5.3 Sensing and Virtual Worlds – A Survey of Research Opportunities

Sensing and Virtual Worlds – A Survey of Research Opportunities

Dana Moore
SAIC
dana.moore@saic.com

Abstract: Virtual Worlds (VWs) have been used effectively in live and constructive military training. An area that remains fertile ground for exploration and a new vision involves integrating various traditional and now non-traditional sensors into virtual worlds. In this paper, we will assert that the benefits of this integration are several. First, we maintain that virtual worlds offer improved sensor deployment planning through improved visualization and stimulation of the model, using geo-specific terrain and structure. Secondly, we assert that VWs enhance the mission rehearsal process, and that using a mix of live avatars, non-player characters, and live sensor feeds (e.g. real time meteorology) can help visualization of the area of operations. Finally, tactical operations are improved via better collaboration and integration of real world sensing capabilities, and in most situations, 3D VWs improve the state of the art over current "dots on a map" 2D geospatial visualization. However, several capability gaps preclude a fuller realization of this vision. In this paper, we identify many of these gaps and suggest research directions

1.0 INTRODUCTION

Virtual worlds can add value to many domains. It is possible to draw many Venn diagrams that feature virtual worlds serving as the presentation or interaction layer for a wide variety of use cases. In this paper, we suggest that by combining live sensing, three-dimensional (3D) virtual worlds and social networking capabilities, we can derive a command and control capability that significantly improves communication, situation awareness, and situation response. Figure 1 depicts this conjunction graphically (C4ISR = command, control, communications, computers, intelligence, surveillance, and reconnaissance).

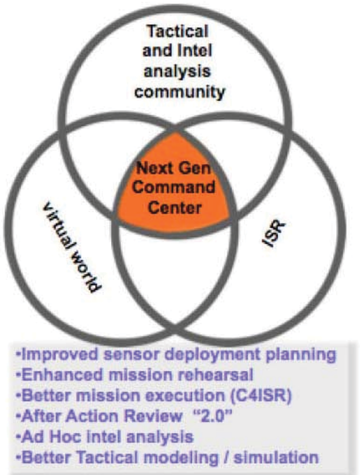
Tactical and Intel analysis community
Next Gen Command Center
virtual world
ISR
•Improved sensor deployment planning
•Enhanced mission rehearsal
•Better mission execution (C4ISR)
•After Action Review "2.0"
•Ad Hoc intel analysis
•Better Tactical modeling / simulation

Figure 1: Intersection of ISR and virtual worlds

2.0 AN ILLUSTRATIVE USE CASE

Live tactical sensing field trials and exercises have become extremely expensive to stage and conduct, and even when the logistics and system integration go without a hitch, the after-action "hot washes" are often unrevealing—results are often hard to characterize, replay and data mine sufficiently well to arrive at meaningful conclusions. Often, the quality of logistics preparation is blamed for many of the problems in exercise setup, and the lack of customer focus on strong system integration. Both are commonly cited as the cause of often indifferent and hard-to-interpret results.

We contend that while these both may be true, they are aspects of execution of a specific strategy that relies exclusively on live execution as its mechanism.

A different approach, one we will propose here, should be obvious to the modeling and simulation community: Use the real devices to stimulate the model and provide a high-fidelity simulation, high enough in many cases to render the actual exercise optional or even unnecessary. In this way, travel and logistics become moot, and many more exercises can be run within the same time and funding constraints. In fact, it may be possible to use simulation for command and control of the actual in situ as a deployed system. Nowhere would the benefits be more keenly felt than in large, logistically complex exercises. We take as an illustrative use case the National Level Exercise, yearly mandated by the White House. While this exercise is pure simulation, not involving real, deployed sensors, it does serve to illustrate the aspects of exercise complexity and the utility of virtual environments, and lay the foundation for later assertions we will make.

