VRFire: an Immersive Visualization Experience for Wildfire Spread Analysis

William R. Sherman*, Michael A. Penick*†, Simon Su*, Timothy J. Brown*, Frederick C. Harris Jr.†

* Desert Research Institute, Center for Advanced Visualization, Computation and Modeling (CAVCaM), Reno Nevada

† University of Nevada, Reno, Department of Computer Science and Engineering

Abstract

Wildfires are a frequent summer-time concern for land managers and communities neighboring wildlands throughout the world. Computational simulations have been developed to help analyze and predict wildfire behavior, but the primary visualization of these simulations has been limited to 2-dimensional graphics images. We are currently working with wildfire research groups and those responsible for managing the control of fire and mitigation of the wildfire hazard to develop an immersive visualization and simulation application. In our visualization application, the fire spread model will be graphically illustrated on a realistically rendered terrain created from actual DEM data and satellite photography. We are working to improve and benefit tactical and strategic planning, and provide training for firefighter and public safety with our application.

CR Categories and Subject Descriptors: I.3.8 Computer Graphics Applications; I.3.7 Three-Dimensional Graphics and Realism – Virtual Reality; I.6.6 Simulation Output Analysis.

Additional Keywords: Applied Virtual Reality; Physically Based Simulations; Wild-fire Visualization; Scene-Graph.

Figure 1: A wildfire chars the landscape billowing smoke into the air.

1 Introduction

The dynamic nature of wildfires and the damage they can cause have prompted many researchers to study the phenomenon [e.g. 4, 15, 20]. Obviously, the dynamic nature of fire makes starting a real fire for the purpose of research too costly and dangerous. However, through the observation of unplanned fires to develop and verify mathematical models [3], researchers are learning how to analyze various fire scenarios, setup training situations, as well as perform forensic evaluation on their computers.

Virtual reality technology [13] has long been used to produce realistic simulations for training and analysis, and it has proven to be a viable tool in many cases [8]. Recent advances in both simulation and supporting technologies now offers the possibilities of creating realistic, real-time, large-scale fire simulation for research. Advancement in both the graphics and computational power now provide the capacity for realistic simulations that heretofore have been difficult to achieve.

We are developing VRFire, a virtual reality tool for visualizing wildfire scenarios. VRFire is intended to allow users to visualize wildfires from different perspectives to facilitate analysis and the experience of wildfires. Wildfire scenarios are constructed from remote sensing data combined with a computational simulation of fire spread and simulated atmospheric conditions. Viewing an actual wildfire up close, and in person, is extremely dangerous. Fire-protected cameras can give a fair approximation of the experience of a fire passing by, but they can only be statically positioned, providing only a limited and non-immersive sensation. Therefore, we hope that by providing realistic, immersive, visual reconstructions of wildfire scenarios we will help professionals in the field gain insights into and understanding of wildfires, as well as benefit training, prediction and policy decisions related to wildfire management.

2 Related Work

There have been many efforts to computationally and/or visually represent destructive fires; however, we are not aware of any efforts to provide an immersive interface to a wildfire simulation. This may be due to the considerable amount of computational power required both for realistic modeling of the spread of the fire, and also for realistic visual rendering of how the wildfire would appear.

Much of the research on computational fire simulation has been to simulate fire within the confines of a building [1, 5]. There are some instances of virtual reality technology used for firefighter training scenario [6, 7, 9, 10, 16]. Most of these research efforts were not focused on creating realistic fire within the simulations but rather for using the simulation for logistics training – often a dense obscuring fog-like smoke layer was sufficient. One similar, though non-immersive, effort focused on the creation of an outdoor forest scene based on GIS data, combined with a computational fire spread generator [18], In this case, however, rather than interfacing to a computational fire spread simulation in use by local wildfire researchers, they developed their own elliptical spread model.

3 Hardware and Software Environment

Our basic approach to virtual reality hardware and software is to use standard and open solutions as much as possible. Thus we use fairly generic virtual reality displays which can accommodate applications ranging from walking through real spaces, to the visualization of molecules in solution. Likewise, we strive to use standard libraries, open-source if possible, for our software development. Ideally this software will also run on a wide range of immersive and non-immersive visual displays.

