Advanced SAR simulator for ATR and AI database generation

|  |  |
| --- | --- |
| Simone Placidi, simone.placidi@metasensing.com  MetaSensing AP Pte Ltd, Singapore | Alessandro Vetere, alessandro.vetere@metasensing.com  Eugenio Pino, eugenio.pino@metasensing.com  Adriano Meta, adriano.meta@metasensing.com  MetaSensing, Milan, Italy |

*Abstract* — Automatic Target Recognition (ATR) with Artificial Intelligence (AI) plays a key role in modern surveillance and reconnaissance activities. The enabling technology for ATR is the availability of a database that is representative of realistic situations including targets and backgrounds. When applied to radar imaging the problem of collecting enough data of targets becomes extremely complicated and unaffordable. At MetaSensing we are tackling this problem with a recently developed SAR simulator able to estimate the complex 3D RCS of a target and to model the SAR collection and processing stages to build a synthetic yet realistic database of target-background-images. This paper provides a description of the SAR simulator and examples of synthetic data generated.

Keywords — SAR, Synthetic Aperture Radar, Simulations, EM Simulations, Ray Tracing, Artificial Intelligence (AI), Automatic Target Recognition (ATR), GPU, Raytracing

# Introduction

Synthetic Aperture Radar (SAR) is a critical remote sensing imaging technology which is receiving a lot of interest thanks to the NewSpace developments. There are multiple applications which can benefit from SAR imaging ranging from Automatic Target Recognition (ATR) in the Intelligence, Surveillance and Reconnaissance (ISR) domain to structural stability monitoring in the geomatics domain. Moreover, SAR can be used for agriculture, forestry, biomass, and generation of Digital Elevation Models.

Within the ISR domain, SAR with its ATR technique plays a very important role providing advanced information about the detected targets in the SAR images. The combination of SAR, ATR and AI is pivotal for the best SAR-based ISR system which allows to identify, classify, and recognize different targets in different conditions. The enabling technology for ATR is the availability of a high-fidelity realistic database of SAR images for backgrounds and targets under different configurations which can be used to match the observed real targets for the recognition purpose.

Although more and more satellite SAR are deployed providing a lot of data, there are still several drawbacks and barriers in the generation of databases of target signatures in SAR images. First, costs are one of the main limiting factors in creating a collection of SAR images with labelled targets to be used as training dataset or ATR database, second real SAR data are subject to multiple noise sources including system errors and accuracies, position errors, viewing geometries, data processing, uncontrolled scenarios on ground. Besides the above problems, there is also the problem of having access to the real targets and the problem of the amount of data and experiments required. The RCS (or the image) of a target varies drastically due to the characteristics of the radar (resolution, operational band) and the aspect angles (in azimuth and depression angles). Other complications come from the near background scene surrounding the target that must be added to the database. The configuration space becomes extremely large.

All those barriers make difficult to have a reliable collected data usable for SAR-based ATR and AI applications hence simulated synthetic data are a much-preferred choice thanks to the possibility to control all the simulation spaces and configurations and generate less expensive dataset.

