An Interactive, physics-based unmanned ground vehicle simulator leveraging open source gaming technology: Progress in the development and application of the virtual autonomous navigation environment (VANE) desktop

Citation: Rohde, Mitchell M. et al. "An interactive physics-based unmanned ground vehicle simulator leveraging open source gaming technology: progress in the development and application of the virtual autonomous navigation environment (VANE) desktop." Unmanned Systems Technology XI. Ed. Grant R. Gerhart, Douglas W. Gage, & Charles M. Shoemaker. Orlando, FL, USA: SPIE, 2009. 73321C-13. © 2009 SPIE--The International Society for Optical Engineering

As Published: http://dx.doi.org/10.1117/12.820069

Publisher: SPIE

Persistent URL: http://hdl.handle.net/1721.1/52682

Version: Final published version: final published article, as it appeared in a journal, conference proceedings, or other formally published context

AN INTERACTIVE, PHYSICS-BASED UNMANNED GROUND VEHICLE SIMULATOR LEVERAGING OPEN SOURCE GAMING TECHNOLOGY: PROGRESS IN THE DEVELOPMENT AND APPLICATION OF THE VIRTUAL AUTONOMOUS NAVIGATION ENVIRONMENT (VANE) DESKTOP

Mitchell M. Rohde*[a], Justin Crawford[a], , Matthew Toschlog[a], Karl D. Iagnemma[b] Gaurav Kewlani[b], Christopher L. Cummins[c], Randolph A. Jones[c], David A. Horner[c]
[a]Quantum Signal LLC, 3741 Plaza Drive, Ann Arbor, MI, USA 48108; [b]Robotic Mobility Group, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA USA 02139,
[c]US Army ERDC, 3909 Halls Ferry Road, Vicksburg, MS USA 39180

ABSTRACT

It is widely recognized that simulation is pivotal to vehicle development, whether manned or unmanned. There are few dedicated choices, however, for those wishing to perform realistic, end-to-end simulations of unmanned ground vehicles (UGVs). The Virtual Autonomous Navigation Environment (VANE), under development by US Army Engineer Research and Development Center (ERDC), provides such capabilities but utilizes a High Performance Computing (HPC) Computational Testbed (CTB) and is not intended for on-line, real-time performance. A product of the VANE HPC research is a real-time desktop simulation application under development by the authors that provides a portal into the HPC environment as well as interaction with wider-scope semi-automated force simulations (e.g. OneSAF). This VANE desktop application, dubbed the Autonomous Navigation Virtual Environment Laboratory (ANVEL), enables analysis and testing of autonomous vehicle dynamics and terrain/obstacle interaction in real-time with the capability to interact within the HPC constructive geo-environmental CTB for high fidelity sensor evaluations. ANVEL leverages rigorous physics-based vehicle and vehicle-terrain interaction models in conjunction with high-quality, multimedia visualization techniques to form an intuitive, accurate engineering tool. The system provides an adaptable and customizable simulation platform that allows developers a controlled, repeatable testbed for advanced simulations. ANVEL leverages several key technologies not common to traditional engineering simulators, including techniques from the commercial video-game industry. These enable ANVEL to run on inexpensive commercial, off-the-shelf (COTS) hardware. In this paper, the authors describe key aspects of ANVEL and its development, as well as several initial applications of the system.

Keywords: robotics, simulation, virtual, unmanned, teleoperation, interface, mechatronics, video games

1. INTRODUCTION

Engineering methods have substantially and continuously evolved over the past 40 years. In the past, engineered products such as vehicles were designed, prototyped, and tested in a hardware environment. End-users were typically considered separately from the vehicle or device being engineered, leading to results that in some cases did not meet operator expectations or requirements.

Driven by ever-increasing performance requirements, customer expectations, global competition, and regulations, the product development process has evolved markedly. Ad-hoc methods have given way to robust design methodologies, mechanical drawings to three dimensional CAD representations, hardware-focused testing to testing in "virtual worlds," and the operator is now considered part of the overall system. This transition has been enabled by enhanced fundamental engineering knowledge, ever-increasing computing power, and improved engineering software.

Unmanned Systems Technology XI, edited by Grant R. Gerhart, Douglas W. Gage, Charles M. Shoemaker,
Proc. of SPIE Vol. 7332, 73321C · © 2009 SPIE · CCC code: 0277-786X/09/$18 · doi: 10.1117/12.820069

Traditionally, a weak point in the "virtualization" of the development process is in the area of real-time, on-line interactive simulation. Such simulation capabilities are useful for engineering visualization, collecting user/customer feedback, marketing, training of end user and logistics/service personnel, and hardware-in-the-loop simulation.

Historically, for real-time applications, engineers were forced to choose between simulation tools geared toward high-fidelity analysis and those geared toward graphically realistic interactive environments. This division took place because of the dichotomy between computing products developed for visualization that used high-end hardware and software for multimedia rendering and those developed for engineering simulation/computation (e.g. FEA, crash simulations, etc). The latter were developed and optimized for number "crunching" and used hardware and software display techniques for after-the-fact data display with limited graphics.

Fortunately, the barriers to integration of accurate engineering models and interactive, multimedia simulation have dropped tremendously in the past 20 years. COTS computing platforms can be obtained for less than a few thousand dollars that far outperform workstations or supercomputers costing millions of dollars in the recent past. The supercomputers of Cray and rendering hardware of Evans and Sutherland have given way to the high-end workstations of Silicon Graphics, and later to PCs with advanced, multi-processor video cards. Machines with 3+ GHz processors, gigabytes of RAM, advanced graphics and sound capabilities, and 100 Mb/s communications are found in many households across the country. Moreover, the internet has made instantaneous, almost transparent networking available at extremely low cost. The direct result of this is that most engineers have access to hardware sufficient to run simulations with intensive computational requirements.

