3

Sonar Data Simulation

Didier Guériot, Christophe Sintes
Institut Télécom ; Télécom Bretagne ; UMR CNRS 3192 Lab-STICC
Technopôle Brest-Iroise, CS 83818, 29238 Brest Cedex 3
Université européenne de Bretagne
France

1. Introduction

1.1 What is the need for sonar data simulation ?

At-sea daily costs for performing data collection are high and acquiring specific data to either validate hypotheses or try new algorithms is often too expensive to be achieved. Thus, due to operational constraints for underwater data acquisition, simulating realistic sonar data, like images and swath bathymetry profiles, is crucial for designing and tuning detection and classification algorithms according to sensors settings, sea-bottom nature and topography. To sum up, an accurate and realistic simulation tool is a very good asset to enhance system & algorithm performances, thus appearing as a complement to real data collections.

1.2 Requirements for simulating realistic sonar data

The robustness of any performances estimation or prediction algorithms can be greatly improved as soon as such an accurate simulation tool provides a way to introduce within the simulation: a modular and flexible underwater world representation, multiple kinds of sensors, multiple kinds of environments (propagation, . . .) and multiple kinds of acquisition conditions (including sensor trajectories). Indeed, controlling the full data acquisition process through a highly modular sea-world representation, allows to produce many datasets with various dedicated configurations.

However, another axis of modularity is required as different computational engines may be used depending on simulation cases (ray theory, normal mode or parabolic equation), all solving the Helmholtz equation with specific approximations (Burdic, 1984).

Starting with this equation and assuming high frequency signals, the first approximation leads to the eikonal equation that allows acoustic wave to be modeled as rays following a trajectory bent according to sound celerity gradients. The engines that will be described in this chapter derive from this model. Relying on wave guide propagation mechanisms, the second approximations open into modes solutions. Assuming extra constraints (specific stratification), the modes equation may become parabolic (Brekhovskikh & Lysanov, 1982) or using Perkins model, admit trigonometric function based solutions (Perkins & Baer, 1982). Green functions may also be introduced leading to Fourier transforms (Harrison, 1989). However all these methods (based on differential equations) are computationally expensive as numerical approaches (FDTD, TLM techniques) are required to compute their final solutions.

1.3 Chapter objectives

Therefore, responding to the presented needs, we have developed a framework for simulators allowing both underwater scene design and computational simulation engine choices. More precisely, this generic framework provides the reader with a common and simple software system in which various sensors, environments and computational engines can be plugged in.

The first section of this chapter presents the framework architecture pointing at the interactions between all its components. Subsequent sections of the chapter focus on specific computational engines describing all the phenomena to be considered and the problems to be answered in order to produce realistic simulated sonar data. Section 3 details how ray and tube engines collect exhaustively all the backscattered acoustic waves resulting from scene interactions with the transmitting acoustic wave.

Using these two engines and considering acoustic waves and scene interactions as energy based only, Section 4 presents realistic simulated results for several imaging systems (sidescan and front-looking sonars).

However, several phenomena such as speckle effect or diffraction, have a great impact on collected sonar data and they cannot be simulated using energy only: simulating full sonar signals (including the phase component) is absolutely required. Section 5 describes the difficulties such a complete simulation involves, and proposes solutions to model and implement the needed local signal/scene interactions. Finally, two standard phase-based sonar processing (beamforming an interferometry) fed with simulated full sonar signals, successfully produces output exhibiting very good behaviors compared to real measurement.

2. A multi-sensor simulator framework with engine plugins

The main goal of this simulation framework is to propose a common environment for developing modular simulation tools. The keyword "modular" stands for the simulation objects describing the underwater scene but also for all the algorithms developed to deal with acoustic propagation or object interactions (reflection, refraction . . .). Thus, a user can pick among several families of components to tune its simulation environment. Some of these components are relative to static properties of the environment while others are meant to represent dynamic phenomena. Figure 1 offers a synthetic view of the software components of this framework.

2.1 Scene components

The scene components constitute the basic and common blocks for describing the simulation underwater environment; all the processing engines will operate on these blocks. Several families of scene components can be listed:

- *Objects*: All the geometric features of the scene are described through objects either using mathematical definitions (sphere . . .) or lists of facets which are a more flexible way of description. Objects mean real objects like wrecks or containers but also stand for sea bottom elevation maps (or Digital Terrain Model). Vehicles supporting sensors are also considered as objects, as their geometric descriptions may impact the simulation.
- *Media*: Such medium elements gather all the intrinsic properties of the water (sound velocity profiles . . .) because of the impact on the propagation (Ziomek & Polnicky, 1993).
- *Sensors*: Different kinds of sensor are defined like sidescan sonar or multibeam echosounders. The purpose of these sensor objects is to model their specific behaviors

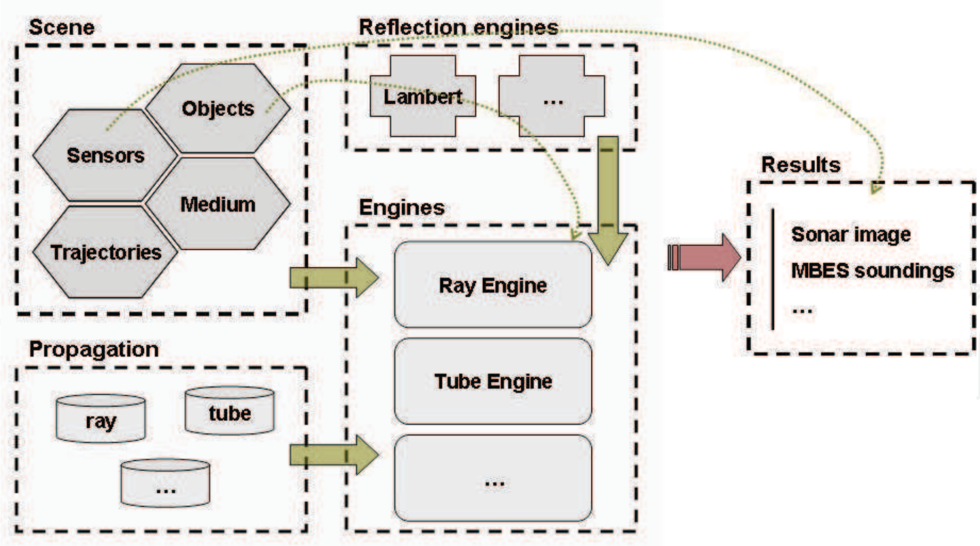


Fig. 1. Framework main software components

especially when aggregating the incoming acoustic waves and reconstructing the output "signal" (Bell & Linnett, 1997).

- *Trajectories*: Objects may be associated to trajectories (typically, vehicles supporting sensors during a survey). These trajectories allow a time varying environment to be taken into account.

This framework design aims at proposing flexibility for developing new items within these component families, allowing reuse and specialization of existing components.

2.2 Propagation

The propagation package gathers all the models available for simulating acoustic wave propagation within a medium (like rays or tubes, for instance). The medium properties greatly affect the propagation behavior: sound velocity variations according to depth may bend the rays depending on the sign of the velocity gradient (Urick, 1983).

2.3 Engines

The engines part relies on a scene description and a propagation model to propose a full simulation of the acoustic propagation within the environment. Therefore, it main goal is to generate a propagation history from the transmitting to the receiving sensor, gathering along the propagation path, all the interactions between the acoustic wave and the scene elements. Preserving such a propagation history allows each receiving sensor to aggregate all the incoming contribution to produce specific sonar simulated data.
In fact, one key feature of this framework is to separate the rendering part (through the various sensors) from the acoustic propagation simulation, thus allowing more flexible updates (with new sensors or simply for testing several sensor settings).

3. A tube engine for propagation history building

Within the global framework presented in the previous section, two engines have been developed in order to validate the required computational modularity. These first two engines basically rely on a ray approach to model the acoustic field propagation: the first considers an acoustic wave as a series of rays while the second uses these rays as support for a volume description of the insonified area.

Previous works on simulating sonar images through acoustic ray tracing techniques have already shown interesting results (Bell, 1997). From these developments, the core engine dealing with ray propagation has been extracted and redesigned to fit in a framework compatible *ray engine*. The results obtained with this new implementation concur with simulations from the previous version (Guériot et al., 1998).

3.1 Why a volume approach ?

The main challenge of such an engine lays in the volume approach of the acoustic wave to be simulated. The previous engine represents the wave propagation as a series of rays always orthogonal to the current local wave front. The involved sampling of the studied volume, for both the propagation itself and the wave interactions with the scene elements, introduces artifacts as each ray has only a very localized interaction with the environment. Thus, tuning this sampling has a great impact on the quality of the simulated data (missing objects falling between two neighboring rays, missing object characteristics like refraction or border effects, artificial backscattering density due to the chosen sampling . . .). It is then obvious that the key feature for this new engine is to implement a volume model of the acoustic wave along with propagation and scene interaction algorithms in order to guarantee an exhaustive energy transport.

3.2 Describing tube propagation

Figure 2 presents the geometrical sensor-centered reference, used for all the computations. Within this reference, $\wp$ corresponds to the selected acoustic propagation model used to build the rays supporting each tube.

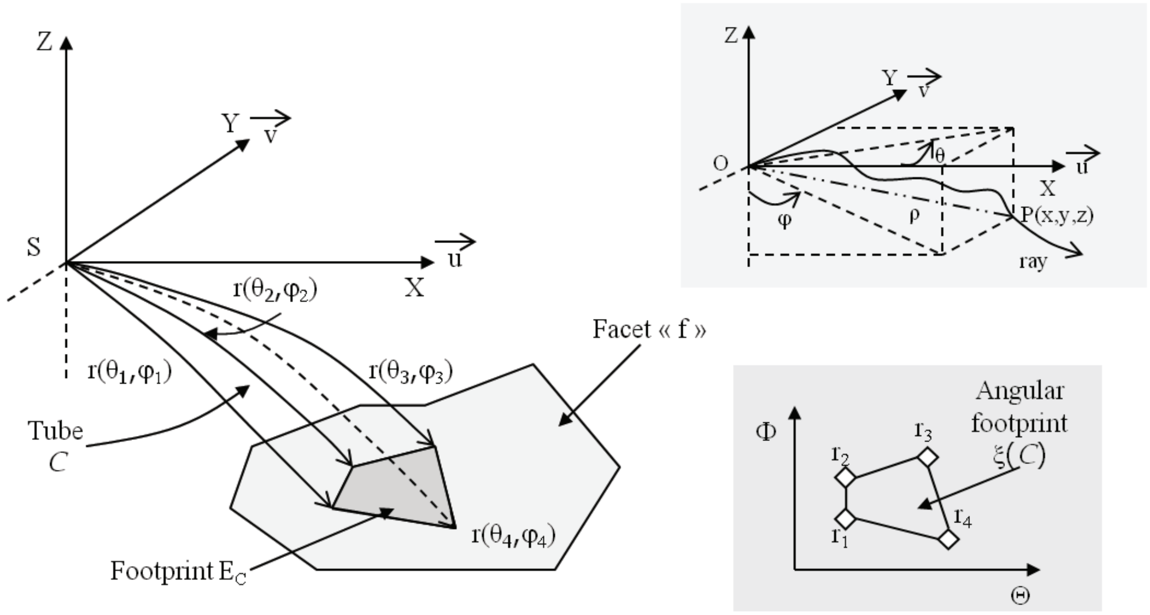


Fig. 2. From rays to tube definition

All the following definitions will be useful to clearly define the basic principles of the tube engine algorithm.

