



EXPLORING THE APPLICATION OF VIRTUAL REALITY TO REMOTE ROBOT OPERATIONS

Sharon Stansfield [{{contact info}}](#)

Nadine Miner

Christopher Cooke

Sandia National Laboratories

Albuquerque, NM

Editor: Mel Siegel

ABSTRACT

This paper presents two components of a prototype system exploring the application of virtual reality (VR) and telepresence to remote robotic cleanup of hazardous materials. The goal is to provide an interface between the human operator and the robots and sensors used for handling such materials that is both easy to learn and intuitive and natural to use. Sandia National Laboratories has developed a model-based control system which uses graphical models of a remote environment to allow the operator to both control the robots and to monitor the operations as they occur. The work presented here looks at extending that paradigm to allow the operator to build the required graphical models interactively and to control the system via a multi-sensory, immersive VR interface.

1. Introduction

Applications such as the retrieval of hazardous waste from deteriorating underground storage tanks create a need for new methods of programming and remotely operating robotic systems. Because such tanks contain highly radioactive material, it is not possible to allow human workers to directly carry out retrieval tasks. These operations must be done remotely, using robots. Unfortunately, both the tanks and the robotic systems are very complex. The tanks, for example, are filled with obstacles, such as cooling pipes, risers, pumps and waste materials. The consistency of this waste may range from viscous fluid to solid. The exact contents of any specific tank is not known a priori. The robotic systems employed will consist of several different manipulators, tools, and sensor devices. Operating such complicated systems is difficult, if not impossible, using traditional telerobotic techniques.

Researchers at Sandia National Laboratories have developed a graphical control system which allows the operator to program and operate robotic systems using a supervisory mode of control [5] and [10]. The operator interacts with a simulation of the robot and environment, sending commands and monitoring operations via a graphical interface. The work presented here looks at extending this graphicsbased controller in two ways. First, it explores the use of VR and telepresence to interactively build a graphical model of the remote site. The success of the control system relies heavily on the accuracy of the model being used to program robot operations. Since there is little a priori knowledge about the contents of a given tank, building an accurate model is essential for generating motion plans for the manipulators and tools which will enable them to carry out the specified operations without damaging either the devices or the tank structure in the process. The second component of this work looks at using VR to create a more intuitive, easily learned interface to this control system. The goal is to decrease the cognitive load on the operator by allowing him/her to utilize multiple senses to interpret information and by addressing some of the limitations of the traditional flatscreen and mouse interface, such as the lack of depth perception.

The goal tasks for this work are twofold: To interactively add an existing, but unmodeled, object to a partial graphical model of piping; and to control a robotic simulation of the Sandia robotic testbed via a VR interface. The mechanisms for down-loading verified control commands from the simulated robots to the actual robots exists as part of the Sandia graphical control system. Hooks to this existing software have been developed, but this aspect of the system has not been fully integrated. The reader is referred to [5] and [10] concerning this system.

2. The Prototype System

2.1 HARDWARE

The hardware used for this work consists of the following: A Silicon Graphics, Inc. (SGI) Crimson/ Reality Engine, an SGI Indigo sound server, a Fakespace Labs BOOM2C VR viewer, a Fakespace Labs MOLLY camera platform, a BG Systems Flybox, an RGB Spectrum image digitizer, a Dragon Systems DragonWriter voice recognition system, and an ultrasonic range sensor. The BOOM is a six degree of freedom, stereoscopic viewer. It is mechanically tracked, producing both low latency and high accuracy in tracking the user's motion for graphics updating. The resolution of the BOOM, which is CRT-based, is 1280x1024 pixels per channel providing a clear image and wide field of view to the user. MOLLY is a three-degree-of-freedom platform with two monochrome CCD cameras mounted on it. MOLLY's motion is slaved to the yaw, pitch and roll of the BOOM. Slaving the camera platform to the motion of the user's head gives the operator a sense of immersion in the remote environment. The stereo images also provide depth perception to the operator. The frame grabber is used to capture digitized video images and to provide graphical overlay capabilities. The ultrasonic sensor is mounted on the MOLLY platform and has a range of 6 inches to 60 feet. The Flybox provides a 3D joystick which is used to control a graphical pointer. The voice recognition system is used for interpreting vocal commands from the user to the interface.