3.0 NATIONAL LEVEL EXERCISE

Each year, the U.S. executive branch mandates an exercise called the National Level Exercise (NLE)[1], which is done only as a virtual exercise. One reason given for this "virtual-only" approach is that the dynamics of the exercise, which models the kinetic event, is of sufficient magnitude that putting the exercise together live would be prohibitively expensive and time-consuming. According to the U.S. government's own literature, the NLE is a part of the *"National Exercise Program (NEP), which serves as the nation's overarching exercise program for planning, organizing, conducting and evaluating national level exercises. The NEP was established to provide the U.S. government, at all levels, exercise opportunities to prepare for catastrophic crises ranging from terrorism to natural disasters."*

The 2011 NLE concluded in May simulated a 7.7 magnitude earthquake on the New Madrid fault near Memphis, TN. The geo-specific terrain in the simulated environment encompassed 10 square kilometers with over 2,100 structures at sub-meter photo-real high resolution.

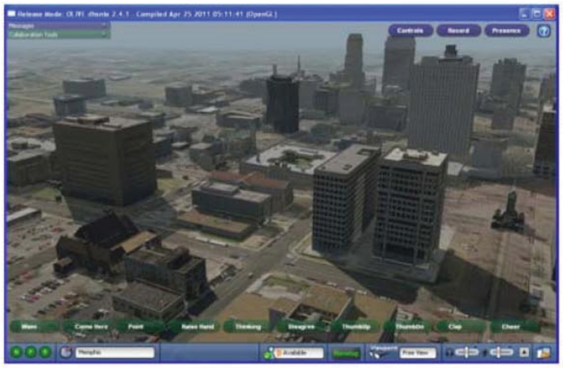

Figure 2: NLE 11. High overhead view of downtown Memphis, TN

Figure 1 shows a screen capture of the 3D virtual environment in which NLE was implemented.

Pre-exercise, the commercial On-line Interactive Virtual Environment (OLIVE) was used for over four months as a collaboration, exercise development and systems integration environment. During the exercise, the 3D environment was used as the command and control nexus for 14 command groups whose participation included several participants, including U.S. state, U.S. federal, U.S. Army Northern Command (NORTHCOM), and U.S. Coast Guard participants. A control room with participants in remote, real-world locations, but co-located in virtual space, and several simulated system feeds, again from disparate sources in multiple data clouds, can be seen in Figure 3.

By any standards of assessment, NLEs have been well received, produced significant data, and by design, exercised all aspects of emergency response from low-level tactical to the highest echelons of state and federal government that we assert might not be possible to otherwise conduct.