3.1 Hardware

Our immersive visualization facility hardware includes both a four-screen CAVE™-like Fakespace FLEX™ display, and a single-screen Visbox-P1™. The FLEX display is driven by an SGI Prism™ running SuSE™ 9.0 Enterprise edition, with four active-stereo capable graphics channels. The Visbox display is driven by a dual Opteron™ PC system running the openSUSE™ 10.0 Linux variant, using both outputs of an nVidia GeForce 6800GT card to drive two projectors producing a passive stereo output.

Our FLEX uses an Intersense IS-900™ with wireless head and wand units for position tracking. The Visbox-P1 uses two forms of tracking. The first form is an Ascension Flock of Birds® which is used to track a standard multi-input gamepad controller as part of the hand interface. A proprietary infrared video system wirelessly tracks the user's head position through image processing.

3.2 Rendering and Interface Software

We built our application on the open-source FreeVR [14] and OpenSceneGraph (OSG) libraries. The FreeVR virtual reality integration library is a cross-display VR library with built-in interfaces for many input and output devices. It allows programmers to develop on a standard desktop machine, with inputs and display windows that simulate a projection or head-based immersive system. The application can then run on either the Visbox-P1™ or FLEX™ displays, or the display of a collaborator on just about any type of VR system.

The OpenSceneGraph library is used to help with world rendering. OSG allows 3D objects to be hierarchically organized within the environment, and also provides a system that optimizes the rendering through the use of various culling and sorting techniques.

A considerable amount of our effort thus far has been in writing the software interface between FreeVR and OSG. FreeVR works naturally well with OpenGL and other lower level graphical rendering libraries. However, when interfacing a VR integration library with a higher level rendering APIs there are many issues that need to be addressed. Four major issues are: 1) dealing with the perspective matrices, 2) shared memory allocation, 3) multi-processing, and 4) windowing & input device interfacing. A software interface between FreeVR and the SGI Performer™ [12] scene-graph library already existed, so we felt confident that the similar OSG library would not be too difficult. Performer itself was avoided due to its closed-nature, and expected lack of future support.

While the OSG scene-graph system is somewhat based on the efforts of the Performer library, there are two major differences between the OSG implementation and Performer: 1) OSG does not double-buffer the scene-graph, requiring the update traversal to avoid making changes to the scene-graph while a cull traversal is in progress, and 2) because many people contribute new node types to the open-source OSG, there is no strict enforcement of the rule preventing scene-graph modifications taking place outside the "update" traversal. Neither of these issues is typically a concern for desktop applications running on a single CPU system, but for multi-screen immersive systems, they are problematic.

To address these implementation issues, we must specifically avoid the modification of the scene-graph when the multiple-renderings are taking place. FreeVR provides a semaphore-based locking/barrier system that we used to exclude writes to the scene-graph data during culling. Furthermore, when we discovered that some of the node-types (e.g. the particle system node) used the culling traversal to make additional modifications to the scene-graph, we had to specifically insert extra locking code into those modules. The end result is a system that works satisfactorily, but the addition of each barrier results in lower frame rendering rates.

3.3 Fire simulation software

Along with the creation of various graphics objects (i.e., fire, smoke, 3D buildings, and vegetation) and the landscape, the VRFire simulation includes visualization of fire spread created from the FARSITE fire area simulator[4]. FARSITE is a well-established fire behavior and growth simulator developed by the USDA Forest Service and also used by fire behavior analysts from the US Department of the Interior, National Park Service, US Department of the Interior Bureau of Land Management, and US Department of the Interior Bureau of Indian Affairs, as well as many state fire agencies. Its importance and widespread use among fire professionals was a critical factor for choosing to visualize its simulation output.

For our visualization process, we relied on the fire-science researchers to determine the proper parameters, run the simulation, and provide us with data to input into VRFire.