Definition 1 (Ray)**.** *A ray $r(A, \theta, \varphi)$ is defined as an acoustical trajectory $f_{\wp}(t)$ along the time t and according to the selected propagation model $\wp$. θ and φ specify the launching angles of the ray r, from the point A. Thus, the tuple (θ, φ, t) corresponds to the propagation coordinates of a geographical point (x, y, z) within the scene, according to the ray r. It is important to note that a geographical point may have several propagation coordinates.*

Definition 2 (Tube)**.** *A tube $\mathcal{C}(A, r_i)$ is a 3D region of the scene bounded by a series of n rays $r_i(A, \theta_i, \varphi_i)$ such as the polygon $\xi(\mathcal{C})$ defined in $\Theta \times \Phi$, by the vertices list (θ_i, φ_i) is simple. $\xi(\mathcal{C})$ is the tube $\mathcal{C}$ angular footprint. A sector is a 4-rays tube; its goal is to easily split the scene volume space.*

Definition 3 (Tube membership)**.** *A ray $r(S, \theta, \varphi)$ belongs to a tube $\mathcal{C}(S, r_i(S, \theta_i, \varphi_i))$ if the point (θ, φ) defined in $\Theta \times \Phi$, belongs to its angular footprint $\xi(\mathcal{C})$.*

Definition 4 (Elementary tube)**.** *A facet $\mathcal{F}$ in the scene is intersected by a tube $\mathcal{C}$ if it exists a ray r belonging to this tube such as r intersects this facet. In general, a tube may intersect several facets but an elementary tube intersects only one facet of the scene.*

Definition 5 (Tube footprint)**.** *As shown in Figure 2, the footprint $E_{\mathcal{C}}$ of an elementary tube $\mathcal{C}$ on a facet, is the unique polygon which vertices are the intersections between the rays defining the tube and this facet.*

Definition 6 (Footprint intersection)**.** *The footprint intersection $E_1 \cap E_2$ between two elementary tubes $\mathcal{C}_1$ and $\mathcal{C}_2$ is either empty if the footprints E_1 and E_2 do not belong to the same facet, or a set of polygons (general case of intersection between two non-convex polygons).*

Definition 7 (Tube intersection)**.** *The intersection $\mathcal{C}_1 \cap \mathcal{C}_2$ between two tubes $\mathcal{C}_1$ and $\mathcal{C}_2$ is equal to a set of elementary tubes $\Im$ included in $\mathcal{C}_1$ such as it exists no ray belonging to $\mathcal{C}_1$ that has an intersection with the footprint $E_1 \cap E_2$ and does not belong to an elementary tube in $\Im$.*

3.3 Acoustic tube tracing algorithm

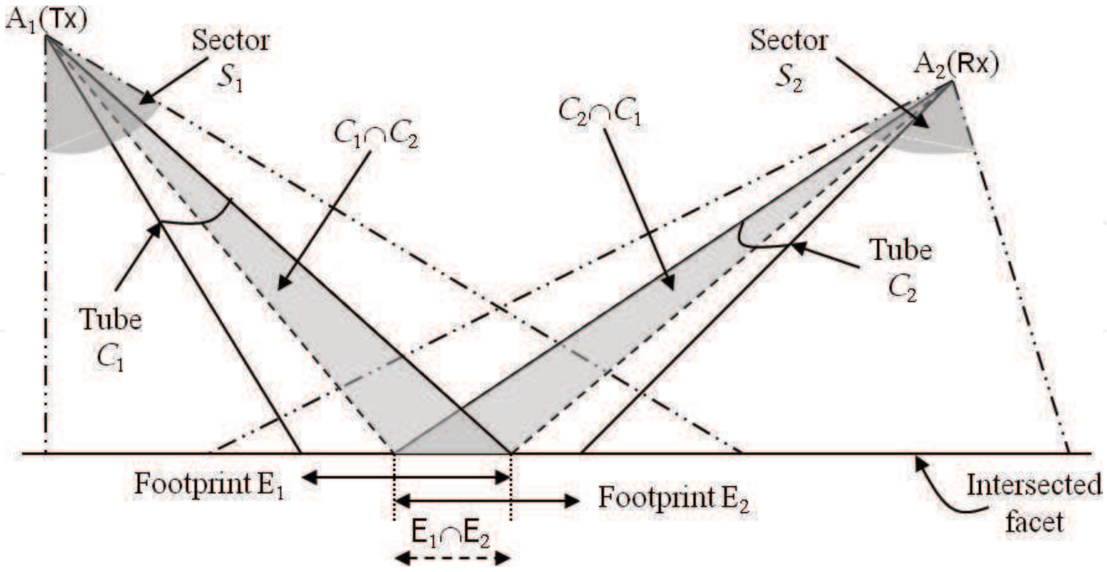


Fig. 3. Geometrical configuration for an elementary tube intersection from Tx to Rx sensors

With all these definitions, the tube engine basic computation algorithm is a 4-step processing (see Figure 3):

- The transmitting sector, defined by the Tx sensor properties, is split as long as it contains non-elementary tubes.
- The same splitting process is performed for the receiving Rx sensor.
- For each intersection between an elementary Rx tube and an elementary Tx tube, the corresponding footprint is computed allowing for the receiver to properly aggregate the transmitter contribution after its interaction with the intersected facet.
- In case of multiple paths simulations, all the footprints belonging to a facet are gathered in order for this facet to become a new transmitter. The amount of levels in this recursive loop defines the number of multiple paths to consider.

3.4 Tube engine generic output

Computing the full 3D acoustic wave propagation history is the major key feature of this algorithm. Each sensor uses it then to output data according to their specific appropriate *aggregation* algorithm (complete signal or amplitude for images...). Next section will use this propagation history in order for several sensors to output simulated data.

4. Multiple sensors simulation data

Based on the previously computed propagation history, this section presents simulated data from three different sensors: sidescan sonar, multibeam echosounder and acoustic camera. Following their specific aggregation algorithms, results are shown with focuses on specific simulator features.

4.1 Simulating sidescan sonar

4.1.1 Aggregation principles

According to Figure 3, the geometrical configuration and properties of the facet supporting an elementary tube, characterize the energy backscattered through that tube. When arriving on the Rx sensor, this temporal backscattered energy is sampled according to the sensor sampling frequency to fill up the corresponding pixels in the output image. Processing all the Rx elementary tubes with respect to their intersection with Tx elementary tubes, completes the full image and involves a full energy preservation from the Rx sensor point of view.

4.1.2 Imaging complex objects

High resolution simulated sonar images have been produced by our simulator when using this tube engine and a *virtual* sidescan sonar to perform the acoustic data collection. An oil well head model has been read from a "Wavefront" formatted file in order to extract the geometrical description of the object (mainly its vertices and facets). Using this facet-based description, a tube engine based simulation was launched assuming the oil well head lies on a flat sandy bottom. Several rotations (none, then 45° along the sonar path, then 30° more across the sonar path) have been applied to the oil well head in order to produce the images shown in Figure 4. The resulting images clearly agree with the 3D model (also presented).

4.1.3 Imaging real sea-bottom

For this test, a Digital Terrain Model built with bathymetric measurements on a small area near the Brittany shores, has been used as sea bottom on which rocky patches appear along with a small ridge and rippled sandy regions. Figure 5 (upper part) shows the obtained simulated image on such a sea bottom and displays the links between the real sea bed features (rocky

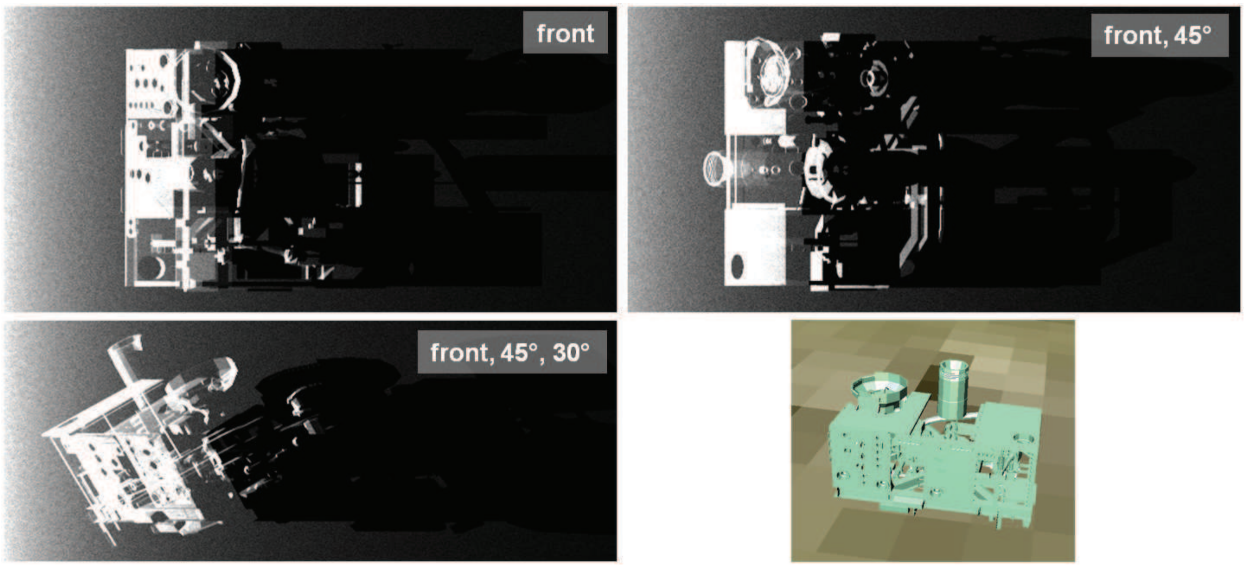


Fig. 4. Oil well head simulated sonar images (according to its 3D model)

patches, ridges, sandy areas limits) and how they are rendered on the simulated image. The main goal of this sketch is to show the consistency between the simulator inputs and the produced data.