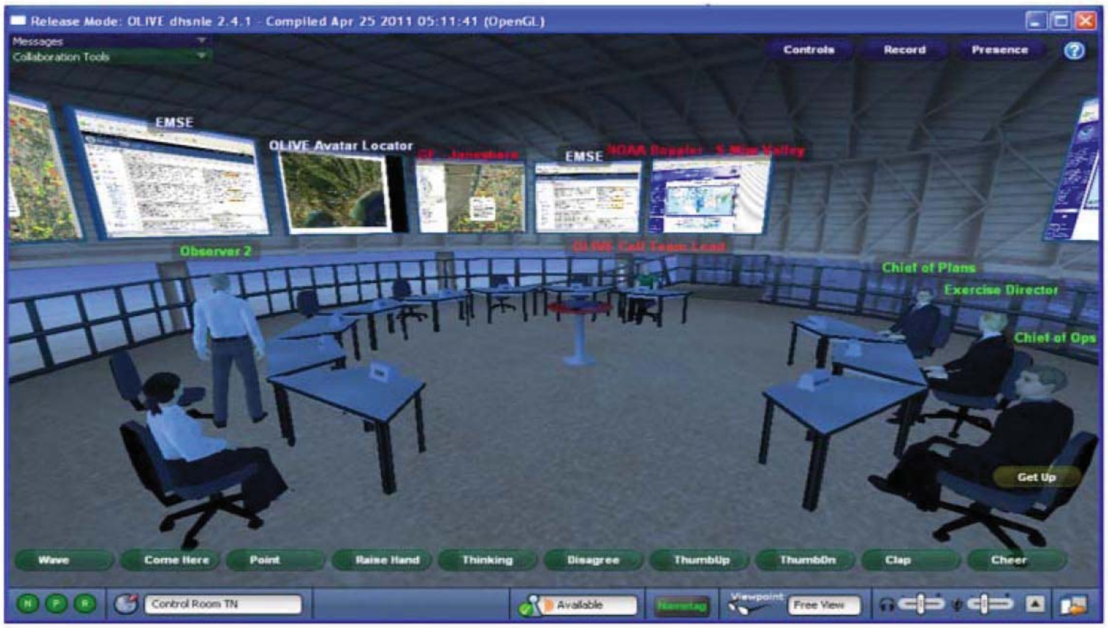

Figure 3: NLE 11. Command and control center with multiple simulated data feeds

For purposes of this discussion, we can consider NLE as an example of large-scale integration, mission preparation and command and control, all staged and performed in a virtual arena. If actual sensors had been used to stimulate the virtual arena rather than synthetic data, the architecture could support a live exercise.

With NLE as a reference implementation, we now have a frame of reference in which we can examine the question of how one might move from a fully virtual arena to a real-world, operational domain. This examination will help characterize technology gaps that have, in the past, prevented technologists from achieving integration of 3D simulations of areas of interest with operational sensing capabilities. These gaps, in turn, represent opportunities for research. It is important to note that significant gaps exist both in the sate of the art in sensing and in 3D virtual worlds that attempt to mirror real-world attributes (which often are called "*mirror worlds*").

To attempt to bound what is meant by "sensors," we suggest restricting our purview to those that have proven relevant in exercises such as NLE and would be commonly used in a tactical setting. Amongst these would be

1. Environmental data, such as temperature, precipitation, wind speed and direction, clouds and visibility, day/night (based on actual ephemeris models), wave heights and riverine flows
2. Unattended ground sensor/unmanned aircraft system (UGS/UAS) data from traditional sources such as infrared, seismic and acoustic sources; magnetics and thermal sensors
3. Video from electro-optical sources, such as unmanned aerial vehicle (UAV) feeds, webcams, closed-circuit television (CCTV), broadcast television
4. News (and other) data from streaming sources, such as Really Simple Syndication (RSS) feeds, broadcast monitoring systems, language capture and translation

To these, we might add a new and interesting, but non-traditional, abstract sensor that might provide insights into the human terrain of an area of interest (AOI) and might, for example, drive models of non-player characters. These might include:

5. Social/cultural indicators, providing real-time, geo-specific indicators of social *climate* or *temperature* based on real-time news feeds
6. Social networking, useful for estimation of the memetic ecology of an AOI. These might be instant messaging; Twitter® (Twitter, Inc.) tweets; email

We begin the discussion by looking at sensing and then move to mirror worlds.

4.0 RESEARCH PURSUITS FOR MODERN SENSOR SYSTEMS

A readily apparent characteristic of sensor systems is that while they are capable of producing copious data, they often yield little actionable data absent much massaging, either by a human in the loop or by automated systems. It is therefore insufficient, we contend, to simply faithfully model a sensor and show some manifestation of its data outputs in a mirror world. Thus, although one approach might be to faithfully model a weathervane, dropping it into a virtual mirror world as shown in Figure 4, the tactical value is very small. There is added value in seeing the sensor *in situ,* with the surrounding environment fully navigable in 3D.

