Some Image Tools for Sonar Image Processing
Stéphane Guillaudeux
Triton Elics International
125 Westridge Drive
Watsonville, CA 95076

Abstract — The trend in seafloor imaging towards using multiple sonar sensors with enhanced resolution has resulted in higher volume data sets. As a consequence, users need better, more efficient tools for data processing and interpretation. This paper describes three data processing tools useful for automating and accelerating sonar image formation. All have been validated on real data and all are TEI application products. The applications support automatic object detection, automatic seafloor classification, and sonar patch tests using backscatter information.

I. INTRODUCTION

As the resolution of sonar sensors increases, the volume of acquired data increases. Coincident with increased data volumes is the requirement for efficient, automatic processing of such data sets to aid our interpretation and understanding of survey areas. Image processing tools provide some useful solutions. A basic example is the image mosaic where the geocoded view leads to a better sonar image interpretation.

This paper illustrates the use of processing techniques originating in the image processing field for three different sonar applications. As a commercial product, the goal was not to develop new processing methods but rather to optimize several classical image tools into an efficient, adaptive, and minimally complex software product.

II. OBJECT DETECTION

A. Purpose

When an object is lying on the seafloor, two major pieces of information can be observed on the sonar image: the echo of the object and its shadow. Echoes are high-level signals reflected to the sonar when the sonar's incident beam hits the object. Shadows exist in the area behind the object where no ensonification occurs (i.e., the area from which no energy returns to the sonar). The length of the shadow is a function of the object height above the seabed and the beam-grazing angle.

The purpose of additional processing is to automatically detect echo and shadow regions that are properties of objects. Here, there is no recognition process; we only take into account sonar basic principles to locate targets. Thus targets can be rocks, wrecks, or man-made objects.

Possible applications of such a tool include object searching (e.g., wrecks, containers, mines), pipeline inspection, obstacle avoidance from forward-looking sonars, and AUV navigation aiding through comparison of information extracted from a sidescan sonar with the location of known features in a database [1].

B. Processing Chain

Object detection requires several image processing stages, each of which must be optimized to achieve real-time processing capabilities (Fig. 1). One key component of our optimization efforts was to implement a ping-by-ping treatment of the data. In other words, the processing algorithms do not take an image as input but rather handle each ping individually. A second key optimization point is the detection efficiency in various configurations. We compared several methods to assess the best trade-off between product complexity and adaptivity.

Fig. 1. Object detection processing chain

1. Despeckling: The despeckle step reduces the noise level. Despeckling is a low-pass filter applied either in the spatial or frequency domain.
2. Range Normalization: Range normalization is a signal amplitude correction designed to conserve the across-track signal dynamic. Because of attenuation in the water column, signal amplitudes decrease with time or distance from the sonar. Consequently, a processing step such as segmentation that is tuned on signal level at the beginning of the swath becomes more inefficient for farther ranges. Range normalization improves the efficiency of the process across the entire signal range. The proposed normalization uses the second order statistics of the signal (mean, deviation). Let $M[i]$ and $S[i]$ be the local mean and local deviation of the signal at the range number $i$. Assume two constant values $R_s$ and $R_m$. Then the normalized pixel at range $i$ is given by (1):

$$
\text{normalized pixel} = \frac{S[i]}{R_m} + \frac{M[i]}{R_s}
$$
As a result, the normalized signal tends to follow both the reference mean, $R_m$, and the reference deviation, $R_s$, throughout the ping range.

3. **Segmentation**: Many segmentation techniques exist. Two analyzed for this project included histogram thresholding [2,3] and fuzzy segmentation [4]. Histogram thresholding is good for defining the gray level borders between regions, whereas fuzzy segmentation has proven to be efficient for sonar images.

Segmentation consists of assigning a class type to each pixel. Class types for this application were echo, shadow, or reverberation. Segmentation is one of the most important steps of the processing chain because it results in an important loss of image information. Our tests showed that the main problem was obtaining an efficient segmentation for both large and small shadow regions; this problem was best addressed by the histogram thresholding method.

4. **Object Detection**: The last step for object detection is image description in terms of regions. At the segmentation step, only individual pixel information is available (each pixel has been assigned to a class). This local information must be grouped in some fashion to lead to a more regional description. The grouping is done by a connected component analysis. The purpose of the connected component analysis is to merge the segmentation classification from two successive pings to progressively create and update regions of the image.

Object detection is based on a region association which aims at identifying possible objects as an echo-shadow pair. The association adheres to some basic sonar image rules: *i.e.*, shadow and echo regions are not too small, they are close to each other, shadow is behind echo, shadow and echo are aligned. Those criteria validate some segmented regions as possible objects.