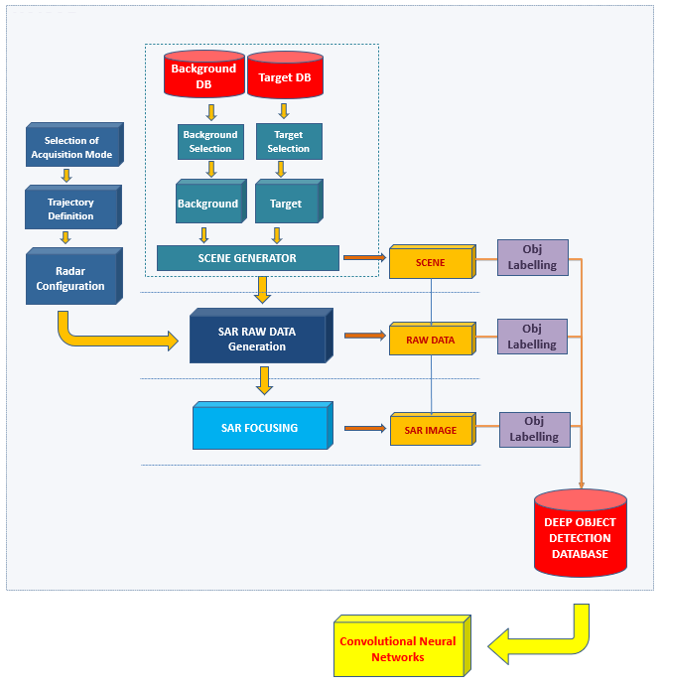


Fig. 1: Flow chart model of the KAISAR

MetaSensing has developed a realistic fast NVIDIA GPU hardware accelerated SAR simulator, called KAISAR, with the goal to support different applications. The first application is to generate large databases of SAR images of targets and backgrounds under different configurations to enable ATR. The second application is to generate SAR images with labelled targets to support deep learning developments for object detection in SAR images. The third application is to provide a fast SAR simulation environment for generation of SAR raw data for system design and new algorithm developments.

KAISAR is based on NVIDIA RTX GPU card technology to provide hardware acceleration for the raytracing in the EM-Solver module, and the back-projection algorithm in the SAR raw data and SAR images simulations of the SAR module.

In this paper, the KAISAR simulator developed by MetaSensing is presented along with some examples of simulations.

# KAISAR Architecture

The KAISAR simulator is based on two cores: an asymptotic ElectroMagnetic Solver (EM-Solver) and a SAR simulator including the image focuser.

Fig.1 depicts the flow chart of the KAISAR from the two databases of backgrounds and targets, which are used to create a scene to be simulated and included in the deep object detection database which can then be passed to the convolution neural networks (CNN). The EM-Solver generates the 3D Radar Cross-Section (RCS) maps of a 3D scene composed of user-selected background and targets. The scene is created by the user from high-resolution 3D models of targets placed on 3D realistic background models. The scene is then used by the scene generator module which, by applying the EM-Solver, generates the 3D RCS maps. The 3D RCS maps are passed to the SAR simulator that implements a Reverse Global Back Projection to generate the SAR raw data considering the different configurations set by the user related to the acquisition mode (spotlight, stripmap), the trajectory and the radar parameters. The simulated SAR raw data are then focused with a Global Back Projection to SAR Single Look Complex images. The SAR simulator is all implemented in CUDA on GPU cards to reduce the processing time, while the EM-Solver is based on the DirectX 12 and DirectX Raytracing (DXR) technologies. All the targets in the three domains (RCS-map, SAR Raw data, SAR images) can be labelled allowing for quick finding and use in training database. All the simulated data are then saved in a database which can be used for ATR or as AI training dataset.

Fig.2 depicts the main modules of the KAISAR including the EM-Solver and the SAR simulator. The EM-Solver based on ray-tracing techniques takes care of modelling the real-world scenario composed of realistic background and targets. The EM-Solver generates the 3D RCS maps which are passed to the SAR raw data generation module which accounts for all the configuration parameters related to the platform and the radar. The simulated SAR raw data are focused with the Global Back Projection to SAR images.

This SAR Simulator, thanks to the use of GPU hardware acceleration, can simulate single targets in high resolution in few seconds. A large database of labelled targets and background both in their RCS form, in the SAR raw data and SAR images can be generated in short time; for instance, it takes about 10 days for 10k scenes with 5 targets at 30cm resolution, on a single Intel i7-8700K, 32GB RAM machine equipped with a NVIDIA RTX 2080 SUPER. Adding another similar machine would half the computation time linearly.

