A Distributed Virtual Reality Prototype for Real Time GPS Data

Roy Ladner[1], Larry Klos[2], Mahdi Abdelguerfi[2], Golden G. Richard, III[2], Beige Liu[2], Kevin Shaw[1]

[1]Naval Research Laboratory, Stennis Space Center, MS 39529
{rladner, shaw}@nrlssc.navy.mil
[1]University of New Orleans, Computer Science Department, New Orleans, LA 70148
{mahdi, golden, lklos, bliu}@cs.uno.edu

Abstract: We describe a prototype that provides distributed, three-dimensional, interactive virtual worlds, which are enhanced with reliable communication and recording of real time events throughout the system. These events correspond to personnel movements in the real world, which are captured from GPS transmissions and are reflected by the movement of 3D human figures within the corresponding synthetic environment in real time.

1 Introduction

The Naval Research Laboratory's Digital Mapping, Charting and Geodesy Analysis Program was actively involved with providing digital mapping and database services to the U.S. Marine Corps Warfighting Lab during its Advanced Warfighting Exercise in San Francisco. During the exercise, positional data was transmitted by Global Positioning System (GPS) equipment carried into the field by key personnel. It was then sent via routine client server architecture to a designated workstation where positions were marked with icons on a two-dimensional digital map. This enabled the user to keep track of the locations of key personnel during the exercise. Among our objectives was the examination of interfacing our work on 3D Geographic Information Systems with the Marines' use of GPS technology.

Efforts to interface with this system during that exercise lead to the development of a prototype, which incorporates research efforts underway at the Naval Research Laboratory and the University of New Orleans. This research is based on an extension of the National Imagery and Mapping Agency’s (NIMA) Vector Product Format (VPF) [1]. VPF is a government standard developed in the 1980’s for large geographic databases. The extended VPF, referred to as *VPF+* [2], is focused on the development of a database framework for 3D synthetic environments. Our prototype provides distributed, three-dimensional, interactive synthetic environments that are enhanced with reliable communication and recorded real time events throughout the system. These events correspond to personnel movements in the real world, which are captured from GPS transmissions and are reflected by the movement of 3D human figures within the corresponding synthetic environment in real time.

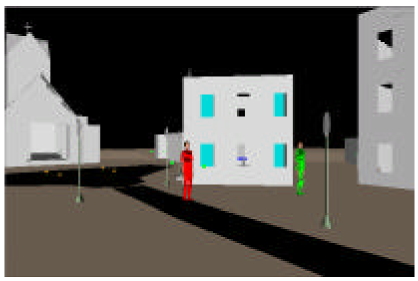
We are able to greatly improve situational awareness of individual viewers through the introduction of a 3D-world view using VPF+. To this we add a communication subsystem, *Julep* [3], which provides a distributed communication environment for message passing over connectionless, ‘unbreakable’ and dynamically reconfigurable channels. Julep also provides a set of fault tolerant components that enable guaranteed message delivery and stable system-wide message logging. These types of services are significant in mission rehearsals, field operations and future replay/analysis from multiple perspectives. The Julep communication environment eliminates substantial programming efforts usually associated with establishing these types of services and their associated communication guarantees.

2 Vpf+

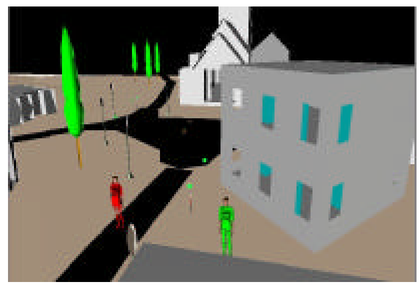
Our prototype deploys synthetic environments generated from a VPF+ enabled 3D Geographic Information System (3D-GIS) [4]. VPF+ makes use of a non-manifold data structure that facilitates the storage of the diverse set of features typically found in a synthetic environment into a single coverage, rather than in disjoint thematic layers. VPF+ also has the added benefit of providing full 3D topology to increase system performance and which should be useful for 3D query needs. For more detail on our work in 3D-GIS the interested reader is referred to [4].