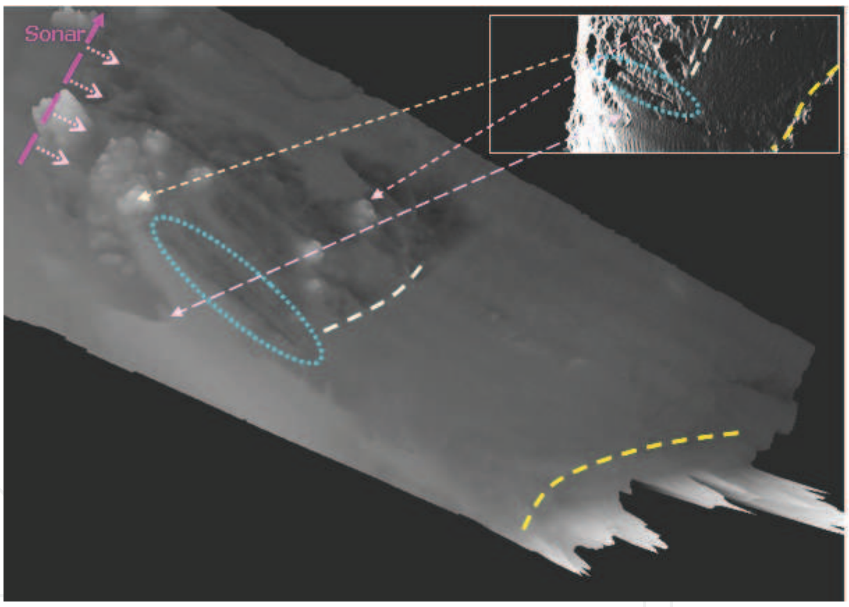


Fig. 5. From sea bottom topography to sonar image

4.1.4 Ray vs. Tube engines

As explained in section 3.3, the tube engine makes a complete energy summation over the whole solid angles. Even if tube borders are defined using rays, the inside volume is taken into account for all scene intersections, removing undersampling artifacts observed with the pure ray engine like missing sea-bottom features as shown in Figure 6. Moreover the medium absorption is directly taken into account through the volume approach involving a more realistic output (before a Time Varying Gain correction).

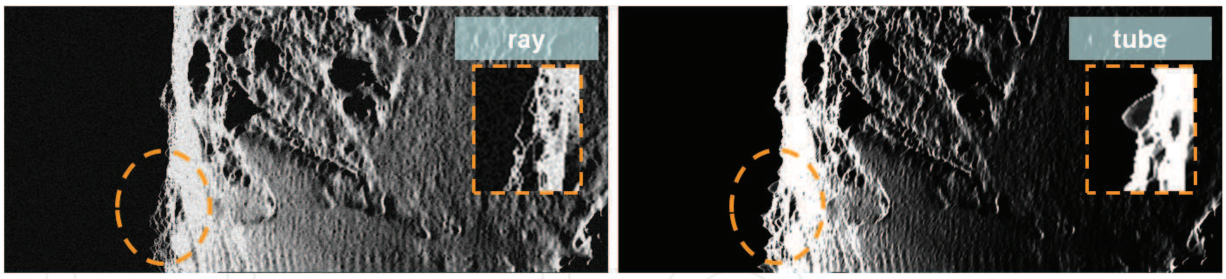


Fig. 6. Comparing ray vs. tube simulated sonar images on the same sea-bottom area (with subsampling effects).

4.1.5 Comparing simulated with real sonar images

In order to check the quality of the simulator outputs, we compared real sonar images with simulated ones using the same acquisition conditions (sensor trajectory & attitude). Data collected for the *Shallow Survey '99* conference (Australia Defense Science & Technology Organization, 1999) in Sydney harbor have been used for this test according to the methodology given in Figure 7. During this survey, several kinds of sensors were used: a RESON 8101 multibeam echosounder for bathymetry data and a Klein 5400 for high resolution sonar images. Extra information such as sound velocity profiles and tide charts were also provided to correct bathymetry measurements in order to build a 2km by 3km DTM of the observed region with a 1.5m resolution cell.

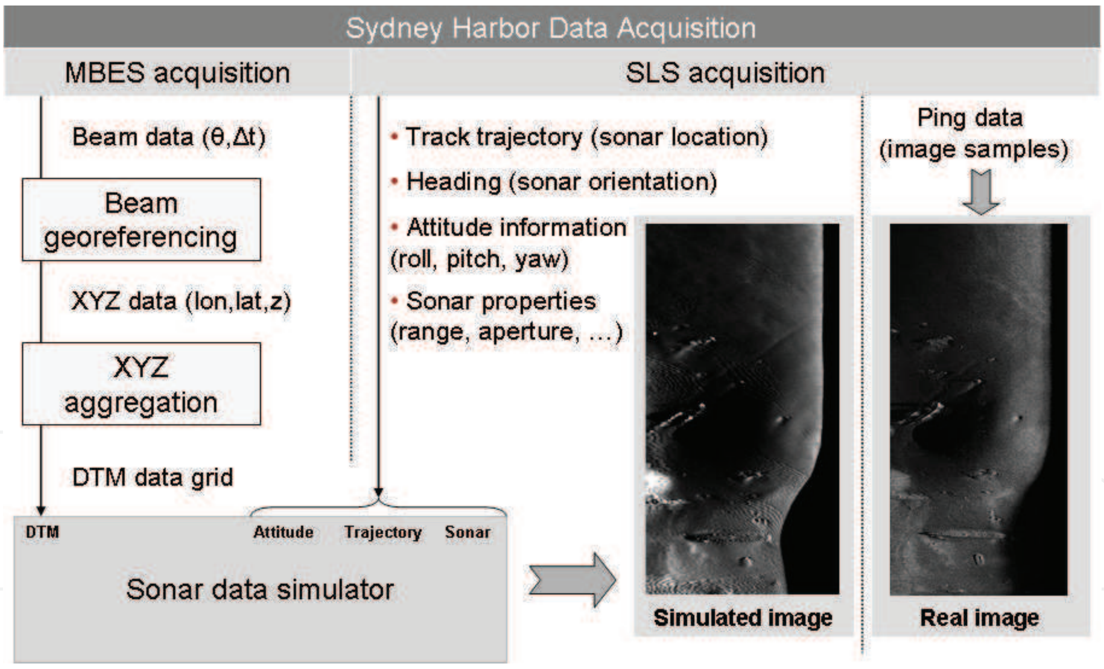
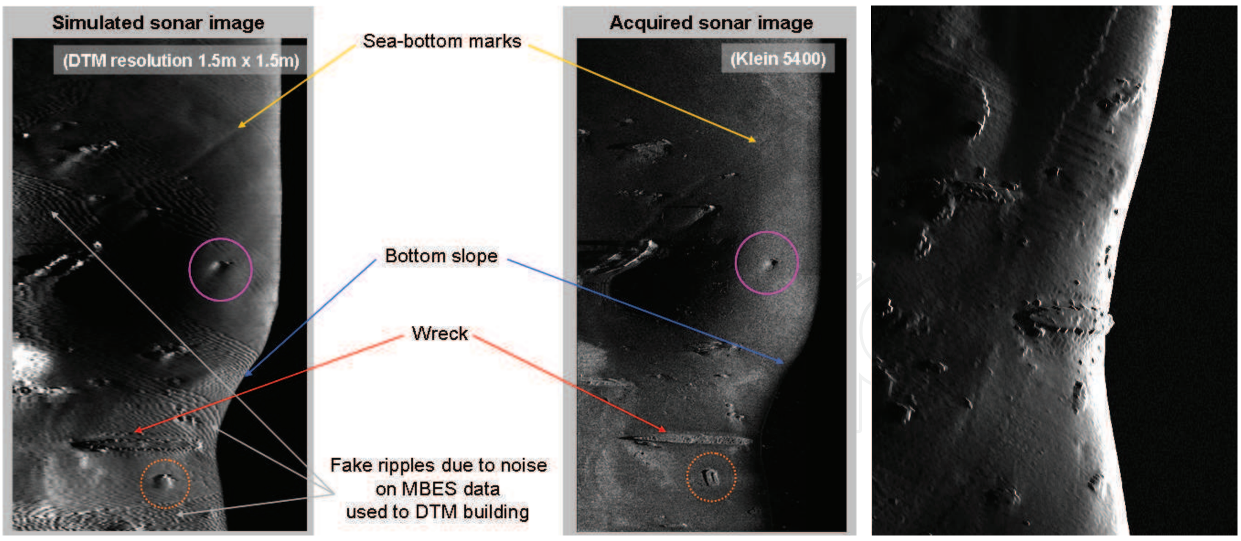

Sydney Harbor Data Acquisition
MBES acquisition
SLS acquisition
Beam data (θ,Δt)
Beam georeferencing
XYZ data (lon,lat,z)
XYZ aggregation
DTM data grid
• Track trajectory (sonar location)
• Heading (sonar orientation)
• Attitude information (roll, pitch, yaw)
• Sonar properties (range, aperture, ...)
Ping data (image samples)
DTM
Attitude
Trajectory
Sonar
Sonar data simulator
Simulated image
Real image

Fig. 7. Comparing simulated vs. real data procedure

The purpose of using such a survey is to compare real sonar images with simulated ones produced by the simulator fed with either raw data (sound velocity profile, trajectories . . .) or preprocessed information (DTM), but excluding the real sonar images. The expectation is to find on the simulated images, the features appearing on the real ones.

A sidescan sonar antenna matching the KLEIN 5400 specifications has also been defined to produce images as output. Figure 8 presents both simulated and real sonar images. A quick



(a) Klein 5400 vs. simulated sonar image (DTM built with 1999 MBES data collection)

(b) Simulated sonar image using a DTM built from MBES data acquired in 2003

Fig. 8. Comparing Klein 5400 sonar images with simulated one on the same sea-bottom area (Sydney bay).

look shows that all the features (wrecks or bottom slopes) even the smaller ones, existing in reality (and shown on the real sonar image), also appear in the simulated data:

- *Objects*: both wrecks (main one on the bottom part, smaller one on the central part, end of range) can be identified. Two other spots corresponding to impacts (purple plain circle) or objects (orange dotted circle) also match.
- *Marks*: even fine sea-bottom marks appear on the simulated image.
- *Backscattering strength*: with recorded sidescan images only, determining the reason for having sandy lighter texture areas is not easy; their appearance in the simulated data involves they come from an elevation change (with a slope facing the sensor), besides sediment differences. This aspect is very interesting from a scene interpretation point of view.

The smoothing effect appearing on features of the simulated image can be explained by the DTM grid resolution (1.5m by 1.5m cell) compared to the resolution of the real or simulated sonar images (a few centimeters).

Some artifacts are also observed on the simulated data. These ripples come from the DTM building phase with not so well-calibrated MBES data. This lack of calibration introduced a few centimeters height difference between XYZ points coming from end-of-range overlapping tracks. As with this survey, only one or two points participated to the height of a DTM cell located at the end of the sonar range, for these cells, the poor amount of data leads to alternating heights between neighboring cells.

Therefore, having an accurate world representation is crucial as the simulation accuracy puts into relief all its details. Thus, when using a DTM built on the same area but with different MBES data (2003 instead of 1999 survey), these artifacts do not show off. Figure 8(b) presents such a simulated image with a sensor trajectory slightly shifted to the left: the sonar path flies just over the main wreck, as revealed by the water column limit.

4.2 Simulating multibeam echosounders

4.2.1 Aggregation principles

Each Rx beam from the multibeam echosounder is described by a specific Rx sector which is split into Rx elementary tubes according to the same process as previously. A series of energy contribution along time is then computed for each Rx beam. Based on amplitude detection, the first temporal return among these contributions, provides the beam sounding characterizing the return time travel between the sensor and the intersected scene.

4.2.2 Results

A RESON 8101 survey has been simulated on the Sydney area using a previously built DTM. On the simulated data, the two wrecks observed on the simulated sidescan images are logically present along with some small rocks over the slopes of the basin.

