

A novel GPU-based sonar simulator for real-time applications

Abstract

Mainly when applied in the underwater environment, sonar simulation requires great computational effort due to the complexity of acoustic physics. For that, simulation of sonar operation allows to evaluate algorithms and control systems without going to the real underwater environment; that reduces the costs and risks of in-field experiments. This paper tackles with the problem of real-time underwater imaging sonar simulation, by using the OpenGL shading language chain on graphics processing unit. This proposed system is able to simulate two main types of sonar sensors: mechanical scanning imaging sonars and forward-looking sonars. The underwater scenario simulation is performed based on three frameworks: (i) OpenSceneGraph reproduces the ocean visual effects, (ii) Gazebo deals with physical forces, and (iii) the Robot Construction Kit controls the sonar in underwater environments. Our system exploits the rasterization pipeline in order to simulate the sonar devices, which are parameterized with the echo intensity, the distance to the target and the angular distortion, being all calculated over objects shapes in the 3D rendered scene. Sonar-intrinsic speckle noise and object material properties are also considered as part of the acoustic image. Our evaluation demonstrated that the proposed method is able to operate close or faster than the real-world devices' frame rate, as well as generating realistic sonar image quality in different virtual underwater scenarios.

Key words: Simulated sensor data, Sonar imaging, GPU-based processing, Robot Construction Kit (Rock), Underwater robotics.

1. Introduction

2 Simulation is an useful tool for designing and programming
3 autonomous robot systems. That allows evaluating robot behav-
4 ior, without dealing with physical hardware or decision-making
5 algorithms and control systems in real-time trials, as well as
6 costly and time-consuming field experiments.

7 When working with autonomous underwater vehicles (AUVs),
8 simulation of facilities are specially relevant. AUVs usually de-
9 mand expensive hardware and perform long-term data gather-
10 ing operations, taking place in restrictive sites. As AUV does
11 not need umbilical cable, and the underwater communication
12 carries on by unreliable acoustic links, the robot should be able
13 to make completely autonomous decisions, even with low-to-
14 zero external assistance. While the analysis and interpretation
15 of sensor data can be performed in a post-processing step, a
16 real-time simulation is strongly needed for testing and evalua-
17 tion of vehicle's motion response, avoiding involved risks on
18 real world rides.

19 AUVs usually act below the photic zone, with high turbid-
20 ity and huge light scattering. This makes the quality of image
21 acquisition by optical devices limited by a short range, and also
22 artificially illuminated and clear visibility conditions. To tackle
23 with that limitations, high-frequency sonars have been used pri-
24 marily on AUVs' navigation and perception systems. Acoustic
25 waves emitted by sonars are significantly less affected by water
26 attenuation, aiding operation at greater ranges even as low-to-
27 zero visibility conditions, with a fast refresh rate. Although
28 sonar devices usually solve the main shortcomings of optical
29 sensors, they provide noisy data of lower resolution and more
30 difficult interpretation.

31 By considering sonar benefits and singularities along with

32 the need to evaluate AUVs, recent works proposed ray tracing-
33 [1, 2, 3, 4, 5, 6] and tube tracing-based [7] techniques to sim-
34 ulate acoustic data with very accurate results, although having
35 a high computational cost. Bell [1] proposed a simulator based
36 on optical ray tracing for underwater side-scan sonar imagery;
37 images were generated by acoustic signals represented by rays,
38 which are repeatedly processed, forming a 2D-array. Coiras and
39 Groen [2] used frequency-domain signal processing to produce
40 synthetic aperture sonar frames; in that method, the acoustic
41 image was created by computing the Fourier transform of the
42 acoustic pulse used to insonify the scene. For forward-looking
43 sonar simulations, Saç *et al.* [3] described a sonar model by
44 computing the ray tracing in frequency domain; when a ray
45 hits an object in 3D space, three parameters are calculated to
46 process the acoustic data: the Euclidean distance from the sonar
47 axis, the intensity of returned signal by Lambert Illumination
48 model and the surface normal; the reverberation and shadow
49 phenomena are also considered in the scene rendering. De-
50 Marco *et al.* [4] used Gazebo and Robot Operating System
51 (ROS) [8] integration to simulate acoustic sound pulses by ray
52 tracing technique, also producing a 3D point cloud of the cov-
53 erage area; the reflected intensity has taken into account the
54 object reflectivity, and the amount of Gaussian and salt-and-
55 pepper noises applied in the sonar image is empirically defined.
56 Gu *et al* [5] modeled a forward-looking device, where the ultra-
57 sound beams were formed by a set of rays; the acoustic image is
58 significantly limited by its representation using only two colors:
59 white, when the ray strikes an object, and black for shadow ar-
60 eas. Kwak *et al.* [6] improved the previous approach by adding
61 a sound pressure attenuation to produce the gray-scale sonar
62 frame, while the other physical characteristics related to sound
63 transmission are disregarded. Guériot and Sintes [7] introduce

64 a volume-based approach of energy interacting with the scene,
 65 and collected by the receiving sonar; the sound propagation is
 66 defined by series of acoustic tubes, being always orthogonal to
 67 the current sonar view, where the reverberation and objects sur-
 68 face irregularities are also addressed.

69 1.1. Contributions

70 This paper introduces a novel imaging sonar simulator that
 71 presents some contributions when compared to the existing ap-
 72 proaches. Instead of simulating the sound pulse paths and the
 73 effects of their hits with the virtual objects, as presented by ray
 74 tracing and tube tracing-based methods [1, 2, 3, 4, 5, 6, 7], we
 75 take advantage of precomputed geometric data during the ras-
 76 terization pipeline to compute the acoustic frame. In addition,
 77 all raster data are handled on GPU, accelerating then the sim-
 78 ulation process with guarantee of real-time response, in con-
 79 trast to the methods found in [1, 2, 3, 4]. Although the mod-
 80 els found in [1, 2, 3, 4, 5, 6, 7] focused on the simulation of
 81 specific sonar device, our model is able to reproduce two kind
 82 of sonar devices: mechanical scanning imaging sonar (MSIS)
 83 and forward-looking sonar (FLS). The intensity measured back
 84 from the insonified objects depends on surface normal direc-
 85 tions and reflectivity, producing more realistic simulated frames
 86 than binary representation as found in [5, 6]. The speckle noise
 87 is modeled as a non-uniform Gaussian distribution and applied
 88 to the final sonar image, next to real sonar operation, differently
 89 to [3, 4, 5, 6, 7]. On the other hand, the additive noise is con-
 90 sidered by the authors in [3, 4]. Finally, we have decided to
 91 extend the physical phenomena in the simulator, obeying the
 92 use in real-time experiments (e.g. decision-making algorithms
 93 and control system tuning). Knowing this real-time constraint,
 94 the high computational cost phenomena such as reverberation
 95 is not included at this point, differently of [3].

