LAYERED AUGMENTED VIRTUALITY

G. Ahuja, G. Kogut, E.B. Pacis, B. Sights, D. Fellars, H.R. Everett
Space and Naval Warfare Systems Center, San Diego
53560 Hull Street, San Diego, CA 92152
{gaurav, pacis, sights, fellars, everett}@spawar.navy.mil, greg.kogut@navy.mil

ABSTRACT
Advancements to robotic platform functionalities and autonomy make it necessary to enhance the current capabilities of the operator control unit (OCU) for the operator to better understand the information provided from the robot. Augmented virtuality is one technique that can be used to improve the user interface, augmenting a virtual-world representation with information from on-board sensors and human input. Standard techniques for displaying information, such as embedding information icons from sensor payloads and external systems (e.g. other robots), could result in serious information overload, making it difficult to sort out the relevant aspects of the tactical picture. This paper illustrates a unique, layered approach to augmented virtuality that specifically addresses this need for optimal situational awareness. We describe our efforts to implement three display layers that sort the information based on component, platform, and mission needs.

KEY WORDS
robotics, unmanned systems, augmented virtuality, multi-robot controller

1. Background

Supervising and controlling autonomous robotic behaviors requires a suitable high-level human-robot interface to facilitate an increased understanding of the robot's actions and intent, better perception of the information provided by the robot, and an overall enhancement of situational awareness. The Robotics Technology Transfer Project (TechTXFR) sponsored by the United States Department of Defense Joint Ground Robotics Enterprise and managed by SPAWAR Systems Center, San Diego (SSC San Diego) is tasked to evaluate and improve both robotic platform and interface technologies to meet emerging warfighter needs. The TechTXFR philosophy is not to develop the needed technologies from scratch, but leverage the investments already made in robotic R&D by building on the results of past and ongoing programs. The technical approach is to identify the best features of component technologies from various resources (e.g. academia, other government labs/agencies, and industry) and fuse them into a more optimal solution. Therefore, instead of focusing on a single technology solution, the outcome is a blend of complimenting ones that can overcome the caveats of individual technologies. For example, to address the obvious need to improve the navigation capabilities of the baseline tele-operated systems, TechTXFR developed an adaptive method[1] to fuse traditional local and global localization algorithms. The end result allows a robot to seamlessly navigate between outdoor and indoor environments. In the same regard, TechTXFR is fusing existing advanced interface methods to develop a layered augmented virtuality solution to facilitate optimal command and control of multiple robotic assets with maximum situational awareness. Existing methods used include a 3-D interface from the Idaho National Laboratory (INL), SSC San Diego's Multi-robot Operator Control Unit (MOCU), and Google Earth. INL's interface and the Multi-robot Operator Control Unit are both capable of incorporating data from a wide range of sensors into a 3-D model of the environment. While INL's interface is designed for high-performance, real-time mixed-initiative and multi-perspective control of a robot[2], SSC San Diego's Multi-robot Operator Control Unit, while also real-time, is focused on being portable and configurable for use in multiple applications. Its underlying modular framework allows for quick swapping of modules, such as communications protocol module, map module, and video link module[3]. Our layered approach integrates the advantages of both interface systems with Google Earth to develop an enhanced augmented virtuality interface. This paper describes the various technologies and component interfaces, followed by an explanation of how they are integrated in this initial proof-of-concept implementation.

Why Augmented Virtuality?

Augmented virtuality, like augmented reality, is a type of mixed-reality user-interface. The taxonomy of mixed-reality interfaces, introduced by Milgram[4,5] describes methods of combining real-world and computer-generated data. While augmented reality involves adding computer-generated data to primarily real-world data, augmented virtuality deals with primarily real-world data being added to a computer-generated environment. Augmented virtuality, rather than augmented reality, is viewed as an ideal tool for a robot-human interface because the latter suffers from the registration problem – aligning the user's location and perspective in the environment with the overlaid data. With augmented virtuality, the robot can use its existing estimate, usually accurate, to display itself registered appropriately in the virtual model, making it easier for the user to comprehend. Another feature that makes augmented virtuality useful for robotics is its flexible ability to display various kinds of data while operating under much lower real-time bandwidth requirements than a video-based augmented reality system. This is of particular importance in military robotics where ideal wireless radio links are often unavailable.

Therefore, augmented virtuality is pursued here to provide a single, unified interface to the user, regardless of the complexity of the robot, environment, or application. It provides an inherently flexible and scalable architecture for data visualization compared to more conventional interfaces such as video and gauges.