At the forefront of the revolution has been the entertainment simulation ("video game") industry. Since the early 1970s, video games have pioneered interactive simulation and laid the groundwork for inexpensive computing that individuals, corporations, and governments benefit from today. Games have made computing truly pervasive in the United States and worldwide, much as other technologies owe their ubiquitousness to their ability to deliver entertainment content [1]. The insatiable appetite for entertainment and ever increasing expectations of the general public have fostered a tremendous level of competition in the video game industry, currently a $2B yearly market (comparable to the entire digital prototyping and simulation market (CAE) in 2003 [2]). This competition has yielded simulation products, software tools and techniques, and hardware platforms that have tremendous performance at extremely low cost.

A large section of the technologies that are useful to video gaming (realistic multimedia A/V, real time interaction, networked multi-player, and more) are highly desirable in simulation for engineering or scientific/technical purposes. This has not gone unnoticed, and software technology and techniques evolved for video games are beginning to have extraordinary impact in engineering simulation software. By combining "serious" simulation technology (e.g. dynamic models and actual/proposed geometries) with game technology, it is, in many cases, possible to create realistic, interactive multimedia simulations coveted by systems engineers in both government and industry

One area that is beginning to benefit greatly from the combination of games and simulation technology is unmanned ground vehicle (UGV) development and systems. UGVs are being used extensively in anti-IED, reconnaissance, countermine, and related operations and have proven to be valuable assets on the battlefield. While UGVs play a strong role in the Army's Future Combat System (FCS) strategy, it is well-recognized that the field is in its relative infancy. The rigorousness of the development, testing, support, and life-cycle management process is nowhere near that of the passenger car industry. While it may seem like a reasonable assumption that all of the virtual product development tools used in the automotive industry can be directly leveraged for UGVs, unmanned vehicles face unique challenges in terms of application, goals, and environments that have not been explored or lack any direct analog (e.g. autonomous driving, amongst other examples). It can be argued that the lack of virtual product development tools for UGVs has hindered the pace of advance. While several tools have appeared in the marketplace (such as Microsoft Robotics Studio [3]) none are comprehensive and/or have become a standard.

Ideally a simulation environment for UGVs would possess both real-time, high-quality interactivity while also maintaining realistic physical models. A developer would, ideally, have the ability to easily manipulate parameters of the vehicle model(s), change aspects of the terrain or world model, add or remove sensors or other components, and interactively test the UGV under a variety of conditions. The authors are participating in multiple efforts that leverage game technology for the simulation of UGVs, their components and subsystems. The Autonomous Navigation Virtual

Environment Laboratory (ANVEL) is one of these efforts, and represents an excellent example of fused gaming and serious simulation techniques.

Figure 1: Examples of ANVEL Visualization

ANVEL is being developed as a real-time desktop interface and development tool for the Virtual Autonomous Navigation Environment (VANE) being developed and used at the US Army ERDC [4]. VANE enables researchers to perform advanced studies in UGV mobility, vehicle-terrain interaction, sensor responses, and autonomous operation in very high fidelity. These high-fidelity, computationally intensive simulations run on a high-performance computing (HPC) based computational test bed (CTB), which is not oriented for real-time, interactive simulation. ANVEL "bridges the gap" between interactive, visualized reduced-order with off-line, high-resolution simulation, and provides an extensible framework to link to related simulation tools and technologies. Figure 1. presents examples of the ANVEL's high quality scene details and graphic rendering. ANVEL, its architecture and makeup, and an example application are described in the following sections.

2. METHODOLOGY

ANVEL is real time simulation software that enables the analysis and testing of autonomous vehicle operation, systems, and subsystems in an enriched, virtual world. It leverages physics-based vehicle and vehicle-terrain interaction models in conjunction with high-quality, multimedia visualization techniques to form an intuitive, accurate engineering tool. ANVEL has been designed to take advantage of "Open Source Software" (OSS) targeted at the video game industry and thus ANVEL yields characteristics uncommon to traditional engineering simulators. The primary advantage of leveraging open source tools is to avoid the extensibility and cost complications common to the defense industry. It is the authors' desire to create a tool that can be used and extended by a wide community of users without closed, proprietary formats and excessive per-seat license fees.

Some of the highlights of ANVEL include:

- Real time rendering of complex environments & vehicle models;
- Realistic terrain interaction and object dynamic models;
- Controlled, repeatable testing for advanced simulations;
- Adaptable and customizable simulation capability;
- Intuitive and consistent user interface.

By interfacing with VANE, ANVEL "bridges the gap" between core simulation and the end user (e.g. R&D engineer) by providing a number of key functionalities:

- The ability to select and place a variety of UGV models in different virtual environments and pilot them in real time.
- The ability to place virtual sensors at different points on the UGV, pilot the vehicle in virtual space, generate a "drive file" for use in simulators such as the VANE HPC CTB, and pass that drive file to the simulator along with other critical data.

- The ability to receive simulated high fidelity sensor (output) data from the external HPC simulator (following non-real-time simulation) and "play back" video of the UGV's motion in synchronization with the simulated high fidelity sensor outputs.
- The ability to repeatedly execute simulation "runs" of a vehicle traversing a terrain while manipulating a variety of parameters (including the terrain itself), and to use/swap vehicle terrain interaction modules or modify their operational parameters.
- The ability to interoperate with other military simulations at different scales (such as oneSAF and HLA via Matrex [5]).