96 The main goal here is to build quality and low time-con-
 97 suming acoustic frames, according to underwater sonar image
 98 formation and operation modes (see Section 2). The shader ma-
 99 trix with depth, intensity and angular distortion parameters are
 100 extracted from underwater scene during the rasterization pipe-
 101 line, and subsequently fused to generate the simulated sonar
 102 data, as described in Section 3. Qualitative and time evalua-
 103 tion results for two different sonar devices are presented in Section
 104 4, allowing the use of the proposed simulator by real-time ap-
 105 plications.

106 2. Imaging sonar operation

107 Sonars are echo-ranging devices that use acoustic energy to
 108 locate and survey objects in a desired area. The sonar trans-
 109 ducer emits pulses of sound waves (or ping) until they hit any
 110 object or be completely absorbed. When the acoustic signal
 111 collides with a surface, part of this energy is reflected, while
 112 other is refracted. The sonar data is built by plotting the echo
 113 measured back versus time of acoustic signal. The transducer
 114 reading in a given direction forms a *beam*. A single beam trans-
 115 mitted from a sonar is illustrated in Fig. 1. The horizontal and
 116 vertical beamwidths are represented by the azimuth ψ and el-
 117 evation θ angles, respectively, where each sampling along the

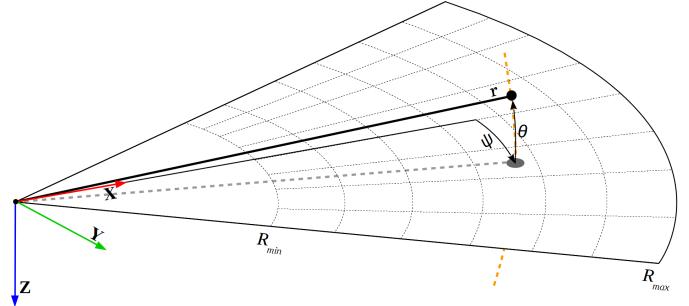


Figure 1: Imaging sonar geometry. By a projection process, all 3D points belonging to the same elevation arc (represented as dashed orange line) will be represented to the same image point in the 2D plane. Range r and azimuth angle ψ are measured, and elevation angle θ is lost. Sonar coverage area is defined by R_{min} and R_{max} .

118 beam is named as *bin*. The sonar coverage area is defined by
 119 R_{min} and R_{max} . Since the speed of sound underwater is known,
 120 or can be measured, the time delay between the emitted pulses
 121 and their echoes reveals how far the objects are (distance r), as
 122 well as how fast they are moving. The backscattered acoustic
 123 power in each bin determines the intensity value.

124 With different azimuth directions, the array of transducer
 125 readings forms the final sonar image. Since all incoming sig-
 126 nals converge to the same point, the reflected echoes could have
 127 been originated anywhere along the corresponding elevation arc
 128 at a fixed range, as depicted in Fig. 1. In the acoustic represen-
 129 tation, the 3D information is lost in the projection into a 2D
 130 image.

131 2.1. Sonar characteristics

132 Although sonar devices overcome main limitations of opti-
 133 cal sensors, they present more difficult data interpretation due
 134 to:

- 135 a) **Shadowing:** This effect is caused by objects blocking the
 136 sound waves transmission and causing regions behind them,
 137 without acoustic feedback. These regions are defined by a
 138 black spot in the resulting sonar image;
- 139 b) **Non-uniform resolution:** The amount of pixels used to
 140 represent an intensity record in the Cartesian coordinate
 141 system grows as its range increases. This fact causes im-
 142 age distortions and object flatness;
- 143 c) **Changes in viewpoint:** Imaging the same scene from dif-
 144 ferent viewpoints can cause occlusions, shadows move-
 145 ments and significant changes of observable objects [9].
 146 For instance, when an outstanding object is insonified, its
 147 shadow is shorter, as the sonar becomes closer;
- 148 d) **Low signal-to-noise ratio (SNR):** sonars suffer from low
 149 SNR mainly due the very-long-range scanning, and the
 150 presence of speckle noise introduced by acoustic wave in-
 151 terferences [10];
- 152 e) **Reverberation:** This phenomenon is caused when mul-
 153 tiple acoustic waves, returning from the same object, are
 154 detected over the same ping, producing duplicated objects.

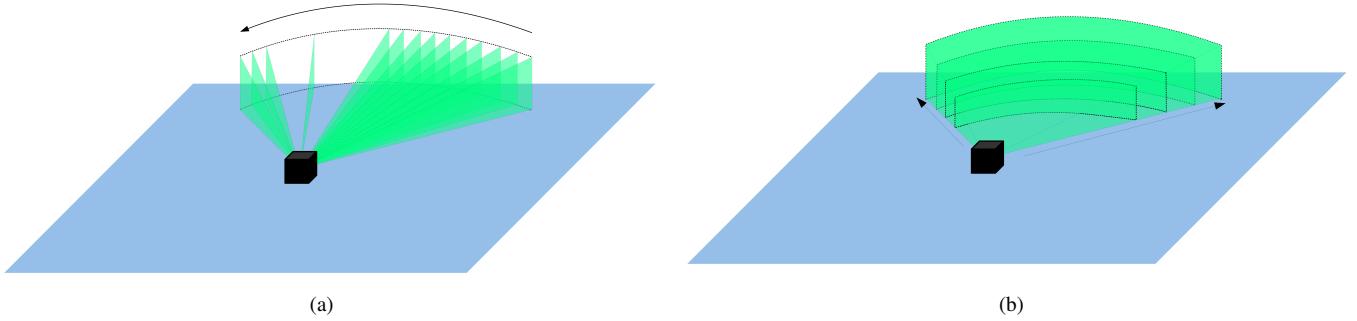


Figure 2: Different underwater sonar readings: (a) From a mechanical scanning imaging sonar and (b) from a forward-looking sonar.

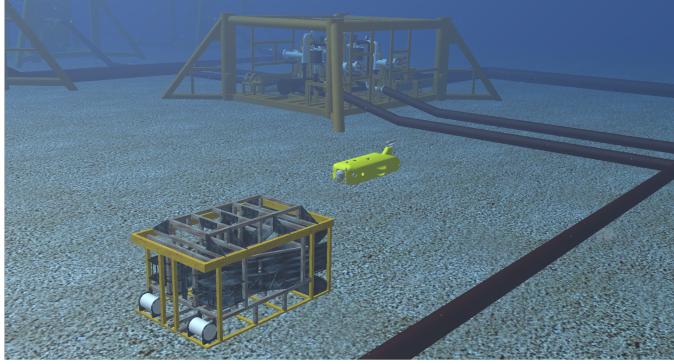


Figure 3: The AUV in Rock-Gazebo underwater scene.

