ARL-TR-7335 ● JULY 2015

US Army Research Laboratory

Visualization Development of the Ballistic Threat Geospatial Optimization

by Stephen Allen, Song J Park, and Dale R Shires

NOTICES

Disclaimers

ARL-TR-7335 ● JULY 2015

US Army Research Laboratory

Visualization Development of the Ballistic Threat Geospatial Optimization

by Stephen Allen
Engility Corporation
Chantilly, VA

Song J Park and Dale R Shires
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED (From - To)**
July 2015	Final	October 2014–February 2015

4. TITLE AND SUBTITLE Visualization Development of the Ballistic Threat Geospatial Optimization	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Stephen Allen, Song J Park, and Dale R Shires	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-CIH-S Aberdeen Proving Ground, MD 21005-5067	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-7335
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report acts as an in-depth guide to understanding the full feature set for visualizing the Ballistics Threat Geospatial Optimization code. This guide goes over the necessary steps to run the visualization and set up various scenarios to demonstrate the capabilities of the ballistics threat code. It also includes a detailed breakdown of how to interpret results from running the visualization.

15. SUBJECT TERMS

world wind, ballistic threat optimization, cooperative guard, visualization, precision Monte Carlo

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Song J Park
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	24	**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified			410-278-5444

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

INTENTIONALLY LEFT BLANK.

1. Introduction

Tactical high-performance computing (HPC) seeks to enable supercomputing for tactical operations. It effectively provides sophisticated and powerful computational horsepower at individual Soldier level. The Tactical HPC project conducts research in emerging architectures, programming models, applications, and interface tools. This report focuses on the interface and visualization component of the Tactical HPC research.

Real-time tactical intelligence could prove vital in complex urban environments. The Ballistic Threat Geospatial Optimization application calculates ballistic threat probabilities and performs mathematical optimization to minimize vulnerability while maximizing surveillance. This application exemplifies the enhanced tactical intelligence obtained with the assistance of a tactical HPC. Leveraging commodity accelerators such as graphics processing unit (GPU) and Xeon Phi, high capacity workstation systems can be constructed for reasonable mobility. For a positive impact in tactical operations, an intuitive user interface needs to interact and display results. An open-source, cross platform, and National Aeronautics and Space Administration (NASA)-developed virtual globe called World Wind was chosen as the interface software.

A multitude of features specific to our algorithm was developed under World Wind, which are conveyed in the following sections. This report presents the development outcomes and a guide to creating and running a reconnaissance scenario.

2. Background

2.1 The Problem

The Ballistic Threat Geospatial Optimization code is capable of finding the optimal placement of friendly soldiers in a given scenario. It produces static images that give a wide array of information from threat and watch point field of view (FOV), to Markov chain Monte Carlo optimizations. The purpose of this visualization is to create an easier means of interacting with and setting up scenarios and a more visually impactful way of showing the results of the code's optimization algorithm. To achieve this level of interactivity we leverage the NASA World Wind SDK. Using custom code written in Java, we extend the capabilities of World Wind to meet our visualization needs.

2.2 NASA World Wind

World Wind is an open-source virtual globe first developed by NASA in 2003. World Wind features a fully 3-dimensional (3-D) virtual world that can be manipulated and navigated through. The SDK is built in Java and uses the Java implementation of OpenGL (JOGL) to handle the creation of the 3-D assets. The program overlays NASA and US Geological Survey satellite imagery, aerial photography, topographic globes, Keyhole Markup Language (KML), and Collada files. World Wind gives the user the ability to import 3-D models and navigate through them within an interactive 3-D space. Source code access and further information can be located on the World Wind Java SDK website.[1]

2.3 System Configuration

The demonstration code is executed on a workstation equipped with 4 GPUs. The workstation contains 2 Intel Xeon X5675 3.06 GHz hexa-core CPUs, 4 XFX Radeon HD 6970 GPUs, and 24 GB DDR3 memory running CentOS operating system. The system has a 1,400-W redundant power supply. In terms of software, the following packages were installed on the system to run the demonstration:

- Java-1.7
- JMF2.1.1e
- NASA World Wind-1.5.1
- Coprthr-1.6.1
- Libgeotiff-1.4.0
- Libelf-0.8.13
- Libconfig-1.4.9
- Libevent-2.0.21
- AMDAPPSDK
- GCC-4.9.0
- GDAL-1.11.0
- Gluegen-rt
- JOGL-1.1.1

3. Related Work

A cooperative guard concept for tactical operations in an urban environment has been developed to showcase the capabilities of a mobile HPC. At the core of the ballistic threat algorithm resides a compute-intensive calculation based on ray-tracing.[2] For determining optimal positions given threat and observation points, a Monte Carlo method was employed for locations of minimized threat and maximized surveillance.[3] Furthermore, support for dynamic mission scenario construction and rapid extensibility via XML-driven C++ framework was presented in Ross et al.[4]

World Wind offers vast capabilities for displaying and interacting with geographic information. Demonstrations by examples and applications for World Wind are accessible at the goworldwind web site.[5] Our work, however, incorporates heterogeneous computing with the visual functionality of World Wind. World Wind serves as input and output interface and the ballistic threat geospatial optimization is computed in an OpenCL-compatible accelerator.

