

Maritime Surveillance Scenario Simulator

Mitja Vardjan and Jan Porekar

Abstract—As in many areas, typical scenarios are used to aid development of a maritime surveillance system. A software tool was developed to visually present high level aspects of a wide area surveillance scenario. Two basic types of objects are used in a scenario: threats and assets that are searching maritime area for threats. Assets include airborne, space-borne, maritime and stationary objects. Threats are maritime objects such as smuggling boats or oil pollution. An asset or threat is described by its speed, sensor description, detectability, trajectory, etc. From such input data the tool calculates dynamic behavior and any threat detections using spherical Earth model and simple radar equations. Each detection is associated with time, objects involved, text description and an image. The produced output is suitable for visual, animated and interactive presentation in an external geobrowser such as Google Earth.

Index Terms—Animation, detection, simulation, surveillance.

I. INTRODUCTION

There are various illegal activities being conducted in the sea, for example, smuggling, trafficking, polluting, illegal immigration, illegal, unregulated and unreported fishing, piracy and other terrorism. In continuous attempts to prevent such acts the seas are under surveillance by means of stationary radars, satellite images, sea and air vessels, operated by coast guard, police, or other authority. In order to detect illegal activities on time, the surveillance may not be limited to territorial waters, but include also international waters and cover a large area on Earth.

To tackle present and future security needs, there are projects, civil or military, that aim for more integrated and automated surveillance of large maritime area, possibly using unmanned aerial vehicles (UAV) [1]-[3]. In project [1] “scenarios are used to model the tasks and operational environment in which the system should operate”. They also help end-users to analyze system requirements [4]. Scenarios include potential threats, UAVs and other assets searching for threats, geographical area, environmental conditions, etc. As assets patrol the area, they detect threats and report detections to ground control center, which can leave the assets to either continue patrolling or approach the threat to collect more data. In this paper most assets are expected to be unmanned and the role of assets is limited to observation and data collection.

For improved presentation of scenarios to other people, scenarios need to be simulated and the simulation visually appealing. The scene can be set anywhere in the world and the user should be able to change the view during simulation.

The simulation should also be easy to distribute. Therefore, the file produced by simulation tool should be small enough to be sent by email and users should be able to open it with a widely available software tool. Video files encoded with a widespread video codec are an example that meets the last requirement, but they can be too large for email distribution if high quality is preserved and they also lack interactive viewing.

II. RELATED WORKS

There are few virtual globes or geobrowsers available that provide most of the required functionality, some of most developed and famous being Google Earth and NASA World Wind [5]. Geobrowsers do not provide the surveillance simulation, but many of them can load and visualize standard Keyhole Markup Language (KML) files that are typically from few kB to few MB in size and could contain description of the whole simulation.

There are some projects that are using geobrowsers for simulation, but are either not for surveillance simulation purpose [6] or not freely available [7]. For graphical presentation of scenarios own software tool was developed, inspired by [6] and named S³ Tool (SETCCE Scenario Simulation Tool). The tool has been previously presented in shorter version [8].

Additionally, many tools exist that simulate aircraft flight and airborne surveillance from the pilot or cockpit point of view. Some are even based on NASA World Wind [9]. Because such simulators are focused on a single asset only and the camera view during simulation is extremely dynamic, they do not show the overall surveillance picture and are out of scope of this paper. However, they can nicely complement a scenario simulation.

III. SIMULATION TOOL ARCHITECTURE

In order to develop a simulator with limited resources and still meet given requirements a non-standard approach was used. The presentation or visualization layer is strictly separated from the underlying physical model of the system, as shown in Fig. 1. Common standards are used for all visualizations. The maritime surveillance simulation tool exploits an existing geobrowser, namely Google Earth, to visually construct a scenario and save it to a Google Earth native file format, KML. KML 2.2 is an XML language and an official standard for expressing geographic annotations and visualization for two-dimensional and three-dimensional Earth browsers. It was adopted and approved as an official standard by Open Geospatial Consortium (OGC) in 2008 [10]. Scenario can be alternatively saved to a KMZ file, which is compressed KML and can include additional

Manuscript received May 21, 2012; revised June 24, 2012.