These capabilities, while relatively simple in nature, provide a powerful base platform upon which to research a variety of mobility, sensing, and other UGV capabilities. We will now discuss key aspects of the ANVEL software and its development.

2.1 ANVEL Architecture and Features

2.1.1 Architecture

An overview of the ANVEL architecture is shown in Figure 2. The central block represents the "core" of ANVEL, including the dynamics, environment, vehicle-terrain interaction (VTI), and vehicle dynamics simulation. This is tied to the visualization system (upper left block) that renders the virtual landscape, vehicles, and other features onscreen, as well as the windowed ANVEL development environment. The user input block (lower) allows a variety of controllers to be used and provides interface to external JAUS compliant controllers. The system incorporates sensor simulations (left) that are both internal (i.e. ANVEL core product sims that operate in real-time) as well as external on- or off-line sensor sim modules developed by third parties. ANVEL interfaces to the offline VANE CTB (right) via drive, sensor, and terrain files, and also supports connections to external entities via the external controller plug-in.

ANVEL is architected with a philosophy of modularity, thus simulation components of varying fidelities and performance levels can be easily integrated as the needs of users change or expand. For instance, this allows the VTI code or rendering engine to be changed or replace without significant effort.

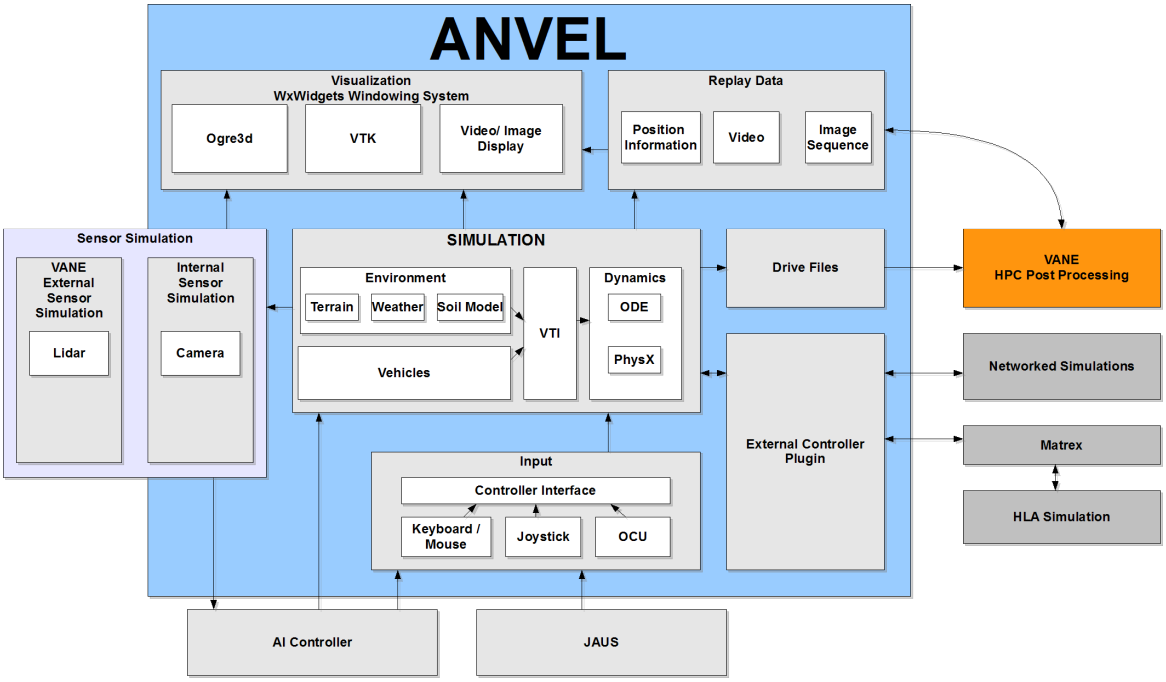

Figure 2: ANVEL Architecture Overview

2.1.2 Features

2.1.2.1 Visualization

At the heart of ANVEL are its high resolution visualization capabilities. ANVEL is designed with maximum flexibility in this regard; through the use of powerful OSS resources, ANVEL allows individual components to use different rendering systems based on the type of data that needs to be displayed. For example, three dimensional output, such as the main simulation world view, is currently handled by Ogre3D, which excels at advanced 3D graphical display. Additionally, certain sensor data is rendered by Visualization ToolKit (VTK), an advanced data visualization tool focused on 2D representations. ANVEL leverages "best of breed" open source software tools for the specific task, thus delivering the best possible performance and features wherever possible. ANVEL has the following main areas of graphical output:

Main Graphical User Interface

ANVEL consists of one main onscreen window with a variety of child windows that offer the user a choice of many different data displays. The Main Graphical User Interface provides an intuitive mode for interacting with all of these children windows, while maintaining a consistent look and feel. WxWidgets, described in section 3.1.2.1 is the graphics toolkit ANVEL uses to achieve this goal.

3D Visualization

The ANVEL simulation displays a detailed 3D environment that includes ground, sky, background, obstacles, and other elements that are relevant to UGV activities or contribute to maintaining an immersive environment. ANVEL allows the user to analyze the environment from any angle (via positionable virtual cameras) or see exactly what the vehicle cameras would see. Some example screenshots from the current ANVEL environment are shown in Figure 3, 5, 7, and 10.

Sensor Visualization

Any or all sensors in the ANVEL world can display the data they are currently sensing either in Real Time Mode or Replay Mode. This can be overlaid upon the main world view or rendered in a separate window at the

user's convenience. Example data could be LIDAR output, graphs from environmental sensors, infrared visualization, or many other possible data displays.