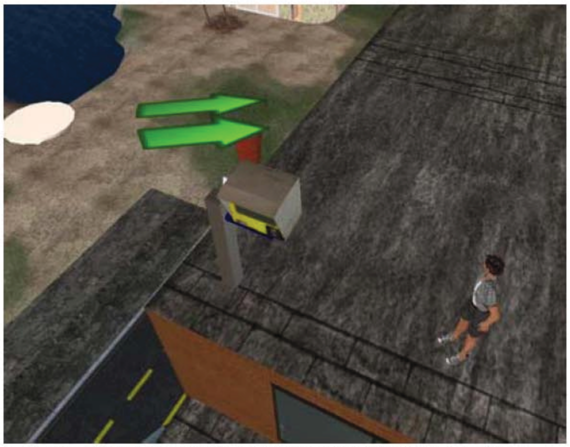

Figure 4: Wind direction and velocity sensor model fed by real-world source (Harvard School of Public Health CitySense Project)

However, while having multiple perspectives might be useful and can add situation understanding, the data from the sensor itself has a limited utility, constrained by the timeliness of the reported data. Were the sensor less "dumb" it might be able to yield some notion of its operating data (e.g., its remaining duty cycle, or its error envelope), or trend data, long term or short term, or the inter-relationship with other environmental physical sensors.

One can argue that long-term persistence of such data and meta-data, or converting it into useful intelligence, is the responsibility of systems seeking to integrate sensor data, but therein lies the root of a perennial problem. Traditional sensor systems are at once too "dumb"—often producing too much data, and yet too little actionable intelligence. A sensor field might incessantly report the footsteps of a few goats on a hillside, but be unaware of its mission context, which might be to report not goats, but men traveling in formation.

This aspect of the sensor field's mission might be well understood by humans receiving the data, but because of the incessant reporting, might come to be ignored as noise source, especially problematic when the real formation of men comes across the hillside.

In part, this situation exists because traditional sensor systems, especially physics-based sensors, have been designed, implemented, and fielded as a part of a bespoke system. Such systems are heavily "*silo-ed,*" useful in very narrowly defined contexts, with interfaces dictated by the size constraints, communication and power budgets of traditional micro-electronics components. The value of modeling such a sensor field in a 3D environment would be essentially nil, but would entail considerable software development cost.

However, owing to a virtuous trend of micro-componentizing compute and communications capability, functionality narrowly constrained by package size or power limitations need no longer be the case. Piece parts such as the *Gumstix*® [2] (Gumstix, Inc.) embeddable computer, running accustom Linux® (Linus Torvalds) stack are readily available. 3GS and WiFi piece parts are likewise easily available. Were these simple piece parts a commonplace in physical sensors, a paradigm shift would shortly occur.

Thus, we can identify the first of many research pursuits: **Add standardized local computation, and communications capability to sensors.** Doing this by creating a common sensor reasoning and communications platform and creating standard interfaces would benefit an entire creative community, but especially the 3D command and control virtual tactical operations center.

One may argue that abstract, non-physics-based sensors (e.g., RSS feeds) do not suffer the same lack of local computation capability. Although this is essentially correct, the current state of the art is that even for networked abstract sensors, while there may be compute resources attached, these are seldom used to add value, to change data into intelligence.

Hence, our second suggested research pursuit: **Add local intelligence, reasoning and context understanding to sensors.** Reasoners (e.g., C Language Integrated Production System, CLIPS [3]) can be packaged and deployed in quite small packages and placed adjacent to individual sensors or as a "super node" in sensor fields, but the challenge is to develop better reasoning capabilities beyond venerable reasoning suites founded on Bayesian-style logic.

The need for better sensor reasoning is predicated on the belief that a common sensor vernacular or ontological representation can exist. Currently none does, although one may argue that SensorML [4] could be the foundation of such a data description language or, at the very least, a useful conceptual antecedent.

In any case, it does point out the potential for a research pursuit: **Develop sensor agnostic description languages that might become the basis for query and description.** Innovators in the 3D virtual environment would use description and query languages to place sensors automatically in the 3D scene graph and query them for timely update; show information flows among sensors and command and control paths; and automatically add and operate non-player characters or other automata into the scene graph.

Another capability that would dovetail nicely is the addition of marketplace mechanics to sensors or sensor fields. This was the intent of the short-lived and fondly remembered JXTA® (Oracle America, Inc.) [5] protocol and framework, wherein small form-factor data producers (e.g., sensors) offered their content into a marketplace and market mechanics played a strong part in steering the mission. The JXTA architecture was at once elegant, comprehensive and compact. A research pursuit derived from this observation: **Develop a marketplace mechanic for sensors and webs of sensors.** A marketplace mechanic would enhance the capability to create a virtual operations center (as well as enabling the capability to more easily control sensor fields from mobile devices).