4. World Wind Operation

4.1 Code Execution

A shell script named **ballistic-demo.sh** is available to compile and run the code. To run the code, type **./ballistic-demo.sh** in the Linux console. Once the code is compiled, a JFrame window will open displaying the World Wind globe as depicted in Fig. 1. Configuration has been programmed to automatically zoom into the map area with added 3-D building representations. World Wind allows users to navigate the modeled city and set up scenarios as inputs to the simulation.

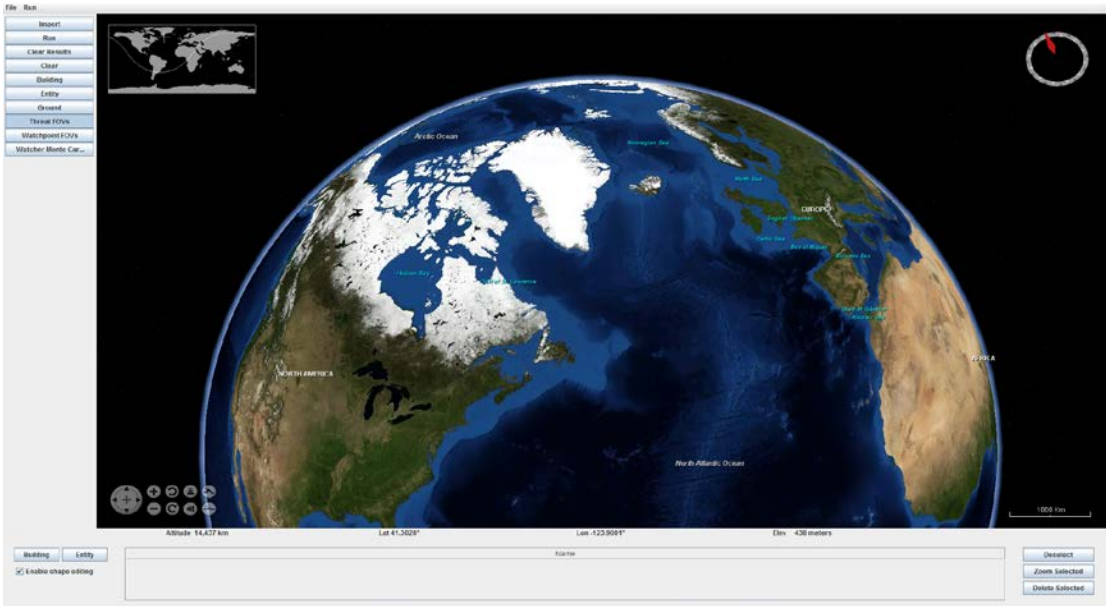

Fig. 1 The initial World Wind viewport at the launch of execution

4.2 Navigating the Globe

World Wind acts as an intuitive user interface to visualize the computationally challenging ballistic threat geometric optimization solved via GPU-accelerated computing. There are several means to navigate the World Wind globe. Options include a keyboard interface, onscreen controls, and the mouse. The mouse controls are the easiest and most responsive to use, so this documentation will focus solely on that option. Table summarizes mouse functionalities.

Table. Mouse controls

Functionality	Mouse Operation
Pan	Left mouse button click and drag—all directions
Zoom	Use the scroll wheel on the mouse
Tilt	Right mouse button click and drag—up and down
Rotate	Right mouse button click and drag—left and right. Note: crossing the top and bottom half of the screen while rotating will change direction.

5. Scenario Execution

5.1 Threats and Watch Points Setup

A placement feature to position entities as inputs to the ballistic threat geospatial optimization algorithm was implemented in World Wind. To run a ballistic threat scenario, a combination of threat and watch point entities must first be placed on the map. These selected positions using World Wind are communicated to the geospatial optimization code. To place entities on the map, you must press one of the **Entity** buttons located at the left and lower-left of the frame as illustrated in Fig. 2. A left mouse click on the map will now place an entity at that position. Entities are represented by a colored semisphere on the globe. The list at the bottom of the view will be updated with the name of the entity once it is placed. You can change an entity's position by selecting it with the left mouse button and dragging it across the map. A selected entity appears white on the map and its name will appear highlighted in the list. By default, any entity placed on the map is a red enemy shooter, but this can be changed by bringing up the entity's context menu.