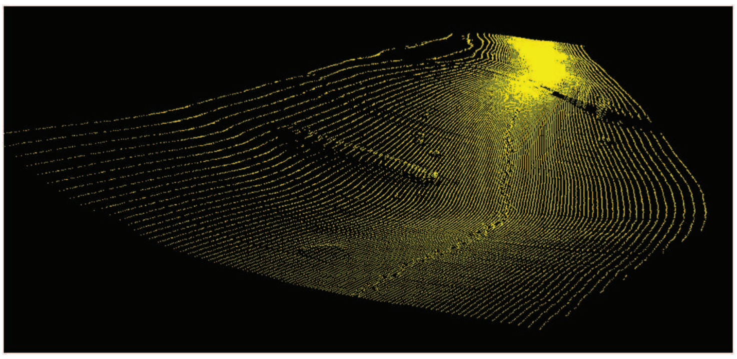


Fig. 9. Multibeam echosounder simulation on the Sydney harbor area

4.3 Simulating front-looking imaging sonar : DIDSON front looking camera

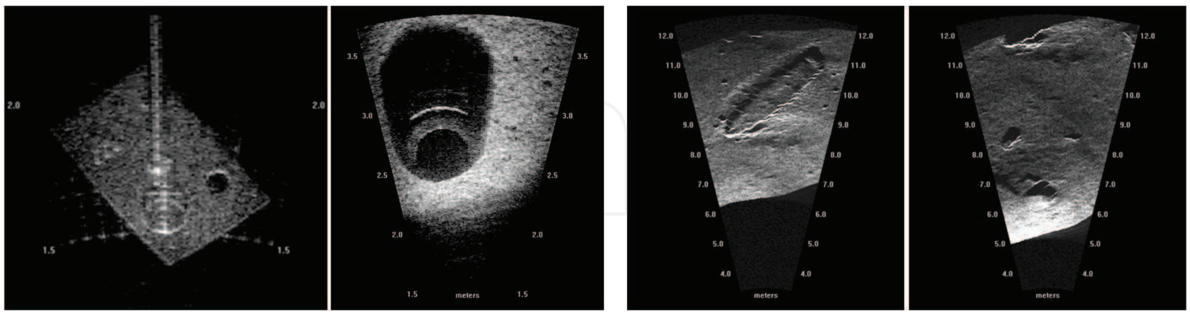
Based on high frequency sonar, DIDSON acoustic cameras provide short range, high resolution images of underwater environments when optical systems get blind mainly due to water high turbidity. Moreover, their image refresh rate can be rather high providing several views of quite the same area, allowing various applications like inspection (Belcher et al., 2001), navigation or surveillance missions (Asada et al., 2007).

4.3.1 Aggregation principles

Besides the generic propagation history building, some DIDSON specificities need to be taken into account to create realistic images. DIDSON system uses acoustic lenses to perform direct and energy saving beamforming. The antenna length is split into 48 or 96 cells or beams, each beam being 0.3° width with spacing ranging from 0.3° to 0.4°, for a full 29° azimuth coverage. Sampling along time the energy focused inside each cell provides per-beam backscattered signal according to range (Belcher et al., 2002). A DIDSON image gathers all these collected backscattered signal samples per beam and per range.

However, according to the system frequency (high or low resolution), a full DIDSON image (48 or 96 beams) is built from the reverberation received by 4 or 8 sets of 12 non-adjacent beams fired simultaneously. This interlacing technique requires several Tx/Rx cycles to produce a complete image but aims at preventing beams cross-talk. Indeed, the acoustic beam pattern width for one beam is almost 2°, showing that two neighboring (0.3° spaced) beams cannot be used within a single cycle. Nevertheless, depending on the shape and nature of the observed object, cross-talk phenomena are still observed on real images (see Figure 10(a)) as side lobes

may gather enough energy contributions from round-shaped objects that reflect transmitted energy to all directions.
All the built images are stored within a DDF file (DIDSON format for sequences of recorded images) as frames.



(a) Real DIDSON images (metal plate and tire)

(b) Simulated images extracted from a simulated sequence on Sydney bay

Fig. 10. Examples of real and simulated DIDSON images.

4.3.2 Results

The component in charge of building DIDSON images from the propagation history gathers the backscattered energy described through the lists of Rx elementary tubes according to sensor Tx and Rx patterns. Moreover, depending on the considered cycle, only sectors corresponding to the fired beams are used for data collection.

4.3.2.1 Image simulations

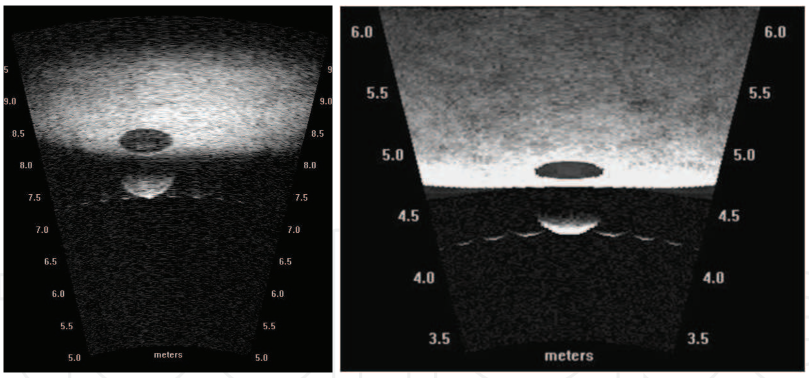
Figure 10(b) exhibits two frames extracted from a simulated DDF file using Sydney harbor survey data (DTM). These images clearly show the big wreck and the rocky patches already displayed by both the previous sensors.

4.3.2.2 Observing cross-talk phenomenon

When performing simulation on round-shaped objects, cross-talk phenomena are observed the same way as with images coming from a real DIDSON camera. Figure 11 presents both real and simulated images on a scene containing a sphere lying on a flat bottom. Cross-talks are logically observed on both real and simulated images as round objects reflect the transmitted acoustic wave to all directions. This realistic result clearly shows the ability of the simulator to take into account the actual sensor beam pattern and its specific aggregation algorithm, without questioning the propagation history principle.

4.4 Why using georeferenced perturbations ?

Using facets or DTM cells for describing items inside an underwater scene defines the resolution of the geometrical features seen by the sensors during any simulation. However, many phenomena come from higher resolution scene features. For instance, on real DIDSON images collected in a flat bottom pool, some patterns between consecutive images are observed, besides the fluctuating noise. Being clearly georeferenced, these patterns show that modeling a flat bottom by a flat DTM only, is not enough. Indeed, they reveal the bottom response occurs at a finer resolution than the one used for DTM description.
Next sections will present two approaches to allow a more detailed scene description, producing either *macro-textures* or *micro-textures*.



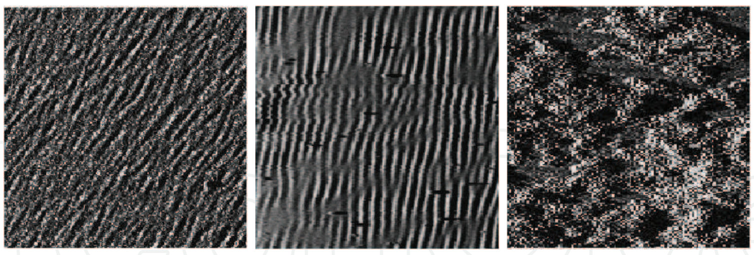
(a) Real (b) Simulated

Fig. 11. Real & simulated DIDSON images with crosstalk phenomenon on a sphere lying on a flat bottom

4.4.1 Macro-textures

This first approach introduces georeferenced perturbations to the elevation surface modeling the observed scene, by decomposing each facet into smaller ones (sub-facets). Adding the global elevation coming from the underlying facet (DTM or object) to a small variation produced by a local and finer elevation map, gives each sub-facet elevation. Thus, each sub-facet produces a specific response giving a more realistic characterization of the encompassing facet.

Various techniques may produce adequate local elevation maps depending on the textures to be simulated. These textures clearly depend on the scene sea-bottom sedimentology. Rocky sea-bottoms may be obtained using standard fractal techniques like fractional Brownian motion (Pentland, 1984). Moreover, shifting fractional Brownian motion spectral components allows producing oriented textures; the direction and amplitude of the produced ripples depend on the direction and length of the spectral translation (Guillaudeux & Maillard, 1998).



(a) Soft ripples (b) Sharp ripples (c) Rocky area

Fig. 12. Using macro-textures to simulate sea-bottom elevation features

Figure 12 shows examples of simulated texture images using these techniques to produce local elevation maps. Thus, without any modification of the simulator core, introducing local georeferenced perturbations to the global elevation map, allows producing more realistic sea-bottoms textures (with structured shadows) reflecting local seabed altitude variations.

4.4.2 Micro-textures

The second approach that produces *micro-textures* does not involve elevation perturbations but introduces georeferenced perturbations to the facet reflection properties. In that case, the

standard backscattered energy transmitted by each Rx elementary tube is modulated by a local contribution linked to both the statistical properties of the facet (sediment nature, material. . .) and the location on this facet.
Figure 13 illustrates the impact of such georeferenced backscattering modifications on simulated images with various micro-textures corresponding to several sandy sea-bottoms. Indeed, modifying the backscattered energy at the elementary tube resolution allows tuning image granularity inside facets, but unlike the macro-texture approach, this micro-texture approach cannot produce textures with shadows.

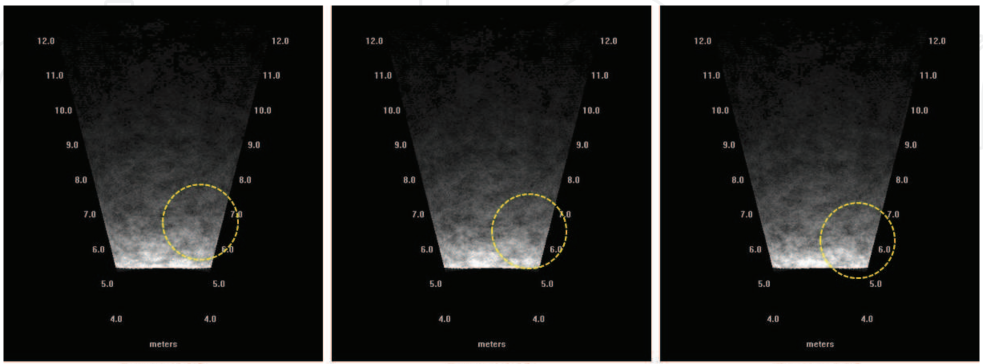
(a) Mud (b) Sand (c) Gravel

Fig. 13. Using micro-textures to simulate various kind of sandy sea-bottoms

As for macro-textures, micro-textures can be generated using existing maps or georeferenced perturbation functions computed during the simulation. The underlying micro-texture used for the next simulation results has been generated online using Perlin noise (Perlin, 2002) and counted for 25% of the diffuse reflection component.