Video Replay

Video replay is used to display pre-computed videos corresponding to a certain sensor output and/or the vehicle trajectory in the virtual environment. This might include a stream of JPGs of, for example, a high-resolution LIDAR output from the VANE CTB, or a video of a real-world analog to the virtual space encountered.

Toolbars and Utilities

Not all graphics displays in the ANVEL tool require advanced rendering techniques. For configuration dialogs, text input and output, and similar functionality, ANVEL presents a standard "Windows" style look and feel that is familiar to most users. Since VANE will adapt to any operating system, Windows users will see common Windows dialogs, and Mac users will see traditional Mac dialogs. The GUI toolkit also allows easy addition of new, complex user interface elements through a simple API for programmers, or XML files for non programmers. This allows ANVEL to adapt as users wish to work with new sensors, or as new simulation methods are required.

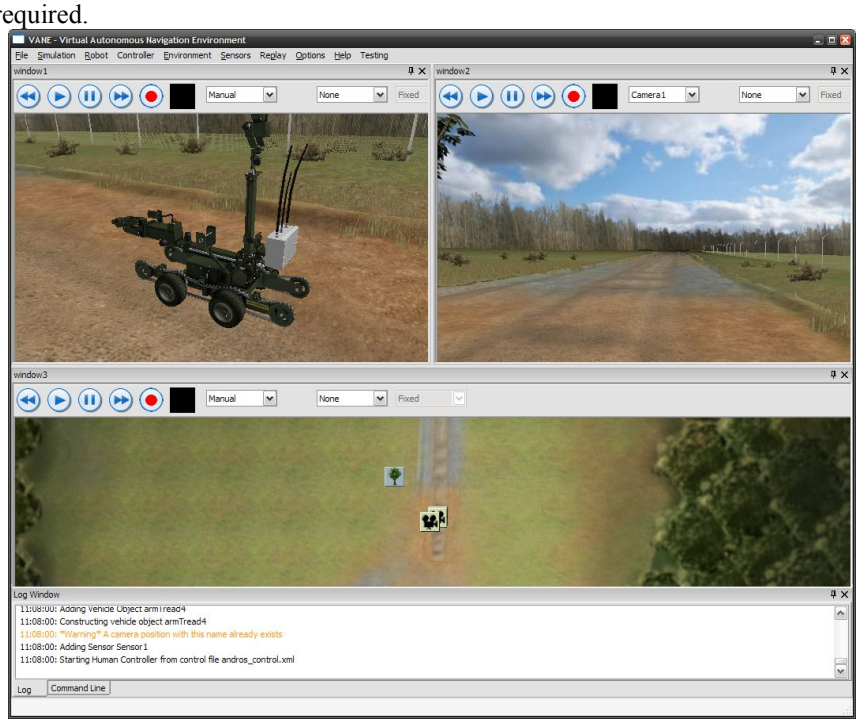

Figure 3: Example ANVEL Screenshot Demonstrating Third-Person View (upper left), First-Person (UGV) View (upper right), and Overhead View (bottom). The windowed ANVEL environment can be reconfigured by the user to display various combinations of views, sensors, and other data.

ANVEL uses a variety of multi-platform OSS graphical solutions to achieve its rendering goals:

- WxWidgets [6]– An open source cross-platform GUI framework that allows one code base to create applications that will utilize the native operating system's graphical style, and will tailor the application's internal structure to match the requirements of the intended operating system.
- Ogre3D [7]- An open source "Ogre3D" engine for real-time three dimensional visualization. Ogre3D is multiplatform graphics framework that allows real-time generation of high quality graphics. It takes advantage of multiple graphics libraries, including DirectX and OpenGL, which are standard within the hardware accelerated rendering communities. Ogre3D also contains a large set of useful routines and systems that is

leveraged for use by VANE. For example, Ogre includes an efficient and powerful math library that harnesses SIMD (Single Instruction, Multiple Data) instructions to fully utilize the processor.

- Visualization Toolkit (VTK) [8]- An open source rendering toolkit that focuses on advanced visualization techniques of many kinds of data. It is widely used in the scientific community for graphical analysis, 2D and 3D simulation. It is not targeted toward a real-time environment, so it is not practical as the main 3D visualization package. It is, however, an ideal resource for sensor data visualization.

2.1.2.2 Controls

For maximum flexibility, ANVEL is designed to use swappable "Controller Interfaces" to provide input to UGVs in the system similar to the controller interface displayed in Figure 4. Some examples are a human operator at an operator control unit (OCU) or keyboard, an autonomous navigation system, or a replay of prior simulation run(s). With minimal effort, new Controller Interfaces can be added that can work in the simulation environment via a "plug-in" API. ANVEL uses the Object-Oriented Input System (OIS) [9] to gather input from standard computer input sources, namely the keyboard and mouse, and also compatible joysticks, wheels, or game pads. OIS is another OSS project that is distributed with Ogre and easily integrates with WxWidgets. Standard OCUs will also be able to function as input devices through the JAUS protocol.

Autonomous navigation systems, also termed autonomous behavior kernels, are integrated as Controller Interfaces. A typical autonomous navigation controller processes data received from vehicle sensors along with user-described waypoints to calculate desired controls for the vehicle. A key goal of the ANVEL effort is to allow kernel developers to test their creations in the virtual environment.