M. Vardjan and J. Porekar are with SETCCE, Research Department, Ljubljana, Slovenia (e-mail: mitja.vardjan@setcce.si; jan.porekar@setcce.si).

resources like images and icons.

Java based S³ Tool reads all scenario parameters from the KML or KMZ file, calculates movement of all objects through time and models asset to threat interaction. Detections are calculated during this stage and are modeled in form of scenario events. The simulation generator includes them as additional visual notifications and rewrites the scenario in a format suitable for direct graphical presentation (KMZ file). Resulting KMZ file is of typical size of up to a few MB which can be sent to others via email. User opens the file in a geobrowser that can visualize standard KML code, e.g., Google Earth.

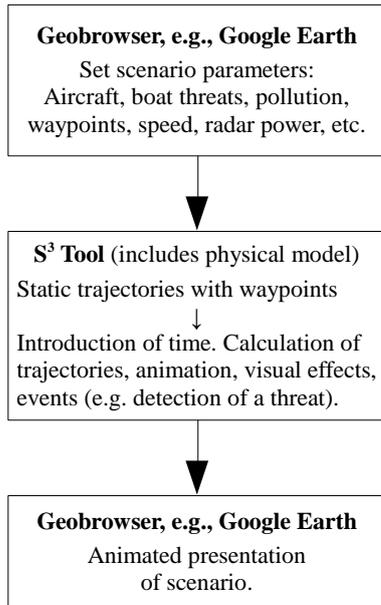


Fig. 1. Simulation main phases

IV. SCENARIO CONSTRUCTION

Simulation scenario can be quickly and easily constructed using parametrization and is conducted in a geobrowser. Defining a scenario does not require additional programming. Any object involved in scenario can have time associated.

For this the user can use the geobrowser GUI to set the planned trajectories of assets and threats. Properties of each object in scenario such as speed, sensors, radar cross-section, boat height, etc., are entered as key-value pairs in the description field of object's trajectory.

V. PHYSICAL MODEL

Objects are classified in two ways: according to their role as assets and potential threats, and according to their physical properties. The latter classification determines the way they are modeled. This, together with sensor specifications and environment model, determines the simulation course and results. Simulation is deterministic.

A. Environment

Earth is modeled as a sphere. Distances along the surface are calculated with Haversine formula (1)(2) for great-circle

distances.

$$h(\theta) = \text{haversion}(\theta) = (1 - \cos(\theta)) / 2 \quad (1)$$

$$h(d/R) = h(\phi_1 - \phi_2) + \cos(\phi_1)\cos(\phi_2)h(\Delta\lambda) \quad (2)$$

The underlying physical model does not model weather, sea state, terrain or communication time-lag.

B. Sensor types

Each asset has at least one sensor attached. The S³ Tool defines general sensor requirements in form of Java interface and provides implementations for radar and camera sensor types. Additional types can be added by implementing the required interface.

1) Radar

Equation (3) describes a simple and widely used radar model [11]. P_t , G_t , S_{min} are radar characteristics. A_e is effective aperture area which S³ Tool also considers as a radar characteristic. Combined with $(4\pi)^2$ these parameters are described as generalized radar power $P_{generalized}$ in (4) providing simple calculation of maximum radar range R_{max} for a target with radar cross section σ . With radar types like synthetic aperture radar (SAR), A_e may not be constant. While S³ Tool does not directly support such radars, it provides ability to change $P_{generalized}$ along an asset's path to compensate for changing parameters like A_e . Equation (3) does not take into account effects of multipath and shadowing that depend on environment details. The simple radar model (3)(4) is an obvious simplification and in a real-world situation, path-loss effects should also be considered. For the sake of simulation of high level overview of scenarios the approximation is sufficiently accurate.

$$R_{max}^4 = P_t G_t A_e \sigma / ((4\pi)^2 S_{min}) \quad (3)$$

$$R_{max}^4 = P_{generalized} \sigma \quad (4)$$

Radar range is additionally limited by line of sight, given by target height, asset altitude and spherical Earth model, as shown in Fig. 2. Slant range is given with (5) from partial slants originating from radar altitude ($slant_a$) and target height ($slant_h$). Partial slants are calculated from (6), which is a derived version of Pythagorean theorem. The classic Pythagorean theorem suffers from significant numerical errors due to extremely small target height h and radar altitude a compared to Earth radius R .