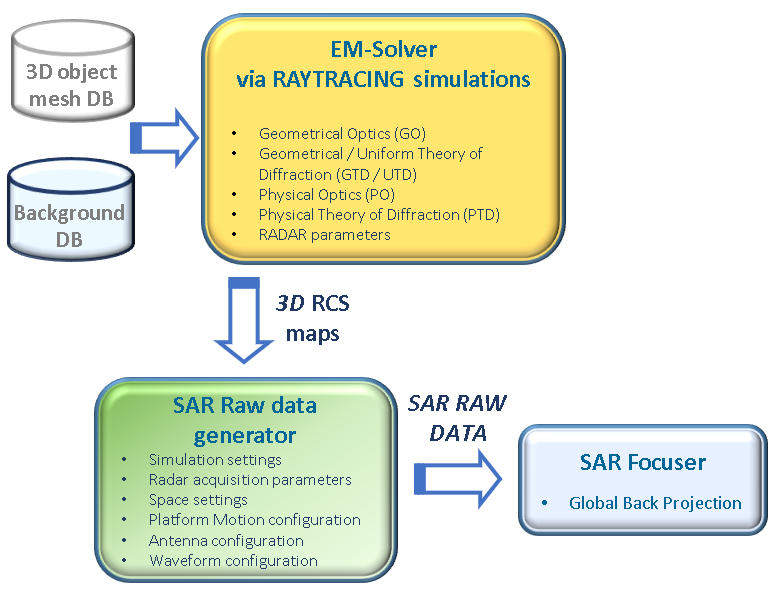


Fig. 2: KAISAR architecture and main modules

The paragraphs below will provide a description of the different modules.

## The input database

The input data for the real-world modeling is composed of two separate databases for background and targets. High-resolution high-fidelity 3D models are used for the targets, and they can be obtained from different sources and formatted in standard .obj file also including information about the materials. For AI training dataset purposes, the target models are divided into five different classes: marine, air, ground, weapon (missile), radar. Each class is then divided into different subclasses as reported in Table 1. Every target is then identified with its own model name.

Table 1: Classes and Sub-classes of targets

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Marine** | **Air** | **Ground** | **Weapon (Missile)** | **Radar** |
| Warship | Fixed wing | Tank | Surface | Surveillance |
| Submarine | Rotary Wing | Artillery | Air |  |
| Cargo ship | UAV | Vehicle | Ship |  |
| Motorboat |  |  |  |  |

The backgrounds instead are also obtained from realistic 3D models of different areas and are divided into six classes: airport, downtown, forest, harbor, mountain, and sea. Each of the class have different subclasses related to its main class.

Table 2: Classes and Sub-Classes of the backgrounds

|  |  |
| --- | --- |
| **Airport** | International airport |
| **Downtown** | City |
| **Forest** | Generic spring forest |
| **Harbor** | Seaport with city |
| **Mountain** | Forested mountain |
| **Sea** | Coastline |

|  |  |
| --- | --- |
|  | Map  Description automatically generated |

Fig. 3: Scene model generated with five artillery tracks in a downtown background.

## The EM solver

The asymptotic EM-solver used in KAISAR is based on both the GO-UTD (Geometrical Optics/Uniform Theory of Diffraction) and PO-PTD (Physical Optics/Physical Theory of Diffraction) implementations of Ray Optics. It is well known that the limitations of performance in time come from the ray tracing stage (Shooting and Bouncing Ray - SBR) but modern GPUs equipped with dedicated hardware units (GPU RTX series) can perform billions of ray casts per second reducing the computation time by thousands of times if compared with scalar CPUs. The problem of SBR is solved natively from frameworks like DirectX which drastically reduces the effort of software implementation. The contribution of GPU can also be exploited during the stage of integration for both PO & PTD.