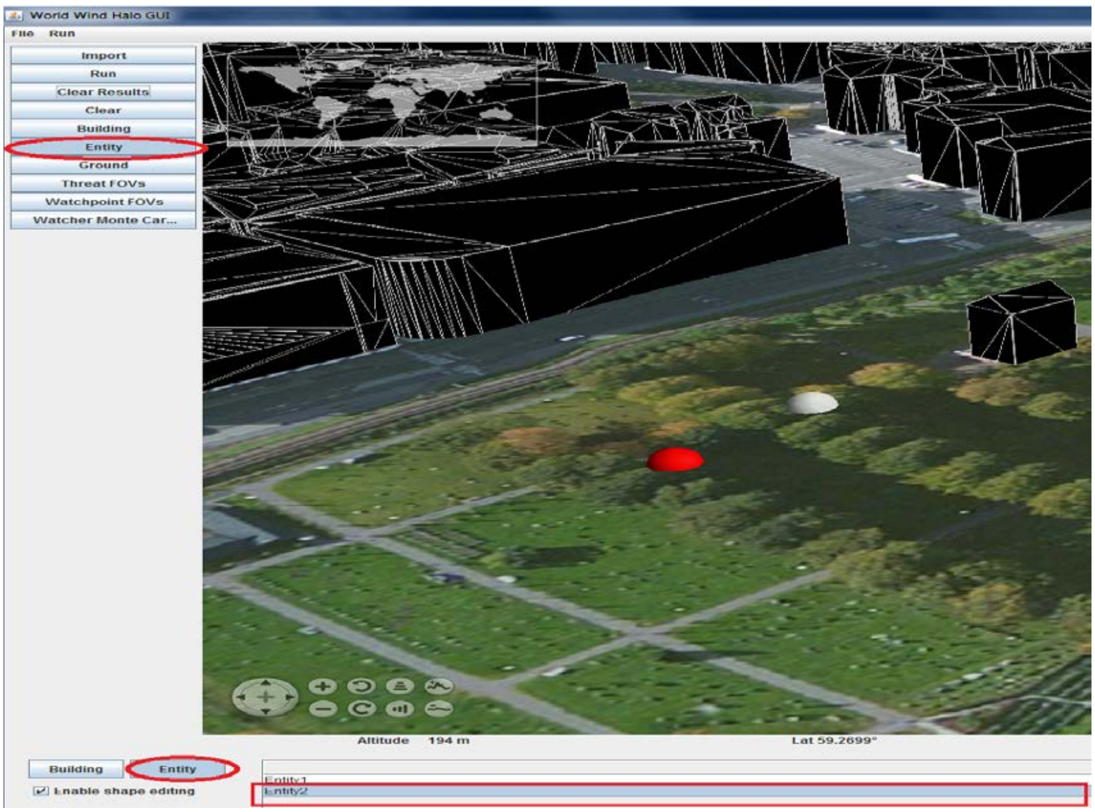

Fig. 2 The position of the entity buttons and the name list on the window

5.1.1 The Context Menu

In addition to entity placement, the context menu developed for entities allows assigning an entity's role and enabling first person view. Right-clicking on an entity brings up the context menu as shown in Fig. 3. There are 3 options provided by the context menu: **First Person View, Soldier Red,** and **Watch Point**. The **Soldier Red** option changes the entity's node color to red and gives it the enemy soldier attributes. The **Watch Point** option changes the entity's node color to green and gives it the watch point attributes. The **First Person View** option brings up the first person view of the selected entity.