Radar can have a dead angle defined, starting and stopping at any angle relative to movement direction of carrying platform or relative to true North in case of stationary radars. This can model physical obstacles: terrain and buildings in case of stationary radar, or limitations from aircraft and radar construction and placement in case of aircraft radars. Dead angle can be used also to model side looking airborne radar (SLAR), e.g., by setting it from 95 to 85 degrees (Fig. 6).

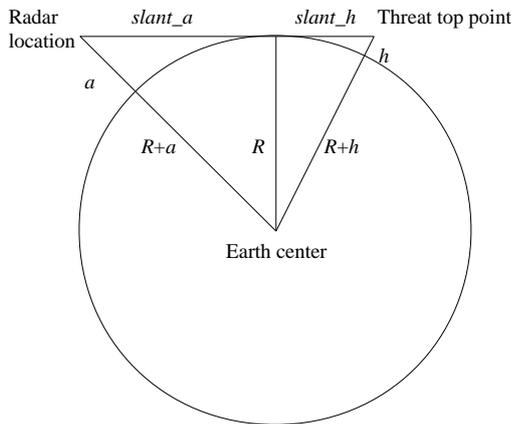


Fig. 2. Line of sight limitation

$$\text{slant range} = \text{slant}_a + \text{slant}_h \quad (5)$$

$$\text{slant}_a^2 = 2 R a + a^2 \quad (6)$$

2) Fixed range sensors

An abstract category of sensors with same range for all threats is supported, but rarely used.

3) Camera

Camera is modeled by the area it covers and by images that are shown during simulation. It usually makes sense to make the area rectangular, but any shape is allowed. Simulator assumes all threats within the area are detected. The picture quality is presented with images, defined in simulator configuration file.

C. Point Objects

Most objects are modeled as point objects without spatial extension. They are either mobile or stationary. The following implementations are provided:

1) Aircraft and Watercraft

A UAV, manned aircraft, patrol boat, etc., is described by its route (using waypoints and optionally containing also altitude data), speed and sensor description. Sensor is usually a radar defined by generalized radar power and optional dead angle. Route can be split into several paths and each path can have its own values for speed and radar characteristics. This can be used as a very simple model for changing weather and sea state.

For each object the physics engine calculates the exact trajectories through time. The calculation is based on a discrete time model. At each moment in time all sensor to threat interactions are calculated. Based on the outcome of interaction comparison the detections and events are created in the model.

2) Threats

Threats are any objects that assets aim to find and observe. These include any boats involved in illegal fishing, smuggling, etc., as well as vessels targeted by search and rescue missions. Point object threats are defined by route, speed, radar cross section (RCS) and height.

3) Stationary radar

Coast based radar stations are defined by position and generalized radar power. They are distinguished from aircraft and watercraft assets by fixed position and thus inability to

change characteristics over time.

D. Oil Pollution

Oil spills are a type of threats, but they are significantly bigger than other objects and are modeled as polygons. With sufficient number of vertices, any 2-dimensional enclosed shape can be modeled.

There are many oil spill models for marine environment, many of them take into account weather, wind, sea currents [12] and are too complicated for this simulator as they would require additional effort from user to enter all environmental parameters to accurately describe the oil spread in scenario definition. A simple model from [13] was used. The model (7) was developed for spreading on placid water. The extent of spread R is a function of sample dependent constant m and time in seconds t . Exponent n is independent of spill sample and size and is 0.87 [13]. Constant m is entered for each pollution in scenario description.

$$R(t) = m \cdot t^n \quad (7)$$

The S^3 Tool models effects of wind, waves and sea current by additional translational movement of oil spill. An oil spill is expected to move much slower than most other objects in a scenario and travel only short distances during a scenario lifetime. For performance reasons, flat Earth model is used for the translation of oil spill.