2.2 SOFTWARE

The modeling and simulation software used for this work is SILMA's CimStation. CimStation is a robotics simulation package which can be extended and customized to provide added functionality. Extensions can be written in SIL, CimStation's native programming language, or linked in with code written in C or Fortran. This feature allows users of CimStation to customize their working environment. Hardware devices are interfaced to CimStation by linking driver libraries. CimStation was chosen for this work because of its extensibility and high-level graphics functions. Using SIL, graphical objects (e.g. block, cylinder, sphere) can be created and positioned during execution. Pre-existing models can be loaded using a menu-based interface. In addition, CimStation provides the capability to graphically model complex environments and robots. The robot models provide accurate kinematics and simulate the motions of the real robots correctly. Real-time collision detection between graphical objects in a model are also provided.

2.3 APPLICATION TESTBED

The application testbed for this work is Sandia's Advanced Controls and Manipulation Laboratory/ Underground Storage Tank (ACML/UST) mockup [7]. The ACML/UST facility was developed to facilitate design, integration and testing of waste remediation technologies. It consists of two robots (Cincinnati-Milacron 786 and Schilling Titan II hydraulic manipulator attached to the wrist of the CM 786), a gripper tool attached to the end effector of the Titan II, two inch and four inch hydraulic cutting tools with ultrasonic docking sensors, optical sensors and force measurement sensors. A tank mockup consisting of a rectangular wooden box containing pipe stands, pipes and sand (to simulate the radioactive waste) is used to simulate the UST environment.

3. Interactively Building Models of a Remote Site

Graphical models convey geometric information. Such models are usually created using tools such as computeraided design (CAD) packages. However, building graphical models with a CAD system requires that the operator have geometric information for the objects to be modeled. This section presents a prototype system for building and verifying graphical models of remote sites that does not require the operator to have a priori information about the objects in order to model them. The system provides an interface between the operator, the computer and a camera system located at the remote site. This interface utilizes both VR and telepresence to provide the capability to interactively create graphical models of the site where the cameras are located.

Previous, related work includes that of Cooper, et al. [6] who use laser and video cameras to interactively create wireframe models of objects. Operator input is mouse and keyboard. Graphical overlays are possible. Stereo views and positioning of the camera are not addressed. Bon et al. [3] use stereo cameras to produce a stereo display on two flatscreen monitors. Wireframe models are interactively created. A joystick is used to mark points on the object in both views. The joystick is also used to control the view. Oxenberg et al. [14] use multiple cameras and a wireframe overlay to match remote video to a known model by interactively moving the graphical objects. Milgram et al. [11] use shutter glasses to display stereo views of video and graphics. The purpose is to study human fusion of video and graphics. No models are created. This work is unique in that it provides a stereo view of both the video and the graphics, with motion slaved to the motion of the user's head, thus immersing the user in the video/graphical model. It also provides interaction in the form of voice commands and voice

recognition, and utilizes texture mapping and shaded graphics to produce a more realistic model.



Figure 1: BOOM/MOLLY System

3.1 SYSTEM COMPONENTS

Figure 1 shows the BOOM stereo graphics viewer and the MOLLY camera platform. Figure 2 shows a diagram of the configuration of system components for this work.

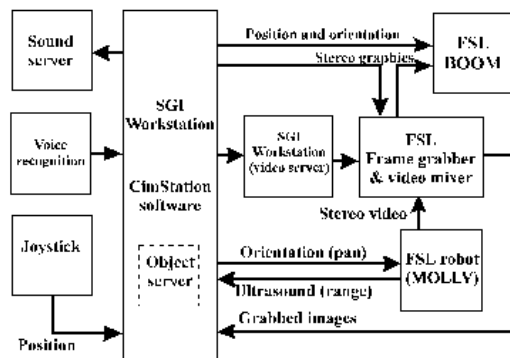


Figure 2: System diagram

The following additional components were also developed for this work.

3.1.1 Computer Vision Software

The computer vision software used was specifically developed for this prototype system. It uses information input by the operator, such as rough object position and object identity, to guide computer vision processing. Edge elements, or edgels, are extracted using an algorithm developed by Canny [4]. The edgel information is used for stereo matching and triangulation to extract geometric and positional information about the object being modeled.

3.1.2 Initialization of Video and Graphics

Before an accurate model can be created from the remote video both the cameras and the graphical display must be initialized. The Molly's two CCD video cameras must be adjusted so that the parameters of both cameras

match as closely as possible. These parameters include focal length, brightness, and contrast. Setting up the cameras in this manner allows the operator to fuse the two images into a 3-D view. In addition, the camera separation cannot be too large. Camera separation greater than ten cm makes it difficult for the operator to fuse the images.