4.5 Simulation results

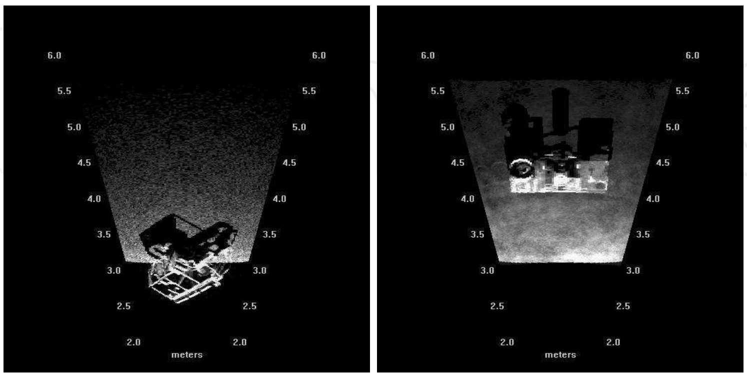
As previously mentioned, consecutive DIDSON frames exhibit a meaningful level of correlation between images. Micro-textures, being georeferenced by definition, induce such cross-correlations between simulated images observing same scene areas as illustrated by Figure 14 which shows several consecutive images extracted from a sequence simulated with a flat sea-bottom with a straight constant speed trajectory. The yellow dotted circle outlines a visible pattern between the three frames. As the sensor moves along its trajectory, the pattern slowly drifts towards the sensor water column (first return recorded by the sensor).



Frame 138 Frame 142 Frame 146

Fig. 14. Neighboring frames (with consistent micro-textures) extracted from a simulated DIDSON sequence with a straight constant speed sensor trajectory

Finally, Figure 15 compares two simulations performed with the oil well head scene. As the sea-bottom from the left image is only modeled by a flat DTM with no micro-textures, no structured pattern appears: the backscattered energy is only affected by random and uncorrelated noise. On the contrary, the right image integrates georeferenced perturbations into the reflection: the underlying flat bottom produces uneven intensities across the image. Moreover, as micro-textures adds spatially structured contributions to the basic backscattering, some smoothing effect is also observed.



(a) No cross-talk, no micro-texture (b) Cross-talk, micro-texture

Fig. 15. DIDSON simulated images and georeferenced perturbations using Perlin noise

5. Simulating sonar full signal

Previous sections have explained how realistic images can be produced by simply gathering incoherent contributions from the observed sea-bottom. No speckle effect although very obvious on sidescan sonar image, can be produced with this approach. However, extra contributions looking like speckle noise may be added to the simulated images, after simulation. Partial diffraction effects for specific configurations may also be integrated to simulated images, using the same kind of shortcuts. However, simulating phase signal is absolutely required in the general case and clearly to feed standard sonar processing like interferometry or beamforming. Simulating images only is not enough.

Simulating the phase of a sonar signal is a matter of aggregation resolution. Indeed, when simulating sonar images, the scale of aggregation is the sensor sampling cell which defines the scale used to aggregate the backscattered contributions. However, coherent phenomena involving the signal phase component, occur at the signal wavelength scale. Simulating the phase signal thus requires the simulator to process the propagation history at this scale. Hopefully, this propagation knowledge is already available.

The notion of *coherence* is the key feature for any realistic signal simulation. Indeed, the sampled backscattered signal is the result of coherent summations of contributions from sea-bottom tiny parts. These tiny reflectors behaving according to many parameters such as local sea-bottom properties, the summation is strongly linked to the observed sea-bottom small area. Indeed, estimating the correlation between two distinct sea-bottom patches, returns 0 while the correlation of two partially overlapping areas is only proportional to their common surface: the specific identity of every sea-bottom part affected by the transmitted signal, controls the coherent summation.

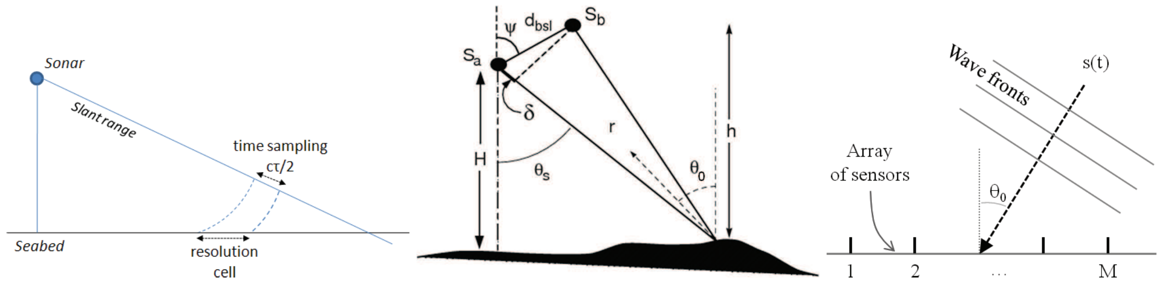
Thus, any simulation engine looking forward to being used as input for sensor array processing (beam processing or interferometry) has to deliver a georeferenced phase associated to every sampling cell using the underlying coherent summation. The challenge is to find a sea-bottom sampling model which is computationally simple enough to be used for full track simulations. Next paragraphs will describe a proposed attempt to get realistic phase simulation. First, the required formalism is introduced to fully explain the approach. Then, a Brownian-based optimization is proposed to lighten the computational burden needed to get reflectors behaving according to wished statistics properties.

5.1 Requirements for building a coherent sea-bottom simulation

The first engine plugged in this simulator uses ray tracing and only involves geometrical interactions between the acoustic wave and the scene. These local interactions are closely related to punctual scatterers but how to use these rays to simulate coherent backscattering still remains a big issue since two major parameters of the acoustic wave are not taken into account: the signal wavelength and the signal bandwidth which quantifies the quantity of information transported by the carrier frequency and thus, the sensor resolution.

5.1.1 Sonar resolution cell

Each time sample of the backscattered signal (after sonar demodulation) corresponds to a footprint on the sea-bottom, also called resolution cell. The energy level contained in such a sample results from a very intricate combination of several contributing parts within the resolution cell. Figure 16(a) illustrates such a resolution cell according to the sampling interval τ and the sound speed c. Thus, the challenge of this simulator is to use the acoustic propagation history in order to create a summation process, representative of the coherent aggregation.



(a) Resolution cell definition (b) Interferometry (c) Beamforming

Fig. 16. Basic principles for interferometry and beamforming

5.1.2 Sonar processing involving phase signal: interferometry and beamforming principles

Simulating full sonar signal including the phase component is absolutely required to feed standard sonar processing like interferometry or beamforming with signal generated by the simulator. This section quickly describes how the phase component is crucial for such processing and clearly shows that phase simulation is intrinsically linked to the sea-bottom coherence.

5.1.2.1 Interferometry principles

Interferometry is a widely-used method that provides accurate bathymetric information by measuring the phase difference between two sensors (Lurton, 2003)(Denbigh, 1989). Figure 16(b) depicts the standard geometrical configuration for interferometric sonar systems where θ_0 is the direction of arrival (DOA) of the backscattered signal and ψ, the steering angle. Assuming a non-dispersive medium hypothesis (i.e. the phase and group velocity have a linear connection), Equation 1 gives the searched phase difference $\Delta\varphi$ which represents indeed a backscattered signal delay between the sensors.

$$\Delta\varphi = \frac{2\pi}{\lambda}\,\delta = \frac{2\pi}{\lambda}\,d_{bsl}\;\cos(\theta_0 + \psi) + 2\pi k \tag{1}$$

where λ is the acoustic wavelength, d_{bsl} measures the spacing (or baseline) between the sub-arrays centers and k corresponds to the number of 2π phase ambiguities as $\Delta\varphi$ can go beyond 2π for high angles and large baselines, while the measured phase difference (the cosine term) remains trapped within the $[0, 2\pi]$ interval.

5.1.2.2 Beamforming principles

Beamforming is also widely used to build the backscattered signal from the receiving antenna considered as an array of several elementary sensors with wide aperture. Basic beamforming thus consists in an in-phase summation of the backscattered signal collected by all the M elementary sensors. As shown in Figure 16(c), when assuming the incoming signal $s(t)$ comes from a far field source with a direction of arrival θ_0, Equation 2 gives the signal $\vec{y(t)}$ arriving on each sensor of a d-spaced line array.

$$\vec{y(t)} = a(\vec{\theta}_0)s(t) + \vec{b(t)} \tag{2}$$

where $\vec{b(t)}$ represents noise contribution on each sensor and $a(\vec{\theta}_0)$, the goniometric vector

$$a(\vec{\theta}_0)^T = \left[e^{2\pi j \frac{0}{\lambda}\sin(\theta_0)}, ..., e^{2\pi j \frac{d.(M-1)}{\lambda}\sin(\theta_0)}\right] \tag{3}$$

Building the final backscattered signal $y_{BF}(t)$ through beamforming consists in introducing a phase correction on each sensor to compensate for the $a(\vec{\theta}_0)$ effect before summing all the elementary contributions $y_k(t)$ from $\vec{y(t)}$. Equation 4 gives this in-phase summation.

$$y_{BF}(t) = \sum_{k=0}^{M-1} e^{-2\pi j \frac{d.k}{\lambda}\sin(\theta_0)} y_k(t) \tag{4}$$

5.1.3 How to model the phase brought by resolution cells ?

Many works model various properties of these resolution cells and how they contribute to the backscattered signal in terms of phase behavior (Mourad, 1989)(Kuo, 1964).

5.1.3.1 Exact and continuous models

Two main approaches try to exactly describe resolution cells backscattering behavior using continuous models. The first one takes benefit from the rough surface theory with Kirchoff condition while the second one deals with diffraction phenomena through *micro-roughness* with Bragg wavelength (Jackson et al., 1986). These models heavily rely on finding Green function solutions to solve the propagation equation making the computation of one resolution cell backscattered signal very complex and time consuming. Moreover, all these models make strong assumptions about isotropy or angle of incidence.

5.1.3.2 Single scatterers approach

A third approach considers a resolution cell as a set of point scatterers. Then the backscattered signal F of a resolution cell is equal to the coherent summation of the incoming signal $s(t)$ transmitted back by the N single scatterers belonging to this resolution cell. As shown in Equation 5, this atomistic model is very appealing when the amount of scatterers remains small as each scatterer i receives the incident signal $s(t)$ and transmits it back with modified power a_i and phase ϕ_i corresponding to the scatterer properties.

$$F = \sum_{i=1}^{N} a_i e^{j\phi_i} s(t) \tag{5}$$

As within a resolution cell, the amount N of scatterers, the amplitude a_i and phase ϕ_i of every scatterer are random, it easily explains the reverberation diversity such a model is able to produce. However, finding basic single scatterer properties fitting both experimental and theoretical models is the main drawback of this model as the involved mathematical developments may become rapidly complicated due to the manipulation of complex random variable summation.

Moreover, when performing interferometry, several sensors observe almost the same sea-bottom from different points of view. To each sensor corresponds a series of resolution cells. A small patch of sea-bottom is then covered by one resolution cell per sensor, each one having slightly different boundaries on the sea-bottom due to the various points of view. Spatial and angular decorrelations are then introduced between these resolution cells as they do not include exactly the same set of single scatterers (Sintes et al., 2010). The spatial distribution of the scatterers is thus a crucial point to consider in order to perform consistent correlation between the resolution cells and compute for instance, their phase difference.

Thus controlling both the spatial correlation and random properties of single scatterers are fundamental to obtain a realistic full sonar signal simulation.

5.2 Simulating phase for sensor resolution cells

Generally speaking, the key feature to create a simulated signal (both intensity and phase), is to make interfere different parts of the resolution cells.

