Rocinante, a Virtual Collaborative Visualizer*

Michael J. McDonald
Sandia National Laboratories
Intelligent Systems and Robotics Center
Albuquerque, NM 87185-1004

Lisa G. Ice
The University of New Mexico
School of Engineering, Computer Science Department
Albuquerque, New Mexico

I. Abstract

With the goal of improving the ability of people around the world to share the development and use of intelligent systems, Sandia National Laboratories' Intelligent Systems and Robotics Center is developing new Virtual Collaborative Engineering (VCE) and Virtual Collaborative Control (VCC) technologies. A key area of VCE and VCC research is in shared visualization of virtual environments. By sharing kinematic information across the Internet, collaborators with similar geometric models can visualize and share decision making on complex, moving equipment both before and while it is operational.

This paper describes a Virtual Collaborative Visualizer (VCV), named Rocinante, that Sandia developed for VCE and VCC applications. Rocinante allows multiple participants to simultaneously view dynamic geometrically-defined environments. Each viewer can exclude extraneous detail or include additional information in the scene as desired. Shared information can be saved and later replayed in a stand-alone mode. Rocinante automatically scales visualization requirements with computer system capabilities. Models with 30,000 polygons and 4 Megabytes of texture display at 12 to 15 frames per second (fps) on an SGI Onyx and at 3 to 8 fps (without texture) on Indigo 2 Extreme computers.

In its networked mode, Rocinante synchronizes its local geometric model with remote simulators and sensory systems by monitoring data transmitted through UDP packets. Currently, device positions and device "joint" or kinematic values are shared. As each new packet arrives, Rocinante updates its internal database and displays all of the updates during the next rendering cycle. While typical data rates are under 100 packets per second (pps), rates of 6K pps have been achieved on local area networks without impacting rendering speed.

Rocinante's scalability and performance make it an ideal VCC tool. Users throughout the country can monitor robot motions and the "thinking" behind their motion planners and simulators. Combined with network video solutions, users can verify that the model is in agreement with reality and then use the model's enhanced representation capabilities to assess plans and actions. In addition, the VCV can be used as a tool by both software and hardware developers to determine how their equipment interacts with specific environments. In these ways, Rocinante supports both collaborative development and use of intelligent systems.

Introduction

Use of the Internet for engineering collaboration is increasing dramatically with the result being improved problem solving effectiveness and speed for geographically distributed teams. Sandia has found that graphic-rich Virtual Collaborative Environments significantly improve the quality and performance of remote collaboration for both engineering (via VCEs) and control (via VCCs). Leveraging engineering collaboration technologies, Sandia routinely uses of the Internet to let researchers access and collaborate on intelligent machine systems.[1] More recently, Sandia has shown that VCCs can dramatically improve operational efficiency of many intelligent systems tasks.[2]

VCCs allow combining the specialized skills and tools of varying disciplines to create highly effective teams. For example, the special problems that can occur during hazardous waste remediation can best be solved by combining skills of regulatory and waste source experts, operational supervisors, robot engineers, and operators. As a result, Sandia is now focusing its VCC efforts to build systems that support the information flow between experts to effectively and efficiently bring together their varied skills. The Rocinante VCV, described in this paper, is being built to study these information flows.

*This work was supported by the United States Department of Energy under Contract DE-AC04-94AL85000.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Background

Collaborative Engineering tools are being developed to allow several engineers to interact through virtual environments. These environments are created by either distributing the interface of a single program set or by linking groups of programs through live data feeds.

