# SE-Workbench-RF: Performant and High-Fidelity Raw Data Generation for Various Radar Applications

Nicolas Douchin<sup>\*</sup>, Christian Ruiz<sup>\*</sup>, Jonathan Israel<sup>\*\*</sup>, Henri-José Mametsa<sup>\*\*</sup>

\*OKTAL Synthetic Environment Vigoulet-Auzil, FRANCE email: nicolas.douchin@oktal-se.fr

\*\*ONERA-DEMR (Radar Department) Toulouse, FRANCE email: Jonathan.Israel@onera.fr

Abstract: Guidance of weapon systems relies on sensors to analyze targets signature. Defense weapon systems also need to detect and then identify threats also using sensors. One important class of sensors is radar systems that are very efficient for all weather detection. High frequency radars can produce high quality images with very complex features such as dihedral and corner reflector high scattering contributors, shadows and layover effects. Besides, image quality is very dependent on the carrier velocity and trajectory. Such sensors systems are so complex that they need simulation to be tested in a large variety of operational conditions.

This paper presents a state-of-the-art solution, called SE-Workbench-RF, for generating raw data dedicated to radar simulation based on the exploitation of synthetic environments, which means physical modelling of targets and backgrounds (terrains, buildings, vegetation and other entities).

The paper gives an overview of the models and their implementation in SE-RAY-EM, which is the rendering module of SE-Workbench-RF, including specific features related to radar simulation in the maritime environment with targets moving on the dynamic sea surface. Several technical topics are also discussed, such as the rendering technique (ray tracing vs. rasterization), the implementation (CPU vs. GP GPU) and the tradeoff between physical accuracy and computational performance.

*Key words:* Radar simulation, synthetic environment, asymptotic methods, sea modelling, ship wakes, GPU

### 1. Introduction

Radars are considered for their all-weather capacities to achieve target detection, recognition and even identification, in the frame of surveillance, targeting and guidance systems. Radar systems are governed by very complex features such as dihedral and corner reflector high scattering contributors, shadows and layover effects. The mechanisms that govern such sensor systems are so complex that simulation is needed to assess their performance in a large variety of operational conditions. And the synthetic environment, which means physical target and its background (terrains, buildings, vegetation and other entities) rather than just the target itself, is very import for such sensors system simulation.

This paper presents a solution, developed jointly by OKTAL-SE and ONERA, dedicated to the simulation of raw data of large-scale environments including complex targets for several kinds of radar. After a brief presentation of the basic concepts that are implemented in this simulation

software, a focus is made on the GPU implementation that allows to reduce the computation time considering a massively parallel design of the computation. Reducing the computation time so much is very convenient for parametric studies where the same scenario is considered a large number of times by just changing one parameter value.

The capacity of modelling and rendering the sea environment is also of great importance. In this paper the modelling of the dynamic sea surface and ship wakes is presented along with its application to synthetic SAR image generation of sea targets travelling at the sea surface.

## 2. State of the art

SE-RAY-EM is an electromagnetic simulation code based on ray tracing [1] and asymptotic methods [2]. These methods are less physically rigorous than "exact" methods that are strictly based on the resolution of Maxwell equations. However, asymptotic methods as they are used in SE-RAY-EM enable to handle complex scenes that are very large compared to the wavelength with enough accuracy. In such situations, asymptotic methods provde main contributions of the electromagnetic field which are correctly represented.

In the case of complex targets, SE-RAY-EM gives results very similar to the "exact" method for a much lower computation time. Where 600 MHz is a high limit for "exact" solutions used on this type of objects, it is almost a low limit in terms of physical validity for asymptotic methods. The standard computation range addressed by SE-RAY-EM is more between 1 to 100 GHz on far more complex scenes (natural terrain on several kilometres with several complex targets).

In SE-RAY-EM, rays are traced from transmitters towards reception points. These rays are grouped four by four in beams. Rays are traced from transmitters through a grid (figure 1). The intersections of theses beams are computed. There are two types of interactions (figure 2):

- Geometrical Optics (GO), when the beam is reflected by a metallic or dielectric surface
- Physical Optics (PO), when part of the beam energy is scattered towards the reception points at each interaction.