Registration of the stereo cameras with the stereo graphics is also required so that the graphics and the video match. Registration is accomplished by matching the graphical viewing parameters to the parameters derived by calibrating the two CCD cameras. A least squared error technique is used to create a 4x3 camera matrix [1]. Once a matrix for each camera is calculated, the camera parameters can be extracted [8]. The camera matrix also provides a mapping from world points to image points. Using the parameters derived from the camera matrices, the stereo graphics can be registered with stereo video.

3.1.3 Graphical Pointer

A graphical pointer is used to mark objects in the live video. The pointer is controlled by the FlyBox joystick. Calibration, as discussed above, is used to ensure that the graphical and world positions coincide. The pointer is initially positioned using the ultrasonic sensor. The object in the center of the field of view is ranged, and the graphical pointer is placed at the depth of the object. This gives the operator an initial position measurement for the object. The operator then uses the joystick to control the position of the pointer, allowing him to mark points on the object in 3-D space.

3.2 OPERATOR INTERFACE

Verbal commands and/or Flybox buttons control the state of the modeling system. The computer informs the operator of the status of the system via audio feedback. Audio is also used to prompt the operator for input. The system operates in two modes: verification and modeling. In the verification mode, MOLLY and the graphics are slaved to the BOOM, with the graphics overlaid onto the live video from the cameras mounted on MOLLY. As the operator examines the remote environment, visual verification of the graphical models is possible. The graphical models may be displayed in wireframe or solid model form.

In modeling mode, the operator freezes the graphics and camera motion using a voice command. By using the joystick to control the graphical pointer, the operator can mark points on an object. After marking the object the operator must identify the object (e.g. box or cylinder) by issuing the appropriate verbal command. The computer vision system uses this information to extract positional and geometric information about the object. This information is then used to create a model of the object, which is placed in the graphical model of the environment.

3.3 COMPUTER VISION PROCESSING

The computer vision system is composed of two servers: the image server and the object server. The image server is responsible for grabbing the video images and performing edge detection. The object server then takes the images from the image server and extracts the dimension and pose (position and orientation) of the objects. The image server is informed when modeling mode is entered. It then grabs a frame of video from each of the video cameras and performs edge detection on those images. This provides a 3x3 direction derivative for each pixel. The edgel value and direction value are stored. A 3x3 direction derivative may be used because the high-level image processing (location and identification) is being performed by the operator.

In the modeling mode, the user identifies the object as a cylinder or a box and marks four points which bound the object. This information is passed to the object server, along with the current orientation of the camera platform. The object server translates the four world points into eight image points (four for each image). These image points bound the edgels of interest. If the object is identified as a cylinder, an edge is located that more precisely bounds the object. This edge is located by searching within the bounded area of the image for an edge that is closed. To find such an edge, an edge following routine was written as outlined by Schalkoff [15]. The direction associated with each edgel of interest is stored in an array. This array is then processed to find possible corners. A possible corner is detected whenever there is a direction change of more than twenty degrees with respect to the edgel next to it. The four corners are determined by comparing the distance of all possible corners with each image point provided by the user. The four corners in both images are then triangulated to give four world points. An assumption is made that the radius of the cylinder is smaller than its height. This assumption is not necessary if a constraint is placed on the user regarding the order in which the object is marked. With the four points and the above assumption, the dimensions and pose of the cylinder may be extracted. This method does not work well if the user is looking at the top or bottom of the cylinder. Currently, we assume that the user is able to correctly place the camera to model the object, and so this is not a problem.

If the object is identified as a box, all bounded edgels in the left image are extracted. The world points provided are then used to approximate the depth and the width of the object. For each edgel extracted, an epipolar line of pixels is calculated, in the right image, which would fall within that volume. Stereo matching in the form of auto correlation [15] (10x10) is then performed on the edgels and the line of epipolar pixels. The best match is then compared with a threshold to determine if the match is valid. This provides an array of world points that are associated with that object. These points are then bounded and model dimension and position are sent back to the graphics program. The bound is determined by the maximum and minimum in x, y, and z. The matched pixels in the left image also provide a bound on the image for texture mapping. By sorting the array for maximum and minimum image space (u, v) values, a texture may be created and added to the face of the object model. This texture is obtained from the digitized image of the object. Complex objects within the remote environment can thus be stored within the graphical model as bounding boxes with a digitized image of the actual object texture-mapped onto them. This provides added information and realism to the graphical environment.