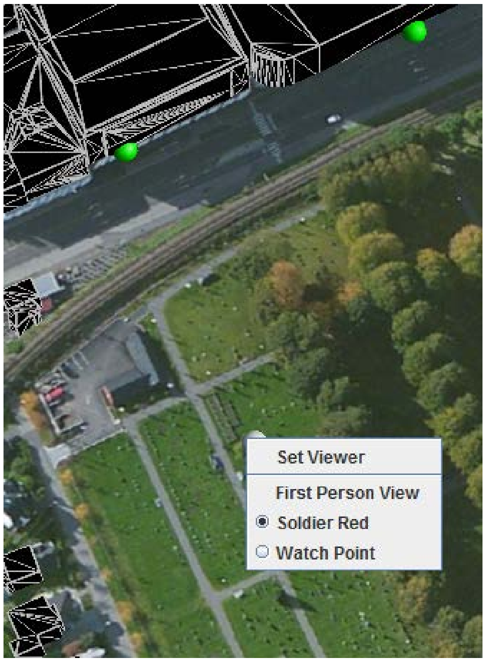

Fig. 3 The context menu for an entity placed on the World Wind globe

5.1.2 The First Person View

The implementation of the first person view feature adds animation to the demonstration that enhances understanding of visualization. After selecting the first person view from the context menu, a **JDialog** box will appear on top of the screen. This new window shows the world from the view of the selected entity (Fig. 4). The view is labeled with the name of the entity and the type of entity (**Watch Point, Shooter**). The view for enemy soldiers (red nodes) pans horizontally from side to side with the center of the arc being a watch point node. This gives a 180° view of the area surrounding the node. The view for watch points is a stationary camera facing enemy soldiers. The views are also dynamic, meaning that as changes occur within the main World Wind globe, they will be reflected in the first person view.

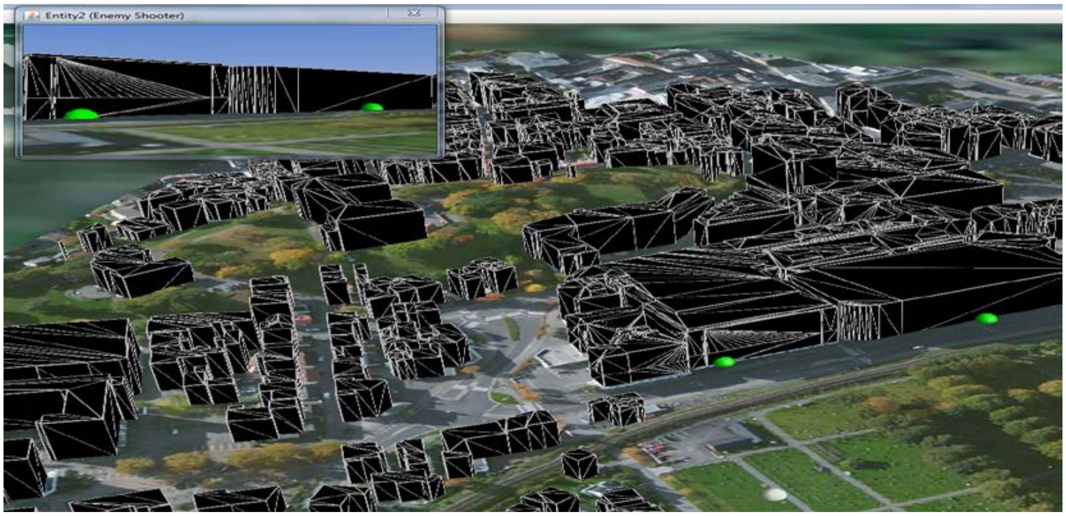

Fig. 4 The first person view of the white node shown at the upper-left corner

5.1.3 Video Capture of the First Person View

This implementation adds a video capture feature to the demonstration. After the first person view is started, each frame of the animation is stored in a memory buffer as a *.jpeg* image. A folder with the name of entity is created on the file system if it is not already present. After the first person view window is closed, the images stored in memory are then converted to a QuickTime movie (.MOV). The video will be placed in the folder and labeled with the entity name, and an index based on the number of videos present will be automatically generated. The video can be played back in any software capable of playing QuickTime movies.

5.2 Execution and Visualization of Results

5.2.1 Initiating Computation

To run a scenario, a minimum of one threat and one watch point entity is required for multiobjective geometric optimization. Selecting the **Run** button from the side panel or the **Run** option from the menu opens the **Server Info** dialog box as shown in Fig. 5. The **Server Info** dialog box is asking for the address of the system in which the core ballistic threat and optimization computations will take place. This server/client model allows for less capable systems, such as handheld or low-end laptops, in a tactical environment to send compute requests to a nearby powerful system. For the case of using the same system, the default values of localhost and port 1234 are selected. After pressing **OK,** the field data (shooter, watch point, and building locations) are sent to the server to determine the placement of watchers (reconnaissance Soldiers) on the map.