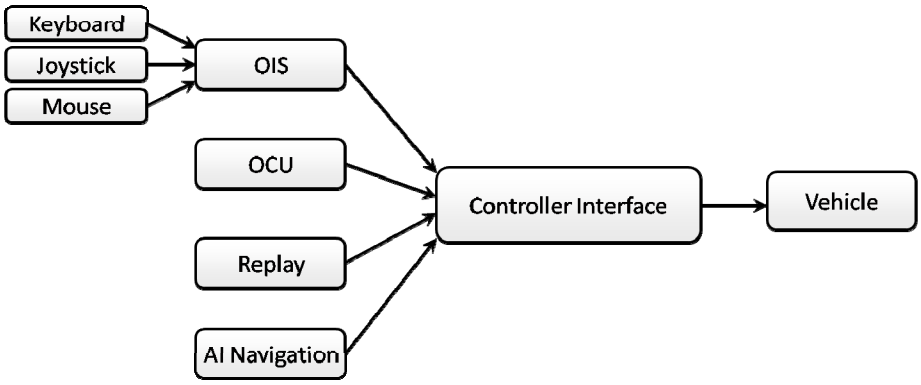

Figure 4: Summary of Controller Interface

2.1.2.3 Vehicle Modeling

ANVEL is not intended to replace computationally expensive modeling packages such as ADAMS; rather, it is intended to allow for physics-based virtual vehicles with reasonably realistic dynamic performance. These simulations consist of two parts: the underlying physics model that emulates the physical ruleset of the real world in the virtual one, and one or more models that define how an individual vehicle behaves. In the current implementation, ANVEL leverages the Open Dynamics Engine (ODE) [10], a rigid-body dynamics engine that is open source, cross-platform, and has been used extensively in various academic and commercial products. ODE performs accurate collision detection, models complex multibody structures, and offers various interaction models. It was designed for real time simulation and gaming, and thus is optimized for fast execution and does not "bog down" easily. Of course, the modularity of ANVEL supports other dynamics engines, and PhysX [11] has also been adapted.

The creation of vehicle models to date has focused on supporting current VANE users and not on providing a set that is complete and validated. A simple, lumped-parameter wheeled vehicle dynamics model has been created and used, and various parameters of this model can be adjusted to meet specific needs. Onboard joints (for manipulators, etc) have also been implemented, as have articulating motion appendages (such as the PackBot flipper and mini-Andros articulated track section as shown in Figure 5). Graphical models for a number of robots have been created, though

accompanying validated dynamics models are still in progress. Of course, users are able to modify or create new models as they desire and use them with the rest of the ANVEL package.

Figure 5: Examples of Vehicle Models (Packbot on left, Mini-Andros II center, Crusher right).

2.1.2.4 Terrain Interaction Models

An important feature of ANVEL is the ability to modify, extend, or add new models for vehicle-terrain interaction (VTI). VTI lies at the heart of many applications for the ANVEL simulator, and appropriate modeling of VTI ensures accuracy when investigating UGV mobility. Currently, there is support for a classical terramechanics based model which uses Bekker values [12] and a rigid tire model, but alternate models (such as those based on the cone index parameter) can be integrated. ANVEL enables users to add new VTI code as a simple module. The modularity of the ANVEL is shown in Figure 6.

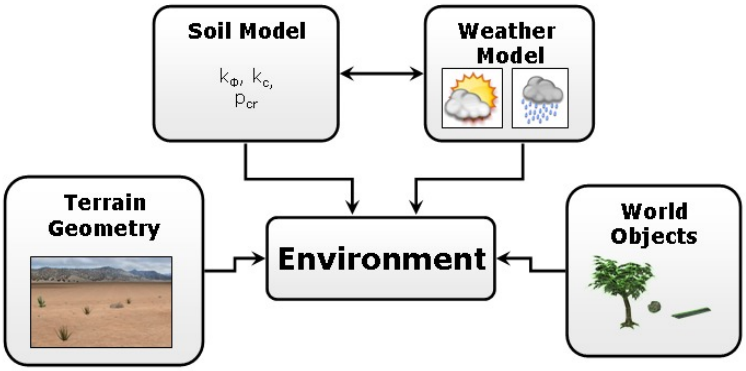

Figure 6: Overview of Environmental Subsystem

2.1.2.5 Terrain and Environment

One of the most prominent features of ANVEL is the detailed, enriched environment. The basic environment begins with a terrain geometry created from a 3D mesh or a height map. Meshes allow detailed artist interaction, while height maps allow for quick importing of a wide range of descriptive geographic information. World objects, such as trees, walls, or obstacles can be added to the environment separately, allowing detailed control of every scene. Currently, environments are generated in a manual process by digital artists leveraging external tools (e.g. Maya).

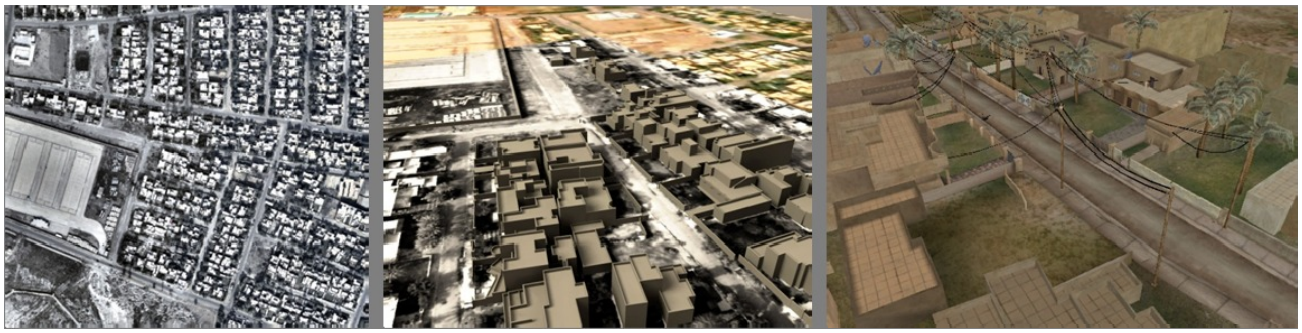

Figure 7: Example Terrain Creation. Digital artists transform satellite images (left panel) into a visualized urban environment (right panel). This example of a Baghdad neighborhood is 70 buildings with 52k polygons, and took approximately two weeks to construct (one artist). The ANVEL development roadmap includes tools for creating such terrain from standard data sources.