Sharing interfaces is an effective way to allow multiple experts (who all understand the way a particular program presents its information) to work together. Several commercial tools let user teams simultaneously launch, see and/or interact with a single set of programs from their respective computers.[3,4] Whiteboards and video conferencing are available to let users efficiently communicate textual, graphical, and pictorial information.[5] Some companies, including Deneb, are developing custom VCE interfaces to support the special needs of their highly graphical systems.[6]

The Department of Defense sponsored Distributed Interactive Simulation (DIS) and follow-on High Level Architecture (HLA) standards are being developed to link many (several hundred) diverse defense simulation tools, each run by separate users, to form a single high-performance simulation.[7] Unlike sharing an interface, DIS shares model information at an informational packet level where each packet represents an action (e.g., creation, position, motion change, interaction with, or destruction) of an agent (e.g., military unit, device, or solder). Individual users can control the behavior of predetermined groups of agents and monitor the actions of other agents. Multi-participant Virtual Reality (VR) applications have been built using similar agent technologies.[8,9]

DIS simulations have been run with several hundred users in mixed computing environments. Platforms ranging from 2D battlefield drawings can easily cooperate with 3D flight, tank, or other device simulators. In several examples, real military equipment has been linked to simulations to validate simulations, provide an optional input format, or to use the simulation environments for battlefield C3I.

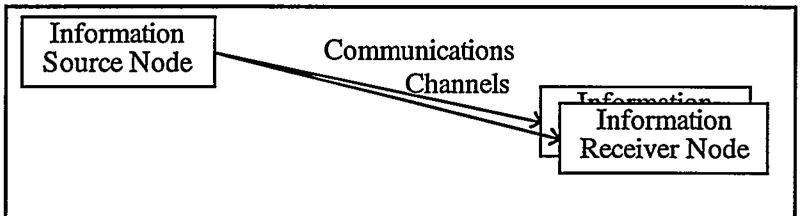

Figure 1: General communications system used by Rocinante

Work is ongoing to support kinematic devices and VR agent protocols in DIS and HLA.[10] However, because the needs are different, achieving high performance VR or distributed robot simulations requires DIS or HLA extensions.

The Rocinante Virtual Collaborative Visualizer

Rocinante is a VCV developed for collaborative engineering and control of robot or kinematically describable machine systems. Rocinante allows many people to simultaneously view the geometric world of rapidly changing robotic simulation and control environments.

Every attempt has been taken to minimize the custom software to produce and use Rocinante. Rocinante is built on Silicon Graphic's (SGI) Performer visualization software library, it uses the WaveFront non-proprietary, industry standard file format to describe the bulk of its geometric model, and it reads information updates as text packets transmittable from any Internet-compatible computer.

Likewise, Rocinante interoperates with engineering tools that can be extended with user-compiled routines (several are described below). As a result, users are not forced to switch to new tools in order to utilize its functionality. Because Rocinante uses Internet communications protocols, these users need not be in the same physical facility. Thus, for example, Rocinante can let several engineers simultaneously visualize a robot motion plan from workstations distributed throughout the world. Teamed with other Internet communication tools (e.g., whiteboard, voice, and audio transmission), Rocinante allows cross-institutional engineering teams to effectively solve problems without travel or relocation.

Rocinante Information Sharing

A communications approach was developed to let information source nodes broadcast kinematic state information to receiving nodes through communications channels. Figure 1 shows the general configuration. Information is distributed on a subscription basis and receivers can start or stop reading any time after a subscription is established. Source nodes periodically rebroadcast static information to assure that receiver nodes get all slowly changing and static information (much as stock market information servers rebroadcast unchanged stock values). A communications protocol was chosen that only allows one direction of information flow on communications channels. Each node in Figure 1 can be located in a different facility and multiple nodes can be configured to provide multi-way communications.

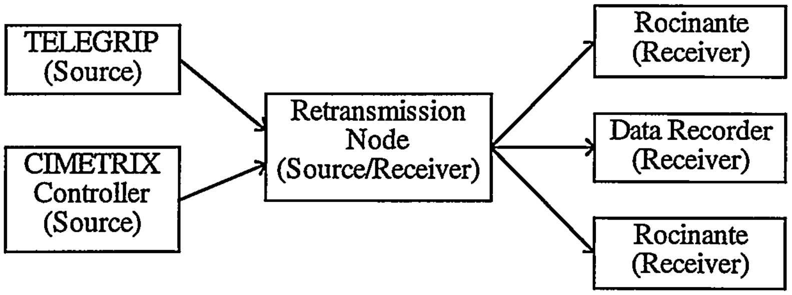

Figure 2) Typical communications configuration. Here, two data sources are being combined and retransmitted to two copies of Rocinante and one data recording receiver.

