Invited Review: The Synergy Between Virtual Reality and Robotics

Grigore C. Burdea, *Senior Member, IEEE*

(*Invited Paper*)

***Abstract*—To a large extent the robotics and the newer virtual reality (VR) research communities have been working in isolation. This article reviews three areas where integration of the two technologies can be beneficial. First we consider VR-enhanced CAD design, robot programming, and plant layout simulation. Subsequently we discuss how VR is being used in supervisory teleoperation, for single operator-single robot systems, single operator multiplexed to several slave robots, and collaborative control of a single robot by multiple operators. Here VR can help overcome problems related to poor visual feedback as well as system instabilities due to time delays. Finally, we show how Robotics can be beneficial to VR in general, since robots can serve as force feedback interfaces to the simulation. Newer back-drivable manipulators offer increased safety for the user that closely interacts with the robot. The *synergy* between the fields of Robotics and Virtual Reality is expected to grow in years to come.**

***Index Terms*—Haptic feedback, off-line programming, teleoperation, virtual manufacturing, virtual reality.**

I. INTRODUCTION

THE reader is familiar with the reliance of Robotics on allied technologies such as multimodal sensing or neural networks which have become part of modern robotic systems. The need to incorporate such technologies stems from the ever increasing expectations on robot performance in both industrial and service applications.

Virtual reality (VR) is a high-end human-computer interface allowing user interaction with simulated environments in real time and through multiple sensorial channels [1]. Such sensorial communication with the simulation is done through vision, sound, touch, even smell and taste. Due to this increased interaction (compared to standard CAD or even multimedia applications), the user feels "immersed" in the simulation, and can perform tasks that are otherwise impossible in the real world.

Fig. 1 illustrates a typical single-user VR system [1]. Its main component is the "VR engine" which is a multiprocessor graphics workstation. Interactions between the user and the VR engine are mediated by input/output (I/O) devices which read user's input and feedback simulation results. Joysticks or three-dimensional (3-D) trackers [2] are used in simple tasks, while sensing gloves such as the "CyberGlove" [3] are used in more dextrous interactions. Feedback from the VR engine is typically through stereo head-mounted displays (HMD's) [4], large-volume displays such as the CAVE [5], spatially-registered sound (also called 3-D sound) [6], or force/touch feedback [7].

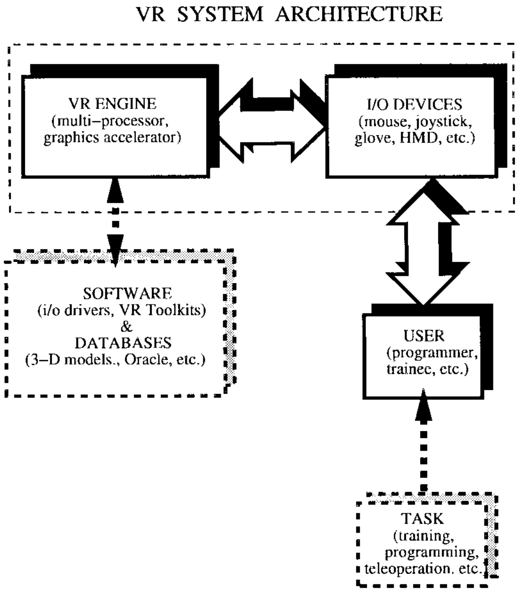


Fig. 1. VR System block diagram (adapted from [1]). © Editions Hermes. Reprinted by permission.

The VR engine responds to user's commands by changing the view to, or composition of, the virtual world. This synthetic world is modeled off-line using dedicated software libraries [8] and databases [9]. Subsequently the models have to be rendered in real time (at 30 frames/s) which limits the geometrical complexity of present virtual worlds. Else the scene takes too long to be displayed and the feeling of immersion is lost.

Another computation load for the VR engine is related to object dynamics, collision detection and contact force computation for interaction between the virtual objects. These computations compound such that it may be too much to be handled by a single computer. In that case several computers are networked, each having in charge some aspect of the simulation [10]. Thus the VR system becomes a single-user distributed one, over a communication media such as the Internet, or Internet2. Alternately, several remote users can

---

Manuscript received October 13, 1997; revised January 1, 1999. This paper was recommended for publication by Associate Editor K. Tanie and Editor V. Lumelsky upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation, Grant BES-9708020, a grant from the Center for Computer Aids for Industrial Productivity (CAIP), Rutgers University, and a grant from the Whitaker Foundation.

The author is with the Electrical and Computer Engineering Department, Rutgers University, Piscataway, NJ 08854 USA.

Publisher Item Identifier S 1042-296X(99)03548-X.

share the same virtual world, in a multiuser distributed VR environment [11].