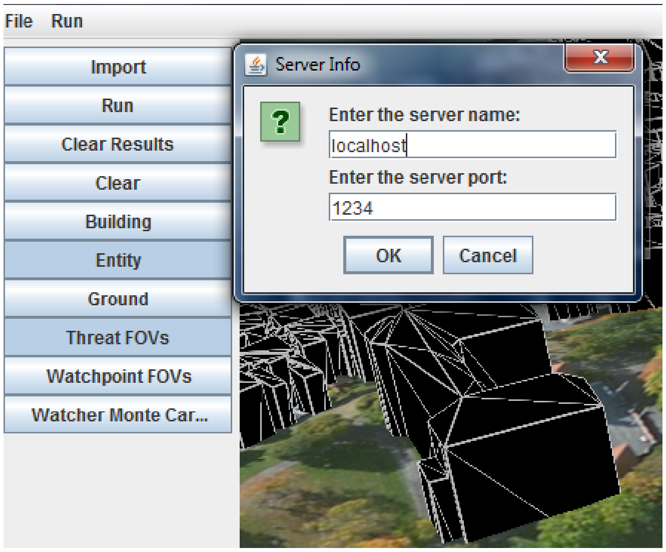

Fig. 5 The server info dialog to enable server/client model of operation

An animation was developed to relay the ray-tracing ray-to-plane intersection calculations (Fig. 6). For each ballistic threat unit, the animation simulates how the server-side code determines the hostiles' FOV. The animation stops when the server has sent the results back. Once the results are returned from the server, a blue node will be placed on the map at the location the code has deemed as the optimal solution to the scenario. The number of blue nodes returned is formulated to have a 1:1 correlation with the number of watch points on the map. The blue nodes cannot be moved from their initial location and do not have a context menu.

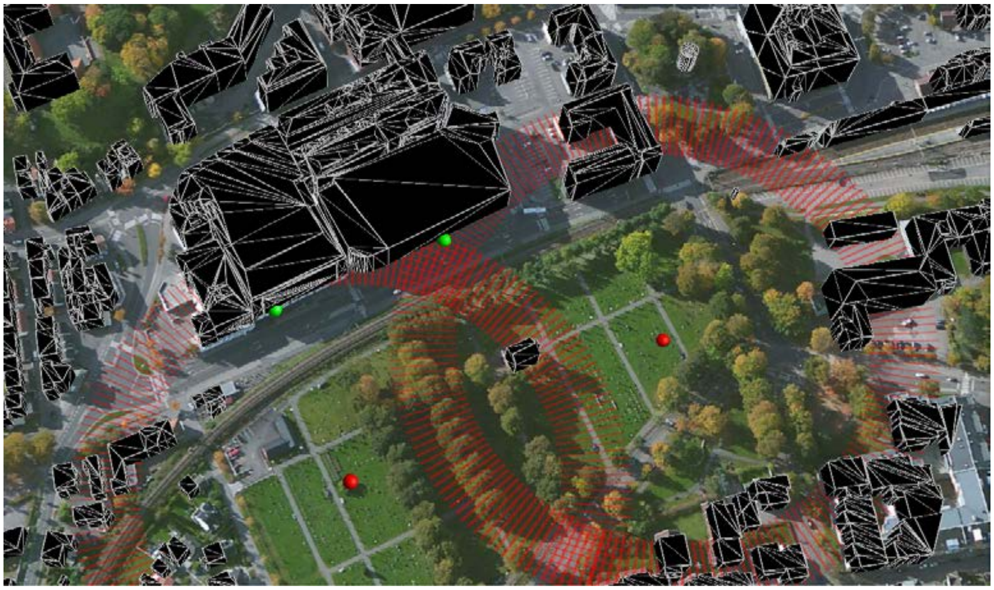

Fig. 6 The ray-tracing animation uses red rays that start at the enemy soldier's position and fan out in all directions interacting with the terrain and the buildings in the area

5.2.2 Images

The server-side code computes and generates several *.png* images and results are overlaid on the World Wind globe for an intuitive view at how the scenario's solution was derived. The output images can be accessed from the **Images** submenu from within the **Run** menu or by selecting their corresponding buttons located on the left button panel. The options include **Ground**, **Threat FOV, Watch Point FOV**, and **Walkers** as displayed in Fig. 7. Clicking the buttons or checking the checkboxes toggles the visibility of the image in question. The **Ground** option exposes the ground plane for the Markov chain Monte Carlo algorithm. The **Threat FOV** is the aggregated FOV of enemy units on the map as depicted in Fig. 8. The **Watch Point FOV** is the aggregated FOV for watch points with cutoff radii to limit extreme proximity to observation targets as shown in Fig. 9. The **Threat** and **Watch Point FOVs** generated are the result of the code's ray-tracing algorithm. The Walkers illustrate a snapshot of the sampled random points for the Markov chain Monte Carlo method as portrayed in Fig. 10.