Properties and dynamics of an abstract polygonal object are implemented separately. Any similar object type or merely different oil spread model can be added by another implementation of the polygon object.

Oil spill is detected as reduced radar reflection from the sea due to smooth oil surface [14]. S^3 Tool models oil spill detectability with RCS.

E. Satellites

Satellites are equipped with camera type of sensor. They are defined by area they cover and by speed and direction of movement of that area. When any type of threat comes within the area covered by a satellite, a special kind of event is generated. Another type of event is generated when a threat moves out of area covered by satellite.

F. Time

The time step of simulation can be set to any positive value. Decreasing the value improves smoothness of simulation (Fig. 3) but also increases output file size and time required for calculation and file processing. Extremely large time step on the other hand can result in errors in form of missed detections (Fig. 4). In most cases this is due to inappropriately configured simulation, however, it can also be used as a feature when simulating sensors with low frequency of data acquisition (high time delay between subsequent acquisitions Δt) and low sensor range, compared to asset speed v (Fig. 4).

If an object has start time associated, it will start moving at that time. Start of simulation is the earliest time of any object or current time if no object has start time set.

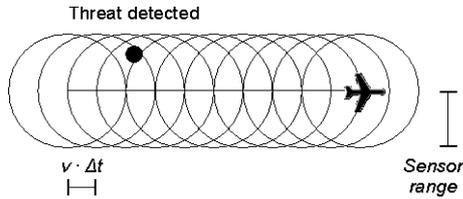


Fig. 3. Sufficiently short time step

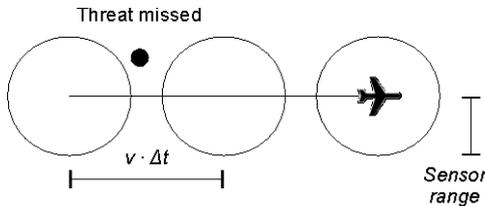


Fig. 4. Simulating low frequency of data acquisition

When an object reaches its final position, the simulator can consider it to either just stop at that location or cease to exist, e.g., boat sinks, UAV lands, etc.

VI. IMPLEMENTATION

The simulator is implemented as a command line Java tool and can be easily invoked from scripts and with various parameters like the time step. Various options like visualization details and text descriptions are specified in the configuration file. Any icons and images can be specified, including online resources. Images available on the internet do not need to be embedded in the resulting scenario file which further reduces the file size.

VII. VISUALIZATION

The visualization is based on KML standard and does not use any Google Earth extensions. Any KML compatible geobrowser should be capable of visualizing the scenario, but it has been found that Google Earth 5.1 performs best. Surprisingly, even newer Google Earth versions (current release is 6.2) do not play the animation as smoothly.

Both terrain overlays and sea overlay maps are provided by the geobrowser. Modeled objects, assets, threats, trajectories, detections, radar ranges at predefined cross-sections (RCS), events, etc. are stored as graphical overlays with additional time and other related information. Because of this they can be animated in the geobrowser. User can choose to play scenario simulation both in real-time and fast forward modes and may control the stage of animation with a slider that denotes the progress of the animation (Fig. 6). The geobrowsers use 3D visualization where camera, view angle and zoom can be interactively controlled by the user.

Any colors, opacity levels, icons, images and text can be easily adjusted. Fig. 6 and Fig. 7 show visualizations in geobrowser using simple comprehensible icons and icons from [15], respectively. All objects and events are listed in the tree view where visibility of any group of objects or events, individual objects or events, or object feature can be

toggled. The map shows current situation. Map in Fig. 7 shows a stationary radar, a patrol boat, an UAV, area covered by a satellite (the semi-oblique white rectangle covering most of map), an oil spill and two point object threats (fishing boats), as well as the trajectories, maximum radar ranges for targets with RCS of 5 and 15 m² and white lines from UAV 3 to all threats currently visible by that UAV. When an asset detects a threat, an event icon is shown on the map in location where the threat was found (the plus icons in Fig. 5 and Fig. 6). If user clicks on this icon, additional information about the event is displayed (Fig. 5 and Fig. 6).