155 2.2. Types of underwater sonar devices

156 The most common types of underwater acoustic sonars are
157 MSIS and FLS. In the former, the sonar image is built for each
158 pulse, with one beam per reading (see Fig. 2(a)); the resulting
159 sonar images in MSIS are usually depicted on a display pulse by
160 pulse, and the head position reader is rotated according to mo-
161 tor step angle. After a full 360° sector reading (or the desired
162 sector defined by left and right limit angles), the accumulated
163 sonar data is overwritten. The acquisition of a scanning image
164 involves a relatively long time, introducing distortions caused
165 by the vehicle movements. This sonar device is generally ap-
166 plied in obstacle avoidance [11] and navigation [12] applica-
167 tions. As illustrated in Fig. 2(b), the whole forward view of
168 an FLS is scanned and the current data is overwritten by the
169 next scanning in a high frame rate, with all beams being read
170 simultaneously; this is similar to a streaming video imagery for
171 real-time applications; this imaging sonar is commonly used
172 for navigation [13], mosaicing [9], target tracking [14] and 3D
173 reconstruction [15].

174 3. GPU-based sonar simulation

175 The goal of our work is to simulate the two types of under-
176 water sonar, discussed in Section 2.2, by vertex and fragment
177 processing and low computational cost. The complete pipeline
178 of the proposed simulator (from the virtual scene to the simu-
179 lated acoustic data) is detailed in the following sections. The

180 sonar simulator is written in C++ with OpenCV [16] support as
181 Rock packages.

182 3.1. Rendering underwater scene

183 In Rock-Gazebo framework [17], Gazebo handles with phys-
184 ical forces, while Rock's visualization tools are responsible by
185 the scene rendering. The graphical data in Rock are based
186 on OpenSceneGraph framework, an open source C/C++ 3D
187 graphics toolkit built on OpenGL. The osgOcean library is used
188 to simulate the ocean visual effects. In our case, Rock-Gazebo
189 integration provides the underwater scenario, allowing also real-
190 time hardware-in-the-loop simulation with a virtual AUV.

191 All scene aspects, such as world model, robot parts (in-
192 cluding sensors and joints) and other virtual objects are defined
193 by simulation description files (SDF), which use the SDFFor-
194 mat [18], a XML format used to describe simulated models and
195 environments for Gazebo. Visual and collision geometries of
196 vehicle and sensor robot are also described in specific file for-
197 mats. Each component described in the SDF file becomes a
198 Rock component, which is based on the Orocó real-time tool-
199 kit (RTT) [19], providing I/O ports, properties and operations
200 as communication layers. When the models are loaded, Rock-
201 Gazebo creates I/O ports to allow real world or simulated sys-
202 tem components interacting with the simulated models. A re-
203 sulting scene sample of this integration is illustrated in Fig. 3.

204 3.2. Sonar rendering

205 The rendering pipeline can be customized by defining GPU
206 shaders. A shader is written in OpenGL Shading Language
207 (GLSL) [20], a high-level language with a C-based syntax which
208 enables more direct control of graphics pipeline, which avoids
209 the use of low-level or hardware-specific languages. Shaders
210 can describe the characteristics of either a vertex or a fragment
211 (a single pixel). Vertex shaders are responsible by transform-
212 ing the vertex position into a screen position by the rasterizer, gen-
213 erating texture coordinates for texturing, and lighting the vertex
214 to determine each color. The rasterization results in a set of pix-
215 els to be processed by fragment shaders, which manipulate their
216 locations, depth and alpha values, and interpolated parameters
217 from the previous stages, such as colors and textures.

218 In our work, the underwater scenes are sampled by a virtual
219 camera (frame-by-frame), whose optical axis is aligned with
220 the intended viewing direction, the coverage range and the

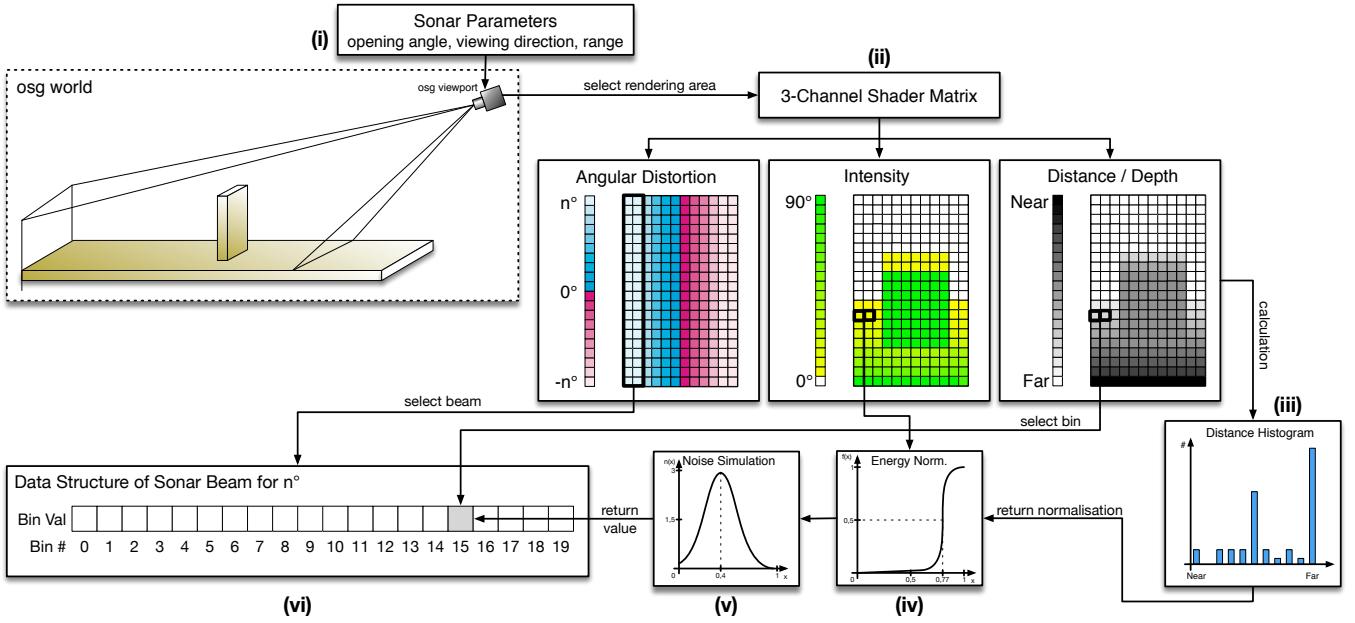


Figure 4: A graphical overview of the imaging sonar simulation process: (i) a virtual camera, specialized as the sonar device, samples the underwater scene; (ii) three parameters are calculated by shader rendering on GPU: **angular distortion**, **intensity** and **depth**; the shader information is split into beam parts, according to the angular distortion values, and the bin depth and the intensity are defined by: (iii) distance histogram and (iv) energy normalization, respectively; (v) the speckle noise is applied to the final sonar data; (vi) and the simulated acoustic data is presented as Rock's data type.

221 **opening angle** of the simulated sonar device (see Fig. 4(i)).
 222 To simulate the sonar imaging by using virtual camera frames,
 223 three parameters are computed in fragment and vertex shaders,
 224 during the rendering pipeline. This way, we are able to use the
 225 precomputed geometric information during the image rasteri-
 226 zation process on GPU. The three parameters to simulate the
 227 sonar using a virtual camera are, as illustrated in Fig. 4(ii):

- 228 • **Distance / Depth** is the camera focal length, calculated
 229 by the Euclidean distance to the object surface point;
- 230 • **Intensity** presents the echo reflection energy based on
 231 object surface normal angle up to the camera;
- 232 • **Angular distortion** is the angle formed from the camera
 233 center column up to the camera boundary column, for
 234 both directions.