In Sandia's collaboration systems, source nodes are typically engineering simulation systems, such as Deneb's Telegrip. Receiver nodes are typically visualizers like Rocinante or engineering tools like Telegrip. In addition, databases can receive data for retransmission or *playback*, rebroadcasting nodes can provide efficient wide area network support, and filtering nodes can be used to reduce data traffic to some users.

Figure 2 shows one configuration of source, retransmission, and receiver nodes demonstrated at the 1996 Robotics Technology Development Program Forum. Here, a graphical programming system (that uses Telegrip) was used to plan and control one robot's motion,[11] another robot was driven with a hand controller (programmed on a Cimetrix controller) and monitored with Rocinante, while a third person used Rocinante to monitor the overall system for safety issues. As the demonstration progressed, software systems joined in and logged out of the system without interrupting others.

Rocinante Information Sources

Sandia has modified several robotic simulation and control systems, including Deneb's Telegrip robot simulator, a Sandia-developed micro-robot simulator, Penn State's Virtual Tools[12] software, Cimetrix's open architecture robot controller, and the SMART telerobotic control system to function as information source nodes that interoperate with Rocinante. For systems that manage few kinematic devices (e.g., Cimetrix, SMART, and Virtual Tools), the needed modification involved adding network transmission code to transmit particular device's joint angles to the programs' main event loops.

More sophisticated algorithms are used to track information from Telegrip. Telegrip allows kinematic devices to be added, modified (moved), and deleted with great frequency. To support these dynamics software was developed that inspects the internal database at each simulation update, determines which changes have occurred, and transmits the new information with each change. In addition, the software rebroadcasts static and slowly changing data periodically to recover lost data (as can occur with UDP) and allow nodes to *come on line* in mid-session. Because Telegrip supports very rapid data access routines, this complex procedure has negligible computational overhead.

Other information sources developed include database, filtering, and rebroadcasting utilities. A database was built from one module that stores packets as they are received and another that reads the data and transmits the records as packets. A rebroadcasting utility was built to allow data to be directed toward several workstations and across the Internet in a Wide Area Network (WAN) efficient manner.

Rocinante Internet Transmission

The two basic Internet protocols, TCP and UDP were investigated for VCV communications. The TCP protocol offers reliable transmission at the cost of increased network traffic and reduced throughput. UDP, a lower-level protocol that TCP is built from, allows higher rates of transmission at the cost of possible data loss. Due to round trip network delays, it was determined that use of UDP could provide a factor of 20 improvement over TCP on typical Internet connections. In VCVs, a higher priority is placed on having current information than reliably transmitting all information. As a result, UDP was chosen for Rocinante.

Three transmission methods-directed, broadcast, and multicast-were investigated. Broadcasting is inappropriate for WAN use. Multicasting was recognized as the appropriate mechanism when several receivers are used on the same network node. Directed transmission is the easiest to configure and does not add significant network overhead with modern network topologies. Directed transmission was completed first and multicasting is planned for future implementations.

Rocinante Program Flow

Rocinante, architected according to standard SGI Performer application development strategies, uses a pipelined programming loop that includes an application, a cull, and a draw block. The cull and draw blocks together optimize the rendering of the database onto the user's