3.4 PRELIMINARY RESULTS AND DISCUSSION

The prototype system described here demonstrates that interactive graphical modeling of remote sites is viable. By interacting with both the live video of the site and the incomplete graphical model, operators can extend the graphical model to contain objects which were not previously known to exist. By using virtual reality and telepresence, an intuitive, easy-to-use system may be developed which requires little operator training. The prototype system was tested on several naive operators over the course of its development, and most of them were able to interactively create graphical models with only minor explanation and guidance. The vision system is able to successfully extract and model cylinders and boxes. The box model may be used as a bounding box for more complex objects. A description of the experimental set-up used follows:

An optical table, 52 inches by 43 inches, is used for calibration and for exact placement of objects. The cameras are approximately 25 inches from the table, or 25 to 75 inches from the objects in depth. The objects were 3 inches to 12 inches in height. Positions and dimensions of objects were measured and then compared against the values extracted by the vision system. In these preliminary experiments, objects were modeled with an accuracy within 0.25 inches of their actual dimensions. Depth placement gave the largest error due to the small camera separation, but was within 1.5 inches of actual depth. This accuracy is more than adequate for initial positioning of robot manipulators and tools.

The current system is limited to camera motion about the z axis (pan) only. This is due to positioning inaccuracies in the MOLLY camera platform, which is belt-driven. A more accurate motion platform for the cameras would alleviate this problem. The initial system configuration also contained a magnetic tracker, mounted on the operator's wrist, rather than a joystick, to control the graphical pointer. This provided a more intuitive method for controlling the position of the tracker (via natural movement of the operator's arm.) However, again due to the positioning inaccuracy of the device, the magnetic tracker was replaced by the joystick. Depth placement of the graphical pointer was also difficult for some operators who were not able to fuse the two images into a 3-D view. This problem was alleviated by adding the ultrasonic sensor which provides an initial depth (range) measurement for the object and automatically places the pointer in a reasonable starting position.

4. An Operator Interface for Controlling Remote Robots

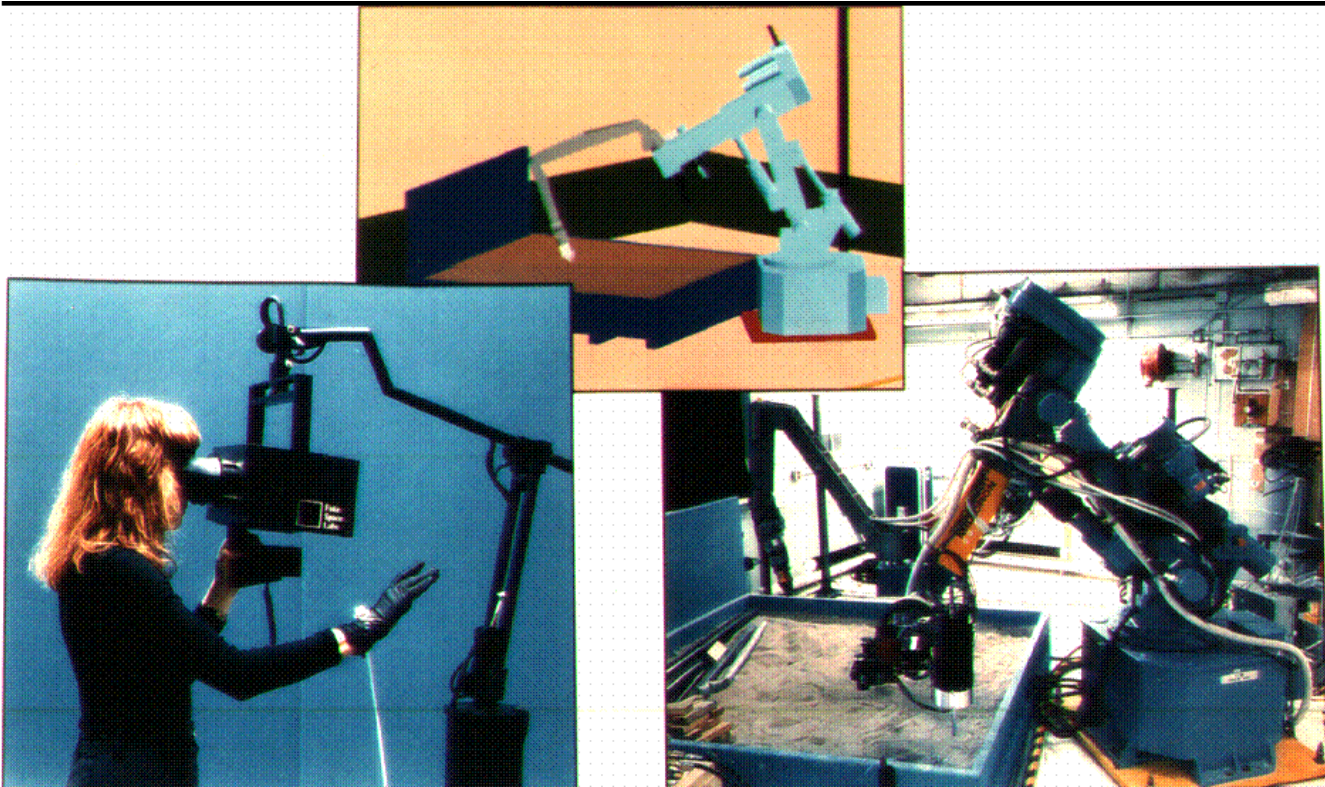
Once a graphical model of the remote site has been created, it may be used to program and control the robots during cleanup operations. This section presents a prototype system which utilizes VR techniques to interface the operator to the remote site so that he/she may operate the robot system from a safe distance. The ACML/UST testbed is used as the application environment. The interface immerses the operator in a 3-D simulation of the robotic system and allows him/her to move about in this environment, to issue verbal commands and requests, and to get feedback from the system via spoken messages. Robots are operated using task-level, rather than primitive, commands. This natural interaction paradigm and task-level orientation has the potential to make operator training and robot control faster, safer and less expensive. Operators may train using the interface without the real robots in the loop and then control these robots via the same graphical interface. Robot commands may be generated, previewed and executed rapidly without the need to learn the control languages for all of the robots and tools within the system.