Any other graphical items defined in scenario are transferred into simulation as they are, such as area of interest defined by closed area modeled as closed polygon.

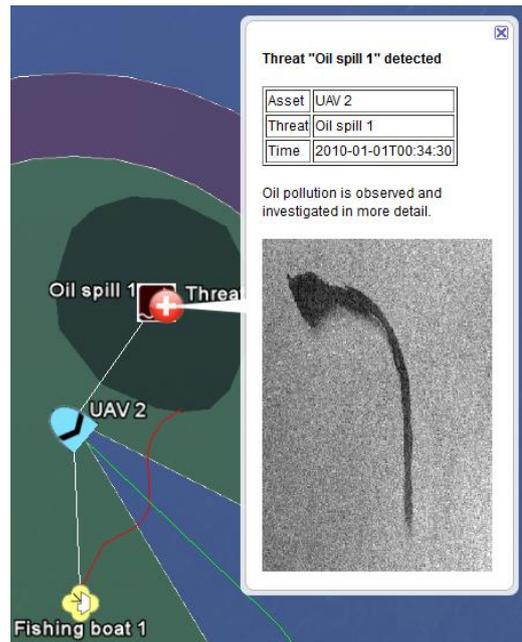


Fig. 5. Visualization of an event. When an event's icon (the plus symbol) is clicked, a balloon with event details is shown.

VIII. SCALABILITY

Two distinct processing bottlenecks shall be distinguished: 1) calculating the trajectories and detections, and 2) visualization in geobrowser.

The former is a one-time process which is done in advance when the scenario is built. The processing time increases approximately linearly with number of threats, number of all sensors on all assets and time step reciprocal. One of the reasons the relation is usually not exactly linear is that various sensor types are associated with various calculation complexities. However, for sensors of same type the relation is linear and the physical model used in calculations is scalable.

More problematic is visualization in geobrowser, because it affects performance every time the scenario is loaded and played. One of the issues with KML based animation is that all moving visible items are described with XML syntax repeatedly for each time point. As threats, assets and time points increase in numbers, the XML-based KML file gets bigger in size, more time consuming to parse and requires more memory to store the parsed data. Fortunately, the

relation is again linear, but the processing in geobrowser may still introduce unpleasant delays when opening a scenario file. For example, a few years old Intel Core 2 Duo based computer with Google Earth 5.1 installed takes 3 seconds to open a scenario with 4 mobile assets, 1 fixed asset, 10 threats, 1 oil spill, 1 satellite, 100 generated events of threat detection

and threat loss, and provides about 30 seconds of smooth animation. Memory usage increases by 60 MB. Even though the time and memory usage increase only linearly with scenario complexity, the tool is inappropriate for scenarios with hundreds or even thousands of threats.