235 By default, the shader encodes the raster data in 8-bit color
 236 channels for red, green, blue and alpha (RGBA). We take this
 237 image data structure to store the intensity, depth and angular
 238 distortion parameters in the RGB channels. The intensity pa-
 239 rameter follows the real sonar common representation as 8-
 240 bit values; the depth is replaced by the native GLSL 32-bit
 241 depth buffer to avoid precision limitation during the distance
 242 histogram (see Fig. 4(iii)), described in subsection 3.3; as the
 243 projection angle range on shader is 0° to 90° , for both direc-
 244 tions, the angular distortion is represented by 8-bit values with-
 245 out loss of meaning. Also, all these three parameters in the
 246 shader matrix are normalized into the interval $[0,1]$. For the
 247 intensity parameter, zero means no energy and one means max-
 248 imum echo energy; for depth, the minimum value denotes a
 249 close object, while the maximum value represents a far one,

250 limited by the sonar maximum range; angle distortion is zero in
 251 image center column, and increases for both borders to present
 252 the half value of horizontal field of view.

253 In real world sensing, surfaces usually present irregulari-
 254 ties and different reflectance values. To render that in a virtual
 255 scene, the intensity values can also be defined by **normal maps**
 256 (see Fig. 5) and material property information (see Fig. 6).
 257 **Normal mapping** is a perturbation rendering technique to simu-
 258 late wrinkles on the object surface by passing **textures**, modify-
 259 ing the normal directions on shaders. This approach consumes
 260 less computational resources for the same level of detail, com-
 261 pared with the displacement mapping technique, because the
 262 geometry remains unchanged. Since **normal maps** are built in
 263 tangent space, interpolating the normal vertex and the texture,
 264 tangent, bi-tangent and normal (TBN) matrices are computed to
 265 convert the normal values into the world space. **The differences**

266 by applying **normal mapping** are illustrated in viewing scene
 267 (see Figs. 5(a) and 5(c)), in the shader representation (see Figs.
 268 5(e) and 5(b)) and the final sonar image (see Figs. 5(d) and
 269 5(f)). The reflectance allows to properly describe the intensity
 270 received back from observable objects in shader processing, ac-
 271 cording their material properties. For instance, aluminum has
 272 more reflectivity than wood and plastic. When an object has its
 273 reflectivity property defined, the reflectance value ρ is passed to
 274 the fragment shader and processed on GPU. So the final pixel
 275 intensity represents the product of surface normal angle by the
 276 reflectance value ρ . The reflectance affects the shader represen-
 277 tation (see Figs. 6(a), 6(b), 6(c) and 6(d)) and the final sonar
 278 image (see Figs. 6(e), 6(f), 6(g) and 6(h)). At the end of sonar
 279 rendering step, the shader provides a 3-channel matrix com-
 280 posed by intensity, depth and angular distortion parameters.

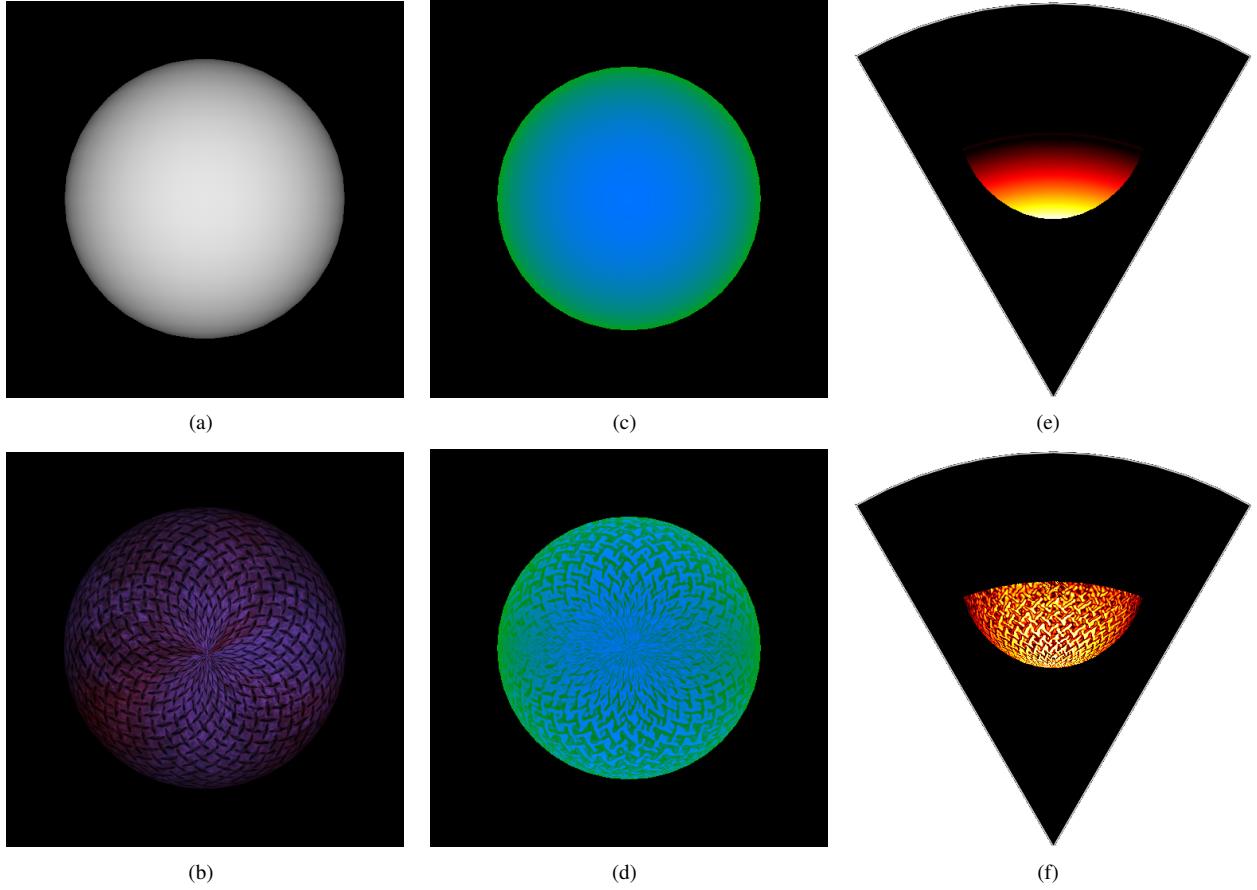


Figure 5: Example of shader rendering with [normal mapping](#): A sphere without (a) and with texture (b); respective shader image representations of the spheres in (c) and (d), where the blue area represents the intensity parameter, while the green area represents the depth parameter. The final acoustic images are depicted in (e) and (f). By using [normal mapping](#) technique, the textures changes the normal directions, and the sonar image details the appearance of object surface, like in real world sensing.