Related work by others includes that of Takahashi et al. [18], who proposed a virtual reality interface for robotic assembly-task teaching. Tasks were executed by an operator wearing a VPL DataGlove (TM), and hand gestures were recognized and translated into robot commands. Extensive telerobotics research using VR techniques has been done by NASA-Ames Research and JPL for control of remotely deployed robots [17], [2]. The VR simulation system described here differs in that it takes a task-level, supervisory approach to the problem of robot interaction and control. Penn State is developing methods for interweaving virtual reality tools with live video

scenes to direct robots [19]. While similar to this work, operation of the real robots is more directly under the lowlevel control of the operator who is using the video as a guide.

4.1 SYSTEM COMPONENTS

Figure 3 shows the BOOM graphics viewer:



the CimStation graphical model of the ACML/UST testbed, and the actual ACML/UST environment. Figure 4 shows a diagram of the configuration of system components.

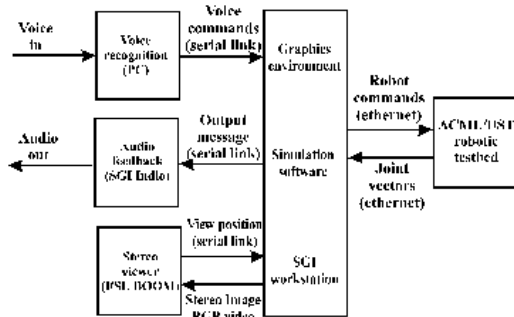


Figure 4: System diagram

The following interaction techniques were designed and implemented for this system:

4.1.2 Voice Command Interaction

The voice command set consists of high-level, speaker-independent commands forming an application specific vocabulary. The vocabulary has been designed using the DragonWriter voice recognition system running on a PC platform. As words and phrases are recognized, ASCII characters are sent from the PC to the SGI workstation through a serial link. Voice commands are processed on the SGI once per second so that voice command response is timely yet real-time graphics updates are not too severely impacted. When command sequences are obtained,

the main application loop is interrupted and the voice command is verified. Once the commands are confirmed, appropriate sets of robot instructions are generated.

For the ACML/UST application there are a total of 32 words and phrases comprising the user input command set. The voice system was trained on five female and five male voices to obtain a speaker-independent voice template; thus, new users of the application need not retrain the vocabulary. Preliminary results indicate the speaker independent voice template consistently recognizes ~85% of the voice commands, including operators with noticeable foreign accents. ASCII output is suppressed for unrecognized voice input. To minimize misinterpreted commands, the vocabulary has a hierarchical structure so that sequences of words are required in order to initiate command actions (for example "open" "gripper".) Using such phrases also helps to distinguish similar commands.

The vocabulary consists of task-level voice commands which make the system easy to learn and use. Using the task-level approach permits the VR application to remain independent of the particular robot performing the task. Commands are generic up until the point where specific robot control sequences must be communicated to the robot system for execution. This allows various robot systems to be used to carry out the tasks simply by changing the low-level robot instruction generation routines. The interface remains unchanged.