Interwoven with representation are issues of access. As mentioned previously, accessing and extracting data from sensors has been traditionally difficult. Currently, most virtual worlds can interface with almost anything that responds to an Extensible Markup Language Hypertext Transfer Protocol (XMLHTTP) Request [6] or XML remote procedure call (RCP) protocol [7] requests and responses. Real-world sensors are not amongst the data sources that respond to these protocols.

This is unfortunate, since there is general agreement that Web and rich Internet application ("cloud") frameworks have redefined popular thinking about how distributed applications are imagined, architected and implemented. This suggests yet another research pursuit: **Create a capability to add small footprint, agile, language-based web frameworks** (e.g., Pylons [8], Ruby on Rails® (David Heinemeier Hansson) [9], TurboGears [10]) **to sensors and sensor fields to enable them expose their data both as web servers and web clients**. While there are open source sensor web architectures for composition of large-scale sensor networks such as 52North [15], many consider this an extremely heavyweight approach.

The modeling and simulation community would especially benefit if the goal of incorporation of real-world inputs is ever to be realized. Not having to write custom interfaces for each sensor type would represent major forward progress. Yet, although these tools have been available for almost a decade, little thought has been applied to using web technologies in sensing.

Agile languages can play another part in putting real-world sensors in play in virtual worlds. Most onboard sensor algorithms are still coded in low-level languages such as Verilog® (Gateway Design Automation Corporation) [11], or even cross-compiled C. In contrast, most virtual worlds use agile (sometimes referred to as "scripting") languages to provide the interaction points in their environment. Bigworld® (Bigworld Pty Limited), a popular game engine uses Python [12]; Unreal® (Epic Games, Inc.), another popular engine, uses UnrealScript, a Java® (Oracle America, Inc.) variant; Unity 3D® (Unity Technologies APS), a hugely popular game and virtual world engine uses JavaScript® (Oracle America, Inc.) and C#; and finally Second Life® (Linden Research, Inc.), a well-regarded virtual world engine uses its own C-family-derived compiled scripting Linden Scripting Language (LSL).

It is clear that there is a gap here—there is no useful mutual language that facilitates a common expression of algorithms and business logic in a lightweight, highly expressive language (e.g., Python® [Python Software Foundation], PHP, Lua™ [Lua Technologies, LLC.] or Ruby® [Faculdades Catolicas]). One may argue that as long as the ability to support web paradigms such as Representational State Transfer (RESTFul) [13] interactions amongst participants in a sensors and virtual worlds mashup, there is no compelling need for common languages.

Yet, the evolution of Java, and hence of object-oriented programming, was predicated on the belief that a single compute language could facilitate common expression across many platforms, scaling from supercomputers down to the very small. A mantra of the 1990s became "Write once—run anywhere," and while this proved to be a goal, there were seams and gaps in implementation detail, it still serves as a brilliant reminder of the art of the possible.

Thus, a next suggested research pursuit: **Create a common high-level language that spans the requirements of sensors, sensor fields, and virtual worlds.** Since both environments are inherently event-oriented, a lightweight language supporting concurrent programming (co-routines, or lightweight threads), callback registration, and a high level of abstraction would be ideal. Loads of Python and other lightweight languages exist for embedded Linux platforms, such as the aforementioned Gumstix single-board computer, so this suggested pursuit may be regarded as within the realm of the