The benefits of VR technology have been recognized by scientists and engineers, with applications ranging from architectural modeling, manufacturing plant layout, to training in servicing equipment, etc.. Within the scope of this review article we are interested in the synergy between VR and robotics and manufacturing. This is a vast area of research as evidenced by the subsequent articles in this Special Issue of the R&A Transactions. Space limitations however require that the coverage in this article be limited to three areas of interest. The first is the use of VR in manufacturing, including CAD design, robot programming and plant simulation. This is the topic of Section II. Section III describes the use of VR in teleoperation (with poor visual or delayed feedback), as well as in supervisory and collaborative control of remote manipulators. Conversely, Section IV shows how robots can be beneficial to Virtual Reality in general, by serving as haptic feedback devices which enhance simulation realism. Concluding remarks are given in Section V.

II. VR IN MANUFACTURING

The ever increasing demands for short production cycles and high product quality require increased process flexibility. This results in a substantial design effort for parts, equipment (including new robot types), as well as plant layout optimization. Furthermore the programming effort for programmable numerical controllers (PNC's) and robotic cells becomes important. The time required by all these tasks can be shortened through the use of VR simulations, as discussed below.

A. CAD Design

The CAD design process is a continuum, starting with the initial "concept" stage, followed by detailed design and ending with the prototype [12]. Concept design focuses on overall functionality without regard to exact dimensions, and is typically done today with pencil and paper. What is needed is a human-computer interface which allows natural gesture and voice interaction, resulting in easier modification of the concept shape. Chu and his colleagues at University of Wisconsin-Madison developed a multimodal VR interface for the generation, modification and review of part and assembly design [13]. Input to the system is through hand gestures (measured by sensing gloves), voice commands and eye tracking. Output from the simulation is through visual feedback (graphics), auditory feedback (voice synthesis) and tactile feedback (allowing the user to feel the parts he is designing). Subjective human factors studies were conducted to evaluate the usefulness of these interaction modalities in various combinations. Results showed voice and gesture input to be superior to eye tracking, while visual output was the most important output modality for shape design. The researchers note the lack of reliable force feedback technology that may be used in CAD design.

Force feedback plays an important role in another study involving part assembly simulation in VR done by Gupta at Schlumberger in collaboration with Sheridan and Whitney at the Massachusetts Institute of Technology, Cambridge [14]. A pair of PHANToM force feedback interfaces [15] allow the designer to grasp the part being designed with the thumb and index and feel resistance due to contact forces. The multimodal simulation incorporates speech and finger position inputs with visual, auditory and force feedback. Experiments comparing handling and insertion assembly times in real and virtual worlds showed that force feedback was beneficial in terms of task efficiency. However, results were hampered by the spatial discrepancy between visual and force feedback (hands were out of sight), as well as differences between physical parameters of real and virtual parts (such as hardness).

Virtual reality may also be beneficial in the prototyping stage of the CAD process. Many companies such as Northrop or Rockwell International are working to replace physical prototypes with virtual ones with the aim of shortening the rapid prototyping process [12]. Boeing is using virtual aircraft prototypes in order to analyze accessibility for maintenance. Wilson and colleagues at University of Nottingham [16] review the use of rapid prototyping as well as training using lower-cost PC-based VR systems. A survey of 350 potential industrial user companies showed a clear preference for desktop VR over more immersive HMD-based systems. The researchers cite several projects using virtual prototyping ranging from design of a thermostat housing for a compact car to submarine engine servicing in confined spaces.

B. VR Aid to Robot Programming

Another area of robotics where VR can be beneficial is in programming of manipulator tasks. Industrial robot programming takes place in either of three forms, using teach pendants, off-line and at task level. While a substantial research effort is directed toward task-level programming, most industrial robots are still programmed using a simple pendant. This approach has the advantage of simplicity since it does not require programming skills on the part of the factory technician. However, teach pendants are ill suited for tasks involving complex manipulator trajectories, or when there is increased reliance on outside sensing.

Yanagihara and his colleagues at NTT (Japan) developed a "multimodal teaching advisor" (MTA) for use in seam welding of complicated car chassis [17]. As shown in Fig. 2, the system consists of a seven DOF manipulator (Mitsubishi PA10), a laser range finder (Fanet FLP-400), a human operator with the teach pendant, a see-through HMD (Virtual I/O i-glasses) and a video tracking system with two CCD cameras. The MTA is hosted by a PC which receives voice commands from the operator and is interfaced with the other controllers through ethernet. The MTA is first given task specifications, such as tolerances and kinematic constraints. It then calculates the difference between the modeled path and that taught by the operator, and provides remedial feedback to the operator in graphical and audio modalities. Tests showed that the use of the MTA produced better taught trajectories even for the novice operators, while the duration of the teaching time was reduced.