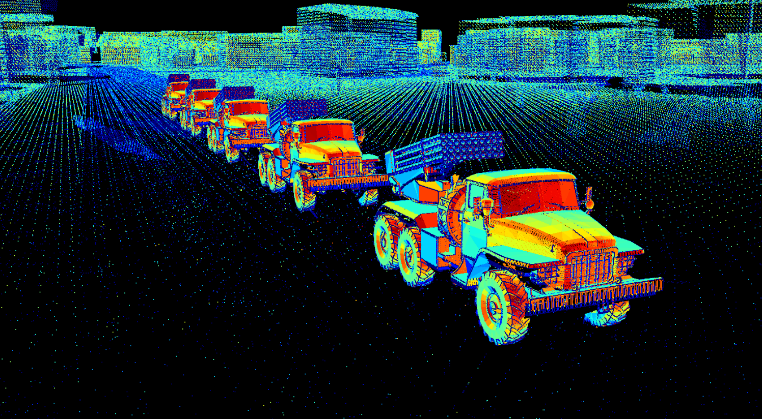


Fig. 4: Normalized RCS map of the scene with five artillery tracks in a downtown background

## The SAR Simulator

The SAR simulator is based on a two-stages module: first the SAR raw data are simulated using the Reverse Back Projection technique with the configurations related to the platform trajectory (airborne or satellite), the SAR parameter configurations and the viewing geometry. Afterwards the SAR raw data are focused with the Global Back Projection algorithm to generate the SAR images. Both steps are implemented in GPU-CUDA language to exploit the parallel processing and increase the simulation speed.

|  |  |
| --- | --- |
|  |  |

Fig. 5: Range compressed (left) and SAR image with labelled target (right) resulting from simulations.

## The Target Labelling

The targets in the KAISAR can be all labelled in the GeoJSON file standard to allow an easy-to-use dataset as training dataset for Machine Learning and Deep learning applications. The labelling, besides the location of the target in the images, includes the class, sub-class and model of the target simulated.

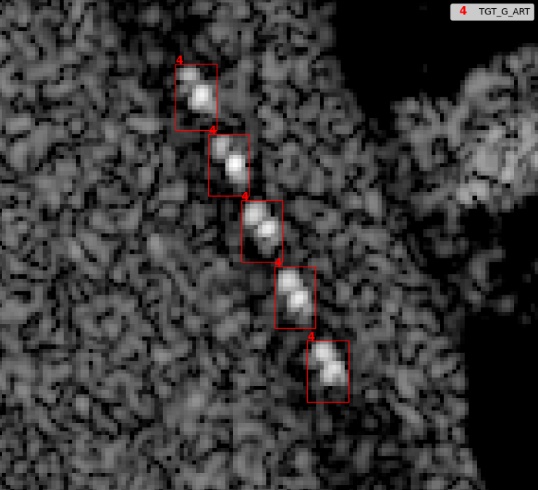


Fig. 6: Zoom in on the labelled targets in the SAR image.

## Polarimetry and Multi-Frequency

The KAISAR simulator can simulate polarimetric images including amplitude and phase generating the SAR images in both polarimetric domains: co-polar (HH-VV) and cross-polar (VH-HV). Fig. 7 shows the polarimetric output of simulated SAR images for a scene with an aircraft carrier and a submarine.