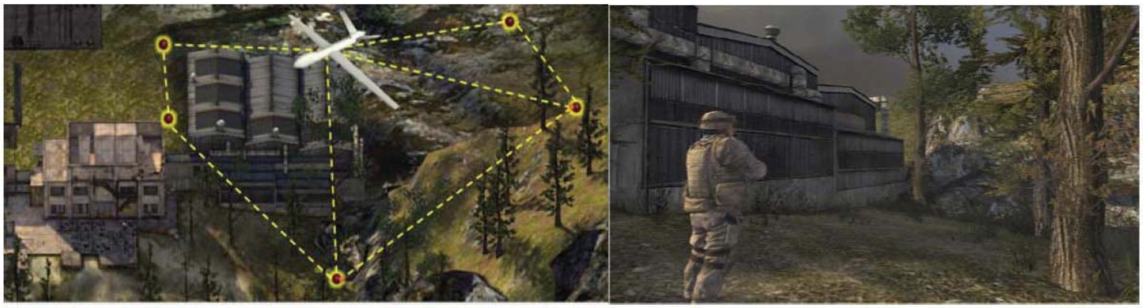

Figure 5: (A) Virtual world lookdown onto resources and AOI(B) A zoomed-in view over the soldier's shoulder

possible and not require a completely fresh conceptual start.

One may argue that had there had been a fully evolved Internet prior to the 1990s, the need for a common language would not have been as compelling or would be even a moot point, given a number of game engines and virtual world platforms that support some form of RESTful interaction. We would counter, however, that having a common object description approach can potentially lead to more efficient interaction and less impedance mismatch.

5.0 RESEARCH PURSUITS IN VIRTUAL WORLDS AND GAME PLATFORMS

There are many gaps in current sensing design and implementation that make a useful integration with virtual worlds more difficult and prevent engineers from achieving the kind of useful integration that creates the tactical operations center of the future. Figure 5 suggests what a tactical officer's view of the AOI might look like in our vision of the future. Notice that resources are visible in their relationship with the AOI in both the high aerial view and the dismounted soldier's perspective, and further, zooming in to perceive greater detail is seamless. This is not dissimilar to what one might experience in a "Google® Earth" (Google, Inc.) view of an AOI, in which one can zoom in from a very high altitude down to a greater level of detail; hence, one might ask, Why is it better and what does the virtual world contribute to the tactical perspective? There are several perspectives on this question, but let us address a few of them in this paper.

First, let us consider tactical displays based on 2D geospatial platforms (e.g., ArcGIS® [Environmental Systems Research Institute, Inc.]) as shown in Figure 6, or 2.5D visualizations (Google Earth, e.g.) as shown in Figure 7.

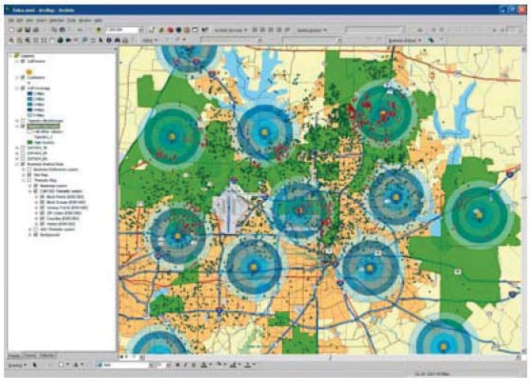

Figure 6: Typical Geographic Information System (GIS) display

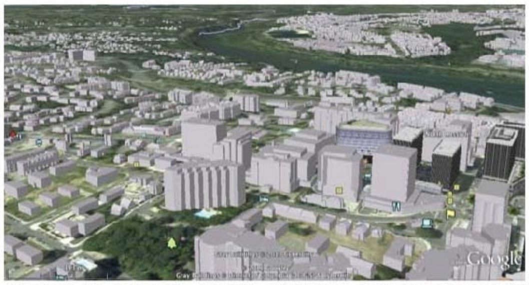

Figure 7: Google Earth display augmented with "grey box" structure.

In Figure 6, useful data have been overlaid onto a 2D topological display, and in Figure 7, “grey box” structure and terrain have been overlaid onto a Google Earth representation. In the case of the 2D display, the data layer overlay can and often does obscure the “big picture” of the terrain and other detail. Arguably, the end user can turn layers on and off in either of these representations, but doing so creates a context switch in which details from now-obscured data layers.