281 *3.3. Simulating sonar device operation*

282 The [3-channel](#) shader matrix is used to compute the corre-
 283 sponding acoustic representation. Since the angular distortion
 284 is radially spaced over the horizontal field-of-view, where all
 285 pixels in the same column have the same angle value, the first
 286 step is to split the image into a number of beam parts. Each
 287 angular distortion column is correlated with a respective beam
 288 vector, according to sonar bearings, as illustrated in Fig. 4(vi).
 289 **One beam represents one or more columns.** Each beamed sub-
 290 image is converted into bin intensities using the depth and in-
 291 tensity [shader matrix parameters](#). In a real imaging sonar, the
 292 echo measured back is sampled over time and the bin number
 293 is proportional to the sensor range. In other words, the initial
 294 bins represent the closest distances, while the latest bins are the
 295 farthest ones. Therefore a distance histogram (see Fig. 4(iii))
 296 is computed in order to group the sub-image pixels with the re-
 297 spective bins, according to the depth [parameter and number of](#)
 298 [bins, and calculate the accumulated intensity of each bin.](#)

299 Due to the acoustic beam spreading and absorption in the
 300 water, the final bins have less echo strength than the first ones,
 301 because the energy is twice lost, in the environment. To tackle
 302 with that issue, sonar devices use an energy normalization based
 303 on time-varying gain for range dependence compensation, which

304 spreads losses in the bins. In our simulation approach, the ac-
 305 cumulated intensity in each bin (see Fig. 4(iv)) is normalized
 306 as

$$307 \quad I_{bin} = \sum_{x=1}^N \frac{1}{N} \times S(i_x), \quad (1)$$

308 where I_{bin} is the intensity in the bin after energy normalization,
 309 x is the pixel location in the shader matrix, N is the distance his-
 310 togram value (number of pixels) of that bin, $S(i_x)$ is a sigmoid
 311 function, and i_x is the intensity value of the pixel x .

312 **Finally, the sonar image resolution must be big enough to**
 313 **contain all information of the bins.** For that, the number of bins
 314 **involved is directly proportional to the sonar image resolution.**

315 *3.4. Noise model*

316 Imaging sonar systems are disturbed by a multiplicative noise
 317 known as speckle, which is caused by coherent processing of
 318 backscattered signals from multiple distributed targets. **This**
 319 **effect degrades** image quality and visual evaluation. Speckle
 320 noise results in constructive and destructive interferences, which

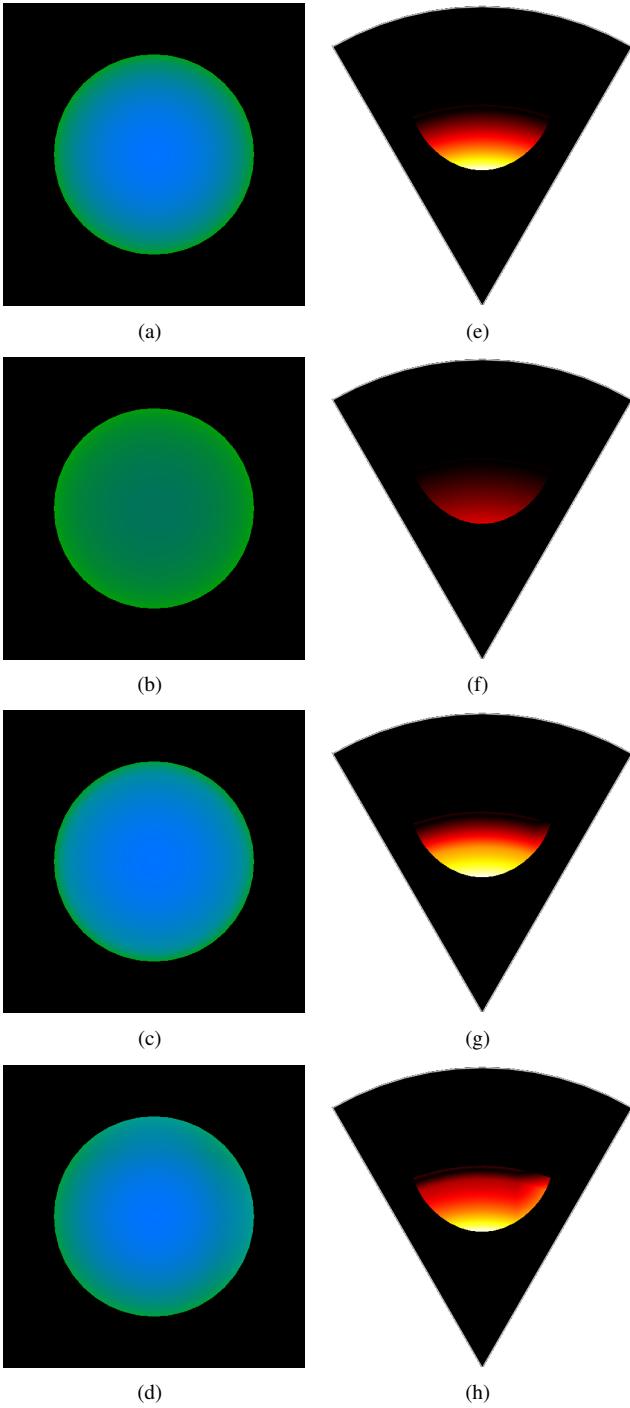


Figure 6: Examples of different reflectance values, ρ , applied in shader image representation of the same target, where blue is the normal channel and green is the depth channel: (a) raw image; (b) $\rho = 0.35$; (c) $\rho = 1.40$; and (d) $\rho = 2.12$. The following acoustic images are presented in (e), (f), (g) and (h).

are shown as bright and dark dots in the image. The noisy image has been expressed as [21]:

$$323 \quad y(t) = x(t) \times n(t), \quad (2)$$

324 where t is the time instant, $y(t)$ is the noised image, $x(t)$ is the
325 free-noise image, $n(t)$ is the speckle noise matrix, and \times symbol
326 defines an element-wise multiplication.

Table 1: Sonar device configurations used on experimental evaluation.

Device	# of beams	# of bins	Field of view	Down tilt	Motor Step
FLS	256	1000	120° x 20°	20°	-
MSIS	1	500	3° x 35°	0°	1.8°

327 This type of noise is well-modeled as a Gaussian distribu-
328 tion (see Fig. 4(v)). The physical explanation is provided by
329 the central limit theorem, which states that the sum of many in-
330 dependent and identically distributed random variables tends to
331 behave as a Gaussian random variable [22]. A Gaussian dis-
332 tribution is defined by following a non-uniform distribution,
333 skewed towards low values, and applied as speckle noise in the
334 simulated sonar image. **This noise simulation is repeated for**
335 **each virtual acoustic frame.**

336 3.5. Integrating sonar device with Rock

337 After the imaging sonar simulation process, from the vir-
338 tual underwater scene to the degraded acoustic representation
339 by noise, the resulting sonar data is encapsulated as Rock's
340 sonar data type (see Fig. 4(vi)) and provided as I/O port of a
341 Rock's component, allowing the interaction with other simu-
342 lated models and applications.