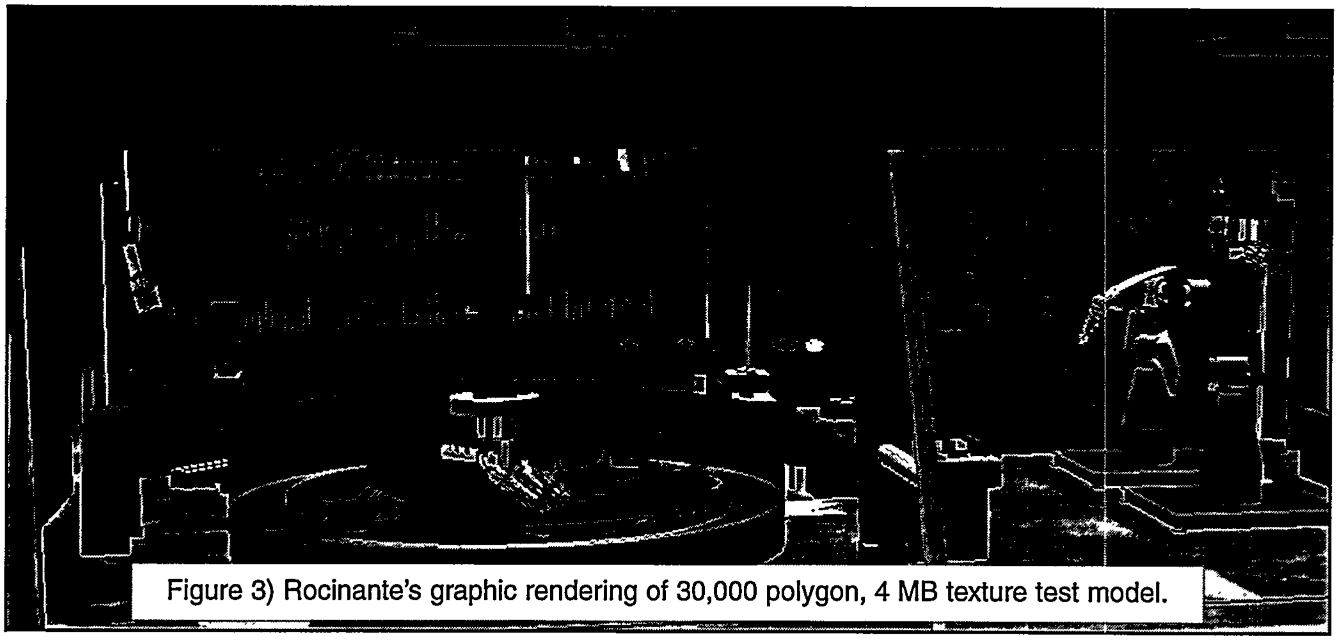

Figure 3) Rocinante's graphic rendering of 30,000 polygon, 4 MB texture test model.

display. In multi-processor machines, the application, cull, and draw blocks each run as separate processes and communicate through shared memory. In addition, a unique data interface module runs as a forked process to receive and interpret each data packet for rapid insertion into the application block.

Flow control in the data interface module is a simple loop. A blocking read is posted, data is read and interpreted, and the result is written into a shared memory ring buffer. The blocking read allows efficient operating system context switching between this and other processes.

At initialization time, the application block creates a run-time database by reading a combination of standard geometry description files (in Wavefront format) and custom files that map geometry to device kinematics. At run time, the application block performs needed run-time database manipulations to achieve the kinematic transformations read from the ring buffer. In addition, the application block services user input including, for example, viewpoint changes and menu calls. Unlike DIS, Rocinante does not simulate motion between packet updates and unlike the engineering simulators, it does not perform other simulation computations. As a result, this application block completes very quickly and Rocinante runs very efficiently on all SGI platforms.

Performance Results

Rocinante was tested on three types of computers and six types of information sources by a variety of users. Graphic performance and packet reception rates were tested. Compatibility with several information sources including Telegrip, Cimetrix, SMART, and several specialized codes was demonstrated. The system was used by six primary users and shown to several dozen other users for informal evaluation.

Maximum usable model complexity is mainly governed by workstation performance. A model with 30,000 polygons and 4 Megabytes of texture (shown in figure 3) rendered at 12 to 15 frames per second (fps) on SGI Onyx RE2 computers configured with dual 150MHz R44000 processors. The same model displayed at 3 to 8 fps (without texture) on a variety of Indigo 2 Extreme computers with 100 to 200 MHz R4400 processors. Models with less than 1,000 polygons performed reasonably (over 5 fps) on 100MHz R4000 Indy computers (without graphic hardware) but larger models rendered unacceptably slowly.