An example of a task-level voice command for the ACML/UST system is "get cutter". When the spoken sentence is recognized, it is sent to the simulation software. If the command is valid in the current system state, a sequence of robot actions occur. The operator receives audio confirmation that the command was received and can observe the command execution in the virtual environment. The graphical representation of the Titan II robot arm is commanded to move the gripper tool to the cutter tool location (stored on the tool bar at a known location), the gripper tool grasps the cutter and the arm is moved safely back to the 'home' position. The paths for robot movement are generated using CimStation's internal trajectory planner. In order to execute this command on the actual robot system, the "get cutter" command would be downloaded to the robot control system and, at this lowest level, be translated into the set of Titan II robot and tool commands. Sensors on the physical robot system would be used for final approach and grasping operations so that inconsistencies between the simulated and physical robot paths would not be critical. The operator need not learn details of the robot kinematics nor the specific steps required to get the cutter tool; they simply need to learn the task-level voice command to initiate the operation. Other examples of task-level commands are: "store cutter", "open cutter", "close gripper", and "cut pipe".

4.1.2 Audio Feedback

During operation, the operator receives continuous audio feedback from the VR system. This serves several purposes. First, the operator obtains confirmation of voice commands. For example, with the "get cutter" command, the audio system responds with the phrase "getting cutter" as the graphical simulation executes the command. In this way, the operator receives auditory, as well as visual command confirmation. The operator may also ask for "help" or system "status" at any time. The voice command "status" gives the status of the workcell and results in an audio sequence such as: "The current status is: gripper closed, cutter open, cutter tool inactive". When the operator requests "help", the current valid commands are provided, for example: "The current valid commands are: get cutter, open gripper, close cutter". Finally, the operator can be guided through use of the system with audio feedback with varying levels of terseness. The audio feedback capability is provided by an Indigo 2 workstation via a socket server. Message pointers are sent to the audio server to initiate appropriate audio feedback messages depending on the users actions and commands.

4.2 SYSTEM INTELLIGENCE COMPONENT

The simulation is controlled via a state machine which maintains the state of the system by tracking valid state transitions as operator commands are received. This method has the advantage of interpreting task-level commands in parallel with command verification. Illegal robot actions are easily detected and prevented. The state machine is designed to run in parallel with the immersive graphics, voice recognition, and robot simulation. A 2-D array defines the state machine operation and is structured so that the current state number equals the row index and the voice command input equals the column index. Figure 5 shows the structure for each state element.

State Definition Record: state_command: command pointer next_state: integer, -1 if invalid function_to_execute: function pointer invalid_message_notice: message pointer invalid_message_trained: message pointer

State Definition Record	
state_command:	command pointer
next_state:	integer, -1 if invalid
function_to_execute:	function pointer
invalid_messageovice:	message pointer
invalid_message_trained:	message pointer

Figure 5: State definition record

When an input command is received, the validity of the command is verified by checking that the next_state value is not -1. If it is -1, an invalid_message is output and the current state is not changed. If the command is valid, the corresponding function is executed and the current state number is updated to the next_state value. Figure 6 contains a simplified state machine diagram which shows the state transition commands.

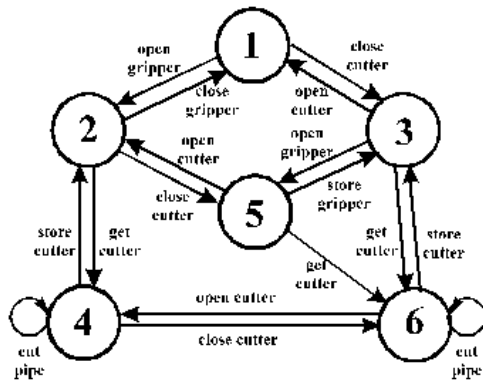


Figure 6: Simplified state machine diagram

Table 1 contains the corresponding state definitions.

State Number	Gripper State	Cutter State	Tool in Use
1	closed	open	none
2	open	open	none
3	closed	closed	none
4	in use	open	cutter
5	open	closed	none
6	in use	closed	cutter
7	in use	open	cutter
8	in use	closed	cutter

Table 1: State definitions.

The state of the VR system is continuously tracked as operator commands are received. The state machine configuration provides immediate verification of commands through direct access of the state record as described above. Operator "help" is quickly obtained by searching through a row of the 2-D array where the row corresponds to the current state of the system. State configuration objects maintain the definition of each state. Thus, system "status" is obtained immediately by using the current state number as an index into the structure containing the state configuration objects. Figure 7 shows the structure of the state objects.