343 4. Simulation results and experimental analysis

344 To evaluate our simulator, experiments were conducted by
345 using a 3D model of an AUV equipped with an MSIS and an
346 FLS. Different scenarios were casted and studied. The sonar
347 device configurations used were summarized in Table 1. In the
348 experimental analysis, as the scene frames are being captured
349 by the sonars, the resulting acoustic images are sequentially
350 presented, on-the-fly (see Figs. 7 and 8).

351 4.1. Experimental evaluation

352 The virtual FLS from AUV was used to insonify the scenes
353 from three distinct scenarios. A docking station, in parallel with
354 a pipeline on the seabed, composes **the first scenario** (see Fig.
355 7(a)); the target surface is well-defined in the simulated acoustic
356 frame (see Fig. 7(d)), even as the shadows and speckle noise;
357 given that the docking station is metal-made, the texture and
358 reflectivity were set, such that a higher intensity shape was re-
359 sulted in comparison with the other targets. **The second sce-**
360 **nario** presents the vehicle in front of a manifold model in a
361 non-uniform seabed (see Fig. 7(b)); the target model was in-
362 sonified to generate the sonar frame from the underwater scene
363 (see Fig. 7(e)); the frontal face of the target, as well the portion
364 of the seabed and the degraded data by noise, are clearly visible
365 in the FLS image; also, a long acoustic shadow is formed be-
366 hind the manifold, occluding part of the scene. **The third sce-**
367 **nario** contains a sub-sea isolation valve (SSIV) structure, con-
368 nected to a pipeline in the bottom (see Fig. 7(c)); the simulated

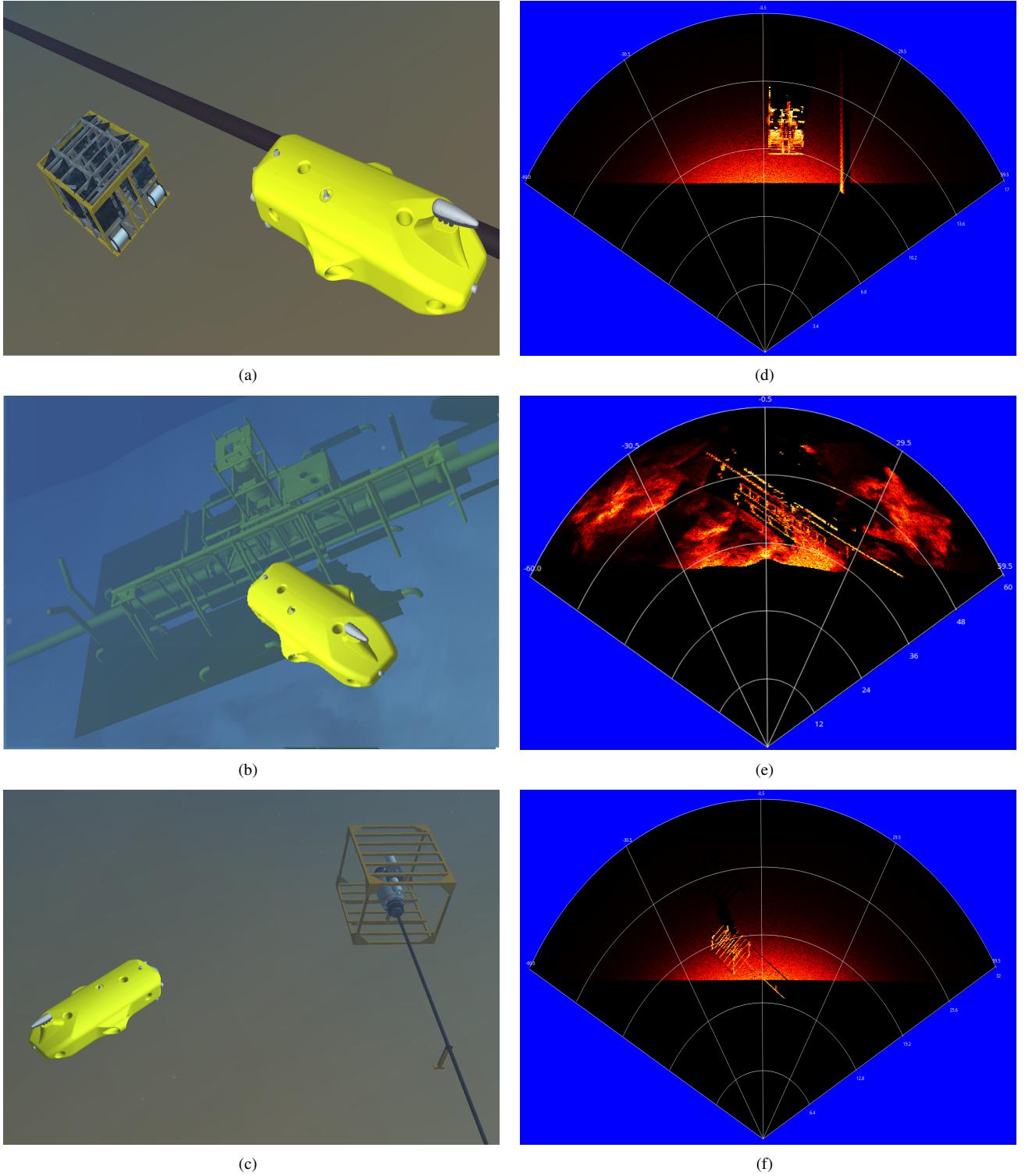


Figure 7: Forward-looking sonar simulation experiments: (a), (b) and (c) present the virtual underwater trials, while (d), (e) and (f) are the following acoustic representations of each scenario, respectively.

369 acoustic image, depicted in Fig. 7(f), also present shadows, ma-
 370 terial properties and speckle noise effects. Due to sensor con-
 371 figuration and robot position, the initial bins usually present a
 372 blind region in the three simulated scenes, caused by absence
 373 of objects at lower ranges, similar to real sonar images. It is

374 noteworthy that the brightness of sea-floor decreases as the sea-
 375 floor is farther from sonar, because of the normal orientation of
 376 the surface.