2. The Component Interfaces

Idaho National Laboratory's 3-D Interface

INL's 3-D robotic control interface was originally developed at Brigham Young University[6] and has helped pioneer the use of augmented virtuality in robotic control. It provides a cognitive collaborative workspace (CCW) as a framework for allowing a user to "share" the same environment as the robot, as well as a unified framework for both data visualization and robot tasking. INL has explored, documented, and tested such functionality with mixed initiative controls and virtual 3-D camera views[7]. Figure 1 below shows the 3-D interface with the laser-based map generated by the robot and signifying the interior walls of the building, and the camera video feed registered in the world model with respect to the camera's current pan and tilt angles.

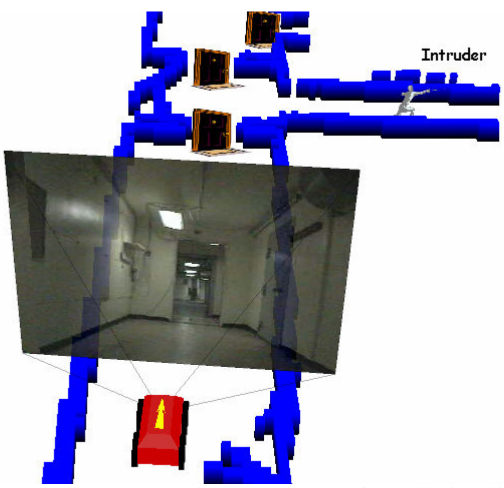

Figure 1. A screen capture of the INL 3-D Interface with the video feed displayed in front of the robot as the robot maps the interior of a building. Door icons represent doorways, and the fencing human icon symbolizes human presence found by the robot.

Multi-robot Operator Control Unit

The Multi-robot Operator Control Unit (MOCU) was developed by SSC San Diego to simultaneously command and control multiple unmanned vehicles and sensors of any type in any environment. For example, demonstrations have been made to simultaneously monitor Unmanned Ground Vehicles (UGVs), Unmanned Surface Vehicles (USVs), Unmanned Air Vehicles (UAVs) and Unattended Ground Sensors (UGSs), as well as command any of those vehicles or sensors individually using its required interface configuration. In monitor mode, the operator can view multiple vehicles, including an overview status window of each (Figure 2). The operator can then select any robot individually at a given time to command that vehicle, whereby the interface configuration becomes specific to that vehicle. In Figure 3 below, the operator selected the USV, which brought up the digital nautical charts, gauges, and other equipment relevant to the USV.

The ability to use MOCU with any robot and any screen interface required is the result of a very modular design, where any developer can add any map format, video format, communications protocol, path planner, etc. as a module. Any preferred layout of the screen interface for any vehicle can also be easily included as a simple XML-based configuration file. MOCU's scalability also allows implementation on various hardware controllers, ranging from handheld devices to multiple rack-mounted computers as required by the vehicle or mission application.

MOCU's basic framework will be used to integrate the three component interfaces together to create a unified one whose utility exceeds that of these interfaces individually. Both Google Earth and INL's 3-D interface will be included as modules under MOCU, where they will appear as overlapping layers of information.

Google Earth

Google Earth is currently the ideal tool for prototyping 3-D user interfaces based on aerial imagery. Google Earth's Keyhole Markup Language (KML)[8] has become a platform-independent Web standard for overlaying 3-D information on Google Earth's impressive aerial imagery. While Google Earth was originally designed to display static information, later versions have become adept at facilitating the display of real-time sensor data, as is required for a robotic user interface. SSC San Diego has adopted Google Earth as an effective and functional rapid prototyping tool for exploring the possibilities of an extensive augmented virtuality robot interface. The core augmented virtuality capability, however, remains independent of the interface tool (e.g. INL 3-D, MOCU, or Google Earth), which allows the interface to be tailored to the application or user.

3. Layered Approach

Our approach currently divides the virtual world model into three layers. The top layer is the big-picture, bird's-eye view of the world showing the various autonomous

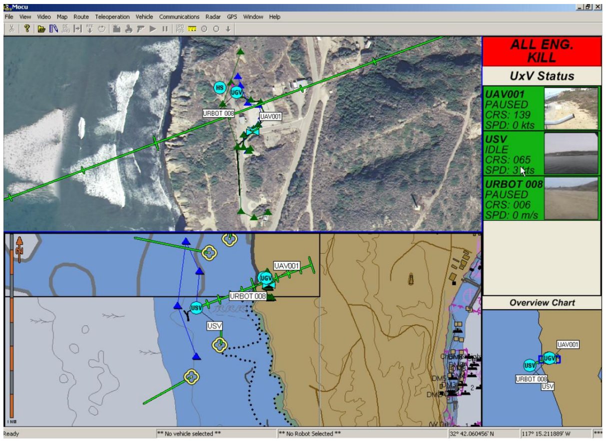

Figure 2. In monitor mode, the operator can view all vehicles (light blue circles) simultaneously. An overview status window (top right) gives basic information, like position, speed, and the video feed from each vehicle.