4 Virtual World

There are several graphical elements involved in the realistic and accurate visualization of wildfires. Fire, smoke and terrain have consumed a large amount of attention in graphics research because of the difficulty involved in rendering these in real-time. The multi-screen rendering aspect of many virtual reality displays also imposes conditions that can be detrimental to the performance of wildfire visualization. Each of these elements is outlined below with the difficulties encountered and the methods used.

The virtual world is constructed from remote sensing and computationally simulated data. Our prototypical database of Kyle Canyon, Nevada, is constructed from a 981x728 cell DEM file with 10m spacing between vertices. A 1m resolution satellite photograph is overlaid on this nearly 10km x 10km region. The FARSITE simulation operates on a grid of 287x203 cells that are 30 meters by 30 meters. This cell size corresponds to the "fire load" (i.e. vegetation) data that is provided as an input.

A typical wildfire will cover less than 100 acres ($0.4 km^2$), easily covered by our sample terrain. Corresponding to roughly 25 million acres, our 10km by 10km test range is a good experimental space to prepare for large scale operations, for which a simulated prediction may be requested to analyze the situation. Requests for this type of analysis would often correspond to wildfires covering tens to several hundreds of thousands of acres ($40\text{-}4000 km^2$).

4.1 Terrain

Terrain is the central point of wildfire visualization. It largely affects the outcome of the fire spread and is an anchor point for everything in the visualization. Wildfire visualization requires the visualizing of potentially very large areas of terrain.

Visualization of large outdoor scenes is extremely computationally expensive. One reason OpenSceneGraph was chosen as the primary rendering engine is that it does a good job of optimizing the rendering of the scenes for real-time visualization. As designed, OSG is an excellent solution for use on multi-screen systems due to its read-only culling and drawing traversals. This allows several rendering threads to run simultaneously along with the simulation update thread with proper locking. OSG is also more suitable for highly dynamic objects (such as particle systems) than other scene-graph systems.

Optimization techniques such as frustum culling and lazy state sorting give OSG a significant and necessary performance increase over traditional pure OpenGL applications. Optimization is achieved through separating simulation computation from updating changes to the scene-graph. In this way, simulation computation and the graphics rendering can take place in parallel.

Traditionally, large terrains are visualized by view-dependent level-of-detail (LOD) algorithms. These methods are not preferred for virtual reality systems because of the difficulty keeping congruency across multi-screen displays. View-dependent LOD algorithms use render-time perspective matrices to adjust the detail of the terrain. In multi-screen systems, the perspective matrix changes for each screen rendered and the terrain must be recalculated for each perspective matrix. Furthermore, because each screen's LOD is calculated independently, the terrain may have different heights when crossing from one screen to its neighbor, destroying the immersion.

We investigated two VR friendly solutions. The first is well suited for small to medium sized terrains with texture sizes that can fit directly into graphics memory. The second method is better for larger terrains that cannot be entirely contained within the system. The implementation of our first method is based on Boer's geometrical mip-mapping method [2]. The second terrain rendering algorithm we implemented uses pre-computed tile-based quad-tree terrain databases, such as the algorithm described by Ulrich [17]. The overhead of paging tiles results in less performance than the first method, but has the advantage of rendering larger, more detailed terrains.

4.2 Fire and smoke

Fire and smoke are complex dynamics systems that that change continuously. We chose to represent these phenomena with particle system effects because they provide the most realistic real-time fire and smoke visualization [11] (Figure 2). OpenSceneGraph provides a particle system node, however this node did not conform to the OSG design rules, and consequently posed problems for multi-pipe rendering. Our solution was to modify the particle system node to push all the scene-graph modifications out of the cull traversal and into the update traversal.

Figure 2: Smoke is rendered using a particle system based node in OpenSceneGraph. A similar method is used to visualize fire.

4.3 Fire spread visualization

The spread of the wildfire in our visualization is controlled by data produced by the FARSITE computational simulation package. FARSITE simulations consume too much time to be suitable for real-time visualization. However, it is possible to visualize the resulting simulation data. Our immediate goal was to determine the fire front at different times during the simulation. This was accomplished by wrapping the smallest convex polygon around areas currently under combustion at particular time intervals as determined by the simulation output. This is important in determining which areas are on fire as the simulation progresses. The FARSITE 2D visualization uses white contours to represent the fire spread as shown in Figure 3.