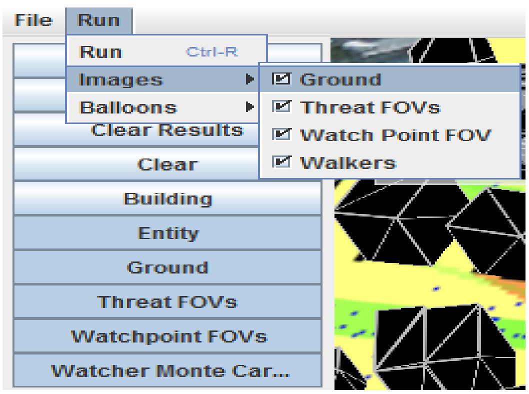

Fig. 7 Submenu from within the Run menu

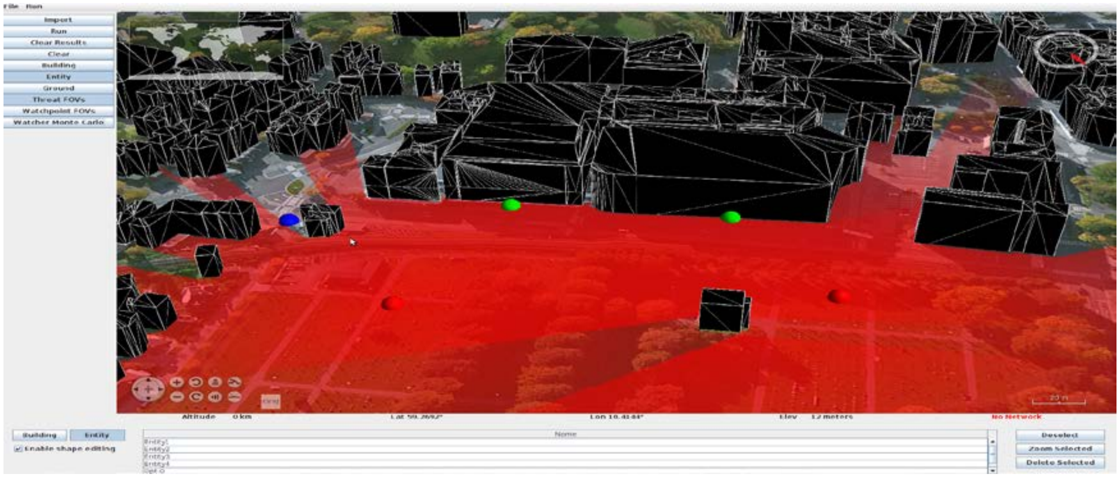

Fig. 8 The Threat FOV visible from hostile shooters on the globe

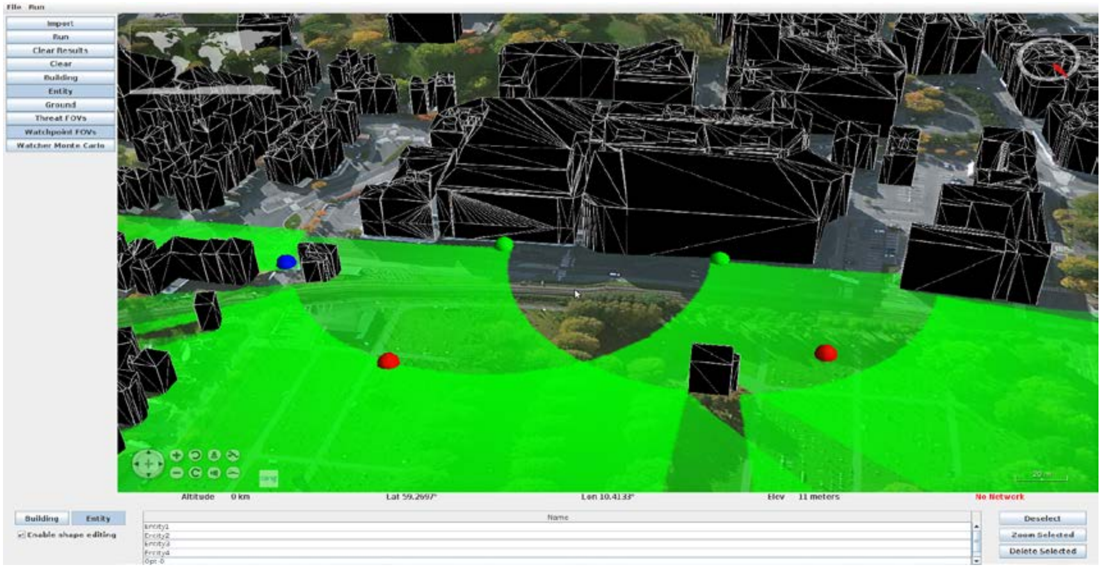

Fig. 9 The Watch Point FOV with cutoff radius

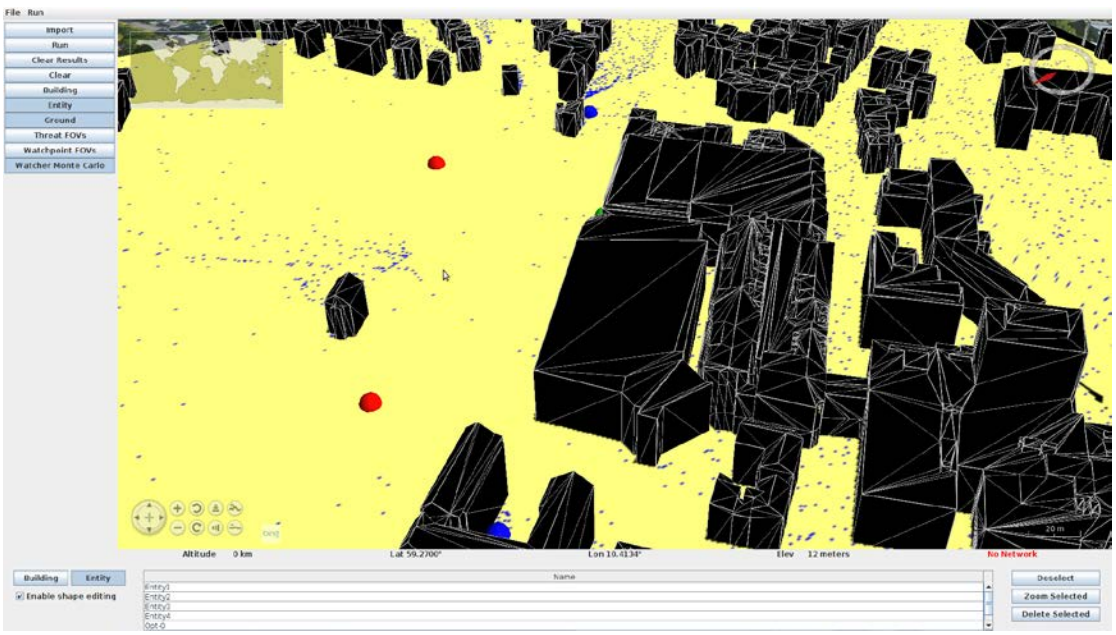

Fig. 10 The ground plane and Watcher Monte Carlo. The yellow area is the domain used in the Monte Carlo simulation. The blue dots are the random points taken from within the domain.

5.2.3 Balloons

Accessing the **Balloons** submenu from within the **Run** menu brings up a series of checkboxes labeled **Shooters, Watch Points,** and **Watchers** (Fig. 11). **Shooters** are for the red enemy units, **Watch Points** are for the green watch points, and **Watchers** are for the blue friendly units. Enabling a checkbox will cause a place marker to appear above all the entities of the type specified for quickly locating entities. Depending on the viewing angle, entities can hide behind building structures. This feature is particularly beneficial in finding the location of optimal blue soldiers placed in hidden areas.

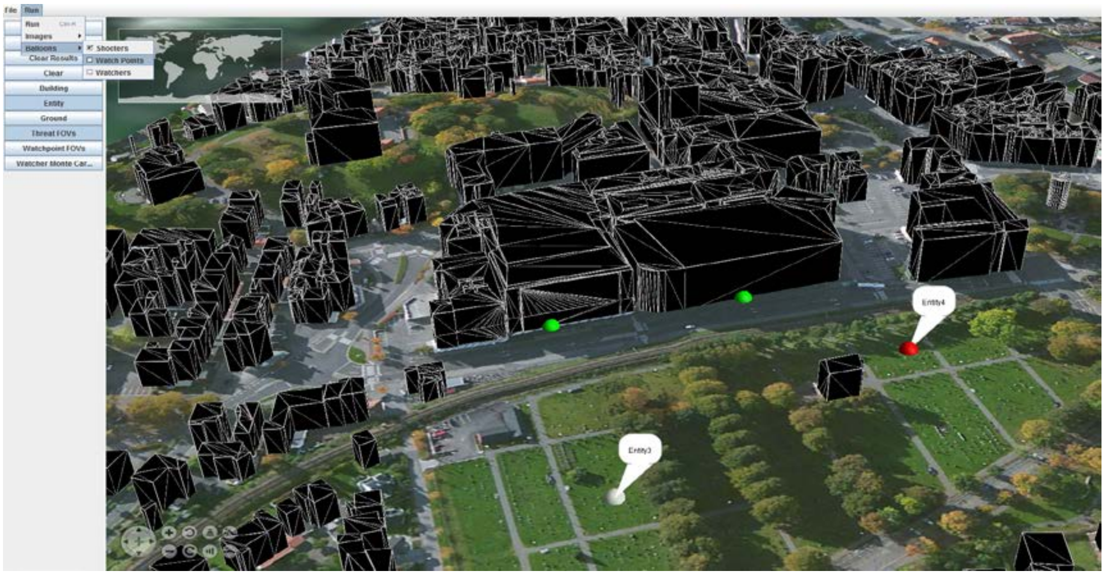

Fig. 11 The Balloons submenu from within the Run menu and the resulting balloon markers from enabling a checkbox