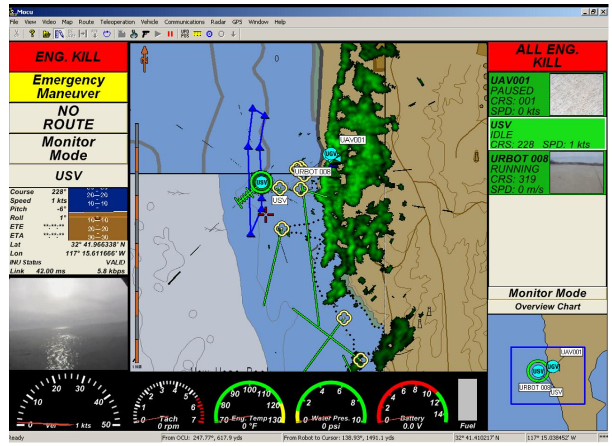

Figure 3. This USV screen interface was configured to display the nautical chart, the radar information (green "cloud" in center window), the live video feed (bottom left), and ship gauges (bottom row).

vehicles and unattended sensors contained in the overall *mission tactical picture*. The middle layer is the detailed zoomed-in view of an area showing maps generated by the autonomous robots and filled with icons of information requested by the operator, constituting a *platform-generated picture*. The inner layer is the *component-focused picture* that could be the interactive vehicle- particular world, sensor-particular information, or other flagged anomalies for which a vehicle found and saved additional information.

Mission Tactical Picture (MTP)

This layer encapsulates the complete tactical picture, displaying all the various unmanned vehicles (UXVs) and unattended ground sensors (UGSs), etc. Along with the current location of each vehicle, it also displays some high level information about each object, including its mission/goal, autonomy level, and health (communications strength, battery power and mission critical sensor status). In addition to displaying crucial pieces of data from each asset in the model, the MTP flashes warnings, errors, or other urgent messages when necessary. This layer is for the most part non-interactive, similar to MOCU's monitor mode, where the operator isn't allowed to influence the UXVs or the UGSs, except for halting or shutting them down in an emergency.

Implementation of this layer involves marrying MOCU with Google Earth, or using Google Earth as a module for MOCU. This approach allows us to not only display 3-D models of our unmanned/unattended assets, but also to better understand the environment these assets are in when visualizing them in a 3-D world model. The effectiveness of this presentation is enhanced by the ability to provide the user with options on what he/she would like shown in his/her world, e.g. the kind of data or anomalies, the types of assets (UGVs, UAVs, etc.,) deployed for a particular mission and so on. The aforementioned items constitute only a small portion of the wide variety of options the user can choose to personalize his/her world dynamically.

In the final implementation of the MTP, the assortment of options will be presented to the user in the form of check boxes and radio buttons generated from a database using KML. The database used here will be created dynamically based on the sensor suite and capability messages sent back from the UXVs and UGSs. The database will also change dynamically as other assets are added or removed from the model based on sensor(s) malfunctions and/or availability of capabilities due to changes in the immediate environment. Consequent to the variations in the database, the options presented to the user will be altered to maintain an optimal situational awareness.

Our current implementation uses static KML files for Google Earth, and XML configuration files for MOCU, to generate these options and screen layout for the UGVs being used in the initial setup.

Platform-Generated Picture (PGP)

Digging deeper into the virtual world by zooming in shows a more detailed picture augmented with specific perceptual information. One of the most useful among these perceptions is the robot-generated 2-D or 3-D map overlaid on Google Earth's virtual world model. Our current unmanned systems build maps of their surroundings using planer 2-D ladars, which are then extruded to a certain height to create pseudo-3-D representations that are displayed as walls on the virtual model. These added maps help in two ways. Map data of the exterior augments Google Earth's virtual world with the latest information, such as perceived walls, buildings, etc. Interior maps overlaid on top of exterior building structures provide accurate blueprint-like representations. Other perceptive information is displayed as icons signifying the availability of additional information. The type of information is again either mission-oriented or user-influenced, based on the options available on the platform. Figure 4 below shows the interior map of an underground bunker displayed on Google Earth, along with specific information icons as requested by the user.