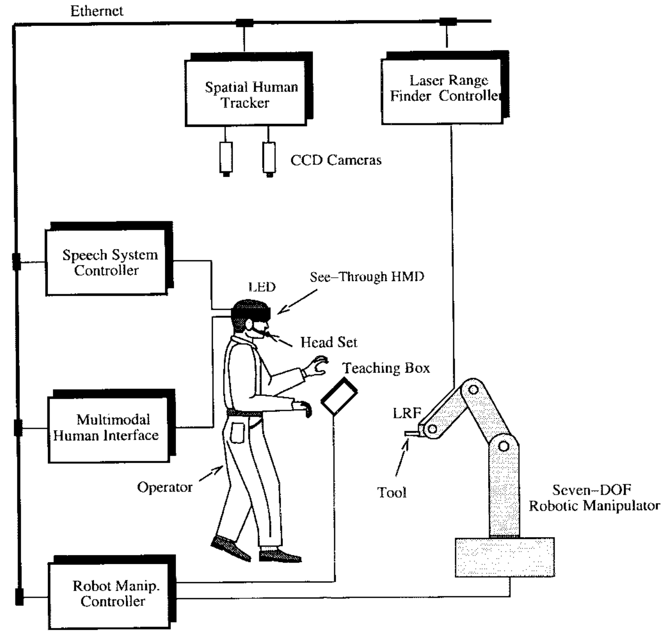


Fig. 2. Robot teaching using a Multimodal Teaching Advisor (adapted from [17]). © 1996 IEEE. Reprinted by permission.

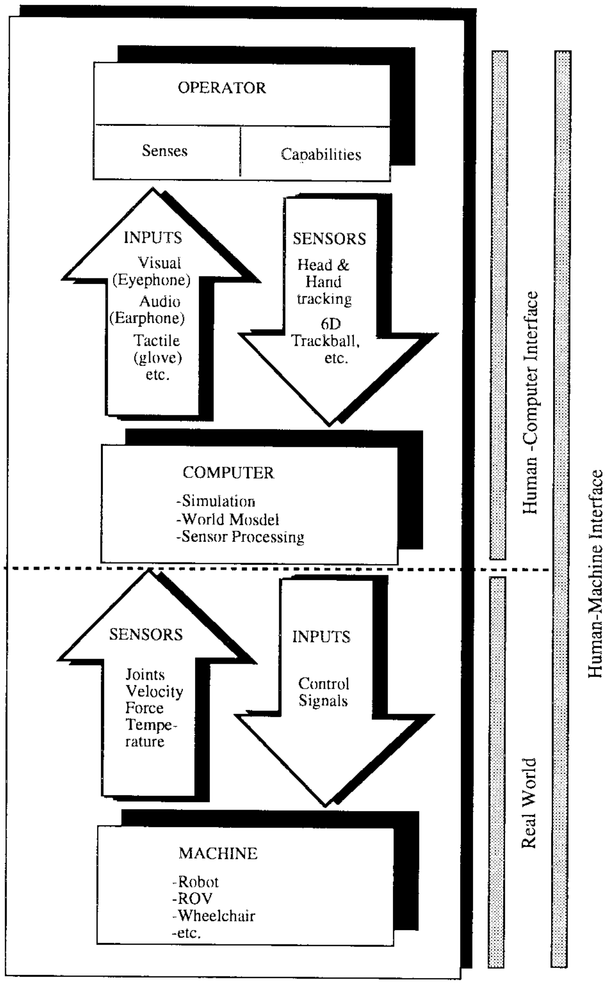


Fig. 3. Block diagram of the IPA robotics VR workstation (adapted from [20]). © Editions Hermes. Reprinted by permission.

Off-line programming is more suited for sensor-intensive tasks, but robotic languages are dependent on the particular manipulator used, and the debugging stage is quite time consuming [18], [19]. One project attempting to advance the state-of-the-art in robot programming has been ongoing at Fraunhofer IPA in Stuttgart, Germany [20]. As illustrated in Fig. 3, programming is done on a virtual robot and a virtual environment, with the programmer interaction being mediated by VR I/O devices (sensing glove, tracker, HMD, etc.). Thus the programmer feels immersed in the application where he can navigate and look at the scene from any angle, and see details that may not be visible in real life. Specifying a trajectory is as simple as a hand gesture. The code is automatically stored using a special-purpose toolkit called "VR4" [21]. This toolkit allows the generation of graphics at high frame rate using variable level-of-detail by reducing the number of polygons of more distant objects (tools, conveyors, etc.). Furthermore, collision detection between the robot end effector and other virtual objects is optimized in a hierarchical way. Initially, a fast but less accurate, bounding box checking is performed, with more computationally intensive methods used only when needed. Databases of preexisting models as well as a graphical user interface (GUI) for custom-made configurations complete VR4 [22]. This GUI allows the user to specify the dynamic behavior of components such as their paths, accelerations, velocities, interpolations, etc. Once the program is completed, it is downloaded to a real robot connected to the same VR engine and the same task is executed. Feedback from the sensors on the real robot is then used to fine tune the program.

A factory floor robot does not act in isolation, rather it is part of a robotic cell which in turn is integrated within the overall assembly line process. Thus the next step of the Fraunhofer project was to integrate VR4 within the logical layers of a plant in order to design and simulate large production facilities. The necessary four layers of such as simulation are (in ascending order) sensor-actuator level, PLC level, robotic cell level and, finally, logistic control level [21]. PLC's are in charge of technical systems such as conveyor belts, while logistic control is done by a single common computer programmed in a high-level language, which monitors overall factory operations. VR4 is used to describe the graphical, kinematic and dynamic characteristics of components, while a software tool called "phoenix" describes their logical behavior. As a test of the simulation capabilities the researchers modeled a distribution warehouse of consumer products, as illustrated in Fig. 4. The model had 150 technical components, with 250 sensors and 350 information links. While the graphic model had as many as 700 000 textured polygons, the simulation cycle did not drop below 20 frames/s on an SGI Infinity computer. The advantage of using such a simulation tool are reduced planning time for new production facilities, increased safety, analysis of economic viability, etc.