377 The MSIS was also simulated in three different experiments.
 378 The robot in a big textured tank composes **the first scene** (see

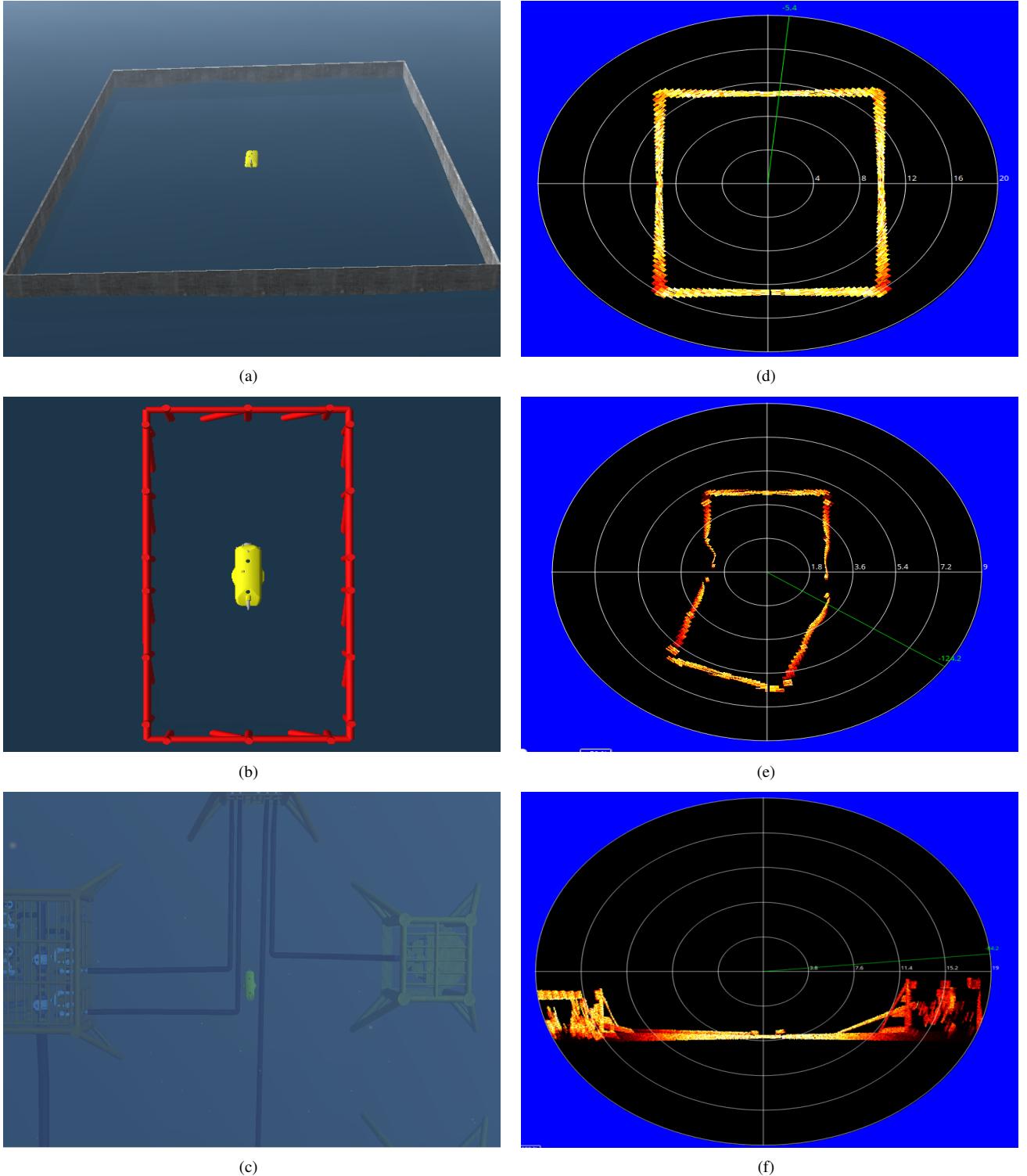


Figure 8: Experiments using mechanical scanning imaging sonar in three different scenarios (a), (b) and (c), and the respective processed simulated frames in horizontal axis in (d) and (e), and vertical axis in (f).

Fig. 8(a)); similar to the first scenario of FLS experiment, the reflectivity and texture were set to the target; the rotation of the sonar head position, by a complete 360° scanning, produced the acoustic frame of tank walls (see Fig. 8(d)). **The second scene** involves the vehicle's movement during the data acqui-

sition process; the scene contains a grid around the AUV (see Fig. 8(b)), captured by a front MSIS mounted horizontally; this trial induces a distortion in the final acoustic frame, because the relative sensor position with respect to the surrounding object changes, as the sonar image is being built (see Fig. 8(e)); in

Table 2: Processing time to generate forward-looking sonar frames with different parameters.

# of samples	# of beams	# of bins	Field of view	Average time (ms)	Std dev (ms)	Frame rate (fps)
500	128	500	120° x 20°	54.7	3.7	18.3
500	128	1000	120° x 20°	72.3	8.9	13.8
500	256	500	120° x 20°	198.7	17.1	5.0
500	256	1000	120° x 20°	218.2	11.9	4.6
500	128	500	90° x 15°	77.4	11.8	12.9
500	128	1000	90° x 15°	94.6	10.2	10.6
500	256	500	90° x 15°	260.8	18.5	3.8
500	256	1000	90° x 15°	268.7	16.7	3.7

Table 3: Processing time to generate mechanical scanning imaging sonar samples with different parameters.

# of samples	# of bins	Field of view	Average time (ms)	Std dev (ms)	Frame rate (fps)
500	500	3° x 35°	8.8	0.7	113.4
500	1000	3° x 35°	34.5	1.6	29.0
500	500	2° x 20°	10.3	0.6	96.7
500	1000	2° x 20°	41.7	3.7	24.0

389 this case, the robot rotates 20° left during the scanning. **The**
390 **last scene** presents the AUV over oil and gas structures on the
391 sea bottom (see Fig. 8(c)); using an MSIS located in the back
392 of the AUV with a vertical orientation, the scene was scanned
393 to produce the acoustic visualization; as illustrated in Fig. 8(f),
394 object surfaces present clear definition in the slice scanning of
395 the sea-floor.

396 All the experimental scenarios provided enough variability
397 of specific phenomena usually found in real sonar images, such
398 as acoustic shadows, noise interference, surface irregularities
399 and properties, distortion during the acquisition process and
400 variance of acoustic intensities. However, the speckle noise
401 application is restricted to regions with acoustic intensity, as
402 shown in Figs. 7(f) and 8(d). This fact is due to our sonar
403 model be multiplicative (defined in Eq. 2). In real sonar im-
404 ages, the noise also granulates the shadows and blind regions.
405 The sonar simulator can be improved by inserting an additive
406 noise to our model. **The impact of additive noise on the image**
407 **is more severe than that of multiplicative, and we decided to**
408 **collect more data before including a specific additive noise in**
409 **our simulator, at this moment.** A second feature missing in our
410 simulated acoustic images are the ghost effects caused by rever-
411 beration. This lacking part can be addressed by implementation
412 of a multi-path propagation model as found in [23]. **Simulating**
413 **the multi-path reflection is computationally costly, thus we need**
414 **to consider the real-time constraint by modeling and including**
415 **the reverberation phenomenon.**

416 4.2. Computational time

417 Performance evaluation of the simulator was assessed by
418 considering the suitability to run real-time applications. The
419 experiments were performed on a laptop with Ubuntu 16.04
420 64 bits, Intel Core i7 3540M processor running at 3 GHz with