Our use of VPF+ has allowed us to store within a single coverage, visually and geometrically accurate 3D features existing in a given location. This includes, for example, a terrain skin and geometrically detailed buildings. Building geometry in this case includes the correct size and shape of the building, the accurate placement of the doors and windows and the correct interior configuration of the floor plan. Specifically, we construct the environment such that the user can *walk* or *fly* across terrain and can *walk* into buildings through doorways or *climb* through open windows. Direct *line of sight* into and out of buildings through open doorways and windows is also possible. Additionally, once inside a building, the user can navigate through the building much the same as though he were in the real world building. This allows the user to gain increased knowledge of the operation by viewing the action from different perspectives. The benefits of this are shown in Figure 1 below.

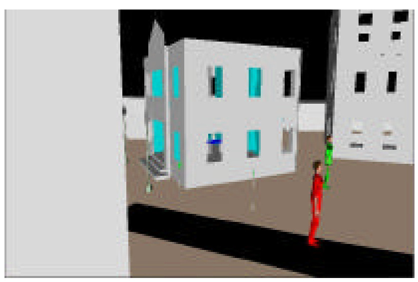
Figure 1 shows several views of part of a synthetic environment used in our prototype, the Military Operations in Urban Terrain (MOUT) facility at Camp LeJeune, North Carolina. The MOUT site comprises a city built by the Marines for training in urban operations. There is also a transportation network and the usual urban features associated with this type of setting such as trees, streetlights and park benches. We implemented the prototype with a subset of the buildings, roads and associated features. Figure 1 shows two icons representing the positions of the red and green team leaders within the MOUT facility during a mock exercise. The impact of allowing the user to view the action from different locations within the synthetic environment should be evident. Figure 1(a) shows the ground level view from a 'following' position, and the remaining figures show the same scene from different viewpoints. Figure 1(b) is from the rooftop across the street. Figure 1(c) is a second floor window view from the adjacent building. Finally, Figure 1(d) gives the view from overhead.



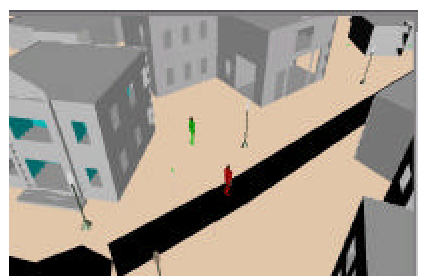
1(a) Ground Track View



1(d) Roof top View



1(c) Window View



1(d) Overhead View

Fig. 1. Some of the Possible Views of the Same Scene from Within Synthetic Environment of the MOUT Site

3 System Communication

Our prototype utilizes the Julep environment for handling system communication. The Julep environment is fully implemented in Java, and runs as a software layer between the distributed prototype application and the operating system. The core components of Julep actually function as a communication subsystem. Julep provides the distributed application with a set of message send and receive primitives, enabling network communication for all distributed nodes registered with the system.

Julep uses the User Datagram Protocol (UDP) connectionless protocol for performance reasons. This takes advantage of UDP's low channel establishment overhead. In the event of node failures, UDP allows a restarting node to immediately begin communicating to peer nodes with no overhead time required for connection establishment. Through Julep we add a protocol layer above UDP that makes system communication reliable and that allows system messages of unlimited size to be transmitted. Handling all system message passing through Julep provides the application programmer with the opportunity to manipulate communication in the distributed prototype as desired, for both fault tolerance and prototype functionality purposes.

Our prototype makes use of several of Julep's configurable fault tolerant components. One of these is an 'unbreakable' communication channel. If a sender task sends a message to a receiver task and the receiver task fails before message reception completes, the sender can be configured to block and retry sending repeatedly. Periodically the sender times out, clears its cache of the receiver's address and re-queries the Manager for a new address. When the receiver task restarts on a new machine and re-registers with the Manager, its new address will then be supplied to the sending task. This allows the message to be successfully transmitted, even with the receiver failing and restarting on a new machine midway through the send operation. This dynamic message rerouting guarantees message delivery and is the basis for the 'unbreakable' communication channel mechanism of Julep.