The current ANVEL development roadmap calls for a variety of new features and functionalities to be added to the environment subsystem in the near future. Soil models that accurately and dynamically describe soil properties and interactions will be added, as this will enhance both the accuracy of simulated vehicle movement as well as visualization. Weather models will allow for environmental variations, such as temperature or wind speed, or drastic changes such as the presence of rain or snow. The weather model will also interact with the soil model, to fully simulate changing traction conditions (Figure 6).

2.1.2.6 Modes of Operation

ANVEL is currently designed to operate in three main "modes"; a real-time simulation mode, replay mode, and "silent" mode. Each of these modes serves a specific purpose:

> *Real-Time Simulation Mode* - The real time simulation mode allows simulation of vehicle travel and interaction in a detailed environment. In this mode the user virtually teleoperates a UGV, "driving" it around a virtual terrain. Environmental data, such as terrain geometry, soil characteristics, and vegetation models, are combined with textures and lighting to present a rich landscape for vehicles to traverse. Current development focuses on building and displaying detailed visual landscapes; eventually this detail will be enhanced via advanced terrain dynamics, including terrain deformation. The addition of such techniques will lend additional physical accuracy to the environment.
>
> In this mode, one can study how variation of various components affect simulation performance, from vehicle performance to high level mission planning routines. Vehicle input and motion, sensor locations, and environmental changes can all be recorded for replay or analysis.
>
> *Replay Mode* - Replay mode allows the user to "replay" movement of the UGV through the environment from a pre-recorded file. Using a replay file, VANE can provide controlled, consistent and repeatable simulations to provide a test bed for any of the UGV system or subsystem simulations. Whether it is the effect of different soil parameters on trafficability, or the efficiency of a new algorithm for autonomous navigation, the test bed environment allows the user to seamlessly integrate new components and evaluate their performance.
>
> A key component of the replay is the ability to overlay post processed sensor data. Ideally, complete, on-line simulation of sensors and sensor response would be part of the ANVEL environment. Because the high-resolution simulation of certain sensor subsystems are highly computationally expensive, ANVEL does not attempt to simulate these sensors in real-time. Instead, ANVEL provides the capability to assist with, and incorporate, simulation of these sensors generated off-line by other simulation systems (e.g. VANE and the HPC testbed). ANVEL allows the user to export drive files with sensor locations to external simulation

systems, and import simulated sensor outputs from these systems. Once imported, this data can be overlaid on the real-time simulated playback and thus the user can visualize all systems working together.

Silent Mode – A simple mode that removes visualization and simply performs physics simulations and stores the results. This is used for experiments with large numbers of iterations and no need to visualize each and every one.

2.1.2.7 Configuration

ANVEL uses XML (eXtensible Markup Language) configuration files whenever possible to allow users the ability to customize most aspects of the program. Environments, vehicles, and physical dynamics can be adjusted by means of XML text file changes. This allows the end users to quickly customize the application to meet their needs without requiring new software or multiple builds.

3. RESULTS

The initial alpha version of ANVEL was completed in early 2008. All of the basic functionalities and features as described in previous sections were implemented and shown to operate as designed. Full 30 fps performance on standard desktop PC (1.X GHz, 2 GB RAM, etc) was achieved, and the system was immediately adapted for two specific projects.

In the first project, ANVEL provided a platform for the development and analysis of a novel algorithm for statistical prediction of small UGV mobility. The algorithm, detailed in [13], is similar in spirit to various statistical methods for manned vehicle mobility prediction that have been developed by the U.S. Army over the past 50 years (i.e. the NATO Reference Mobility Model (NRMM), NRMM II, and others). It exploits the fact that in field conditions, UGVs frequently have access to only sparse and uncertain estimates of important terrain and vehicle parameters (e.g. soil cohesion, vehicle center of gravity location, etc). Thus, in order to accurately predict UGV mobility, an algorithm must explicitly consider the nature of this uncertainty, and correctly propagate it through model-based dynamic analysis.

ANVEL was used to generate "ground truth" mobility prediction results via Monte Carlo simulation. Monte Carlo simulation involves the random selection of a value for each uncertain parameter from its uncertainty range, weighted by its probability of occurrence, followed by model simulation using this parameter set. This process is repeated many times to obtain the probability distribution of an output metric as presented in Figure 8. Scripts were written off-line, containing values of sampled UGV and terrain parameters, and ANVEL was employed to run Monte Carlo simulations in silent mode and display the output metric distributions. Despite the significant computational cost of these simulations, ANVEL performed the task efficiently, allowing iterative study of algorithm performance. Further details on the algorithm development and performance can be found in [13].