An additional benefit of the above system is the possibility to use simulation-developed PLC code in the real facility (in addition to robotics programs). A similar result involving testing work cell control software for agile manufacturing was

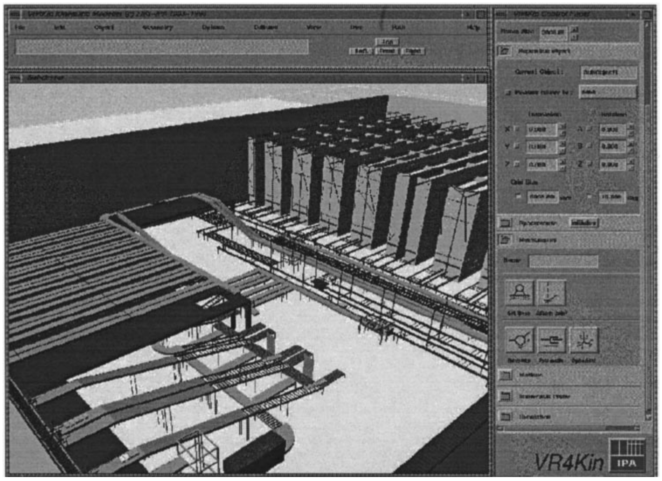


Fig. 4. VR4 graphical user interface for simulation of large production facilities [21]. © Fraunhofer-Institute. Reprinted by permission.

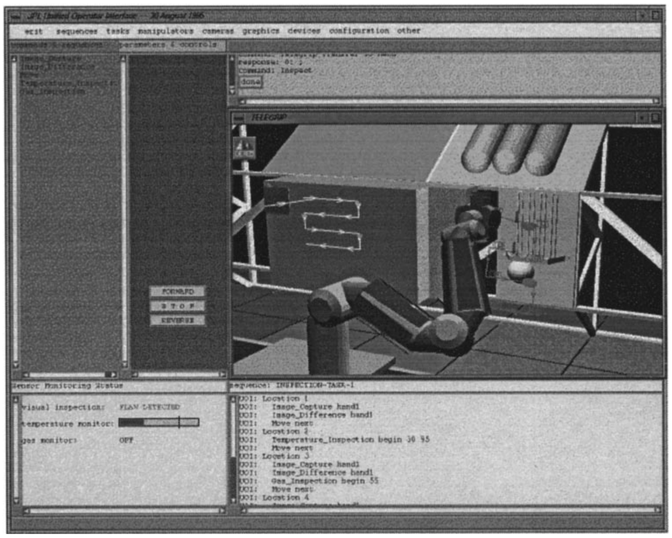
obtained by Jo and his colleagues at Case Western Reserve University [23]. The simulation had the ability to model device failure, which is difficult to recreate on a real work cell, and reduct work cell downtime, as new code was tested off-line. However, the researchers caution on the limitations of virtual testing, and argue that this is a necessary but not sufficient condition for correct work cell operation. Overdependency on the simulation can lead to accidents.

III. VR Use in Teleoperation

The above programming examples are geared toward simulation of industrial facilities where the environment is known *a priori* and well modeled. Such an approach will fail however if the model is inaccurate or if the environment surrounding the robot is changing in an unpredictable fashion (is "unstructured"). In such cases the limited intelligence and adaptability of today's robots require that the operator remain in the loop, if the task is to succeed. Thus the robot slave is controlled at a distance (teleoperated) by the user through a "master" arm. Other situations where teleoperation is necessary relate to adverse environments, such as nuclear plant servicing (or decommissioning), undersea operations, space robotics, explosive environments, etc.. In such cases the robot performs the task for the human operator, and protects him from harm.

A. Supervisory and Cooperative Control

Classical teleoperation requires one operator controlling a single robot end effector. More modern approaches elevate the operator to the role of a supervisor who controls the robot indirectly through some graphical abstraction. Here VR may serve as a predictor of motion commands, before they are actually sent to the remote (real) robot. Blackmon and Stark at the University of California, Berkeley, developed a model-based supervisory control approach using a "task sequence script" list of desired robot sub-goals [24]. Their experimental system consisted of a GUI on the operator's workstation displaying a graphical model of the remote environment. A second workstation at the remote site had a similar model, which was used to control a five-DOF Mitsubishi RM501 robot. The operator could control a graphical model of the remote robot using a pair of two-DOF joysticks, in order to preview trajectories and detect potential collision spots. If unsatisfied with the planned trajectory the operator could edit it interactively by modifying the task sequence script. Human factor tests showed that the model-based supervisory control was superior to manual teleoperation of the same robot by significantly reducing the number of collisions at the remote site.



```
visual inspection:     FLAW DETECTED
temperature monitor:
gas monitor:           OFF
```

```
001: Location 1
001:    Image_Capture hand1
001:    Image_Difference hand1
001:    Move next
001: Location 2
001:    Temperature_Inspection begin 30 55
001:    Move next
001: Location 3
001:    Image_Capture hand1
001:    Image_Difference hand1
001:    Gas_Inspection begin 55
001:    Move next
001: Location 4
```

Fig. 5. Operator Interface with motion guides, task lines and corresponding command sequence [25]. © 1996 IEEE. Reprinted by permission.