A database was used to generate up to 6,000 pps of data (the maximum achievable on 10BT Ethernet) to a visualizer running on an Onyx computer. At 6K pps, it was found that the visualizer's packet interpreter fell behind at a rate of approximately 10-25% but that the display rate did not slow noticably.

Telegrip was loaded with the 30,000 polygon model for several tests. Several 5 minute simulation runs were timed and no noticable difference was found with and without the data source module enabled. In normal use, transmission rates from Telegrip were less than 100 pps. Testing also showed that the visualizer consistantly achieved higher frame rates than Telegrip when similar platforms and parallel models were used. As a result, computational requirements for the visualizer are comparable or lower than the simulator.

System testing was performed under a variety of configurations including those noted earlier. The UDP proto-

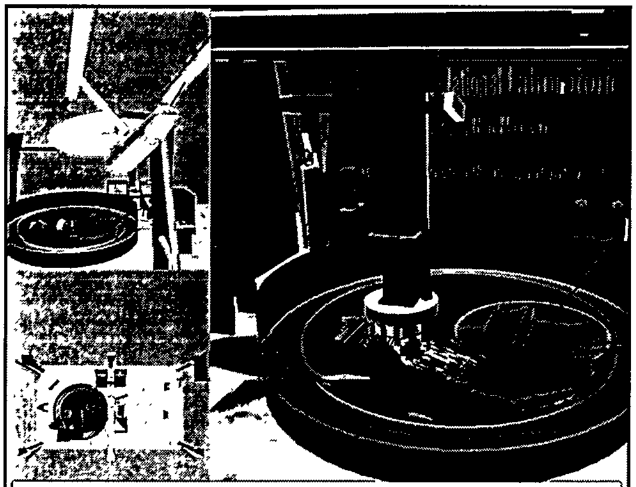

Figure 4) Rocinante with navigation aid windows activated. The eye point of the right frame is attached to the spaceship of the left frame.

col interface was rapidly implemented on several computer platforms including SGI's IRIX 5.3, LYNXOS, and VxWorks 5.2. Directed transmission was found to be very easy to configure as sources and receivers could be started, stopped, and restarted in any order without special coordination.

User evaluations were very favorable. High frame rates, high quality rendering with sophisticated support of textures, and program simplicity were rated highly. Several novel user interface options, including a navigation aid shown in figure 4 (right most windows) were also appreciated. New users successfully and efficiently used the visualizer from remote (far from the robot) VCC control nodes within hours of introduction. Some confusion was noticed when new users who were accustomed to complete simulators such as Telegrip expected the visualizer to perform simulation computations. It was also noted that Performer's standard viewing models (e.g., flying) were sometimes difficult to use with engineering models.

Conclusions

This effort demonstrated the value of graphically displaying kinematic information to users in VCC environments. The effort further showed that existing computational tools and platforms can be combined with sufficient performance to achieve high performance shared graphical environments.

Tests in graphic rendering rates showed that very complex models can be efficiently rendered on computers that have been available since 1993. (We estimate that currently available $10,000 workstations have sufficient performance for most VCV needs.) This shows that high level graphic rendering tools, like Performer, have excellent potential in engineering visualization.

Tests in data transmission and interpretation rates show that the protocol, as implemented, has sufficient performance. While it is believed that interpreter speeds are the major limiting factor in receiving data, performance loss from using text packets is not significant for normal uses. Projecting tests on Ethernet to slower links shows that the packet size will allow a reasonable transfer rate on even slow network connections. (We estimate a 28.8 kBPS V.34 modem will support 20-30 updates/second, fast enough to support two robots)

Future Work

Rocinante provides a good testbed for initial VCV development. Our short term plan is to evaluate its use in it in a variety of situations and identify further VCV requirements.