5.2.1 An atomistic approach : single scatterers and ray tracing

This paragraph gives a short introduction on how to simulate the response of a resolution cell through a single scatterer model and a ray tracing approach within a simulation context.

5.2.1.1 Building a resolution cell response from single scatterers

Equation 6 gives the temporal response $h(t)$ of a single scatterer which distance from the receiver is $r/2$, δ standing for the Dirac function.

$$h(t) = \delta(t - r/c) \tag{6}$$

As shown in Equation 7, we consider the sonar transmits a simple truncated sinusoidal signal $s(t)$ where T is the signal duration and f, the carrier frequency. $\mathbb{1}_A(u)$ stands for the indicator function and is equal to 1 when $u \in A$, 0 anywhere else.

$$s(t) = \mathbb{1}_T(t - r/c).e^{j2\pi f t} \tag{7}$$

The received signal $x(t)$ depends on $s(t)$ with a phase shift depending on the total traveled distance r.

$$x(t) = \mathbb{1}_T(t - r/c).e^{j2\pi f(t-r/c)} \tag{8}$$

Finally, demodulating the received signal $x(t)$ gives $y(t)$ with

$$y(t) = x(t)e^{-j2\pi ft} = \mathbb{1}_T(t - r/c).e^{-j2\pi fr/c} \tag{9}$$

This signal $y(t)$ is sampled in a $z(k)$ vector according to the sampling step (or resolution cell duration) T_e. Assuming the sampling step is close to the signal duration ($T \approx T_e$), Equation 10 computes the contribution of one r_i distant single scatterer i to the sampling cell $z(k+1)$.

$$z(k+1) = \mathbb{1}_T\left((k+1)T_e - r_i/c\right).e^{-j2\pi fr_i/c} = e^{-j2\pi fr_i/c} \tag{10}$$

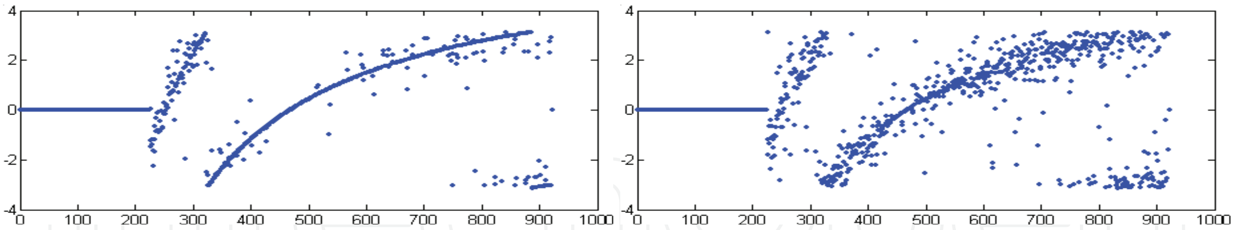
The sampling cell $z(k+1)$ gathers the contributions of all the scatterers i within the corresponding resolution cell: the scatterers i must check $kT_e < r_i/c < (k+1)T_e$ to be summed into $z(k+1)$ as shown in Equation 11.

$$z(k+1) = \sum_i a_i.e^{-j2\pi fr_i/c} \tag{11}$$

5.2.1.2 Simulation through ray tracing

Performing the sensor aggregation step according to Equation 11 is straightforward when using a ray engine. However, the amount of rays hitting each resolution cell has to be dense enough to compute the contribution of every single scatterer within this resolution cell.

Relying on this model, Figure 17 proposes two simulated interferometric signals on a flat sea bottom. In this example, the amplitude of every single scatterer is kept constant. The difference between the two signals lies on the average amount of scatterers per resolution cell, at the end of the swath: 3 for Figure 17(a) and 40 for Figure 17(b). The second simulated signal seems more realistic because of a higher noise level, coming from a bigger amount of contributions per resolution cell. Indeed, as the resolution cell has a finer description (more single scatterers), the spatial decorrelation effect is better described.



(a) 3 single scatterers (b) 40 single scatterers

Fig. 17. Simulated interferometric signal using ray tracing engine

This first attempt is very interesting as it clearly shows the simulator (using the ray engine) can be used to simulate interferometric signals. Nevertheless, some key features remain to be addressed: for instance, choosing a unique constant coefficient of reflection for all single scatterers involves important constraints on the random properties of the simulated backscattered signal. As shown, the random properties of the resolution cell are also affected by the amount of scatterers per resolution cell (i.e. the amount of rays hitting this cell).

A more continuous approach may be used to solve this acoustic wave discretization effect: this atomistic model may be extended for being used by a tube engine. It mainly requires to rewrite the summation of the incoming contributions within a continuous context.

5.2.2 Towards a continuous approach : extended single scatterers and tube tracing

Going from a discrete to a continuous model, Equation 12 gives the contribution z of the observed sea-bottom within a time interval $[t_1, t_2]$: all the sea-bottom parts $u = r/c$, backscattering signal within the interval $[t_1, t_2]$ participate in z.

$$z = \int_{t_1}^{t_2} e^{-j2\pi f u} du = e^{-j2\pi f \frac{t_1+t_2}{2}} . \frac{\sin(\pi f(t_2 - t_1))}{\pi f} \tag{12}$$

Based on the linearity of the integral, a Chasles relation can be derived from Equation 12 between neighboring surfaces sharing the same properties. Equation 13 shows such a relation when considering three instants t_1, t and t_2.

$$\begin{aligned} z &= e^{-j2\pi f \frac{t_1+t}{2}} . \frac{\sin(\pi f(t - t_1))}{\pi f} + e^{-j2\pi f \frac{t+t_2}{2}} . \frac{\sin(\pi f(t_2 - t))}{\pi f} \\ &= e^{-j2\pi f \frac{t_1+t_2}{2}} . \frac{\sin(\pi f(t_2 - t_1))}{\pi f} \end{aligned} \tag{13}$$

5.2.2.1 Building a continuous resolution cell response

The sea-bottom contribution to the $k+1$ sampling cell is easily computed with $t_1 = kT_e$ and $t_2 = (k+1)T_e$ as shown in Equation 14.

$$z(k+1) = e^{-j2\pi f (kT_e + \frac{T_e}{2})} . \frac{\sin(\pi f T_e)}{\pi f} \tag{14}$$

A major benefit from this model is that as the tube engine splits a resolution cell into angular intervals which are different for one sensor to another, Equation 13 reveals it is possible to choose any splitting of one resolution cell, for a subsequent and adequate merging with an other overlapping resolution cell.

However, Equation 14 also points at one major drawback of this approach: no phase signature is associated to each radiating sub-surface composing a resolution cell as $z(k+1)$ only depends on k and T_e. It involves any interferometric processing using such modeled signals on every resolution cell will give no informative correlation (1 everywhere) which is useless. This behavior is logically not observed using the ray engine with a non-uniform distribution of scatterers as each single scatterer returns a phase linked to its distance from the transmitter.

5.2.2.2 Extended single scatterers

The proposed solution is then to give each contribution an unique identity. Associated with this ID, a single scatterer becomes an *extended single scatterer*. Indeed, in order to make backscattered simulated signals interfere, each scatterer should be coherent with itself but not with its neighbors.

Thus, the key idea is to divide the sea-bottom into elementary patches or *extended single scatterers*, with specific properties. Through this local identification, the continuous splitting of the sea-bottom preserves both the conservation of energy when integrating on surfaces and the decorrelation phenomenon between slightly different areas of the sea-bottom.

The flexibility of this approach comes from Equation 13 that allows an extended scatterer overlapping two resolution cells, to be split into two extended scatterers while preserving both phase information and energy. Figure 18 illustrates such a configuration with two sensors slightly shifted in relation to the sea-bottom. In this example, all the extended scatterers have a $T_e/2$ time contribution within the T_e duration resolution cells. Then the integration duration T_e is divided into 4 parts, to take into account all the combination between the overlapping scatterers from sensor #1 and sensor #2 within the two resolution cells but also the time delay between the cells. Each part has its own phase properties. Sensor #1 and sensor #2 signals can then be computed according to Equation 12, over the appropriate integration area. Using the simulated signals from sensor #1 and #2, the interferometric signal (and its phase) can be obtained through standard interferometry processing as given in Equation 1.

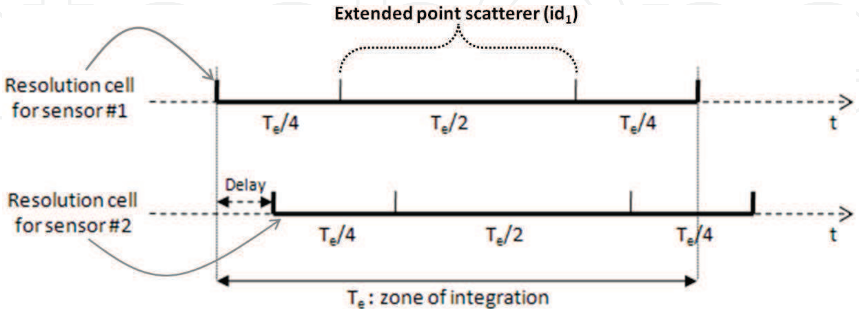

Extended point scatterer (id$_1$)
Resolution cell for sensor #1
T$_e$/4
T$_e$/2
T$_e$/4
t
Delay
Resolution cell for sensor #2
T$_e$/4
T$_e$/2
T$_e$/4
t
T$_e$: zone of integration

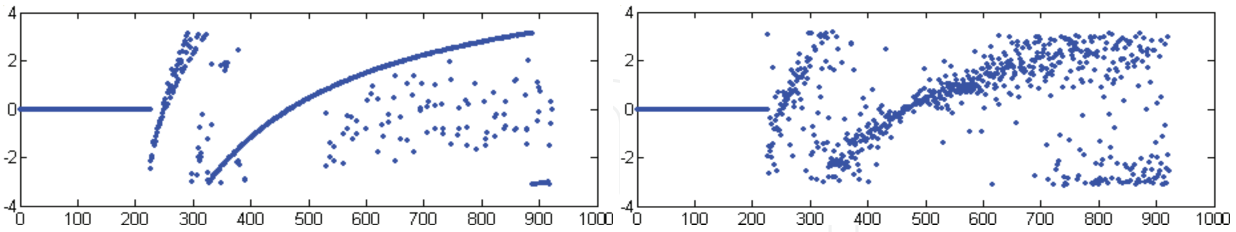
Fig. 18. How the continuous approach with extended scatterers allows taking into account spatial decorrelation effect when integrating backscattered contributions within the final resolution cell

5.2.3 Results

Next paragraphs present some results of full signal simulation (including phase) using the proposed extended scatterer approach. Such signals have been used as inputs feeding two typical sonar processing relying on phase information: interferometry and beamforming.

5.2.3.1 Computing phase differences for interferometry

Figure 19 presents simulated interferometric signals using the extended single scatterers technique. Figure 19(a) is obtained using 3 extended single scatterers, for each resolution cell located at end of the swath. Figure 19(b) presents the same simulation with 40 extended single scatterers, for the same conditions.