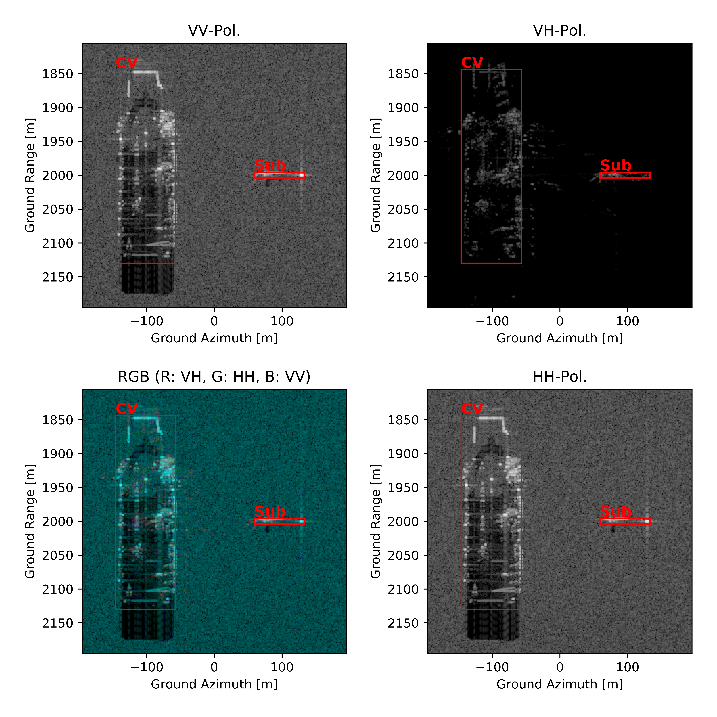


Fig. 7: Example of the polarimetric simulations

Moreover, the KAISAR can simulate RCS, SAR raw data and SAR images for different frequencies from low-frequencies (P-band) to higher frequencies (Ka-band).

# Simulation examples

## 100 Tanks in 1x1 km scene

Fig. 8 shows the flow and description of a simulation of 100 tanks in a 1x1 km scene with flat terrain with a resolution of 15 cm.

This scene is used as reference for testing the simulator speed being a scene with dense targets at high-resolution.

Fig. 8 shows, in the upper row, the four steps of the simulator: RWM: real-world modelling, Raw: SAR simulation, IMG: SAR focusing and LBL: target labelling, with the output image of each step, in the middle row, and the format of the output data in the lower row.