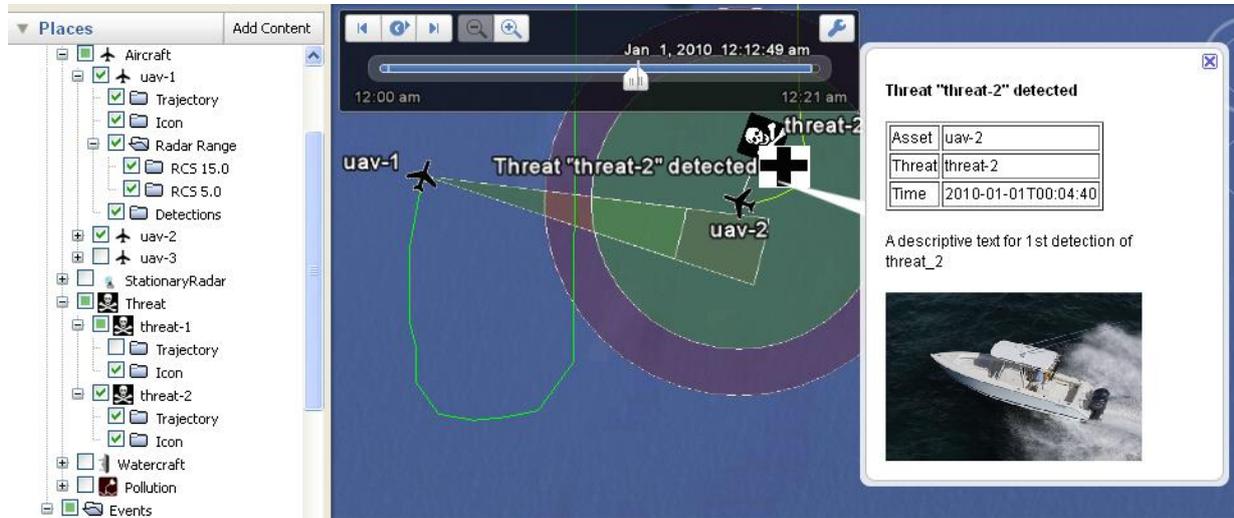


Fig. 6. S³ simulation tool output visualized in Google Earth 5.1. All events, objects and their features are listed in tree view on the left. The tree view is used to toggle appearance of any objects and events in scenario, shown on the map in the middle. On the right, details of an event are shown in a pop-up balloon. Animation (speed, start, pause, etc.) is controlled with toolbar on top center.