Figure 1. SE-RAY-EM emission grid



Figure 2. Principle of beam interactions

Using beams has three main interests. They enable to:

- know the interaction contour of the incident wave with the surface, which is necessary for using PO,
- detect edges,
- detect the main aliasing cases: geometry, material, curvature of the surface.

As shown in figure 3, the adaptive anti-aliasing mechanism of SE-RAY-EM consists in subdividing the computation grid as a "quadtree". So, when aliasing is detected on the path of a beam, it is subdivided in four new beams.



Figure 3. SE-RAY-EM adaptive anti-aliasing principle

Adaptive anti-aliasing is very interesting for EM computations as:

- The size of the primary grid can be coarse since the primary grid resolution has no impact on the physical signal computation. The only constraint is that the grid shall be thin enough in order to interesect all relevant features of the scene entities .
- The accuracy needed for EM computation is linked to the wavelength and object boundaries must be sampled at a fraction of the wavelength. For example, at 100GHz, sampled element size can be less than 1 mm for a target of several meters. Without adaptive anti-aliasing, the computation should use a 100000 x 100000 pixel grid!
- Adaptive anti-aliasing enables to reduce the number of contributors for which EM models are applied. These complex computations have a cost that cannot be neglected and can be done for the processing at different wavelengths. Reducing the number of contributors is then inevitable to reduce computing times.



Figure 4. Concept of EM contributor

The ray tracing based on shooting and bouncing ray technique is used to identify the EM contributors in the scene. In the very simple case illustrated by the figure here above (fig. 4) composed by two flat polygons and a unique ray tube, two EM contributors are identified. Then the amplitude and phase of the EM fields scattered to the receiving point by each EM contributor are computed using PO.

So, at the end of the computation, we can have access to a list of EM contributors. For each EM contributor, we can get many parameters such as: its position, the entity of the scenario it belongs to, its speed, the total path length related to this EM contributor, including the multiple

bounce and the amplitude and phase of the EM fields scattered by the EM contributor to the receiving point.

Access to this information is possible thanks to the EM contributor API developed by OKTAL-SE. Then any kind of processing can be applied on the collection of EM contributors identified in the scene.

The validation of SE-RAY-EM is a permanent process that started in the frame of the PhD works which this solution is based on. A validation dossier is provided along with the software which presents several validation tests from canonical cases to more complex cases. Validation is achieved through comparison of SE-RAY-EM with other codes based on different methods: typically, an ONERA code based on the Method of Moments to solve the integral equation, formerly called ELSEM3D and now called MAXWELL3D. Validation is also based on comparison between simulation and measurements, in cooperation with Fraunhofer FHR in Germany for example, as illustrated hereafter (fig. 5):

Validation through comparison



Figure 5. Example of validation test

## 3. Massively parallel approach adapted to GPGPU implementation

Nowadays, Graphic Processors Unit (GPU) have proven to be very efficient for optimizing General Purpose (GP) computations, and particularly ray tracing applications [3] as SE-RAY-EM. However, the transition to GPU is not straightforward and several issues have to be taken into account.

### **Double versus Float**

Historically, SE-RAY-EM performs all computations using double precision floating point operations (double). This permits to avoid naturally some problems of accuracy without any specific work. However, GPU are very efficient for single precision floating point (float) computation. Even though, double computations are now handled and tends to become faster, they should be used with parsimony. Double data storage and transfer are also more costly and double operations need twice as much registers as float ones. And registers are rare and precious when using GPUs. Our new implementation only uses double when it is absolutely necessary, mostly in the computations that involve the phase of the electromagnetic signal.

#### **Object instances and moving objects**

SE-RAY-EM uses instances of objects in order to share geometries. Such objects can also reference underlying instances, which results in a hierarchical graph. A matrix defines the position of each instance. During raytracing, rays are transformed using these matrices in local space to perform intersection computation with the geometry. The hierarchical definition implies several transformations when a ray traverses the scene, which results in a lot of computation. This process has proven itself inefficient on the GPU. Then, we flattened the instance hierarchy before computation so that a ray is only transformed one time during ray tracing. This enables to handle both instances and moving object on the GPU with a small impact on performances.