Fig. 2 shows an example of detected targets in the TargetPro™ software program.

III. **SEAFLOOR CLASSIFICATION**

Seafloor classification (SFC) is another automated process based on classical image processing tools. SFC aims at separating different seabed types (*e.g.*, sand, rock, pebble, mud) through texture analysis of mosaicked sonar images. Texture can result from one or several motifs repeated on a regular basis and is characterized by some homogeneity criterion. In other words, texture is a region where some statistical or structural properties are constant or change slowly.

Two steps are generally required for texture analysis: information extraction and classification. Information extraction involves the computation of a characterization measurement (*e.g.*, statistical property) from an image sub-area defined as a *textel*. Classification then interprets those measurements and assigns a class type to the image textel. This operation, repeated on each image textel, leads to an image description in term of regions.

In the application developed by TEI, information extraction is derived from co-occurrence matrices, and classification involves a neural network.

A. **Information Extraction**

In this first step the idea is to extract sub-areas with features relevant to the regions we intend to separate. Those features characterize texture regions. In general, the classification algorithms work better and region identification is more accurate as the number of representative sub-areas selected increases.

The textel identification schemes examined for this project included fractal dimensions, frequency analysis, and co-occurrence matrices. Fractals have received considerable attention in the field of underwater sediment characterization and sonar image analysis because of their ability to synthesize natural landscapes [5]. However, while a fractal dimension can quantify seabed roughness, it suffers from a lack of stability and is irrelevant as a unique feature for texture segmentation. Frequency analysis of type-regions proved to be efficient for sea-bed ripple detection [6] but was less successful for other bottom types.

One major statistical representation for texture is the gray level co-occurrence matrix. Co-occurrence matrices estimate the joint probability density of the gray-level of two pixels separated by a given displacement $D$ [7]. In other words, a
co-occurrence matrix $M$ is a $n \times n$ array ($n$ is the number of image gray levels) where each value $M(i,j)$ represents the probability to find a pixel of level $i$ and a pixel of level $j$ separated by a displacement $D$. Many displacements can be used to generate several matrices.

As a product of those matrices, Haralick proposed to compute several statistics that quantify some visual properties of textures. Those statistics include maximum probability, homogeneity, entropy, contrast, and correlation [8].

B. Classification

After information extraction and defining of textel properties, the classification step associates a region label (sand, rock, etc.) to each textel as described by its set of features. The TEI seabed classification application uses a multilayer perceptron neural network because it is robust and it possesses learning capabilities. Other classification tools examined during this project are summarized in [6].

Two modes characterize a perceptron neural network:

- A learning mode where the neural network tunes its own parameters. A “teacher” supervises this stage, providing the network with the solution to find for a given set of input features.
- Once the network has tuned the identification parameters, it can perform classification tasks. For each input set of features, a class is assigned that represents the texture type.

C. Implementation

As noted above, TEI has developed and tested an automated SFC module. The SFC module uses mosaicked images (DDS_VIF format) and produces an output vector file (DXF format) of the resultant region boundaries.

The information extraction step generates 44 parameters from co-occurrence matrices computed over four directions ($0^\circ$, $45^\circ$, $90^\circ$, $135^\circ$). These multiple matrices enhance the robustness of the textel classifications through changing orientation.

The neural network is a three-layer perceptron: the input layer uses the 44 computed parameters, the hidden layer has 50 nodes, and the output layer size depends on the number of regions requiring classification. The network learning stage is performed by a back-propagation algorithm.

An intuitive GUI exists for assisting the user in selecting representative texture regions and for general SFC management. The network is built by selecting representative regions or samples for each seabed type across a mosaicked survey area. Each selected sample is assigned to a seabed type or texture name (up to eight different textures can be classified). Sample size is user-definable. After all type regions have been selected, the classification parameters are computed. These parameters can be saved in a file or recomputed after adding or deleting textures or samples (e.g., if the classification results are not acceptable after the first pass).

Such classification is a local analysis of textures, and consequently a number of isolated regions may appear in the resultant product. A post-processing stage has been added in order to reduce the number of small regions that may confuse the regional understanding of the survey area. This simplification stage merges small regions with larger neighbor regions according to a minimum region size (as specified by the user). The end product is a DXF file of region boundaries that may be overlain on the mosaic of the survey area.