In the above example the operator controls the virtual robot directly. Another approach developed by Backes and his colleagues at Jet Propulsion Laboratory involves the teleoperation of the *trajectory* which the remote robot is constrained to follow [25]. The proposed trajectory is specified as a "motion guide," displayed on the operator station, as illustrated in Fig. 5. The operator can move the motion guide in 3-D using a track ball or other 3-D interface, and specify the direction of motion (forward–backward). Alternately the operator can move the robot on a preexisting motion guide, or tell the robot to keep moving, while he modifies (lengthens) the motion guide line. Various operations that need to be performed by the robot are programmed using "task lines." These are icons selected by the operator and placed at the location of the motion guide where the task is to be executed. Various attributes of a specific subtask can be edited in text mode in the corresponding window of the operator interface, but the use of icons for classes of tasks is intuitive and easily understood.

The abstract symbols used in supervisory teleoperation allow the master (or slave) to be displayed as a virtual hand rather than a robot. This intuitive approach allows the same algorithm to be used for the control of different types of slave robots. One operator can then multiplex between several (dissimilar) slave robots, using a single control workstation. Researchers at the Laboratoire de Robotique de Paris, France, were able to teleoperate in this fashion four robots in France and Japan [26], [27] using a LRP Master [28], as illustrated in Fig. 6. The operator wore a sensorized glove measuring

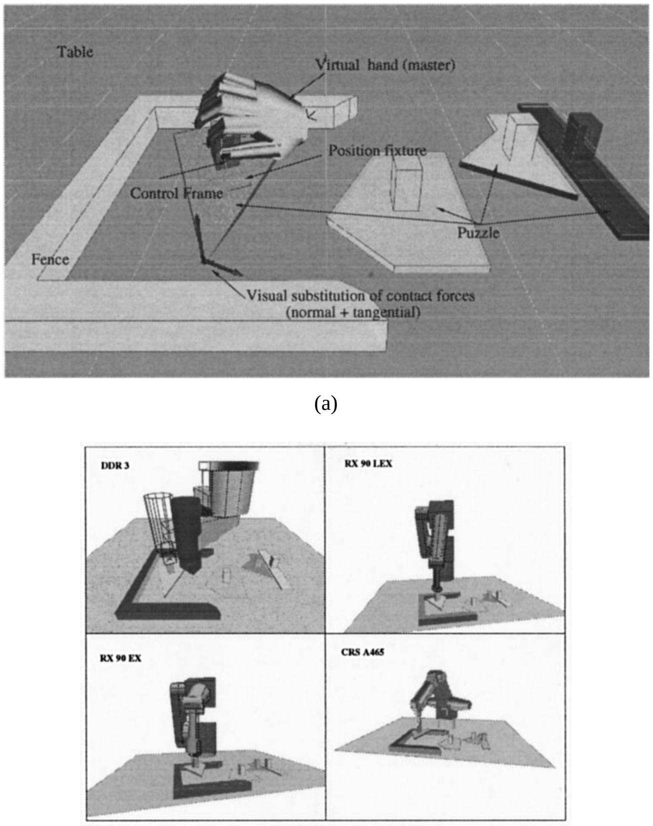


Fig. 6. VR-aided teleoperation of multiple dissimilar robots: (a) functional-equivalent master and (b) task execution by four robots [25]. © IEEE 1997.

hand and finger positions and providing force feedback to each finger. Tasks were done naturally through hand gestures mapped to a graphical model of the remote site, but showing the robot as a hand. A range of commands (some at high level) were then sent to the slave robots which relied on local sensing to perform the remote task.

Supervisory teleoperation using a virtual hand is also used in the collaborative control of a single robot by several operators at different geographic locations. Such as system was developed by Cannon and Thomas at Pennsylvania State University [29] in collaboration with McDonald and his colleagues at Sandia National Laboratory [30]. Each operator in the team had a sensing glove used to control a shared set of "virtual tools." These are icons that are overimposed on live video feedback from the remote robot. When a virtual tool representing the gripper was moved from one location to another in the scene, the corresponding robot trajectory was generated automatically. The motion of a virtual tool controlled by one operator was instantaneously reproduced on all workstations, so that all members of the team (such as experts in other fields) could judge a strategy and eventually reach consensus. Once this happened the final go-ahead was given by a primary supervisor, and the action was executed at the robot site. A pilot experiment was conducted for the teleoperation of a robot used for radioactive site cleanup. Results showed that the equipment utilization rate (ratio of robot motion time to overall task duration) was twice that corresponding to noncollaborative teleoperation. Furthermore, the authors note the potential for better decision making in solving complex problems using a collaborative control strategy than in single-operator systems.