#### **Cone tracing**

Instead of tracing individual rays, we use cones to trace beams. A cone will detect any geometric element in its volume. This way of anti-aliasing is more reliable and small object and edges are detected directly without the need of any transparent wings. Besides the initial resolution can be really coarse. The only reason to increase the resolution will then be imposed if the GPU cannot be loaded enough.

Another advantage of using cones is that they can be processed independently of their neighbors, which is very interesting for parallelization. It is also simple to handle the generated contributors independently from one reflection level to another.

#### Adaptive anti-aliasing

The difficulty when implementing adaptive anti-aliasing on GPUs consists essentially in managing the number of beams. Actually, when performing anti-aliasing, beams are cut in four sub-beams at each level, which can lead to an explosion of the number of beams to achieve the needed accuracy. Unfortunately, these beams need to be stored in buffers to be handled on the GPU, which represents a lot of data.

In order to solve this issue, we use serialization of the beams. Which means that when a beam is aliased, instead of generating the four beams at once, only the first child is generated, and when it is finished, it will launch his brother. With this approach, we ensure that the number of beams to treat never rises. However, if there is enough space left in the beam buffer, some beams may be de-serialized, in order to fill the buffer completely. Indeed, the GPU needs to be loaded enough to be efficient.

#### **Multiple frequency cases**

Multi-frequencies computations consist in computing several (a lot e.g. one to ten thousand) frequencies at the same time. Frequencies are independent in terms of electromagnetic computations, but share all the geometric computations (if we consider that anti-aliasing parameters given for the highest frequency are also valid for the smaller ones). In this new version, we postpone the frequency computations as long as possible. The anti-aliasing process generates all geometrical contributors. These contributors are then used multiple times for applying EM models, one time per frequency.

#### Validation of the GPU implementation

The GPU implementation is tested versus the CPU one in terms of both accuracy and performances.

The first test consists in comparing qualitatively the results between the new implementation and both the standard version of SE-RAY-EM and the reference from the ELSEM3D software of ONERA which is based on Integral Equation EM approach and Moment Method numerical implementation [4]. The test consists in the computation of the Radar Cross Section (RCS) computation of an aircraft for a frequency of 600 MHz (as represented in figure 6). Figure 7 shows the results that are obtained. Some differences exist because signal of a complex object is noisy, but globally results computed by the three codes are very similar.





Figure 6. Aicraft RCS computation setup

Figure 7. RCS of an aircraft computed with ELSEM3D (green), SE-RAY-EM (blue) and GPU version (red)

For testing the performances of the GPU implementation, we consider the computation of the RCS at 10000m of a tank with one or 151 frequencies from 8 to 11 GHz. Figure 8 shows the complex tank used for the simulation and an ISAR image generated from computed multi-frequencies and multiple angles results.



Figure 8. The tank used for test simulation and a computed ISAR image

To compare the performances, the computations have been performed using the standard version of SE RAY EM and using the new parallel implementation with both a CPU and a GPU version. We performed the tests on two distinct computers:

- One laptop: Intel Core I7 2630QM and Nvidia GeForce 540M
- One desktop: Intel Core I5 3470 and Nvidia GeForce GTX Titan.

	Standard Core I7	Standard Core I5	CPU Core I7	CPU Core I5	GPU GeForce 540M	GPU GeForce Titan
Mono-frequency	16.28	12.17	7.49	5.60	0.86	0.111
151 frequencies	109.00	80.40	97.85	70.4	7.62	0.735

 

 Table 1. Computation times for one incidence in seconds with SE-RAY-EM standard version and with new parallel version in both CPU and GPU implementations

 Comparing the times of the standard version and the CPU new version, we can see that the new algorithm is more efficient than the previous one. The GPU implementation is 10 to 100 times faster than the standard SE RAY-EM version.

## 4. Radar simulation in sea environment

## Modelling of the dynamic sea surface

The 3D modelling of the dynamic sea surface is based on the definition of the sea waves spectrum that depends on sea state and on the distance to the sea shore.

In SE-Workbench-RF, two models of sea waves spectrum are available:

- The Jonswap model [5] which is a second-generation model and adapted to open sea conditions (far from the cost)
- The SWAN model, which is a third-generation model and adapted to sea shore conditions [6].