The SSC Google Earth robot display uses the Python interface to the open-source Player/Stage robot architecture[9] to retrieve robot sensor data over Ethernet network connections. Each sensor type is handled by a single Python script. Each script is implemented as a CGI 1.1 script, the data to be served by most conventional Web servers. Each script outputs KML 2.1 code. For example, the script which handles robot position will output the KML code to place a 3D model of the robot in the correct geodetic coordinates for the robot. The master KML file uses the KML *NetworkLink* tag to periodically update expired data by calling the appropriate CGI script to refresh the corresponding KML code segment. Empirical testing shows that the maximum update rates for this method are about 5Hz. The types of Player/Stage interfaces currently supported are robot position and heading through IMU or odometry, GPS, LADAR, sonar, map, image, and power. The update rates for each sensor type are independently configurable. Once the data is received, all further rendering is handled by Google Earth. Each interface is independent of the others and can be displayed depending on mission and robot configuration.

Implementation of this layer is quite a challenge, despite the fact that all of the rendering is handled by Google Earth's internal engine once that data has been provided in KML file format. The non-trivial part of this layer includes transforming local maps from each robot into the global coordinate frame, and then being able to update the KML file continuously as new map data is received from the robot and transformed. At present, local-to-global coordinate transformations are performed by the vehicles' adaptive extended navigation Kalman filter once data has been initially transformed to a geodetic coordinate system. The data is currently being served using the Player/Stage map interface[9]. Any Player client can then receive real-time geodetic map data, which is made available as an occupancy grid. This Player map data is later converted into a list of points representing walls, making the KML code segment relatively compact and easier to draw on Google Earth.

In the current implementation, the SSC San Diego KML file includes a network link that periodically polls the Player server for new data. When a new map is available, the data is transferred, converted to KML format via a Python script, and then rendered. The final implementation will allow for such data to be passed through MOCU's communication protocol, similar to how the rest of the data is currently being made available to MOCU using the Joint Architecture for Unmanned Systems (JAUS)[10] protocol.