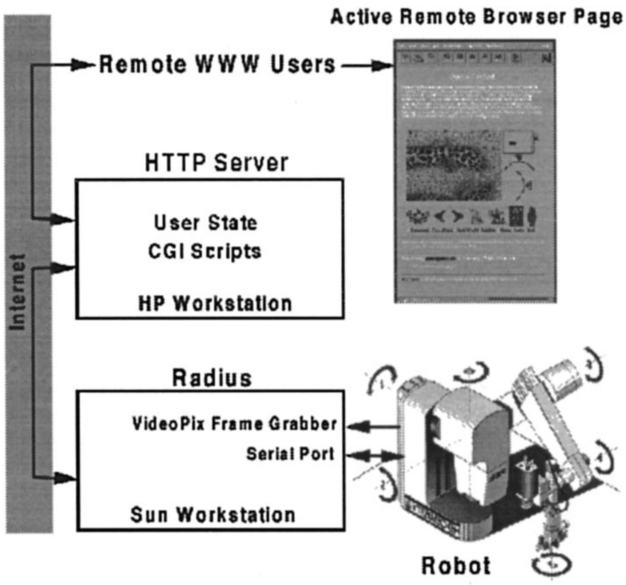


Fig. 7. System architecture for WWW-based multiplexed teleoperation [31]. © 1996 IEEE. Reprinted by permission.

Another system allowing the teleoperation of a single robot by multiple users was developed at the University of California at Berkeley by Paulos and Canny [31]. It uses the world wide web as communication link and serves as telerobotic remote browser of an art museum. The motivation for this research is twofold. First robots can act as physical extensions of users surfing the net allowing them to physically alter rather than just visually "visit" a remote site. Furthermore, in the particular application area envisioned, visiting a real remote museum using a robot-controlled camera makes unnecessary the current digital museum technology which is labor and computer disk intensive. As illustrated in Fig. 7, the CCD camera was mounted on an Intelledex six-DOF robot, and images were digitized by a frame grabber on a Sun workstation. Subsequently the image was converted in JPEG format required by the HTML documents placed on the web. A separate HP workstation served as Common Getaway Interface, namely a front end to the system receiving requests from WWW users and returning documents in HTML format. In order to allow better robot utilization several requests from different users were queued. The user at the top of the queue had exclusive control only during execution of his command to move the robot and grab a new image at a certain location. During the few seconds it took for that image to be processed and placed on the net, the next user took over control of the robot, and so on. Through subsequent requests for images each user could change the view point, zoom in on an area of interest, in other words "visit" the remote museum. While VR was not explicitly present in the above system, VRML-enhanced browsers will supplement WWW-based teleoperation in the near future. Furthermore, the recent

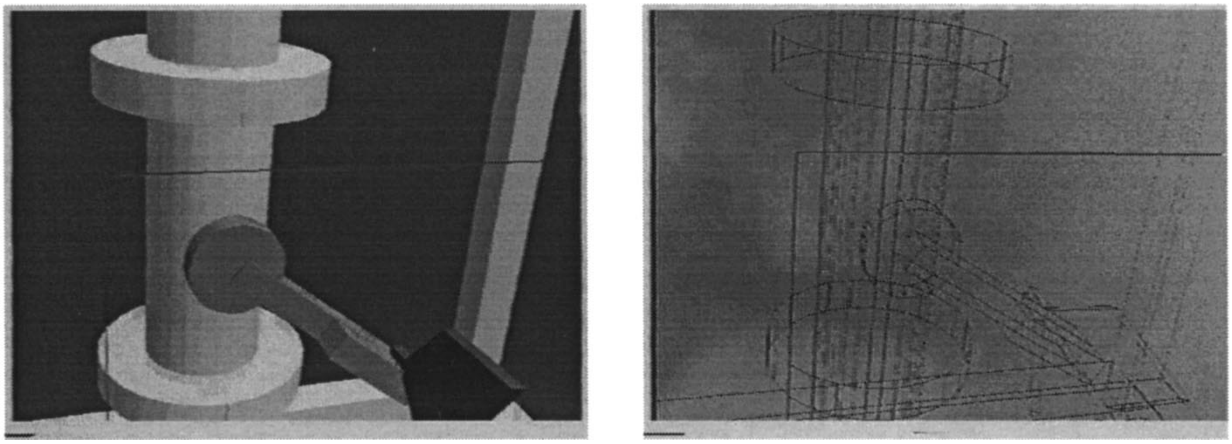

(a) (b)

Fig. 8. Teleoperation in smoky environments: (a) virtual model; (b) real video image [33]. © The MIT Press. Reprinted by permission.

introduction of high-bandwidth Internet2 communication will make possible real-time web-mediated robot teleoperation applications [32].

