A GPU Sonar simulator for automatic target recognition

Jo Inge Buskenesa, Herman Midelfartb, Øivind Midtgårdb

aDept. of Informatics, University of Oslo, P. Box. 1080 Blindern, N-0316-OSLO  
bMÅ FYLLES UT ….

Contact author: Jo Inge Buskenes  
Dept. of Informatics, University of Oslo, P.Box 1080 Blindern, N-0316 OSLO [joibu@ifi.uio.no](mailto:joibu@ifi.uio.no)

***Abstract:*** *Template matching is a common technique used when classifying objects in synthetic aperture sonar (SAS) images. The principle is to isolate an image segment containing an object of interest, correlate it with a set of template images, and assign it to the class of the template yielding the highest correlation coefficient. The challenge is to come up with a suitable set of template images considering that no seabed or object is truly alike.*

*We target this challenge with a sonar simulator that first take as input a seabed model derived from the real sonar image. Then it places an object model on the seabed, renders the scene, and adds the resulting image to the template set. For any object position, alignment, type and material, the procedure is repeated, and a correlation coefficient computed. The faster we are able to perform these simulations, the better we can expect the classification result to be. Therefore the simulator is written in OpenGL and OpenCL and run on graphics processing units (GPUs).*

*The result is a fast performing, mobile and portable on-the-fly template generator which can adapt its behavior to the nature of the current scene. We believe this can prove a powerful tool for mobile sonar imaging platforms such as autonomous underwater vehicles (AUVs).*

**Keywords:** sonar, simulator, ATR, automatic target recognition

1. INTRODUCTION (0.5 pages)

* Why ATR, and current state of ATR in high frequency sonar (skip this?)
* Why template matching?
* Key aspects of template matching and SAS
  + Side –looking. Very high resolution. Resolution and frequency range independence.
* On the use of adaptive templates
  + Key features: Shape of shadow, highlight.
  + Better ATR when trading simulated sonar image accuracy for speed.
* Loading models > adapting camera, render intensity+depth > compute sonar template
* Our work: How SAS image templates can be simulated with OpenGL/CL.
* Improved ATR with improved speed
* Loading models > adapting camera, render intensity+depth > compute sonar template
* Similar work and how this differs from it. Advantages and disadvantages
* Panama City.
* Scotland. Judith Bell. Harriot Watt
* Summary of findings
* Outline
* ???

1. METHODS (2 pages)

* Figure: ATR processing chain and where the simulator belongs in it (skip?)
* How OpenGL works, what it can and can’t do(possible overlap with introduction)
* Design specifics
* Figure: Design build up
* Loading models into OpenGL (assimp, most formats supported)
* Setting up coordinates and transforms (camera+object+seafloor)
* Rendering in OpenGL. What is orthographic rendering and why do we use it?
* Computing sonar template in OpenCL (+figure). How we use depth-map and intensities to form the sonar template
* Short note on speed optimization
* ???
* Where we need to go from here

1. RESULTS (2 pages)

* Case: Flat seafloor, manta mine???
* 3m cylinder, 2m template, flat seafloor
* Submerged Manta mine?
* Cylinder in Isbukta? Difficult since cylinder length varies a lot
* Figure: Viewer images
* Perspective rendering
* Orthographic rendering
* Depth map
* Sonar template
* Speed 50-500 FPS on a 900x800 template
* Figure: ATR performance
* Basic correlation coefficients
* ???

1. DISCUSSION (0.5 pages)

* Were the assumptions sound?
* Simple single scattering model based on Lambert’s law. Sufficient?
* Depth map measures distance to the line of propagation. Is this sound?
* Flat seafloor over-simplifies camera setup. What happens with more complex scenes?
* Relation to previous/other publications.
* Visible interpretation. Good and bad. Elaborates on the assumptions
* Highlight is where it should be and with correct size(?)
* Shadow is where it should be and with correct size(?)
* Transition areas possibly too sharp. Impact on ATR not yet clear
* Moire effects. Suppressed by interpolation. Problematic?
* ATR performance…
* Only two examples. Need more
* What the speed gained from using a GPU allow us to do
* Part of the ATR processing can be done internally in the simulator.
* Templates can be adapted on the fly. Run it in the AUV in the future
* Where we need to go from here
* ???

1. CONCLUSION + REFS (1 pages)

* Summary of findings. Where we wise to use the methods we used?
* Relate to ATR context (“big picture”).
* Promising results(?)
* Remaining challenges and unknown aspects
* Where to go from here
* More on ATR.
* ROC curves