State Configuration Information
state_number: integer
gripper_open: boolean
cutter_open: boolean
gripper_in_use: boolean
cutter_active: boolean

Figure 7: State object structure

4.3 INTERFACING TO REAL ROBOTS

Control of the real ACML/UST robots from the VR environment is still under development although the communication, sensor and control subsystems are already in place. Standard communication tools and protocols developed at Sandia will be utilized to link the VR system with the actual ACML/UST testbed. The robot subsystems of the UST/ACML testbed have been designed to respond to generically-defined commands based on the Robot Independent Programming Environment and Language (RIPE/RIPL) for autonomous systems [12]. A Sandia-developed standardized communication protocol known as GENISAS (GENERALized Interface for Supervisor and Subsystems) [9] will be used to communicate and execute the RIPE/RIPL command set on the actual ACML/UST robot system from the VR environment. The Intelligent System Operating Environment (ISOE) protocol [9] will be used for communication of robot joint vectors, sensor information and tool status from the ACML/UST workcell back to the VR system for user monitoring during command execution. Sensors on the robotic systems, in combination with low-level robot control routines, will carry out the fine robot motions. This sensor-based control reduces the accuracy requirements on the graphical model in the VR environment since inaccuracies in graphical models cannot be avoided. With the VR interface, VR to robot communications, and robot sensors in place, the operator will be able to carry out complete robot operations from programming, to previewing and execution from the VR environment.

4.4 DISCUSSION

The VR system described here provides an intuitive, interactive simulation interface which can be used for control and training on complex robotic systems. The tasklevel nature of the voice activated commands enables users to be quickly trained and allows the VR system to remain independent of the particular robot platform at the highest level. The state machine approach provides a robust control structure for quickly interpreting operator voice commands and responding with appropriate audio feedback. The VR interface allows operators to attain a comfort level with the robotic environment by immersing them in the graphical robot workcell and allowing them to move through and interact with the environment as they might with the real world. This VR system has been demonstrated to a large number of participants throughout its development. Most of the participants grasp the operation of the system quickly and their feedback has served to further improve the system.

There are several areas where the VR system discussed here might be expanded to increase the usefulness and reality of the virtual experience. A position tracker and corresponding graphical pointer will provide the operator with the ability to dynamically select actions and objects in the virtual world. Multi-media information, such as manuals, video sequences, schematics and still photos, accessed from the simulation will improve the training aspect of the system [16]. The monitoring of robot operations will be enhanced by including sensor data in the Robot-to-VR communication such as was done in the collision avoidance/teleoperation work by Novak, et al. [13]. Finally, it is especially important to note that there will be discrepancies between the actual environment, the modeled environment, and the sensor readings. The use of supervisory control by human operators, paired with sensor driven autonomous control of lower lever operations (such as fine positioning) is addressed in [5], [10]. The work described here addresses the issues of intuitive interfaces for robot operators with specific application to two problems: that of creating an initial "rough" geometric model and that of communicating with and controlling the robot system at the supervisory level.

ACKNOWLEDGMENTS

The authors would like to thank several individuals who have helped to make this project a success: Larry Ray, Colin Selleck, and Chris Wilson for use of, and help with, the camera calibration software used for this work. Dave Miller, Jim Tauscher and Brady Davies for their knowledge of and interface support for the ACML robotic facility. And finally, the many demo participants who have contributed valuable insights and enthusiasm. Portions of this work have previously appeared in the Proceedings of the IEEE International Conference on Robotics and Automation, 1994. This work was performed at Sandia National Laboratories and was supported by the U.S.