B. Teleoperation with Poor Visual Feedback

Whether single-operator or collaborative, teleoperation unfortunately suffers from a number of drawbacks. One problem present in all such systems is the degradation of sensorial information about the remote environment due to poor or nonexistent visual feedback (poor visibility, limited field of view, microscopic scale data, etc.). Under such conditions certain tasks may take much longer to be executed remotely, or may be altogether impossible to complete.

Oyama and his colleagues at MITI (Japan) [33] developed a VR-assisted teleoperation system where visual feedback from the remote site is degraded by smoke. The operator controlled a master arm which was kinematically equivalent to a remote six-DOF slave robot. A stereo camera installed on the robot provided visual feedback which was displayed on the master workstation, or on the operator's HMD. The same graphics workstation generated a 3-D virtual model of the slave and its environment. This virtual slave could be teleoperated using the same master arm, while visual feedback from the remote smoky scene was overimposed, as illustrated in Fig. 8. Key to successful teleoperation under these adverse conditions was accurate calibration between the virtual and real remote scenes. The researchers used a manual calibration approach based on least squares by selecting corresponding points in the real and virtual worlds. They reported calibration errors up to 3 cm in the direction parallel to the camera line of sight. Under these conditions the robot could perform low-accuracy tasks such as turning a lever on a pipe in order to stop the flow of smoke. Better calibration methods, such as that proposed by Kim [34] are required for more precise tasks. His method uses up to four cameras to obtain data on the remote site and extends the linear least-squares calibration method with a more precise nonlinear step. Kim's approach reduced calibration errors in a space repair task to only 1 cm.

Another instance of teleoperation with poor visual feedback is underwater navigation for mobile robots servicing offshore platforms. Even experienced operators can get disoriented due to muddy waters and complex pipe structures. In the worst case this may lead to accidents and loss of equipment. Lin and Kuo [35] at the University of Strathclyde (UK) developed a VR-assisted navigation system for underwater tethered robots. The sonar-based robot positioning system matched it's data against a CAD model of the underwater structure being inspected. The simulation displayed the position as a graphical icon registered at the proper location in the 3-D CAD model. Since sonar data was noisy, the robot actual dimensions were increased by a buffer field (called "robot safety domain") designed to prevent collisions or tether entanglement. An optimal navigation path was automatically generated based on the underwater starting location and destination, and displayed in the same simulation scene. This intuitive graphical interface reduced the operator workload, and increased the mission chances of success despite poor visual and sonar sensing data.

In the above examples a "virtual camera" allowed the operator to see portions of the remote scene that were not within direct view, or at a higher level of detail that was not possible otherwise. If motion in the remote environment is at sub-molecular scale then clearly simple visual feedback will not suffice. Taylor and his colleagues at the University of North Carolina at Chapel Hill [36] developed a teleoperation system consisting of an old Argonne II master manipulator controlling the head of a scanning tunneling microscope (STM). Unlike other microscopes, the STM returns an elevation map of the structure being scanned, based on the electrical current from its scanning head to the sample surface. A computer was used to recreate a virtual image of the atomic surface and display it on the user's HMD. Furthermore, spring-like forces were calculated based on the handgrip height at a given surface location and then fed back by the Argonne II manipulator. It was thus possible to "navigate" the hills and valleys of the atomic structure, and even to modify it by generating electrical pulses whenever the user was squeezing the master hand trigger.

C. Teleoperation with Time Delays

Another difficulty associated with teleoperation is the presence of time delays due to communications over long distances. This affects space-based tasks, as well as land-based

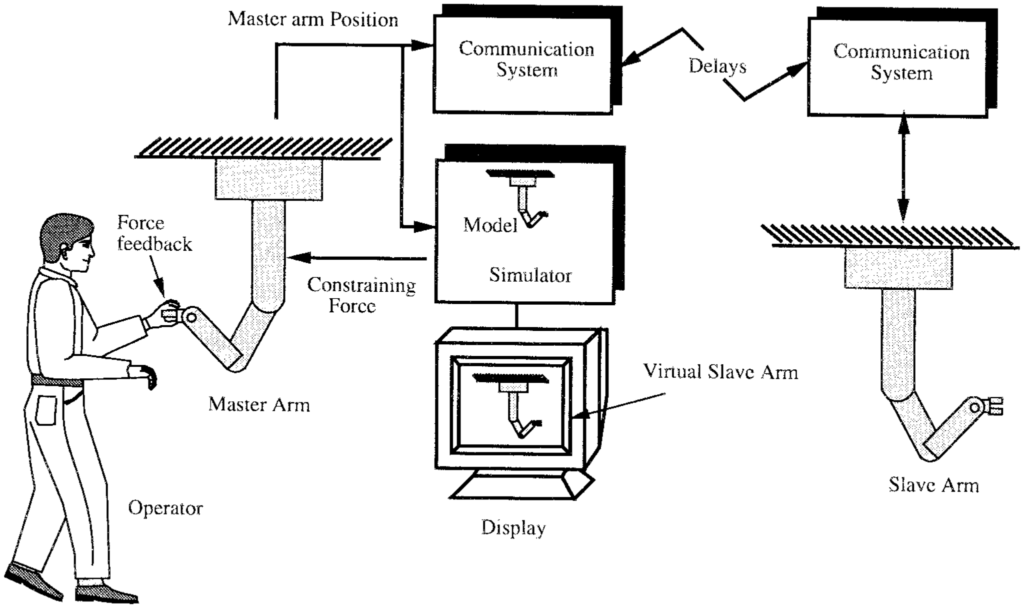


Fig. 9. Telerobotic system using a virtual slave arm to overcome force feedback instabilities due to transmission delays (adapted from [38]). © 1992 IEEE. Reprinted by permission.