![Fig. 3. Classification processing time](image1)

Texture classification can be a time-consuming process because it is a local analysis repeated on each textel (i.e., on each sub-image). Additional time is consumed during the post-processing step of merging small regions into larger neighbor regions. Fig. 4 provides some control over the mean processing times required from SFC. These results were obtained with a standard Pentium III 800 MHz CPU. The x-axis of the graph represents the classification resolution and the y-axis represents the processing time required per Mega-octet. Smaller textel sizes obviously require more processing time, but the primary contributor to the increased time lies in the post-processing stage as the greater number of isolated regions require more merging.

![Fig. 4. Automatic seafloor classification](image2)
IV. PATCH TEST USING BACKSCATTER INFORMATION AND HYPERBOLA DETECTION

Patch tests are routinely used to calibrate multibeam sonar systems through the computation of offset parameters (roll, pitch, latency). Usually bathymetry data is used for such tests, and hence the derivation of offset parameters require a distinct seabed feature (e.g., an isolated object or a slope) that can be easily identified under different survey conditions (variations in survey direction or vessel speed).

A. Methodology

Assuming that the sonar moves along the y-axis, and that slant range distance is along the x-axis, let the target be at position \((x_0, y_0)\). Then the hyperbola equation is given by (2):

\[
R^2 = R_0^2 + (y - y_0)^2 \Rightarrow x^2 - (y - y_0)^2 = x_0^2 \tag{2}
\]

Fig. 5 shows the geometry for this equation. The solution to alleviate this non-linear problem involves a pair of points. Consider a point \((x_m, y_m)\) belonging to an hyperbola for which the target position \((x_0, y_0)\) is unknown. The previous equation gives:

\[
x_0^2 + (y_m - y_0)^2 = x_m^2 \tag{3}
\]

This equation represents a circle of radius \(x_m\) centered on point \((0, y_m)\). Consequently, if two points are detected on the hyperbola, the intersection of two circles given by (3) leads to the target position \((x_0, y_0)\).

Thus an automated process requires not only locating possible points belonging to the hyperbola but also finding the best estimate of the position \((x_0, y_0)\). Identification of points possibly belonging to a hyperbola is achieved by an image segmentation that extracts “bright” points referred to as echoes. Identification of the hyperbola top is achieved through the use of a Hough transform, a powerful shape-recognition tool.

B. Implementation

As stated previously, the first step is to identify and extract image pixels (bright points) that belong to the hyperbola. Those points result from high amplitude acoustic reflections of the target (considered as echoes in the sonar image) and are extracted via a segmentation process. The processing procedures outlined in the ‘Object Detection’ section are implemented here. For the present application, only the echo points are retained, with each point described by its ping number and its range distance. The processed image can be noisy, and there is no guarantee that all extracted echoes belong to the hyperbola of interest. Consequently, another processing step is required to improve hyperbola detection.

The second step involves an Hough transform. An Hough transform is an object-recognition tool used to locate shapes described by a parametric equation. It was introduced to detect lines or circles and was shown to be efficient in noisy environments. The hyperbola detection algorithm implemented in the TEI application is derived from an adaptation dedicated to ellipse detection [10].

The Hough transform is done as follows:

1. Randomly choose two echo points in the image.
2. Compute the target location \((x_0, y_0)\) according to (3).
3. Accumulate computed parameters in a histogram.
4. Repeat the process for some preset number of iterations.
V. CONCLUSION

We have shown how classical image processing tools can be used toward automated object detection, seafloor classification, and patch test implementation. The applications developed by TEI are designed to assist processors and interpreters in quickly and easily manipulating and analyzing the vast volumes of sonar imagery resulting from today's surveys. An additional bonus is that fewer subjective decisions are made with these statistically based analytical procedures, thereby improving the consistency of interpretations from survey to survey. Our goal in the near future is to improve the accuracy and reliability of the resulting data products through the use of other information tools such as databases or a GIS to store and display the information.

REFERENCES


Fig. 6. Automatic hyperbola detection using Hough Transform

The maximum value of the resultant histogram delivers the most probable solution. However, one problem is that the output histogram is often difficult to manage. It requires a large memory allocation since it gathers all the possible solutions; in other words, it is a \( n \times m \) array if \( n \) (\( m \)) is the number of possible \( x \) (\( y \)) target locations. When using an histogram, it is necessary to decide which region of the parameter space the histogram will cover. Increasing region size increases the required memory, whereas decreasing region size decreases the accuracy of the result. Consequently, we implemented a tree structure [10] whereby results are organized in a linked list in which each cell (a possible solution) is created if necessary. This efficient structure improves search time and computational cost.

Fig. 6 shows two target locations (green squares) which show that the detection is accurate despite the noisy environment.