An example of a patch of dynamic sea surface for sea state corresponding to Beaufort 3 is given hereafter:



Figure 9. Patch of 3D dynamic sea surface

For marine application, it is very important to represent the real and complex shape of the sea surface. The wave self shadowing effect is very important. The shadowing of floating objects is also important. We chose the solution of sampling this surface using a "height map" Z(X, Y) i.e. a raster information rather than a polygonal vector information. Nevertheless, the polygon shaping remains important both for coast modelling and to take into account the sea surface roundness. The hypertexture model answers these requirements.



Figure 10. Hypertexture volume definition

Within this volume, the hypertexture contains the additional height modulation to the average altitude given by the polygons. The ray tracing process is applied directly on the hypertexture as illustrated on figure 11.



Figure 11. Ray tracing process in the hypertexture

## Physical characterization of the sea water

The sea water physical material is a bistatic scatterer type of EM physical material meaning that it is defined by its Fresnel reflection coefficients for reflection in the specular direction (coherent component) and by bistatic scattering coefficients for scattering in any other direction including backscattering. The Fresnel reflection coefficients are modified according to sea state. The bistatic scattering coefficients are computed using PO on a piece of 3D sea surface built as a static "realization" of the dynamic sea surface.



Figure 12. Patch of 3D sea surface used for generating the scattering coefficients

### Ship wakes

Two types oh wakes are considered in the radar domain: the Kelvin wake and the dark water wake.

The Kelvin wake is implemented as a 3D local wake height map deformation which is consistent with hypertexture profile.

The dark water wake behind the ship is considered as a zone with lower sea state to simulate the flattening of the sea surface due to the movement of the ship at the sea surface.

### SAR simulation in sea environment

SE-RAY-EM GPU version has been extended to scenarios in the dynamic sea environment, including ships travelling at the sea surface and thus generating wakes. Examples of synthetic

SAR images are given hereafter for two different sea states (Beaufort3 (left) and Beaufort7 (right)):



Figure 13. Synthetic SAR images

### 5. Some other results related to various radar applications

Typical examples of results are shown hereafter as synthetic SAR images, RBGM radar image, Range/Doppler maps or Doppler signatures. They illustrate the capacity of SE-Workbench-RF to generate raw data for almost any kind of radar applications.



Airport 3D DB

Simulated SAR image



Automotive scenario

Range/Doppler map



Doppler signature of helicopter



RBGM radar image of airport 3D DB

### 6. Conclusions

The radar solution presented in this paper has been developed for more than 15 years in cooperation with several partners such as ONERA in France and FOI in Sweden who have participated to both the design of the models and the validation of the software implementation. To make our simulation software solution be adapted to parametric studies, it was needed to reduce the computation time drastically. For achieving that the implementation on the GP GPU based on CUDA was selected as the only relevant solution. Indeed, the solution adopted on the optronic domain for high performance computation, based on rasterization, is not adapted to radar since multipath and phases have to be computed precisely.

In this paper, the solution adopted for SAR simulation in sea environment, including the wakes generated by ships travelling at the sea surface was presented. This solution is now operational but still needs to be further validated, especially by comparison to real measurements. This is a big issue for future works with partners that can bring such measurements.

## References

- [1] T. Whitted, "An Improved Illumination Model for Shaded Display", *Communications of the ACM 23*, 1980, pp. 343-349.
- [2] H.J. Mametsa, S. Laybros, T.Volpert, P.F. Combes, P.N. N'Guyen, P. Pitot, "*FERMAT:* A high frequency EM scattering code from complex scenes including objects and environment", *Progress in Electromagnetic Research Symposium (PIERS 04)*, Pisa, Italy 28-31 March 2004.
- [3] T. Ailan, S. Laine, "Understanding the Efficiency of Ray Traversal on GPUs", *Proc. High-Performance Graphics*, 2009, pp. 145-149.
- [4] P. Soudais, H. Steve, and F. Dubois, "*Scattering from several test-objects computed by 3-D hybrid IE/PDE methods*", Antennas and Propagation, IEEE Transactions on, Vol. 47, No. 4, pp. 646-653, 1999.
- [5] Hasselmann, K. and al., "*Measurements of windwave growth and swell decay*", Ergänzungsheft zur Deutschen Hydrographischen Zeitschrift, 1973
- [6] http://swanmodel.sourceforge.net/