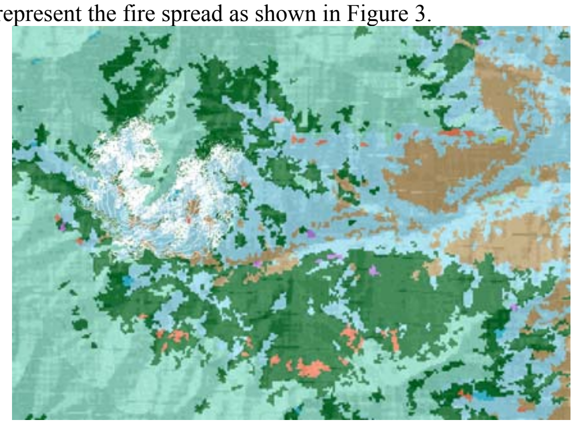

Figure 3: Standard FARSITE 2D visualization output includes white contours that represent the fire boundary in intervals of 30 minutes. Other colors are mapped to different levels of fuel load.

The output data from the FARSITE computation is formatted such that each simulated cell reports the time when the fire spread to its region. This interesting format requires peculiar methods of translating the data into a temporal rendering. Our initial approach uses a convex hull algorithm to find the region encompassed by the fire at a particular moment in time (Figure 4). This works well in general, but does not find the tightest perimeter of the fire. It also does not address the concept of spot fires, where an ember flies beyond the boundary of the fire, starting a new small burn.

Figure 4: The fire boundary is represented as an alpha-blended red wall.

Fuel load data, an input to FARSITE simulations, is also used by the visualization tool to place vegetation, and ignite that vegetation, when found to be within the fire area. The fuel load data controls where fire can travel within the growing fire front resulting in a visual estimation of the real wildfire.

4.4 Combined Visualization

During development we have built our prototype system with terrain and fire simulation of a wildfire scenario for Kyle Canyon, Nevada. The scenario is constructed from remote sensed data including a digital elevation map, satellite photo, fuel load data, and FARSITE outputs. Provided adequate data for a location, we can easily extend our application to other locations and scenarios.

5 Conclusions

VRFire provides a visualization of a fire spread model in a realistic terrain constructed from remote sensing data. The current version of VRFire also provides a framework with great flexibility for future enhancement. Besides supporting fire research visualization efforts, VRFire can also be easily extended into a wildfire planning and training tool for the wildfire professionals.

By using existing software libraries for rendering (OpenSceneGraph), and integration with virtual reality display devices (FreeVR), we have been able to construct a usable application that will run on a variety of computational and display platforms. We have involved fire researchers and wildfire professionals in design and informal discussions throughout the development. The next phase of development will increate the visual realism of the experience and look to improve and formally analyze the user's experience.

6 Future Work

Increasing the accuracy and realism of visualizing wildfire scenarios is a primary focal point. This focus will specifically involve improving the interpretation and parameterization of the FARSITE data, using higher detail vegetation models and adding building models for inhabited land. The realistic visual appearance of the fire and smoke is a top priority for presenting wildfire scenarios.

In order to more accurately reproduce the shape of the wildfire visualization of the FARSITE data, we plan to include spot fire detection and support for multiple fire fronts. Future work will result in the utilization of more FARSITE simulation outputs to control fire duration, crown fire visualization, and fire line intensity. Wind vector data will determine the flow and direction of smoke in the atmosphere.

As the visualization features coalesce, our effort will shift to user interface design, and will include more formal analysis of usability.

Acknowledgements

This material is based upon work supported by the U.S. Army's PEO-STRI, NAVAIR under Contract No. N61339-04-C-0072.

References

[1] R. Bukowski, C. S´equin. "Interactive simulation of fire in virtual building environments." In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH 97)*, pages 35-44, 1997, Los Angeles, California.