While it may be true that either a GIS display or a 2.5D visual representation may be useful at a wide range of regard, they both lose utility when required to yield detail at closer range. Both representations lose human scale. There is not a capability to meaningfully put humans in the scene, without which, understanding becomes abstract and lossy. The ability to walk the scene as one can do in any first- or third-person game or virtual world is missing, as are representations of elevation, an understanding of potential ambush angles, occlusions, and useful vantage points that can best be understood in 3D.

In the notional 3D tactical environment shown in Figure 5A and B, resources can be operated through the on-screen 3D graphical elements and the scene can be navigated in fly-through or walk-through mode, depending on the level of understanding required. Structures can be rotated or walked through such that there are no obscured details. This is especially important in non-rural AOIs or “urban canyons” where entrances and exits, and potential angles of attack are not obvious from drone or other views.

Compared with another popular visual approach, drone video with data overlays outputs (see Figure 8), we would assert that there is less cognitive switching involved with integration of scene elements and data. Scene understanding from drone video is most similar to the understanding gained from viewing a series of flat photographs. Like other 2D and 2.5D views, utility is significantly lower in the urban environment, and occlusions can deceive the user.

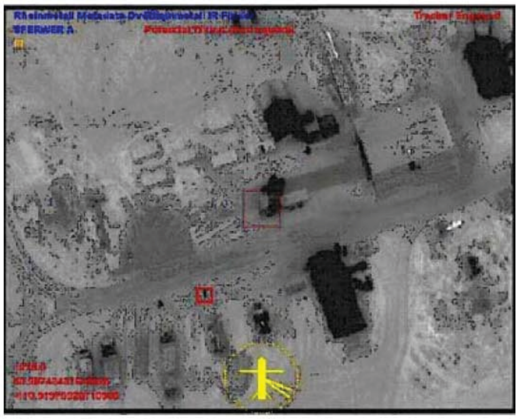

Figure 8: Video drone output

What research pursuits are important to bring the benefits of 3D navigation and immersive experience to the tactical user? Let us consider a few of these. First, the need for object interoperability has become extremely important. A number of model representations, including COLLAborative Design Activity (COLLADA) [16], 3DS Max$^{®}$ (Autodesk, Inc.) [17], FilmBox file format (FBX) [18], and control environments of long-standing, high-level architecture (HLA) [19], distributed interactive simulation (DIS) [20], virtual reality VR federates, currently exist. However, these are not easily composited into a game or virtual world environment. Thus, a first suggested research pursuit: **Create an object interoperability architecture that can enable models from different producers in a game- or virtual world-agnostic way.** Along with this, there would be an ability to distribute an arbitrarily large-scale AOI across multiple game platforms or virtual worlds concurrently.

The need for geo-specific terrain and structure or geo-typical terrain, decorated with specific structure is a critical success factor for convincing tactical users to move to 3D tactical viewers. Thus, a second

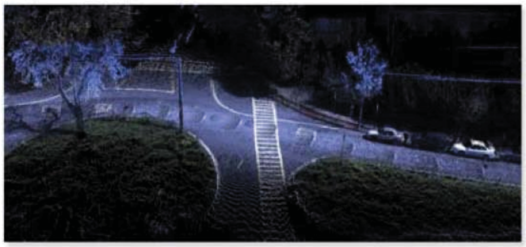

Figure 9: LIDAR point cloud example

suggested research pursuit for virtual worlds: **An ability for (semi) automatic generation of terrain and static structure**. Currently, 360° laser detection and ranging (LIDAR) can perform metrology down to the centimeter level, creating point clouds along with the texture for the point as shown in Figure 9; however, the ability of converting point clouds to meshes is still missing.