REFERENCES

- [1] Ballard, D. H. and Brown, C. M. Computer Vision, Prentice-Hall, Englewood Cliffs, New Jersey, 1982. [{{Return}}](#)
- [2] Bejczy, A. K., "Sensors, Controls, and Man-Machine Interface for Advanced Teleoperation", Science, Vol. 208, No. 4450, pp. 1327-1335, 1980. [{{Return}}](#)
- [3] Bon, B., Wilcox, B., Litwin, T. and Gennery, D., "Operator-Coached Machine Vision for Space Telerobotics," SPIE Conference on Cooperative Intelligent Robotics in Space, Vol. 1387, pp. 337-342, 1990. [{{Return}}](#)
- [4] Canny, J. F., "A Computational Approach to Edge Detection, " IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8 no.6, pp 679-698, 1986. [{{Return}}](#)
- [5] Christensen, B. K., Drotning, W. D., and Thunborg, S. "Model Based, Sensor Directed Remediation of Underground Storage Tanks", Journal of Robotic Systems, pp. 145-159, 1992. [{{Return}}](#)
- [5] Christensen, B. K., Drotning, W. D., and Thunborg, S. "Model Based, Sensor Directed Remediation of Underground Storage Tanks", Journal of Robotic Systems, pp. 145-159, 1992. [{{Return}}](#)
- [5] Christensen, B. K., Drotning, W. D., and Thunborg, S. "Model Based, Sensor Directed Remediation of Underground Storage Tanks", Journal of Robotic Systems, pp. 145-159, 1992. [{{Return}}](#)
- [6] Cooper, E., Vazquez S. and Goode P., "Interactive Scene Analysis Module: A Sensor - Database Fusion System for Telerobotic Environments," SPIE Conference on Cooperative Intelligent Robotics in Space, Vol. 1621, pp. 334345, 1991. [{{Return}}](#)
- [7] Davies, B. R., "Remediating Hazardous Waste Robotically Using a High-Level Control System and Real-Time Sensors", Proc of the SPIE International Symposium on Optical Tools for Manufacturing and Advanced Automation, September 1993. [{{Return}}](#)
- [8] Ganapathy, S. "Decomposition of Transformation Matrices for Robot Vision" IEEE Robotics, pp.130139, 1984. [{{Return}}](#)
- [9] Griesmeyer, J. M., et al., "Generic Intelligent System Control (GISC)", Sandia Internal Report, SAND92-2159, October 1992. [{{Return}}](#)
- [10] McDonald, M. J. and Palmquist, R. D., "Graphical Programming: On- Line Robot Simulation for Telerobotic Control", Sandia Internal Report, SAND93-0003C, 1993. [{{Return}}](#)
- [10] McDonald, M. J. and Palmquist, R. D., "Graphical Programming: On- Line Robot Simulation for Telerobotic Control", Sandia Internal Report, SAND93-0003C, 1993. [{{Return}}](#)
- [10] McDonald, M. J. and Palmquist, R. D., "Graphical Programming: On- Line Robot Simulation for Telerobotic Control", Sandia Internal Report, SAND93-0003C, 1993. [{{Return}}](#)
- [11] Milgram, P., Drascic, D. and Grodski, J., "A Virtual Stereographic Pointer for a Real Three Dimensional Video World," Proceedings of the INTERACT Conference, pp. 695700, 1990. [{{Return}}](#)
- [12] Miller, D. J. and Lennox, R. C., "An Object-Oriented Environment for Robot System Architectures", Control Systems, IEEE, Vol. 11, No. 2, February 1991. [{{Return}}](#)
- [13] Novak, J., et al., "Collision Avoidance During Teleoperation Using Whole Arm Proximity (WHAP) Sensors Coupled to a Virtual Environment", Proc of SPIE International Symposium on Optical Tools for Manufacturing and Advanced Automation, 1993. [{{Return}}](#)
- [14] Oxenberg, S., Landell, B. and Kan, E., "Geometric database maintenance using CCTV cameras and overlay graphics," SPIE Conference on Cooperative Intelligent Robotics in Space, Vol. 1006, pp. 115-123, 1988. [{{Return}}](#)
- [15] Schalkoff, R. J. Digital Image Processing and Computer Vision, Wiley, New York, 1989. [{{Return}}](#)
- [16] Stansfield, S. A., "A Computer-Based Training System Combining Virtual Reality and Multimedia", Proc of

Intelligent Computer Aided Training and Virtual Environment Tech Conference, 1993. [{{Return}}](#)

[17] Stark, L., et al., "Telerobotics: Display, control, and Communication Problems", IEEE Journal of Robotics and Automation, Vol. RA-3, No. 1, pp. 67-75, February 1987. [{{Return}}](#)

[18] Takahashi, T. and Ogata H., "Robotic Assembly Operation based on Task-Level Teaching in Virtual Reality", Proc of IEEE International Conference Robotics and Automation, pp.1083-1088, 1992. [{{Return}}](#)

[19] Wang, C., and Cannon, D. J., "A Virtual End-Effector Pointing System in Point-and-Direct Robotics for Inspection of Surface Flaws Using a Neural Network Based Skeleton Transform", Proc of IEEE International Conference Robotics and Automation, pp. 784-795, 1993. [{{Return}}](#)

CONTACT INFORMATION

Sharon Stansfield
Sandia National Laboratory
PO Box 5800, MS 0978
Albuquerque, NM 87185-0978
Phone: 505-844-1396
FAX: 505-844-2057
Email: sastans@sandia.gov

[{{return}}](#)