Another fault tolerant component the prototype makes use of is Julep's passive status tracking of live group members. If a group member fails during a session, the Manager can be configured to flag the member as *dead*, essentially removing that member from the group. When another task queries the Manager for the locations of all group members in order to send a group message to them, the address of the failed member is not included. This avoids the possibility of the sending process blocking indefinitely waiting for a response from a failed group member that may never come.

One more set of fault tolerant components the prototype takes advantage of is Julep's message logging functionality. The Manager can be configured to log incoming registration records from all hosts in the system, as well as all group management messages. In the event of manager failure, it can be restarted (on the same machine), and it can transparently regain its previous state relative to the rest of the system by reading all logged messages. Also, individual nodes in the system have the ability to log all incoming messages. Apart from fault tolerance purposes, this logging capability acts as a recording device for the system state at all nodes, allowing future playback of events for purposes of analysis. With the prototype's ability to change the perspective being viewed, repeated playbacks can be set to view the same sequence of events from a series of different perspectives as shown in Figure 1.

4 The Prototype Interface

Figure 2 is a screen shot of the prototype user interface showing a portion of the 3D synthetic MOUT site and 3D figures representing two team leaders. The prototype runs as a Java applet in a common Web browser with a 3D graphic plug-in like CosmoPlayer. While the prototype was designed to handle movements of four key Bravo team leaders for the Red, Green, Blue and White teams, it is theoretically possible to track the movements of an arbitrary number of people. Once data reception is initialized, the 3D interface is updated to show the current location of each team leader as she/he moves through the actual MOUT location.

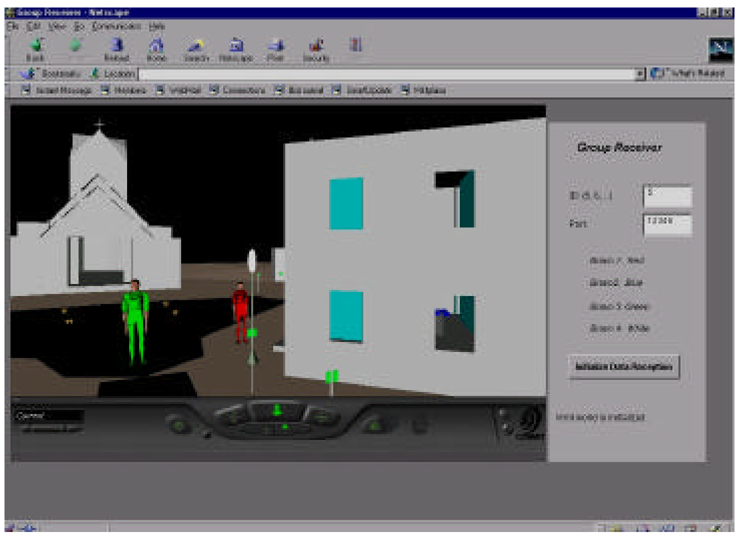


Fig. 2. User Interface to Distributed Virtual Reality Tracking Application

The conceptual organization of the prototype is shown in Figure 3. Field GPS transmitters periodically update field coordinate positions. These new coordinate locations are received and transmitted to multiple recipients using Julep's communication and fault tolerant facilities with a basic process group protocol. The process group protocol allowed us to define target groups for possibly different message delivery guarantees. Julep allowed us to greatly simplify the entire process and provide a robust application.

5 Observations and Future Work

We have described a prototype utilizing research at DMAP and the University of New Orleans into the generation and deployment of 3D synthetic environments using a new extension to NIMA's Vector Product Format known as VPF+. VPF+ defines a new, extended Winged-Edge data structure referred to as "Non-Manifold 3D Winged-Edge Topology" and which is suitable for modeling 3D synthetic environments. Our prototype deploys distributed, interactive 3D synthetic environments that are enhanced with reliable communication and recorded real time events. These events correspond to personnel movements in the real world, which are captured from GPS transmissions and are reflected by the movement of 3D human figures within the corresponding synthetic environment in real time. The prototype makes use of a communication subsystem, Julep, currently under investigation at the University of New Orleans.