(a) 3 single scatterers

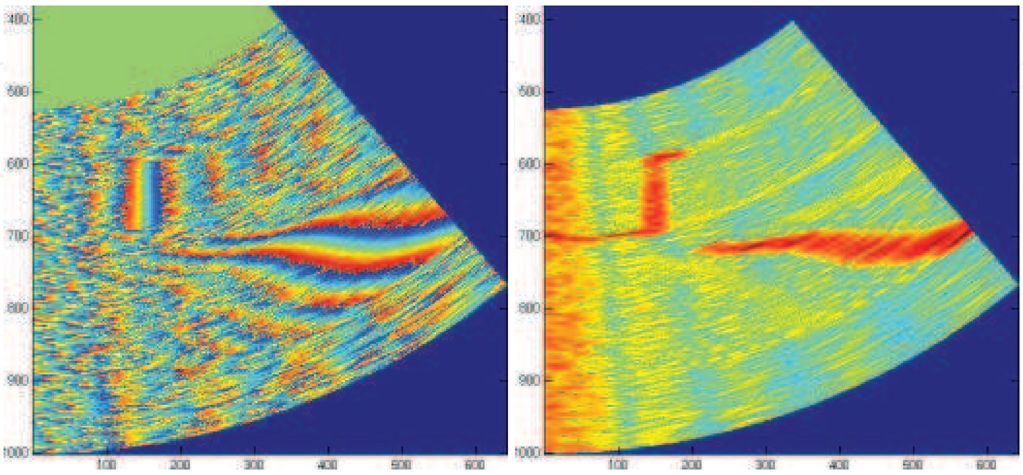
(b) 40 single scatterers

Fig. 19. Simulated interferometric signal using tube tracing engine on a flat sea-bottom

The interferometric signal computed from the simulated full signal for two sensors observing the same sea-bottom clearly indicates the proposed approach embedding extended single scatterers succeeds.

5.2.3.2 Application to beamforming

The extended scatterer approach can also be used prior to beamforming. Figure 20 shows simulated raw signal for a multibeam echosounder consisting of 60 sensors uniformly spaced on a horizontal line. The frequency carrier f is 300 kHz and the transmitted signal is a pulsed wave with $f/\Delta f = 10$. The simulated multibeam echosounder observes a flat sea-bottom with a cube located in the middle of the swath. Both amplitude and phase results are very realistic: for instance, on the simulated phase image, interferometric fringes are clearly observed around the object.



(a) Phase image (b) Amplitude image

Fig. 20. Simulated full signal (both phase and amplitude) using tube tracing engine for a multibeam echosounder observing a small object lying on flat sea-bottom

5.2.4 Discussion

The proposed approach clearly allows the simulation of realistic full sonar signal. However, two features still remain to be studied: what is the best strategy for representing a sea-bottom with unique extended scatterers? what is the influence of extended scatterers distribution in the resolution cell response properties.

It has been shown that the sea-bottom needs to be divided into extended scatterers with unique identifications (like beacons). A global strategy of splitting is required in order for simulated signals over same sea-bottoms to have meaningful correlations. This strategy is independent of sensor trajectory; however, the resolution of the splitting depends on the wavelength of the signal to be simulated. Indeed, at least three extended scatterers are required within a sensor resolution cell, to simulate a realistic signal.

The second question deals with the simulation signal statistics evolution in relation to the amount of extended scatterers per resolution cell. To estimate this impact on a final interferometric signal, a simulation has been performed, considering a constant delay between the two sensors with an amount of extended scatterers varying from 3 to 100. The experience shows the amount of extended scatterers roughly does not affect globally the simulated amplitude. However, having more extended scatterers greatly reduces the variance of the simulated signals from 3 to 20 scatterers per resolution cell. After 20, the variance remains almost stable. This simple analysis clearly shows that to simulate specific random response of a sea-bottom, the statistics nature, distribution and amount of extended scatterers gathered within a resolution cell should be carefully taken into account.

5.3 Optimizing signal simulation through a Brownian sheet technique

As previously mentioned, the extended scatterers approach has two main drawbacks. The first one concerns the statistical properties of the generated signals because they do not correspond to actual phase statistics (Tough et al., 1995). The second drawback deals with the identity assigned to each extended scatterer: although assigning along simulation, a random phase to each single extended scatterer grants good interferences (Sintes & Guériot, 2008), this identification strategy is not robust enough to be used from track to track or survey to survey. Thus, an extended scatterer identity (or individual phase) clearly needs to be linked with the scatterer geographic location in order for several overlapping tracks to correlate when observing the same scatterers, the level of correlation depending on the amount of common observed scatterers. For this two-dimensional problem, the idea is to create a map where each position (or pixel) is a random variable, holding phase statistics that allows such correlation properties.

To produce such a georeferenced map, different techniques may be implemented such as Perlin (gradient) noise (Perlin, 2002) widely used in computer texture graphic technology, or fractional Brownian motion (Pesquet-Popescu & Vehel, 2002) bringing interesting Gaussian properties. Perlin noise main advantage is to compute the value of a given pixel independently of its neighborhood; however, the generated signal has no real statistical sense. Conversely, a Brownian-based solution does permit the generation of uncorrelated Gaussian signals from which it is possible to derive various kinds of statistics.

Thus, the next section discusses the proposed Brownian-based solution and how it has been adapted to generate both amplitude and phase information. Interferometry is then used to check the correctness of such signal simulations.

5.3.1 Assigning specific phase to extended scatterers using Brownian-based model

The effort described in this paragraph consists in assigning each extended scatterer a random phase with specific properties, instead of using a uniform random phase.

Starting from Equation 12, the sea-bottom contribution $z(k+1)$ to the $k+1$ resolution cell can be written in a more general way. Equation 15 proposes such an expression and introduces the function $g(u)$ that gives the random signal associated to any extended scatterer u.

$$z(k+1) = \int_{t_1}^{t_2} g(u) \cdot e^{-j2\pi f u} du \tag{15}$$

Using Equation 15 for a system supporting 2 antennas (#1 and #2), the sea-bottom contribution $z_1(k+1)$ (resp. $z_2(k+1)$) to sensor #1 (resp. sensor #2) $k+1$ resolution cell can be written for each receiver as:

$$\begin{aligned} z_1(k+1) &= H(t_b) - H(t_a) \\ z_2(k+1) &= H(t_d) - H(t_c) \end{aligned} \tag{16}$$

where $[t_a, t_b]$ (resp. $[t_c, t_d]$) defines temporal boundaries of the $k+1$ resolution cell from receiver #2 (resp. receiver #1) as illustrated in Figure 21 and $H(t_x)$ denotes the antiderivative or cumulative function at t_x, of the integrand in Equation 15.

$$H(t_x) = \int_{-\infty}^{t_x} g(u) \cdot e^{-j2\pi f u} du \tag{17}$$

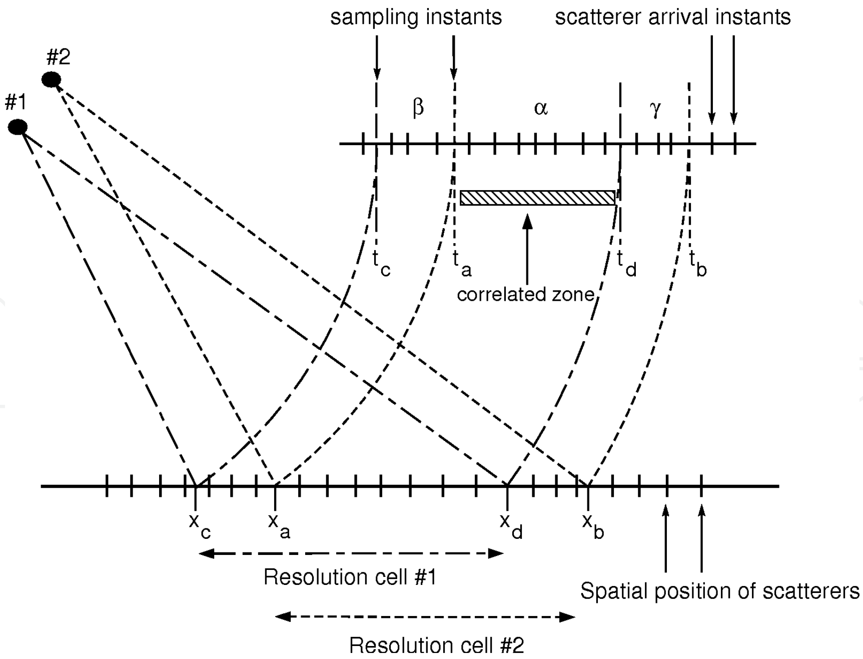


Fig. 21. Splitting integration range due to decorrelation between receivers.

In order to model $H(x)$, let us evaluate the cross-correlation function between the received signals:

$$z_1(k+1).z_2^*(k+1) = [H(t_d) - H(t_c)][H(t_b) - H(t_a)] \tag{18}$$

$$= [\underbrace{H(t_d) - H(t_a)}_{\alpha} + \underbrace{H(t_a) - H(t_c)}_{\beta}].[\underbrace{H(t_b) - H(t_d)}_{\gamma} + \underbrace{H(t_d) - H(t_a)}_{\alpha}]$$

$$= \alpha^2 + \alpha\gamma + \beta\alpha + \beta\gamma \tag{19}$$

with α, β and γ being defined in Figure 21 as the increments between antiderivatives. Thus, α gives the correlated contribution while β and γ represents the uncorrelated contributions between the two resolution cells.

Equation 19 permits the modeling of the received signals. Indeed, in order to ensure both a correct cross-correlation between signals and independent increments, Equations 20 must be checked.

$$\mathbb{E}\{z_1(k+1).z_2^*(k+1)\} = \mathbb{E}\{\alpha^2\} \Longrightarrow \mathbb{E}\{\alpha\gamma\} = \mathbb{E}\{\alpha\beta\} = \mathbb{E}\{\beta\gamma\} = 0 \tag{20}$$

Now, assuming the signal $g(u)$ associated to extended scatterers, is a white Gaussian-distributed noise, its integration in Equation 17 leads the antiderivatives $H(t_x)$ to be a Wiener process also known as Brownian motion (Saupe, 1988). As a result, the increments given in Equation 16 are also Gaussian distributed, producing Gaussian signals at reception.

5.3.2 Brownian sheet simulation

A 1-D Brownian motion can be simulated by creating a zero-mean Gaussian random vector and computing the cumulative sum function. In order to check the properties of cross-correlation between resolution cells, a 1024-length Brownian vector composed of 1024 observations was created. Then, the cross-correlation function was computed between two ten-units length resolution cells. In the case the resolution cells were 3 units apart (resp. 6), the resulting correlations were 0.7 (resp. 0.4) on average, as expected.

For a two-dimensional case, the simulation of a Brownian sheet is carried out by creating a zero-mean Gaussian random matrix, applying the cumulative sum function along one axis direction and then, applying this function again to the result along the other axis direction. As in the previous case, the measured cross-correlations behave as expected.

The main drawback of this model is that computing every pixel of the Brownian sheet requires the whole map to be processed. Moreover, as already said, as the map resolution depends on sonar characteristics (signal wavelength, baselines...), this map may become huge. Thus, one solution is to find a way for computing Equation 15 without requiring each time, a full map processing. Using a series expansion of the integral in Equation 15, it has been possible to simulate coherent signals, bypassing such a costly processing (Llort-Pujol et al., 2010).