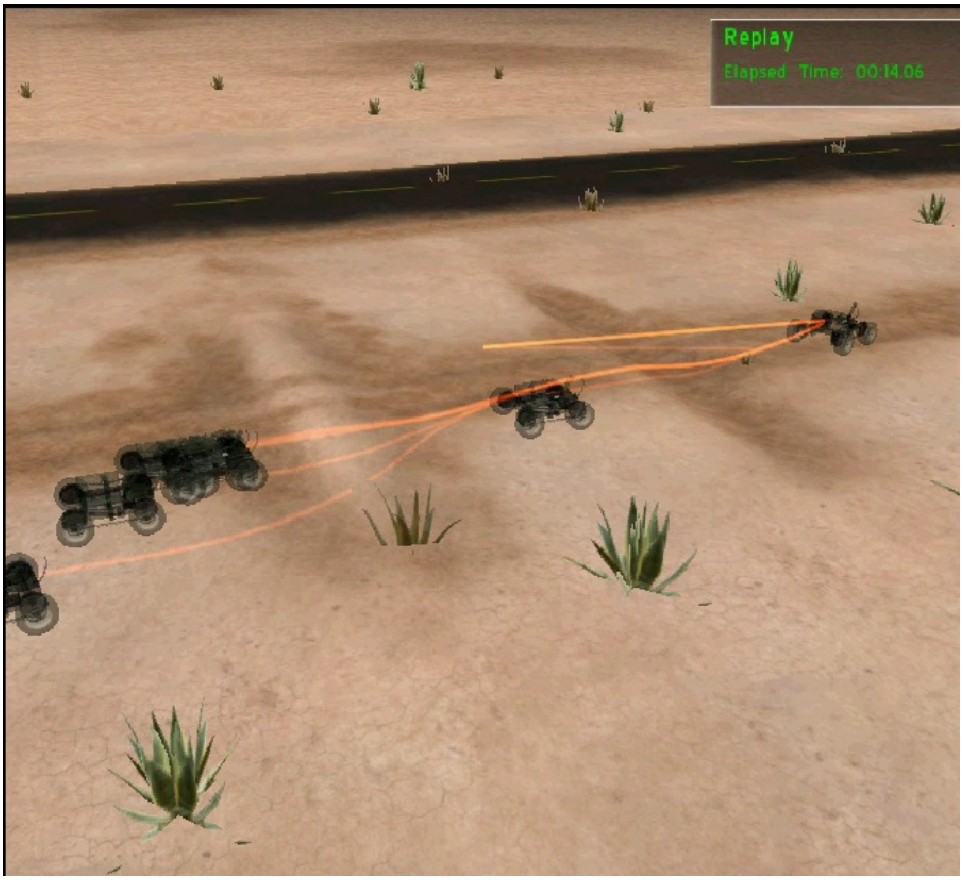

Figure 8: ANVEL Visualization Showing Repeated UGV Mobility Study. The "trails" show paths taken, with a UGV "ghost" remaining where the vehicle could progress no more given the particular constraints of that run.

In the second project, ANVEL was used as part of a proof-of-concept multi-level simulation. US Army ERDC researchers working with United States Military Academy (USMA), created a simulation system spanning high-level strategic simulation (via OneSAF), reduced-order vehicle UGV simulation (via ANVEL), and high-resolution sensor and terrain simulation (via VANE). ANVEL was linked (via integrated hooks/API) to OneSAF and HLA via the MATREX distributed modeling and simulation environment. The combination was used to demonstrate a hypothetical multi-level operational scenario (investigating IEDs) and shown to be effective.

A mission scenario of a Warfighter equipped with a UGV was developed into a simulation use case to evaluate the potential performance of the UGV for IED identification. A specific ANVEL scene was built that would play out an example of the use case narrative. ANVEL was used to evaluate the use case in relation to operation-level experiments and to demonstrate the capabilities of the UGV in a simulated environment. This use of ANVEL also served as an engineering-level experiment in which the interaction of OneSAF and ANVEL could be evaluated. Figure 9 and Figure 10 show details related to the experiment.

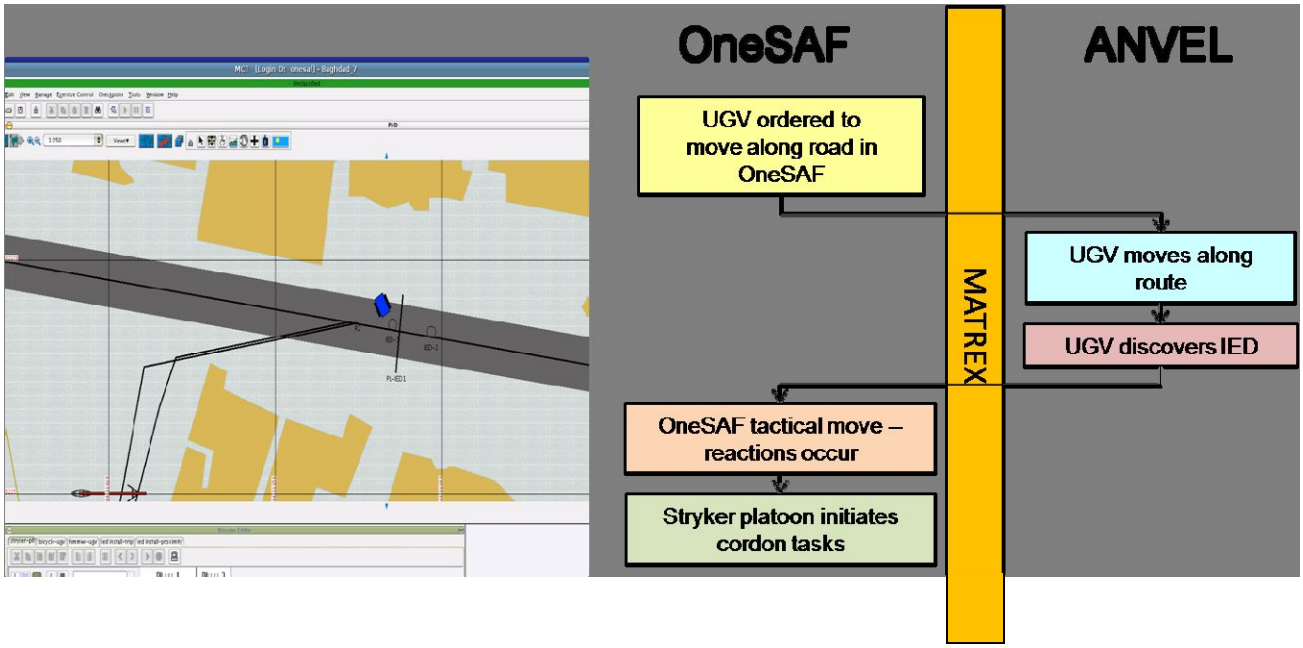

Figure 9: Example Application Exercising the OneSAF linkage. In this case, an operational scenario involving an IED inspection is played out in the operational level (via OneSAF) and the physical street level (via ANVEL). For more information, see [5].