[2] W. H. de Boer. "Fast Terrain Rendering Using Geometrical MipMapping." http://www.flipcode.com/articles/2000 (accessed September 13, 2006).

[3] D. Drysdale. *An Introduction to Fire Dynamics*, Wiley & Sons, 2001.

[4] M. A. Finney. "FARSITE: Fire Area Simulator – Model Development and Evaluation." Research Paper RMRS-RP-4 Revised, USDA Forest Service Rocky Mountain Research Station, pages 47, March, 1998.

[5] J. Govindarajan, M. Ward, J. Barnett. "Visualizing Simulated Room Fires." In *Proceedings of the conference on Visualization '99*, pages 475 – 478, 1999, San Francisco, California.

[6] F. C. Harris Jr., M. A. Penick, G. M. Kelly, J. C. Quiroz, S. M. Dascalu, B. T. Westphal. "V-FIRE: Virtual Fire in Realistic Environments." The 4th International Workshop on System/Software Architectures in *Proceedings of The 2005 International Conference on Software Engineering Research and Practice* (SERP '05), Vol I, pages 73-79, June 27-30, 2005, Las Vegas, NV.

[7] T. U. St. Julien, C.s D. Shaw. "Firefighter Command Training Virtual Environment." In *Proceedings of the 2003 conference on Diversity in computing*, pages 30 – 33, 2003, Atlanta, Georgia.

[8] R. Macredie, S. J.E. Taylor, X. Yu and R. Keeble. "Virtual reality and simulation: an overview." In *Proceedings of the 28th Winter Simulation Conference*, pages 669 – 674, 1996, Coronado, California.

[9] D. Maxwell. "Simulators for training firefighters." *Linux Journal*, Volume 2004, Issue 122, page 1, 2004.

[10] A. Muzy, N. Fauvet, P. Bourdot, F. Bosseur, C. Gouinaud. "A VR platform for field-scale phenomena: an application to fire spread experiments." In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 155 - 158, 2005, Dunedin, New Zealand.

[11] H. Nguyen. "Fire in the Vulcan Demo." *GPU Gems*, edited by Randima Fernando, Addison-Wesley, Boston, MA, 2004.

[12] J. Rohlf, J. Helman. "A high performance multiprocessing toolkit for real-time 3D graphics." In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques (SIGGRAPH 94)*, pages 381-394, 1994, Orlando, Florida.

[13] W. R. Sherman, A.B. Craig, *Understanding Virtual Reality*, Morgan Kaufmann Publishers, San Francisco, 2003.

[14] W. R. Sherman. "FreeVR." http://www.freevr.org/ (Accessed September 13, 2006).

[15] S. Takeuchi, S. Yamada. "Monitoring of Forest Fire Damage by Using JERS-1 InSar." In *Proceedings of the 2002 IEEE Geosience and Remote Sensing Symposium*, vol. 6, pp. 3290-3292, 2002.

[16] D.L. Tate, L. Sibert, T. King. "Virtual environments for shipboard firefighting training." In *Virtual Reality Annual International Symposium (VRAIS 97)*, pages 61 - 68, 215, 1-5 March 1997, Albuquerque, NM.

[17] T. Ulrich. "Rendering Massive Terraines using Chunked Level of Detail Control," *SIGGRAPH Course Notes*, 2002.

[18] Q. Yu, C. Chen, Z. Pan, J. Li. "A GIS-based Forest Visual Simulation System." In *Proceedings of the Third International Conference on Image and Graphics*, 2004, pages 410 – 413, 18-20 Dec. 2004.

[19] T. Zaniewski, S. Bangay. "Simulation and visualization of fire using extended lindenmayer systems." In *Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa*, pages 39 - 48, 2003, Cape Town, South Africa.

[20] Q. Zhu, T. Rong, R. Sun, Y. Shuai. "A study on fractal simulation of forest fire spread." In *Proceedings of the IEEE 2001 International Geoscience and Remote Sensing Symposium,* (IGARSS '01), Vol. 2, pages 801-803, 9-13 July 2001, Sydney, NSW, Australia.