5.3.3 Results

Using the Brownian optimization approach (and the proposed extended scatterers), Figure 22 presents simulated signals for a two-sensors interferometer along with the produced interferogram. Fringes of interference are correctly produced showing adequate correlations properties have been taken into account when simulating the two signals over the sea-bottom.

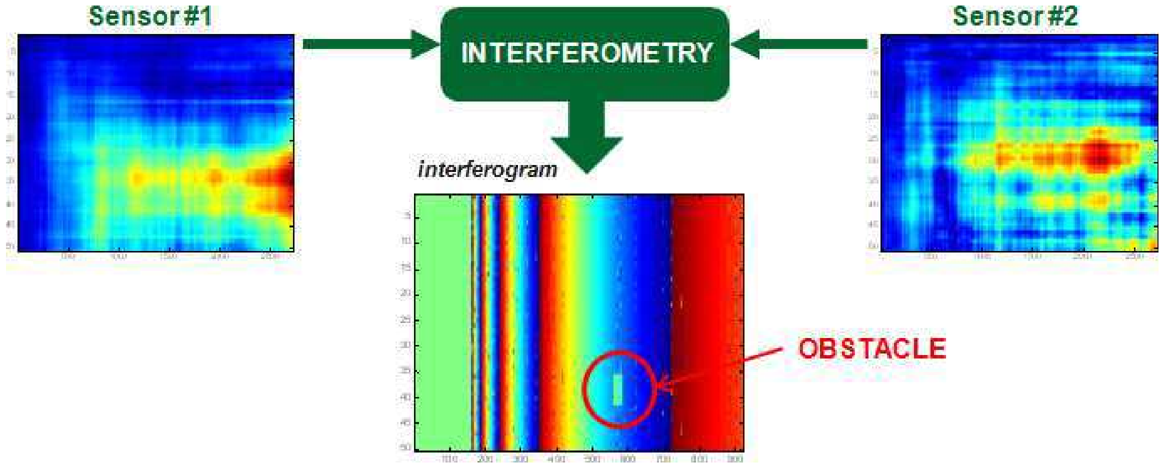


Fig. 22. Simulating sonar signals coming from two different tracks and resulting interferogram.

Nevertheless, the georeferenced grid size remains a major drawback for this method, when computing (once) the Brownian sheet. Indeed, for instance, to generate a 100 meters by 100 meters map for simulating a sidescan sonar which transmits 0.15 ms duration signals, the simulator engine should store the position of about seven million scatterers in order for each resolution cell to encompass only three scatterers. Generating complex signals from the resulting Brownian sheet thus requires a huge amount of both memory and pre-computation processing power.

A more attractive approach, compatible with polygonal ray tubes footprint, would consist in using extensions of the Brownian bridge (Khoshnevisan & Xiao, 2007) to enable on-demand refinement of an already simulated rough grid. The idea is to develop techniques close to Perlin noise generation which can deliver a random number with appropriate properties from its 2D coordinates only.

6. Conclusion

Using tubes for modeling acoustic waves underwater propagation allows the framework we presented in this chapter, to simulate very realistic sonar images. Indeed, the whole energy transmitted by one sensor, can be tracked down through scene interactions, till its aggregation in the receiving sensor. Moreover, this energy-based method has been successfully enhanced to simulate full sonar signal (intensity + phase) developing a coherent aggregation processing relying on the tubes propagation. Indeed, using a sea-bottom representation through extended single scatterers, a coherent summation of their backscattered contributions has allowed a complete backscattered signal to be simulated. The realism of such simulated signals have been proven as beamforming and interferometry have successfully worked on these signals.

Some improvements remains to be integrated within this simulation framework such as more complex signals transmission, Doppler effect checking and diffraction propagation through the tube history. To achieve these new goals, other methods (modes simulation or FDTD techniques) mostly based on analytical calculations, are required to finely model both propagation and interaction of the acoustic wave with the observed scene (sea-bottom, objects...). However, their computational burden is very high and to produce realistic output within an acceptable computation time, they should be used only locally in the scene where specific and impacting phenomena (like diffraction) occur. Such an hybrid behavior has already been successfully implemented for radar simulation, outside this simulator (Lurton, 2010). While electromagnetic wave propagation is performed through rays, interactions with the sea surface rely on FDTD computation. Next developments around this simulator will then consist in integrating this promising hybrid scheme, as the current framework already support multiple engines.

7. References

Asada, A., Maeda, F., Kuramoto, K., Kawashima, Y., Nanri, M. & Hantani, K. (2007). Advanced surveillance technology for underwater security sonar systems, *OCEANS 2007 - Europe* pp. 1–5.

Australia Defense Science & Technology Organization, D. (1999). Common data set, *Shallow Survey 99 - International Conference on High Resolution Surveys in Shallow Water*, Sydney, Australia.

Belcher, E., Hanot, W. & Burch, J. (2002). Dual-frequency identification sonar DIDSON, *Underwater Technology, 2002. Proceedings of the 2002 International Symposium on* pp. 187–192.

Belcher, E., Matsuyama, B. & Trimble, G. (2001). Object identification with acoustic lenses, *OCEANS, 2001. MTS/IEEE Conference and Exhibition* 1: 6–11 vol.1.

Bell, J. & Linnett, L. (1997). Simulation and analysis of synthetic sidescan sonar images, *IEE Proceedings—Radar, Sonar and Navigation* 144(4): 219–226.

Bell, J. M. (1997). Application of optical ray tracing techniques to the simulation of sonar images, *Optical Engineering* 36(6): 1806–1813.

Brekhovskikh, L. & Lysanov, Y. (1982). *Fundamentals of ocean acoustics*, Springer-Verlag Heidelberg, New York.

Burdic, W. S. (1984). *Underwater Sound System Analysis*, Prentice-Hall, Englewood Cliff.

Denbigh, P. N. (1989). Swath bathymetry: Principles of operation and an analysis of errors, *IEEE Journal of Oceanic Engineering* 14(4): 289–298.

Guériot, D., Daniel, S. & Maillard, E. P. (1998). Sonar images and swath bathymetry simulations for environment restitution through acoustical ray tracing techniques, *Fourth European Conference on Underwater Acoustics*, Vol. 1, Rome, Italy, pp. 425–431.

Guillaudeux, S. & Maillard, E. (1998). Seabed simulations with fractal techniques, *OCEANS '98 Conference Proceedings*, Vol. 1, pp. 433 –437 vol.1.

Harrison, C. (1989). Ocean propagation models, *Applied Acoustics* 27(3): 163 – 201.

Jackson, D. R., Winebrenner, D. P. & Ishimaru, A. (1986). Application of the composite roughness model to high-frequency bottom backscattering, *The Journal of the Acoustical Society of America* 79(5): 1410–1422.

Khoshnevisan, D. & Xiao, Y. (2007). Images of the brownian sheet, *Trans. Amer. Math. Soc.* 359: 3125–3151.

Kuo, E. Y. T. (1964). Wave scattering and transmission at irregular surfaces, *The Journal of the Acoustical Society of America* 36(11): 2135–2142.

Llort-Pujol, G., Sintes, C., Chonavel, T., Gueriot, D. & Garello, R. (2010). Simulation on large scale of acoustic signals for array processing, *OCEANS 2010 IEEE - Sydney*, pp. 1 –7.

Lurton, T. (2010). *Modélisation et simulation de l'interaction entre onde électromagnétique et surfaces de mer*, PhD thesis, Dépt. ITI (Télécom Bretagne), France.

Lurton, X. (2003). Theoretical modelling of acoustical measurement accuracy for swath bathymetric sonars, *International Hydrographic Review* 4(2): 17–30.

Mourad, P.D.; Jackson, D. (1989). High frequency sonar equation models for bottom backscatter and forward loss, *OCEANS '89. Proceedings* 4: 1168–1175.

Pentland, A. P. (1984). Fractal-based description of natural scenes, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* PAMI-6(6): 661 –674.

Perkins, J. S. & Baer, R. N. (1982). An approximation to the three-dimensional parabolic-equation method for acoustic propagation, *The Journal of the Acoustical Society of America* 72(2): 515–522.

Perlin, K. (2002). Improving noise, *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, pp. 681–682.

Pesquet-Popescu, B. & Vehel, J. L. (2002). Stochastic fractal models for image processing, *IEEE Signal Processing Magazine* 19(5): 48–62.

Saupe, D. (1988). *Algorithms for random fractals*, The science of fractal images, Springer-Verlag, New York, chapter 2, pp. 71–136.

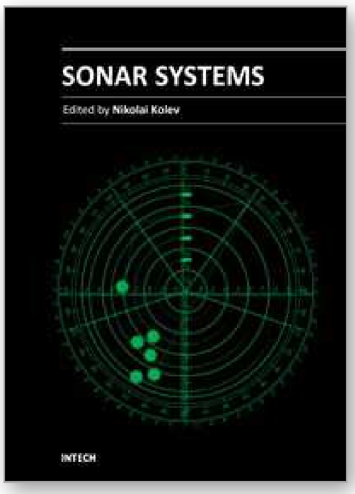
Sintes, C. & Guériot, D. (2008). Interferometric sonar signal simulation through tube tracing, *Oceans 2008 MTS/IEEE Conf. Proc.*, Kobe, Japan, pp. 1–8.

Sintes, C., Llort-Pujol, G. & Guériot, D. (2010). Coherent probabilistic error model for interferometric sidescan sonars, *Oceanic Engineering, IEEE Journal of* 35(2): 412–423.

Tough, R. J. A., Blacknell, D. & Quegan, S. (1995). A statistical description of polarimetric and interferometric synthetic aperture radar data, *Proc. Royal Society* 449: 567–589.

Urick, R. J. (ed.) (1983). *Principles of underwater sound*, third edn, Peninsula Publishing, Los Altos, California.

Ziomek, L. J. & Polnicky, F. W. (1993). The RRA algorithm: Recursive ray acoustics for three-dimensional speeds of sound, *IEEE Journal of Oceanic Engineering* 18(1): 25–30.



Sonar Systems
Edited by Prof. Nikolai Kolev

ISBN 978-953-307-345-3
Hard cover, 322 pages
Publisher InTech
Published online 12, September, 2011
Published in print edition September, 2011

The book is an edited collection of research articles covering the current state of sonar systems, the signal processing methods and their applications prepared by experts in the field. The first section is dedicated to the theory and applications of innovative synthetic aperture, interferometric, multistatic sonars and modeling and simulation. Special section in the book is dedicated to sonar signal processing methods covering: passive sonar array beamforming, direction of arrival estimation, signal detection and classification using DEMON and LOFAR principles, adaptive matched field signal processing. The image processing techniques include: image denoising, detection and classification of artificial mine like objects and application of hidden Markov model and artificial neural networks for signal classification. The biology applications include the analysis of biosonar capabilities and underwater sound influence on human hearing. The marine science applications include fish species target strength modeling, identification and discrimination from bottom scattering and pelagic biomass neural network estimation methods. Marine geology has place in the book with geomorphological parameters estimation from side scan sonar images. The book will be interesting not only for specialists in the area but also for readers as a guide in sonar systems principles of operation, signal processing methods and marine applications.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821