Further work is needed to make VCVs practical in a wide variety of applications. Several emerging standards are making progress toward broad support of the underlying technologies that VCVs rely on. The VRML 2.0 specification, when combined with Java, appears to have sufficient capability to allow development of lower-performance VCVs. HLA and related tools will provide a significant toolset for further VCV communications development.[13] While current VRML renderers do not equal Performer's speed, this situation may change with the evolving COSMO standard. Industry interest for bringing high performance 3D rendering to the home computer, as available on the Nintendo 64 computer, is high. It is hoped that continued development in this direction will make VCVs as common as web browsers.

XII. Acknowledgments

We specially thank Jim Pinkerton and Maria Gallegos-Pinkerton for their significant contributions to the conceptual design and development to the first working prototypes of the Rocinante VCV. A majority of this work was financially supported by the Department of Energy's Office of Science and Technology through the Robotics Technology Development Program on Technical Task Plan AL2-5-C1-61.

Bibliography

1 Harrigan, R. W., M. J. McDonald, and B. R. Davies, *Remote Use of Distributed Robotics Resources to*

Enhance Technology Development and Insertion, **1994, International Symposium on Robotics and Manufacturing (ISRAM'94).**

2 McDonald, M. J., D. Small, D. Cannon, C. Graves *Virtual Collaborative Control to Improve Intelligent Robotic System Efficiency and Quality,* **1997 IEEE International Conference on Robotics and Automation,** April,, 1997

3 Ashby, M. R. and H. W. Lin, *Interactive Collaborative Environments (ICE): Platform Independent X Application Sharing and Multi-Media over Wide Area Networks at Sandia National Laboratories.* **Fifth Intl Symposium on Robotics and Manufacturing,** Maui, 1994

4 Symantic's **PCAnywhere** (http://www.symantec.com/ pcanywhere/index.html), Farallon's **Timbuktu** (http://www.farallon.com/www/ catalog/cattoc.html), & Microcom's **Carbon Copy** (http://www.microcom. com/cc/cc.htm) enable PC interface sharing.

5 Intel's **ProShare** Personal Conferencing and SGI's **InPerson** (http://www.sgi.com/Products /inperson_main.html) incorporate whiteboard interfaces.

6 New Computer Technology Dissolves Limitations to Traditional Design and Engineering Methods. Sandia News Release September 6, 1996. (http://www.sandia.gov/media/vce.htm)

7 DIS and HLA are sponsored by the US DOD Defense Modeling and Simulation Office (DMSO, http://www.dmso.mil), DIS research is reported at the semi-annual Workshop on Standards for the Interoperability of Defense Simulations (see http://stds.sc.ist.ucf.edu).

8 Stansfield S., N. Miner, D. Shawver and D. Rogers, *An Application of Shared Virtual Reality to Situational Training,* **Proceedings of the Virtual Reality Annual International Symposium,** Triangle Research Park, NC., March, 1995.

9 Multi-user VR is a standard feature in Deneb's Telegrip and Envision software MPI option.

10 Macedonia, Michael R., Zyda, Michael J. *A Taxonomy for Networked Virtual Environments* accepted for a future issue of **IEEE Multimedia.** (see also http://www-npsnet.cs.nps.navy.mil/npsnet)

11 McDonald & Small, *Graphical Programming of Telerobotic Tasks,* **ANS 7th Topical on Robotics and Remote Systems,** Agusta GA, April, 1997

12 Wang, C. and D. Cannon. 1996. *Virtual Reality Based Point-and-Direct Telerobotic Inspection.* **IEEE Transactions on Robotics and Automation Journal** 12 (4): 516-531.

13 Brutzman, Donald P., Macedonia, Michael R., Zyda, Michael J. *Internetwork Infrastructure Requirements for Virtual Environments,* **Proceedings of the Virtual Reality Modeling Language (VRML) Symposium,** San Diego Supercomputer Center (SDSC), San Diego, CA, December 13-15, 1995.

*Other Products Mentioned are described at
Deneb Inc. http://www.deneb.com
Silicon Graphics Inc. http://www.sgi.com
Cimetrix Inc. http://www.cimetrix.com

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.