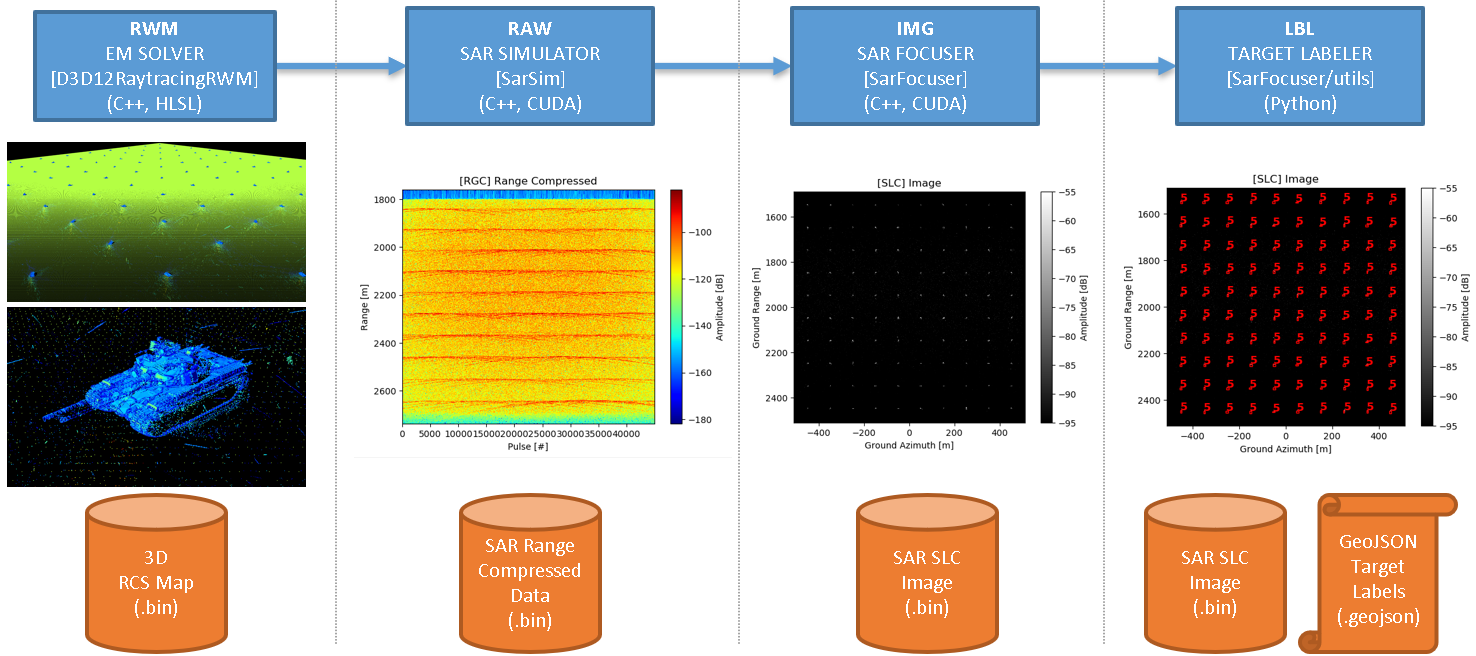


Fig. 8: Simulation example with 100 tanks in a 1x1 km scene

The upper row also shows the software language used for the implementation being C++, HLSL (High-Level Shading Language) for the Real-World Modelling and generation of RCS maps, C++ and CUDA for the SAR simulator and Python for the target labelling.

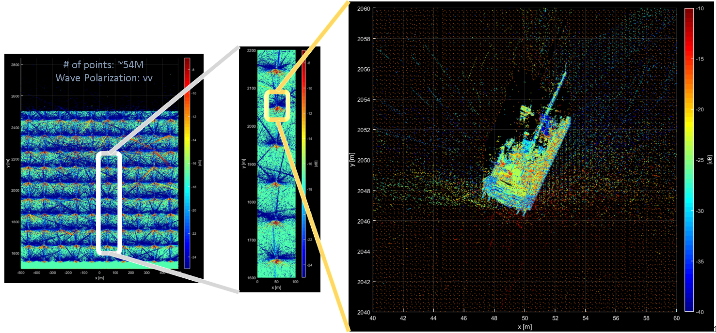


Fig. 9 shows the RCS map of the full scene with the zoom in to one single target. While Fig. 10 shows the simulated SAR image for the single tank.

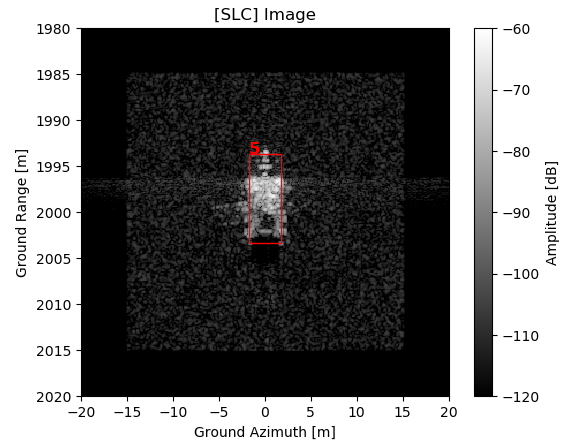


Fig. : SAR image of the tank

The simulation time for the entire scene of 1x1 km with 100 tanks with a resolution of 1 m and a total of 54,221,000 points, is about 17 minutes on the reference machine equipped with GPU NVIDIA RTX 2080 SUPER, 32GB DDR4 RAM and CPU Intel i7-9700K, divided as follow:

|  |  |
| --- | --- |
| RCS map | 430 secs |
| RAW data | 610 secs |
| SAR Image | 16 secs |
| TOTAL | 1056 Secs (~ 17.6 mins) |

Fig. : Computation time for the benchmark scene on the reference machine

## The MSTAR dataset augmented

KAISAR is being used also to simulate some of the targets of the MSTAR dataset and to augment it to provide more information for ATR database. Fig. 12 shows the optical and related SAR images of the MSTAR dataset.

With KAISAR we simulated different targets, namely the T72, BTR70, BMP-2 and we recreated the synthetic dataset which we could use for data augmentation.

Fig. 13 reports the output of the different simulations for the tank BMP-2 of the dataset showing the RCS map, the range-compressed and the SAR image with the labelled target.

Fig. 14 shows the data augmentation for a synthetic scene where two MSTAR targets are placed next to each other and simulated for 72 different heading.