421 16GB DDR3 RAM memory and NVIDIA NVS 5200M video
422 card. The elapsed time of each sonar data is stored to compute
423 the average time, standard deviation and frame rate metrics, af-
424 ter 500 iterations. The results found is summarized in Tables
425 2 and 3. After changing the device parameters, such as num-
426 ber of bins, number of beams and field of view, the proposed
427 approach generated the sonar frames with a high frame rate,
428 for both sonar types, considering a real-world sonar. Given the
429 Tritech Gemini 720i, a real forward-looking sonar sensor, with
430 a field of view of 120° by 20° and 256 beams, presents a max-
431 imum update rate of 15 frames per second, the results allow
432 the use of the sonar simulator for real-time applications. Also,
433 the MSIS data used in the simulator is able to complete a 360°
434 scan sufficiently fast in comparison with a real sonar as Tritech
435 Micron DST. For the FLS device, these rates are superior to
436 the rates lists by DeMarco *et al* [4] (330ms) and Saç *et al* [3]
437 (2.5min). **For MSIS type, to the best of our knowledge, there is**
438 **no previous work for comparison.**

439 According to previous results, since the number of bins is
440 directly proportional to sonar image resolution, as explained in
441 Section 3.3, this is also correlated with the computational time.
442 When the number of bins increases, the simulator will have a
443 bigger scene frame to compute, and generate the sonar data.

444 5. Conclusion and future work

445 **A GPU-based simulator for imaging sonar simulation is pre-**
446 **sented here. The system is able to reproduce the operation mode**
447 **of two different types of sonar devices (FLS and MSIS) in real-**
448 **time. The real sonar image singularities, such as multiplicative**
449 **noise, surface properties and acoustic shadows are addressed**
450 **and represented in the simulated frames. Specially for the shad-**
451 **ows, the acoustic representation can present information as use-**

452 ful as the insonified object. Considering the qualitative results,
 453 the sonar simulator can be used by feature detection algorithms,
 454 based on corners, lines and shapes. Also, the computation time
 455 to build one sonar frame was calculated using different device
 456 settings. The vertex and fragment processing during the un-
 457 derwater scene rendering accelerates the sonar image building,
 458 providing an average time close or better than real imaging de-
 459 vices. These results allowed the use of this imaging sonar sim-
 460 ulator by real-time applications, such as obstacle detection and
 461 avoidance, and object tracking. We are analyzing now a way to
 462 add the reverberation effect to perform a more realistic sensing,
 463 as well as, without significantly effect the computational time
 464 must. We are working now on how to include an additive noise
 465 in the simulation of the acoustic images.

466 References

- 467 [1] Bell JM. Application of optical ray tracing techniques to the simulation
 468 of sonar images. *Optical Engineering* 1997;36(6):1806–13.
- 469 [2] Coiras E, Groen J. Simulation and 3d reconstruction of side-looking sonar
 470 images. In: Silva S, editor. *Advances in Sonar Technology*; chap. 1. In-
 471 Tech; 2009, p. 1–15.
- 472 [3] Saç H, Leblebicioğlu K, Bozdagi Akar G. 2d high-frequency
 473 forward-looking sonar simulator based on continuous surfaces ap-
 474 proach. *Turkish Journal of Electrical Engineering and Computer Sciences*
 475 2015;23(1):2289–303.
- 476 [4] DeMarco K, West M, Howard A. A computationally-efficient 2d im-
 477 aging sonar model for underwater robotics simulations in Gazebo. In:
 478 MTS/IEEE OCEANS Conference. 2015, p. 1–8.
- 479 [5] Gu J, Joe H, Yu SC. Development of image sonar simulator for under-
 480 water object recognition. In: MTS/IEEE OCEANS Conference. 2013, p.
 481 1–6.
- 482 [6] Kwak S, Ji Y, Yamashita A, Asama H. Development of acoustic camera-
 483 imaging simulator based on novel model. In: IEEE International Con-
 484 ference on Environment and Electrical Engineering (EEEIC). 2015, p.
 485 1719–24.
- 486 [7] Guérion D, Sintes C. Forward looking sonar data simulation through tube
 487 tracing. In: MTS/IEEE OCEANS Conference. 2010, p. 1–6.
- 488 [8] Quigley M, Conley K, Gerkey BP, Faust J, Foote T, Leibs J, et al. ROS:
 489 an open-source robot operating system. In: Workshop on Open Source
 490 Software, held at IEEE International Conference on Robotics and Auto-
 491 mation (ICRA). 2009, p. 1–6.
- 492 [9] Hurtós N. Forward-looking sonar mosaicing for underwater environments.
 493 Ph.D. thesis; Universitat de Girona; 2014.
- 494 [10] Abbot J, Thurstone F. Acoustic speckle: theory and experimental analy-
 495 sis. *Ultrasonic Imaging* 1979;1(4):303–24.
- 496 [11] Ganesan V, Chitre M, Brekke E. Robust underwater obstacle detection
 497 and collision avoidance. *Autonomous Robots* 2015;40(7):1–21.
- 498 [12] Ribas D, Ridao P, Neira J. Underwater SLAM for structured environ-
 499 ments using an imaging sonar. Springer-Verlag Berlin Heidelberg; 2010.
- 500 [13] Fallon MF, Folkesson J, McClelland H, Leonard JJ. Relocating under-
 501 water features autonomously using sonar-based SLAM. *Journal of Ocean
 502 Engineering* 2013;38(3):500–13.
- 503 [14] Liu L, Xu W, Bian H. A LBF-associated contour tracking method for
 504 underwater targets tracking. In: MTS/IEEE OCEANS Conference. 2016,
 505 p. 1–5.
- 506 [15] Huang TA, Kaess M. Towards acoustic structure from motion for imaging
 507 sonar. In: IEEE/RSJ International Conference on Intelligent Robots and
 508 Systems (IROS). 2015, p. 758–65.
- 509 [16] Bradski G. The opencv library. *Doctor Dobbs Journal* 2000;25(11):120–
 510 6.
- 511 [17] Watanabe T, Neves G, Cerqueira R, Trocoli T, Reis M, Joyeux S, et al.
 512 The Rock-Gazebo integration and a real-time AUV simulation. In: IEEE
 513 Latin American Robotics Symposium (LARS). 2015, p. 132–8.
- 514 [18] SDF. <http://sdformat.org/>; 2017. Accessed: 2017-04-23.
- 515 [19] Soetens P, Bruyninckx H. Realtime hybrid task-based control for robots
 516 and machine tools. In: IEEE International Conference on Robotics and
 517 Automation (ICRA). 2005, p. 260–5.
- 518 [20] Rost RJ, Licea-Kane B, Ginsburg D, Kessenich JM, Lichtenbelt B, Malan
 519 H, et al. OpenGL shading language. 3rd ed.; Addison-Wesley Profes-
 520 sional; 2009.
- 521 [21] Lee J. Digital image enhancement and noise filtering by use of local sta-
 522 tistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*
 523 1980;2(2):165–8.
- 524 [22] Papoulis A, Pillai S. *Probability, random variables and stochastic pro-
 525 cesses*. McGraw Hill; 2002.
- 526 [23] Huang J. Simulation and modeling of underwater acoustic communica-
 527 tion channels with wide band attenuation and ambient noise. Ph.D. thesis;
 528 Carleton University; 2015.