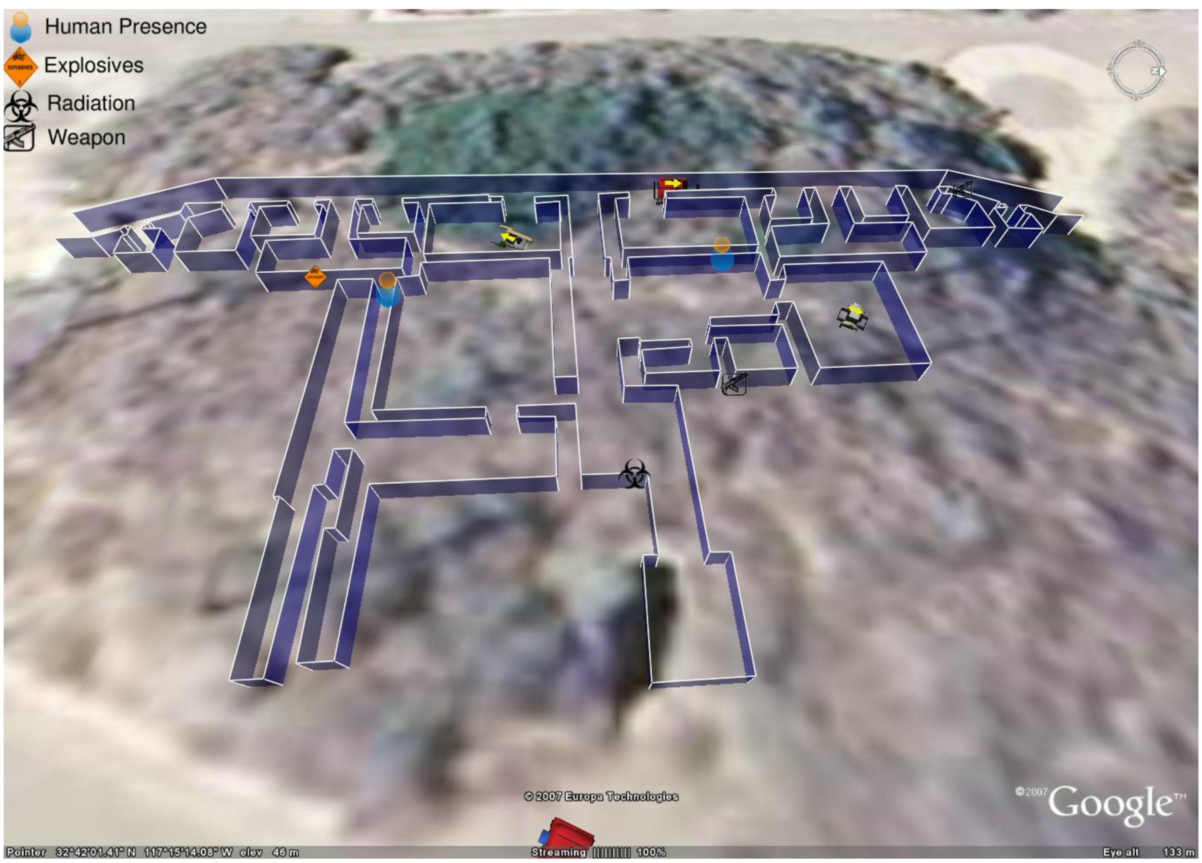

Figure 4. Platform-generated picture of an underground World War II bunker. The view shows all vehicles simultaneously (ATRV Sr., Packbot, Segway RMP and ATRV Mini shown here), the map built by the vehicles, and multiple information icons.

Along with the availability of additional information in this layer, there is also a provision for some high-level interaction with a selected UGV or UGS. Based on the behavior/capability suite available in the vehicle's current environment, the users are provided with additional platform-specific options allowing them to influence the vehicle in certain ways. Some of these commands include providing a list of waypoints to follow, commanding a UGV to explore an unknown area or search a previously explored area for certain anomalies, etc.

Component/Robot-Focused Picture (CFP)

Drilling down even further displays the most detailed interactive world, which is in many ways detached from Google Earth's virtual world model and yet still augmented by it. When a particular UXV is selected for control, a new window is overlaid on Google Earth that provides a first-person perspective on that UXV's world. In this view, the user is only focused on that particular vehicle and is able to influence it in many ways, from simply tele-operating using the keyboard or a joystick, to waypoint navigation by clicking points on the window, or by clicking buttons like "Retro-Traverse" on the window itself. Since this layer is superimposed on Google Earth, the operator is still able to oversee and monitor other assets that are in view.

Double-clicking on a particular information/anomaly icon also overlays a new window on Google Earth, displaying detailed data such as an image showing certain anomalies in the environment, a video capture showing human presence, or a radiation chart for a radioactive source, etc. similar to figure 5 below.

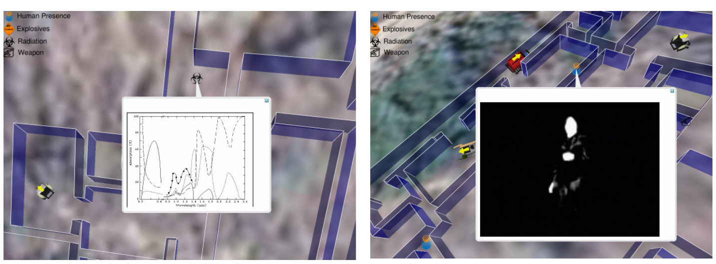

Figure 5. (left) Double-clicking on the Radiation icon pops up a window displaying a radiation chart on this source. (right) Double-clicking on the Human Presence icon pops up a window showing the thermal image captured by the robot's onboard FLIR camera[11,12].

Current implementation of this layer is accomplished by augmenting INL's 3-D interface with aerial imagery. The 3-D interface allows for setting a background image by point correlations between the aerial image and the 2-D map being generated by the UGV. This provides a real-time interactive game-like interface for command and control of UXVs. Figure 6 is a snapshot of the INL 3-D interface with an embedded Google Earth aerial image while an iRobot ATRV (All Terrain Robotic Vehicle) autonomously explores an underground World War II bunker.

Figure 6. Overlaying the robot's world view of the interior of an underground World War II bunker on Google Earth.

The final implementation of this layer will be accomplished by blending the 3-D interface with MOCU and Google Earth. The INL interface will be added to MOCU as a communication/map module allowing for data sharing, mouse-click captures in local or global coordinate frames, and issuing of other commands to the vehicles. Along with the complexities of tying INL's interface into MOCU, the bulk of the work will involve correctly placing the interface at the appropriate location on Google Earth. This will require some non-trivial multiple-coordinate transforms, researching methods for automatic adjustments during zooming and continuously updating both layers as the users change perspectives in either view.

