.10}$$

To determine α, we use the above calculation:

$$\begin{aligned} D(\nu_k|\mu_2) - D(\nu_k|\mu_1) &= E_{\nu_k}\left[\log\frac{\nu_k}{\mu_2}\right] - E_{\nu_k}\left[\log\frac{\nu_k}{\mu_1}\right] \\ &= E_{\nu_k}\left[\log\frac{\mu_1}{\mu_2}\right] \end{aligned}$$

from (A.10), we get

$$\begin{aligned} D(\nu_k|\mu_2) - D(\nu_k|\mu_1) &= \alpha E_{\mu_1}\left[\log\frac{\mu_1}{\mu_2}\right] + (1-\alpha)E_{\mu_2}\left[\log\frac{\mu_1}{\mu_2}\right] \\ &= \alpha\big\{D(\mu_1|\mu_2) + D(\mu_2|\mu_1)\big\} - D(\mu_2|\mu_1) \end{aligned}$$

finally, we obtain

$$\alpha = \frac{D(\nu_k|\mu_2) - D(\nu_k|\mu_1) + D(\mu_2|\mu_1)}{D(\mu_1|\mu_2) + D(\mu_2|\mu_1)} \tag{A.11}$$

References

[1] Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, vol. 27:pages 379–423 and 623–656, July and October 1948.

[2] Mark H.Hansen and Bin Yu. Model selection and the principle of minimum description length. January 1996.

[3] Naïma AIT OUFROUKH. Perception intelligente à partir de capteurs frustres pour la robotique de service. *Université d'Evry Val d'Essone*, décembre 2002.

[4] Lindsay I Smith. A tutorial on principal components analysis. February 2002.

[5] Centre for Neural Science Eero Simoncelli and Courant Institute of Mathematical Sciences. A geometric review of linear algebra. January 1999.

[6] C. Barat and M.-J. Rendas. Benthic Contour Mapping with a Profiler Sonar. Toulon, France, May 2004. Proc of the International Society of Offshore and Polar Engineers (ISOPE) 2004, Laboratoire I3S, CNRS-UNSA Sophia Antipolis France.

[7] M.-J Rendas A. Tenas and J.-P Folcher. Image segmentation by unsupervised adaptive clustering in the distribution space for auv guidance along sea-bed boundaries using vision. Proc OCEANS'2001, Honolulu, Hawaï, Laboratoire I3S, CNRS-UNSA Sophia Antipolis France, November 2001.