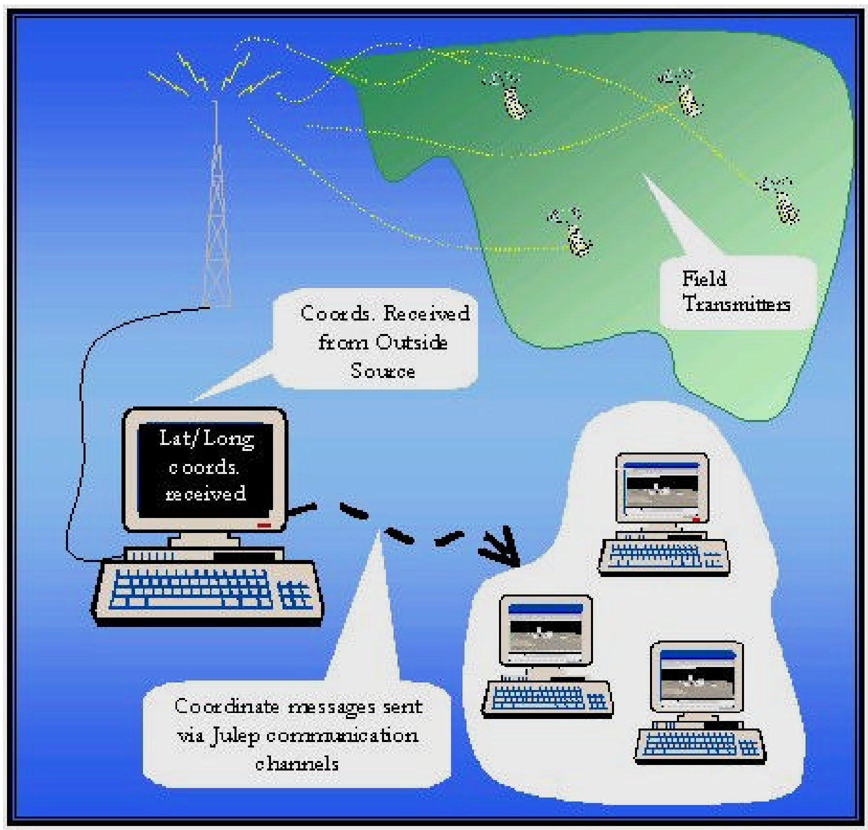


Fig. 3. Conceptual Design of Prototype

Plans for the future include incorporation of VPF+ 3D data into the Naval Research Laboratory's Geospatial Information Database (GIDB) [5]. The GIDB is an object-oriented digital mapping database that employs a CORBA-compliant approach. Data access is by way of a simple Web-browser. A two-dimensional graphical data display is currently available. The incorporation of VPF+ will allow users access to realistic, interactive 3D virtual worlds on the Web with a common browser plug-in.

Planned modifications to the Julep system will allow "group send" events to be implemented via a message multicast, giving substantial savings on network load overhead. The prototype could also be directly modified to enable communication between participant nodes and between any subset of participants.

6 Acknowledgements

The National Imagery and Mapping Agency, the U.S. Marine Corps Warfighting Lab and the Louisiana Board of Regents Support Fund (LEQSF) sponsored this work.

References

1. Department of Defense, Interface Standard for Vector Product Format, MIL-STD 2407, 28 June 1996.
2. Mahdi Abdelguerfi, Roy Ladner, Kevin B. Shaw, Miyi J. Chung, and Ruth Wilson, VPF+: A Vector Product Format Extension Suitable for Three-Dimensional Modeling and Simulation, Technical Report NRL/FR/7441—98-9683, Naval Research Laboratory, Stennis Space Center, MS, 1998.
3. L. Klos, G. G. Richard III, Z. Xu, "Julep: A Framework for Reliable Distributed Computing in Java," University of New Orleans Technical Report UNOCS-TR99-01.
4. Roy Ladner, Mahdi Abdelguerfi, Kevin Shaw, 3D Mapping of an Interactive Synthetic Environment, IEEE Computer, Vol. 33, No. 3, March 2000, pp. 35-39.
5. More information on DMAP and ongoing database research is available at http://dmap.nrlssc.navy.mil/dmap.