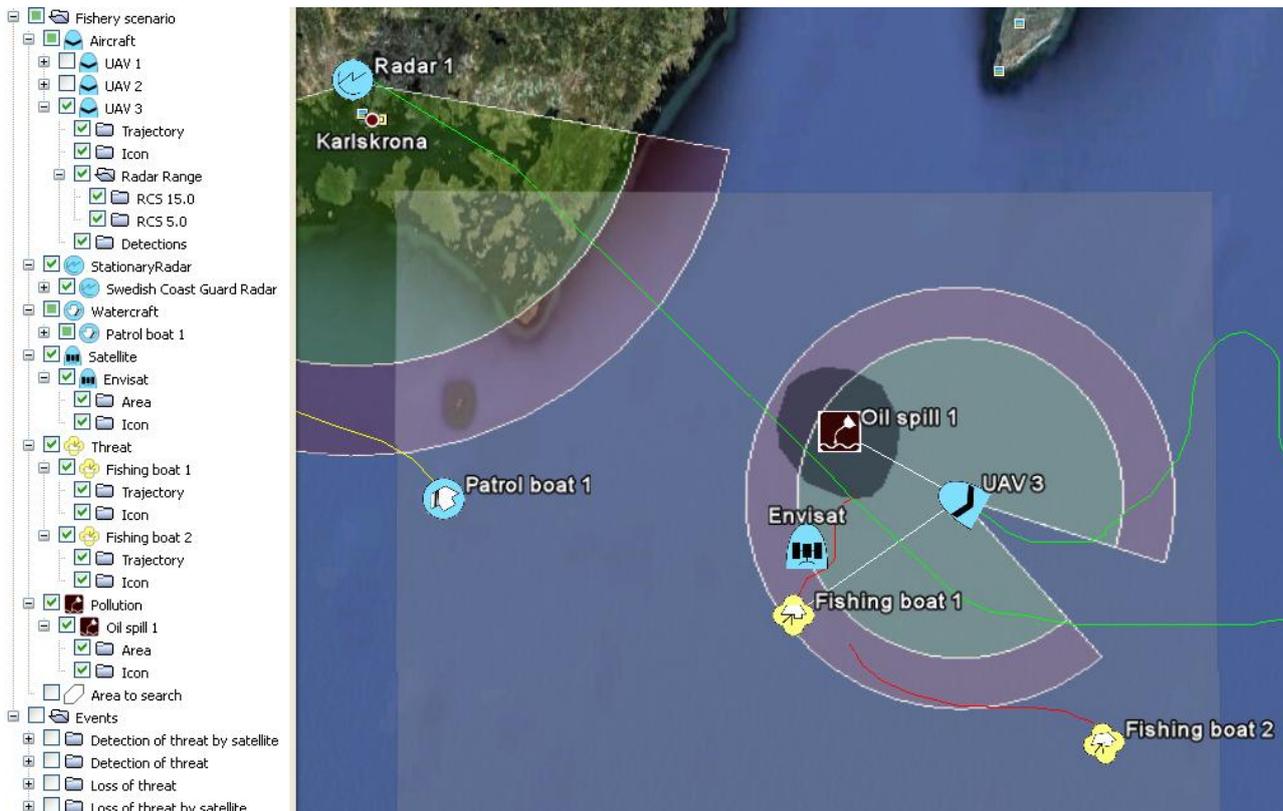


Fig. 7. Various object types shown in the field (right) and in tree view pane (left), using icons from [15]. The map shows rectangular area covered by satellite, a patrol boat with its radar range hidden, a stationary radar, a UAV that is detecting oil pollution and one of the two fishing boats.

IX. CONCLUSION AND FUTURE WORK

Primary intended audience of simulation of illegal fishing scenario are end-users that are being introduced to WiMA²S or other similar projects. As such neither the simulation nor

S³ Tool offers much technical details, but a clear, short, animated, interactive and visually appealing presentation of scenarios. Demonstrating the simulated scenario to external parties helped in improving understanding of complexity of dynamic tasking and automated sensor scheduling

requirements. The detailed simulation of “illegal fishing” scenario also allows non-experts to gain more insight into detailed dynamics of maritime surveillance. In this manner the simulation can also be used as research visual aid for the sensor experts and data-fusion algorithm developers.

Some possible improvements are noted below. Unfortunately, the improvements are either limited or assume a significantly different approach where some of the prominent project objectives would not be met.

A. Optimization of Threat Detection Algorithm

Generation of scenario file could be sped up by intelligent threat detection where any possible detections would not be calculated during each time point, but only when the probability of detection of a particular threat by a particular sensor is greater than zero. The probability could be estimated given both objects’ velocities, the time step, sensor range or coverage, and how “far” was the threat from being detected (or how “well” was the threat detected) in previous time. In a typical scenario, the majority of the threat detection calculations could be avoided with this approach. This improvement does not affect the resulting KML and does not improve visualization performance.

B. Replacing KML with More Efficient Data Representation

To address scalability and visualization performance issues, the scenario data could be stored in a format more tailored to moving graphical objects and by implementing also the visualization part, for example with NASA World Wind Java library or Open GL. However, such implementation would not retain many benefits of the used approach with the KML standard noted in this paper. It would expect the end users to either install the simulation software or to watch only a non-interactive movie file generated by the simulation software.

C. Expanding Usage to Smaller Areas and Micro UAVs

An interesting improvement would be expanding the scope to support airborne surveillance scenarios with small UAVs located in or near buildings where micro-environment with buildings and possibly also three-dimensional terrain would have to be taken into account in the physical model. This could be done either with existing KML approach or with integrated visualization such as NASA World Wind.

ACKNOWLEDGMENT

Authors would like to thank the WiMA²S project consortium [1].

REFERENCES

- [1] Wide Maritime Area Airborne Surveillance (WiMA²S), EU capability project addressing the European Commission FP7 Security Research Call 1 topic “Surveillance in wide maritime areas through active and passive means,” Project No.: 217932, Grant Agreement No.: 217931.
- [2] Broad Area Maritime Surveillance Unmanned Aircraft System (BAMS UAS) program, US Navy.
- [3] S. O’Young and P. Hubbard, “RAVEN: A maritime surveillance project using small UAV,” in *IEEE Conference on Emerging*