4. Future Work

Current capabilities only allow us to augment 2-D maps generated by the vehicles, but with 3-D flash ladar and ranging cameras technologies becoming more mature and miniaturized, the next logical step is to perform the mapping in 3-D. Additional work will be needed to be done to optimize the algorithms and compress the data to make it manageable through a low-bandwidth communications link. The potential for improvement in situational awareness for the warfighter, however, is too great to ignore. We have, therefore, investigated wallpapering the virtual world with live video textures, allowing for a more live realistic view of the world. Much of the work on this approach is being done at INL[13] where additional methods are being researched in this area. We anticipate that such a capability would allow for updating Google Earth's 3-D maps with live images while augmenting the interior maps of building structures.

Another planned feature addition will provide virtual tours of the generated world. Users could easily "fly" through previously explored regions once all the information is stored in a large database. This feature would allow for forensic use of robot sensor data. For example, radiation levels in an explored building could be viewed in both a temporal and spatial context to improve understanding of sensor data by an operator.

5. Conclusion

This paper presents a unique approach to improved situational awareness by using a layered augmented-virtuality interface as the underlying means for improved human-robot interaction. This proof-of-concept effort leverages capabilities from two existing multi-robot controllers, SSC San Diego's MOCU and INL's 3-D interface, as well as a publicly used 3-D virtual world viewing tool, Google Earth. The layered approach is being tested as a solution to sort, present, and comprehend tactical, platform, and component level data that can easily support any new robot, environment, and/or mission to meet the emerging needs for more intelligent robots and sensors.

References

[1] Pacis, E.B, B. Sights, G. Ahuja, G. Kogut, H.R. Everett, "An adapting localization system for outdoor/indoor navigation," SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense Security Symposium, Orlando FL, 17-20 April 2006

[2] C.W. Nielsen, D.J. Bruemmer, D.A. Few, M. Walton. "Mixed-Initiative Interactions for Mobile Robot Search,"American Association for Artificial Intelligence Mobile Robot Workshop. Boston, Massachusetts, July 16-20, 2006.

[3] Bruch, M. H., "The multi-robot operator control unit (MOCU)," SPIE Proc. 6230: Unmanned Systems Technology VIII, Defense Security Symposium, Orlando, FL, April 18-20, 2006.

[4] Milgram, P., Kishino, F, "Augmented Reality: A Class of Displays on the Reality-virtuality Continuum." SPIE Proc. 2351: Telemanipulator and Telepresence Technologies, 1994.

[5] Milgram P, Kishino F, "A Taxonomy of Mixed Reality Visual Displays," IEICE Transactions on Information Systems, Special Issue on Networked Reality, Vol. E77-D, No. 12, December, 1994.

[6] C. W. Nielsen, "Using Augmented Virtuality to Improve Human-Robot Interactions." Dissertation submitted to Brigham Young University, February 2006.

[7] D.J. Bruemmer, D.A. Few, H. Hunting, M.C. Walton, C. Nielsen."Virtual Camera Perspectives within a 3-D Interface for Robotic Search and Rescue," Proc. of the ANS / IEEE 11th Annual Conference on Robotics and Remote Systems for Hazardous Environments , Salt Lake City, UT, Feb. 12-15, 2006.

[8] KML Reference: http://earth.google.com/kml/kml_tags_21.html

[9] Player Project Reference: http://playerstage.sourceforge.net/

[10] Joint Architecture for Unmanned Systems: http://www.jauswg.org/

[11] Kogut, G., Ahuja, G., Pacis, E.B., Carroll, D., Giles, J., Rogers, Dr. B, Nanzer, J, "Sensor Fusion for Automatic Detection of Human Presence", 2006 International Joint Topic: 9th Emergency Preparedness and Response/11th Robotics and Remote Systems for Hazardous Environments, Salt Lake City, Utah, Februrary 12-15, 2006.

[12] G. Kogut, B. Sights, G. Ahuja, Pacis, E.B, H.R. Everett, "Sensor Fusion for Intelligent Behavior on Small Unmanned Ground Vehicles." SPIE Proc. 6561: Unmanned Systems Technology IX, Defense & Security Symposium, Orlando FL, 9-13 April 2007

[13] C. W. Nielsen, B. Ricks, M. A. Goodrich, D. J. Bruemmer, D. A. Few, and M. C. Walton, "Snapshots for semantic maps." Proceedings of the 2004 IEEE Conference on Systems, Man, and Cybernetics, The Hague, The Netherlands, 2004.