6. Clearing the Map

Clearing options were implemented to evaluate different scenarios. Fig. 12 illustrates the 2 clearing options in the left button panel, which are **Clear Results** and **Clear**. The **Clear Results** option will remove all images and blue nodes (friendly soldiers) from the map. Any watch points (green nodes) or enemy soldiers (red nodes) will still remain on the map in their current position. However, pressing the **Clear** button will remove all nodes and images from the map.

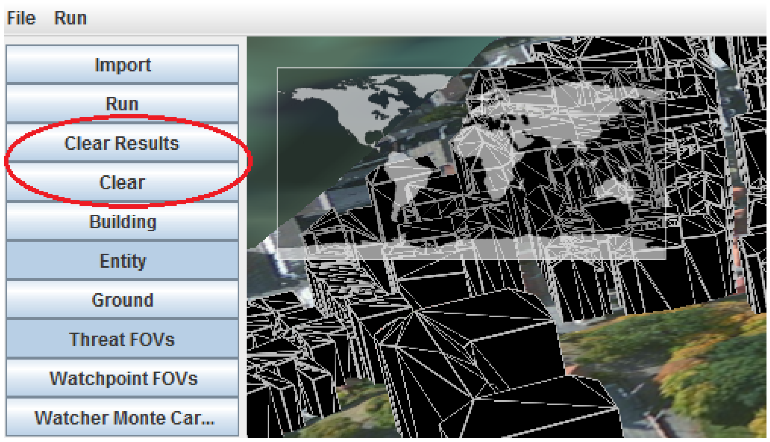

Fig. 12 Clear Results and Clear buttons in the left button panel

7. Conclusion

This documentation provides a detailed guide for using World Wind to demonstrate the Ballistic Threat Geospatial Optimization code. This guide explains how World Wind functions in conjunction with the code and the many options available to the user for setting up and running ballistic threat scenarios. This demonstration illustrates the enabled possibility of tactical HPC in urban environments.

8. References

1. World Wind JAVA SDK. Washington (DC): National Aeronautics and Space Administration; [updated 18 Jul 2011; accessed 26 Feb 2015]. http://worldwind.arc.nasa.gov/java/.

2. Richie DA, Ross JA, Park SJ, Shires DR. Ray-tracing-based geospatial optimization for heterogeneous architectures enhancing situational awareness. CSE 2013. Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering (CSE); 2013 Dec 3–5; Sydney, Australia. Los Alamitos (CA): IEEE Computer Society; c2013. p. 81–86.

3. Richie DA, Ross JA, Park SJ, Shires DR. A Monte Carlo method for multi-objective correlated geometric optimization. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2014 May. Report No. ARL-TR-6928. Also available at http://www.arl.army.mil/www/default.cfm?technical_report=7122.

4. Ross JA, Richie DA, Park SJ, Shires DR, Henz BJ. A class-structured approach to couple application and hybrid core parallelism. PDP 2014. Proceedings of the 2014 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing; 2014 Feb 12–14; Turin, Italy. Los Alamitos (CA): IEEE Computer Society; c2014. p. 681–687.

5. goworldwind.org: Demos. Washington (DC): National Aeronautics and Space Administration; [accessed 2015 Feb 25] http://goworldwind.org/demos/.

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
FOV	field of view
GPU	graphics processing unit
HPC	high-performance computing
JOGL	Java implementation of OpenGL
KML	Keyhole Markup Language
NASA	National Aeronautics and Space Administration

1	DEFENSE TECHNICAL
(PDF)	INFORMATION CTR
	DTIC OCA
2	DIRECTOR
(PDF)	US ARMY RESEARCH LAB
	RDRL CIO LL
	IMAL HRA MAIL & RECORDS MGMT
1	GOVT PRINTG OFC
(PDF)	A MALHOTRA
3	DIR USARL
(PDF)	RDRL CIH S
	S ALLEN
	SJ PARK
	D SHIRES

INTENTIONALLY LEFT BLANK.