- Technologies and Factory Automation*, 2007, doi:10.1109/EFTA.2007.4416878.
- [4] WiMA²S Annex 1 – Description of Work, p. 34.
- [5] NASA World Wind, <http://worldwind.arc.nasa.gov/>, accessed 2012-04-10.
- [6] J. Stafford, “International Flights Animation”, [Online]. http://www.google.com/gadgets/directory?synd=earth&hl=en-US&gl=en-US&cat=featured&url=http://www.google.com/mapfiles/mapplets/earthgallery/International_Flights_Animation.xml, accessed 2010-08-20.
- [7] B. Leininger, J. Edwards, J. Antoniadis, D. Chester, D. Haas, E. Liu, M. Stevens, C. Gershfield, M. Braun, J. D. Targove, and S. Wein, “Autonomous real-time ground ubiquitous surveillance-imaging system (ARGUS-IS),” in *Proc. SPIE Symp. Defense Transformation and Net-Centric Systems 2008*, vol. 6981, 69810H, 2008, doi:10.1117/12.784724.
- [8] M. Vardjan and J. Porekar, “Maritime Surveillance Simulation Tool,” presented at the 3rd International Conference on Computer Modeling and Simulation (ICCMS 2011), Mumbai, India, January 7-9, 2011.
- [9] NASA World Wind Java DiSTI F-16 Flight Simulator, [Online]. <http://www.dist.com/Products/demonstrations/java.html>, <http://video.google.com/videoplay?docid=9190638796406155955>, accessed 2012-04-10.
- [10] Keyhole Markup Language at Open Geospatial Consortium, [Online]. <http://www.opengeospatial.org/standards/kml>, accessed 2010-08-23.
- [11] M. I. Skolnik, “Radar Handbook,” 2nd ed., McGraw-Hill Professional, 1990, p. 1.6.
- [12] M. Reed, Ø. Johansen, P. Johan Brandvick, P. Daling, and A. Lewis, et al., “Oil spill modeling towards the close of the 20th century: overview of the state of the art,” *Spill Science & Technology Bulletin*, vol. 5, no. 1, pp 3-16, 1999, doi:10.1016/S1353-2561(98)00029-2.
- [13] D. O. Njobuenwu and M. F. N. Abowei, “Spreading of Oil Spill on Placid Aquatic Medium,” *Leonardo Journal of Sciences*, issue 12, January-June 2008, pp. 11-24.
- [14] M. F. Fingas and C. E. Brown, “Review of Oil Spill Remote Sensing,” *Presented at Spillcon 2000*, Darwin, Australia, Aug. 2000.
- [15] US Department of Defense interface standard MIL-STD-2525B, Jan. 1999.

M. Vardjan received BSc and M.Sc. degrees in electrical engineering from University of Ljubljana, Slovenia in 2005 and 2008, respectively. Major fields of study included automation, optoelectronics and artificial intelligence.

He worked as Software Developer at VITZ d.o.o., as Research and Development Associate at Optotek d.o.o. and for the last four years he has been a Research Associate at SETCCE, Ljubljana, Slovenia. His research interests include pervasive computing, ambient intelligence, data fusion and simulations.

J. Porekar is the head of research department at SETCCE. He holds a high university diploma in Physics at Faculty of Mathematics and Physics, University of Ljubljana. He has 13 years of experience in software development, system analysis and design, conceptual software design and modelling. He also has extensive experience in leading teams and managing projects in areas of conceptual and applied research, software and system development. During his career he led, designed and developed various pieces of software and IT systems in many different fields application. He was involved in development of both front-end systems with GUIs, back-end services, simulation systems, distributed systems, algorithms design and development as well as real world system modelling. During the last 7 years he was involved in research and development of trust, security, privacy and dependability solutions in complex software development projects. He was the leader of Security and Privacy for DAIDALOS II and leader of the requirements collection activity on the level of business within FP6 project SERENITY and the leader of the central work package within FP7 project PERSIST that has successfully delivered design and prototypes of the Personal Smart Space system. At the moment he is leading the Dependability and Security activity in FP7 SOCIETIES project He is an author and co-author of several papers in the field of information trust, security and privacy in sensor rich socio-pervasive systems.