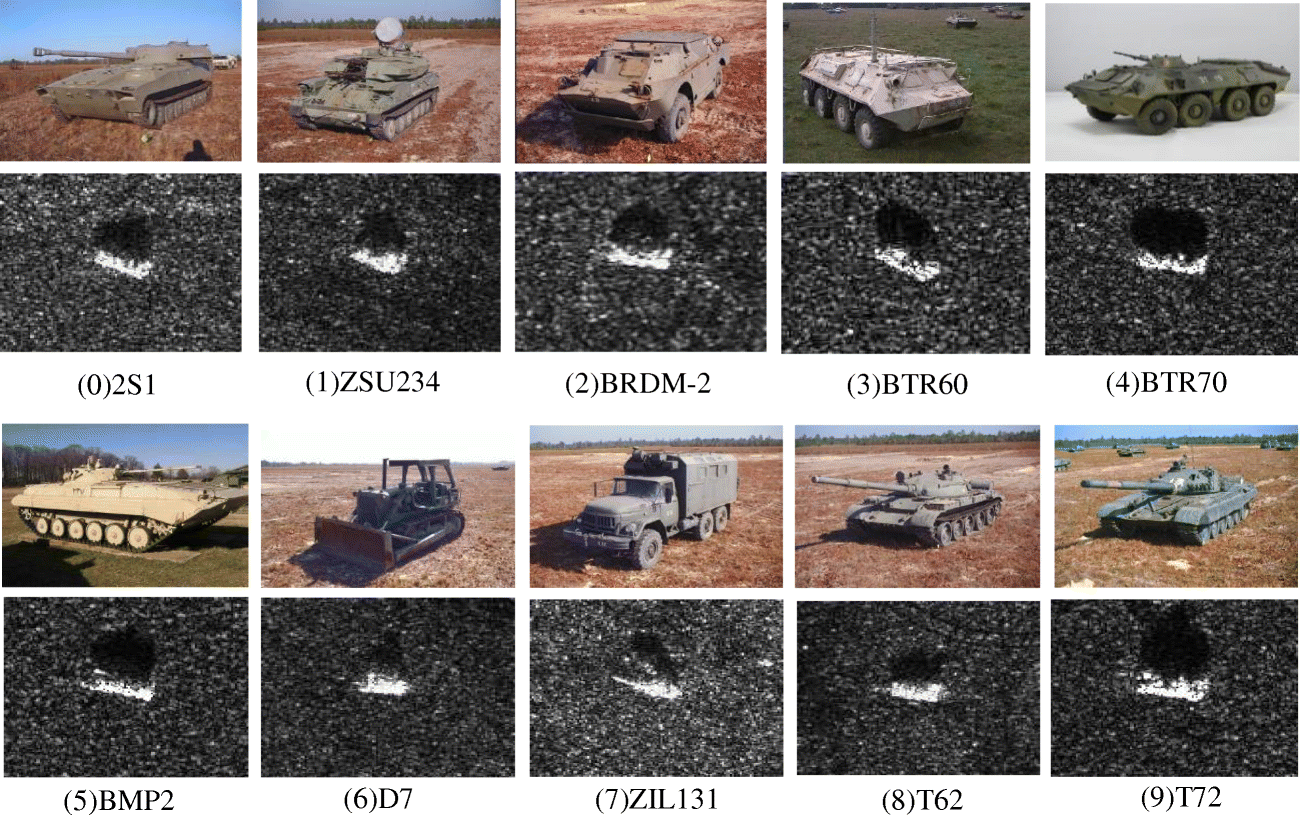


Fig. 12: Overview of the MSTAR targets

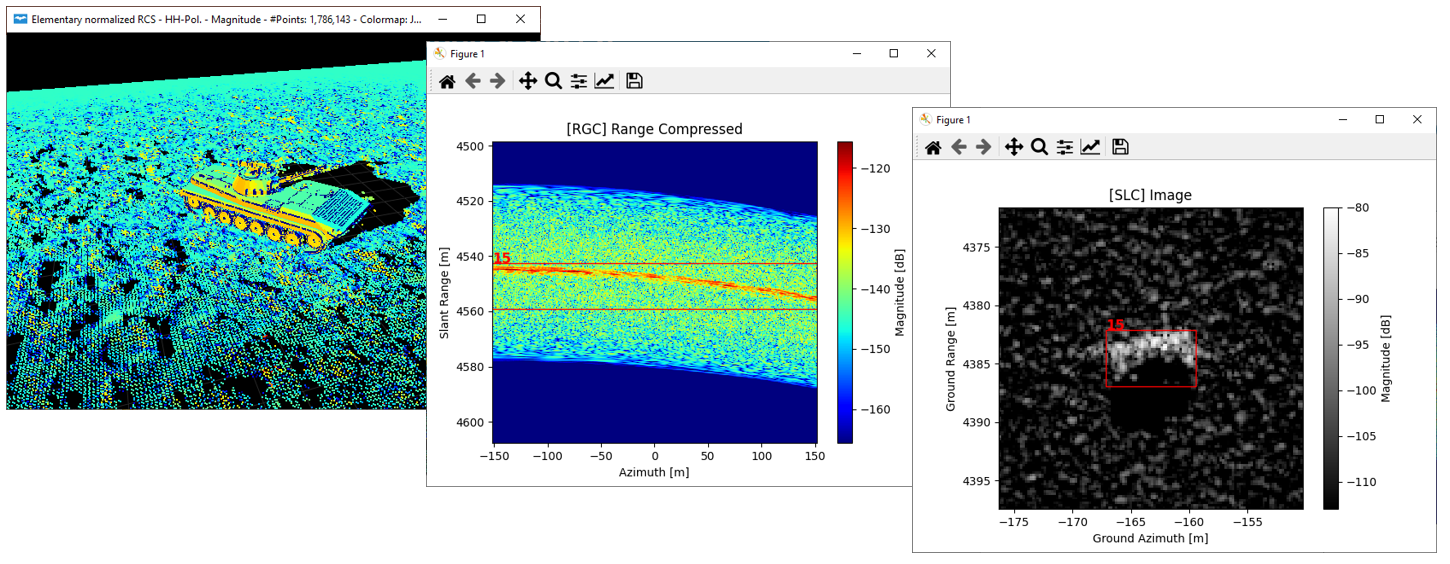


Fig. : Simulated tank (BMP-2) from MSTAR dataset

Table

Description automatically generated with medium confidence

Fig. : SAR images with data augmentation with two targets and multiple look angles to the targets

# Conclusions

This paper provides an introduction to a new realistic fast SAR simulator developed by MetaSensing to provide accurate SAR image databases for ATR as well as large training datasets for AI applications.

The simulator makes use of the new hardware implemented raytracing and hardware acceleration provided by NVIDIA RTX GPU.

KAISAR is being used to simulate some targets of the MSTAR dataset and to provide data augmentation for the different SAR images with some of the targets used in the MSTAR experiment.

The simulation of a benchmark scene with 100 tanks in a 1x1 km scene with 1 m resolution requires around 17 minutes for the complete simulation, while the complete simulation of a single target at 15 cm resolution requires about 15 seconds. The simulation time can be drastically reduced by adding multiple machines or multiple GPU cards.

##### References

1. Walter G. Carrara, Ron S. Goodman, Ronald M. Majewski, Spotlight synthetic aperture radar: signal processing algorithms, Artech House, 1995
2. John C. Curlander, Robert N. McDonough, Synthetic Aperture Radar: Systems and Signal Processing, ISBN: 978-0-471-85770-9
3. Ian G. Cumming, Frank H. Wong, Digital Signal Processing of Synthetic Aperture Radar Data: Algorithms and Implementation, Artech House Remote Sensing Library
4. Jakowatz, Charles VJ, Daniel E. Wahl, Paul H. Eichel, Dennis C. Ghiglia, and Paul A. Thompson. Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach: A Signal Processing Approach. Springer Science & Business Media, 201
5. https://www.sdms.afrl.af.mil/index.php?collection=mstar