In addition to creating structure, there is a need for higher-fidelity graphics in virtual world platforms. The most often voiced complaint from military users when shown a prototype in a commercial virtual world such as Second Life is that it does not compare favorably with console or PC games. It does little good to point out that there is a difference between the open and unlimited vistas that a virtual world must support and the constrained scene graphs that a game engine can support. Thus, a third research pursuit: **Methods and algorithms to deliver higher graphic quality in virtual world engines.** Although Linden Labs are slowly introducing polygonal meshes into their renderers, they drag along a great deal of legacy code with them. Therefore, fresh approaches are needed to deliver higher quality.

Ironically, and at the same time, users clamor for lightweight clients, capable of running in web browsers or on mobile platforms. Hence, a final suggested research pursuit that will yield solid results: **Create virtual worlds platforms capable of being run in the browser, without need for plugins.** Currently, there is much hope for HyperText Markup Language (HTML) 5 [21] as a standard that can host a virtual world within the browser, but much foundational work must be accomplished before this can become a reality.

6.0 CONCLUSIONS

As this paper has asserted, representing and using sensors in massively multiplayer online games (MMOGs) and virtual worlds can produce better tactical command and control, and improved situation understanding. Virtual worlds are an inherently better vehicle for full 3D navigation and, when driven by sensors in conjunction with competent artificial intelligence (AI) engines, for representing forces and counterforces in an AOI.

However, a long legacy of “doing things as they have always been done,” especially in the sensors world, limited their use in 3D contexts or in other advanced interfaces that incorporate augmented reality or full sensory immersion.

On the virtual worlds’ side of the equation, the trajectory of commercial virtual worlds such as Second Life or Teleplace® (Teleplace, Inc.) is such that they have only moderate interest in advancing the state of the art to address the gaps identified. In general, where topics are well aligned with their goals of attracting recreational users, one may expect to see progress benefitting sensor modeling and C2 coming from the virtual worlds community *as a side effect* of general advances in the state of the art. The general case is such that advances in visual representation come at a high investment cost on the part of the platform creator, and are of arguable value. Thus, the introduction of COLLADA meshes (to create more attractive avatars, clothing, and venues) is on the roadmap for Linden Labs and Second Life, but object interoperability is not.

Therefore, in both sensing and modeling disciplines there are a number of excellent opportunities that are "green field" for research exploitation.

7.0 REFERENCES

[1] NLE Fact Sheet http://www.fema.gov/media/ fact_sheets/nle2011_fs.shtm

[2] gumstix: http://gumstix.com/

[3] CLIPS Expert System Description: http://en.wikipedia.org/wiki/CLIPS

[4] Sensor Modeling Language: http://www.opengeospatial.org/standards/sensorml

[5] JXTA: http://jxta.kenai.com/

[6] XMLHTTP History and specification: http://en.wikipedia.org/wiki/XMLHttpRequest

[7] XMLRPC Home: http://www.xmlrpc.com/

[8] Pylons Project Home: https://www.pylonsproject.org/

[9] Ruby On Rails Project Home: http://rubyonrails.org/

[10] TurboGears Web Framework: turbogears.org

[11] Verilog for Electronic Design: http://en.wikipedia.org/wiki/Verilog

[12] Python Language Home: python.org

[13] Representational State Transfer: http://en.wikipedia.org/wiki/Representational_State_Transfer

[14] HTML 5 High Level Description: http://en.wikipedia.org/wiki/HTML5

[15] 52North: http: //52north.org /communities/sensorweb/

[16] https://collada.org/mediawiki/index.php/ COLLADA_Digital_Asset_and_FX_Exchange _Schema and http://en.wikipedia.org/wiki/COLLADA

[17] 3D Studio Max 3ds Format: http://en.wikipedia.org/wiki/Autodesk_3ds_Max

[18] FilmBox File Format description: http://en.wikipedia.org/wiki/FBX

[19] High Level (simulation) Architecture: http://en.wikipedia.org/wiki/High_level_architecture_(simulation)

[20] Distributed Interactive Simulation (DIS): http://en.wikipedia.org/wiki/Distributed_Interactive_Simulation

[21] HTML Overview: http://dev.w3.org/html5/spec/Overview.html