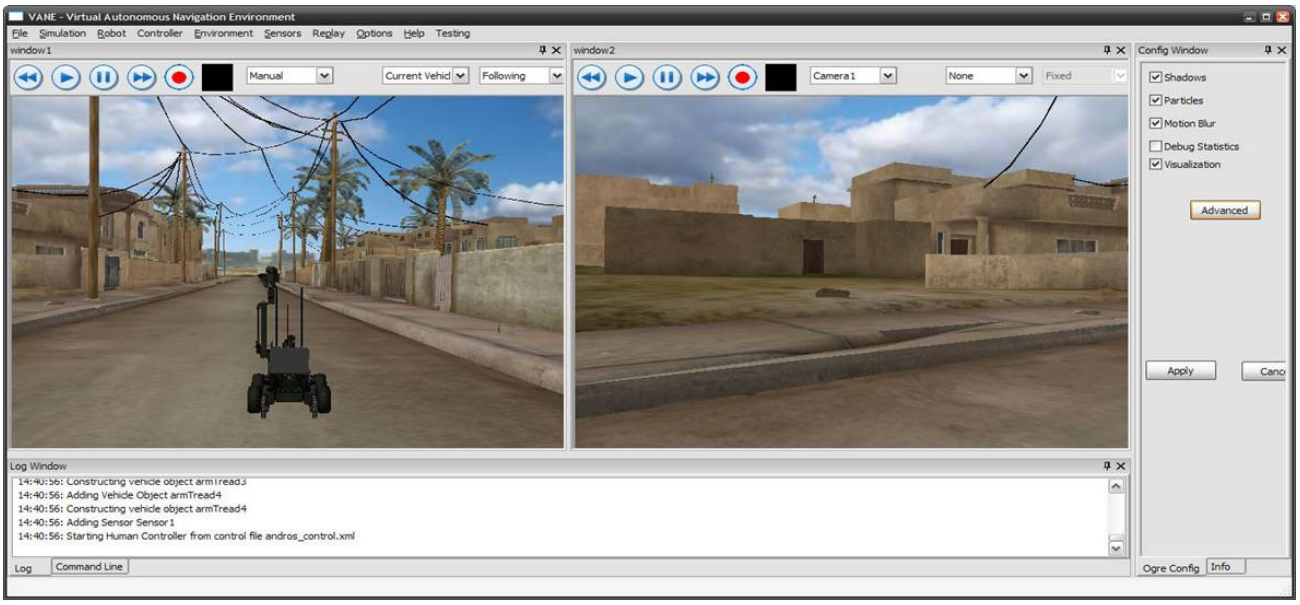

Figure 10: Screen shot of ANVEL from scenario in Figure 9. The left panel is configured to show the third-person view of the UGV, while the right shows a "virtual" camera and what would be seen through it as it sweeps across the side of the street.

4. CONCLUSIONS

In this paper, the authors have described ANVEL, a new simulation tool being developed for use by UGV researchers and developers. The basic architecture, components, and features were described, and several different applications were shown. It is recognized that ANVEL is still in its infancy, and many improvements and additions must be made before it will meet the needs of the wider audience for whom it is intended. Comments, questions, and suggestions regarding ANVEL are welcome.

REFERENCES

[1] Bryant, J., Love, C., "Entertainment as the driver of new information technology", *New infotainment technologies in the home: Demand-side Perspectives*, Lawrence Erlbaum Associates, 1996.

[2] DARATECH, inc. News Release June 9, 2003. Ref: iDPS 2003.

[3] Microsoft Robotics Studio Homepage http://msdn.microsoft.com/en-us/robotics/default.aspx

[4] Jones, R.A., Priddy, J.D., Horner, D.A., Peters, J.F., Howington, S.E., Ballard, J.R. Jr., Gates, B.Q., and Cummins, C.L., "Virtual Autonomous Navigation Environment (VANE)", Proc. of ASCE Earth & Space Conference, 2008.

[5] Cummins, C.L., "ANVEL/MATREX/OneSAF: Experiment 2," VANE Working Meeting, US Military Academy at West Point, 10/29/08.

[6] wxWidgets Homepage, http://www.wxwidgets.org

[7] Object-Oriented Graphics Rendering Engine (OGRE) Homepage, http://www.ogre3d.org

[8] The Visualization Toolkit Homepage, http://www.vtk.org/

[9] Sourceforge OIS site, http://sourceforge.net/projects/wgois

[10] Open Dynamics Engine Homepage, http://www.ode.org/

[11] PhysX Software Library Page http://www.nvidia.com/object/physx_8.10.13_whql.html

[12] Bekker, G., *Theory of Land Locomotion*, University of Michigan Press, 1956.

[13] Kewlani, G., and Iagnemma, K., "A Stochastic Response Surface Approach to Statistical Prediction of Robotic Mobility," *Proceedings of the IEEE International Conference on Robots and Systems, IROS '08*, 2008