operations when communication is done over satellite up links. In time-delayed teleoperation it may take up to a few seconds for the remote robot to receive operator's input and for visual confirmation of the outcome to reach the operator. In classical teleoperation the user has to adopt a "move-and-wait" strategy, which leads to unacceptably long task durations.

One solution to the above problem was the "phantom robot" proposed by Bejczy and his colleagues at Jet Propulsion Laboratory [37]. The phantom robot is a graphical representation of the remote robot that is overlaid to the video image of the real robot. The virtual robot is properly registered with the position of its real counterpart and responds instantaneously to the operator's input. Thus the phantom manipulator serves as a predictor of the remote motion, and allows faster and safer teleoperation. The researchers conducted human factors experiments using a Puma robot to perform a simple tapping task with time delays up to four sec. Results showed about 50% reduction in task completion time when the phantom robot was used vs. simple teleoperation.

Time delay affects other sensorial modalities in addition to visual feedback. Feedback forces normally are beneficial to task performance since the operator has a better feel for the contact forces, contact stability, surface roughness or other physical characteristics of the manipulated object [7]. However, time delays as small as 0.1 sec can make force feedback detrimental and can lead to system instabilities. Kotoku at MITI (Japan) [38] attempted to solve this problem in a similar fashion to the approach taken by Bejczy, namely through the use of a virtual slave arm. As illustrated in Fig. 9, the operator position commands arrive from the master arm to the remote slave robot with some communication delay (in this case 0.5 s). The same master input is used by the computer running the simulation to instantaneously move a virtual model of the slave manipulator. Simulation contact forces during virtual slave interactions are instantaneously fed back to the operator through the master arm. Initial experiments were conducted for a simple planar task in which the operator was asked to trace a rigid barrier while pushing with a constant force. The contact was considered frictionless and the remote slave was modeled as a point object. Subjects carried out the task observing the virtual slave arm (and its modeled contact forces) on the graphics display and the real slave on a video monitor. The master arm trajectory was sent to the remote slave arm with 0.5 s transmission delay. Contact forces were fed back to the subjects in half of the trials. Results showed that the addition of force feedback produced a more stable control. Furthermore, the movement of the slave was three times faster when force feedback was present. When the master arm provided no force feedback subjects moved slower because they used only the graphics display to gauge the contact force.

The above approach, while promising, suffers from several drawbacks. First, the virtual models of the slave, remote environment and task were oversimplified. Second, the approach will fail in unstructured environments which cannot be modeled accurately. Rosenberg at Stanford University [39] proposed to use "virtual fixtures" as an alternative method for solving the problem of teleoperation with time delay and force feedback. These virtual fixtures are abstract sensorial data overlaid on top of the remote workspace, and only interacting with the operator. They can occupy the same physical space as objects in the workspace, without geometrical or physical constraints. Later on Rosenberg [40] used virtual fixtures to enhance the performance of a peg-in-hole telerobotic task. A "fixture board" made of plastic was placed in front of the operator wearing an HMD. Position input from the operator was given by an exoskeleton structure, while feedback forces were felt when interacting with the (unseen) fixture board. The performance degradation was measured as percentage increase in movement time under time delay conditions, compared to the case when no delays existed. Results showed that movement time increased as much as 45% when no virtual fixtures were present and the time delay was 450 ms. The virtual fixtures provided a guidance (or enhanced localization) which had a beneficial effect by reducing this difference to about 3%.

Virtual fixtures were also used in a form of telerobotics called "teleprogramming" developed by Sayers and Paul [41] at the University of Pennsylvania. The master station (arm and computer) "programmed" the slave robot by sending high-level commands only thus coping with low-bandwidth communication lines and with time delays. The master arm was a Puma 250 robot equipped with six-axis wrist force sensor. The user applied forces on the arm which were interpreted as input, while the robot position response provided haptic feedback. A graphics workstation displayed a virtual slave which responds to the user's input, as well as "synthetic fixtures" overlaid on the same graphics. These graphics fixtures were automatically activated by the computer to guide the user in the teleoperation task by increasing precision and speed and decreasing the effect of uncertainties in the world model and command process. The "face fixture," for example, had a central attracting region, surrounded by a repelling region corresponding to the facet edge. In this way the user was pushed away from uncertain contact at the edge of a facet and toward a more certain contact at the center of that facet. Other types of fixtures were task-dependent, but generally speaking the system took a very active role in assisting the operator to complete a given task. The disadvantage of the approach was that it required the system to have an understanding of the task.

High-level task understanding (requiring minimal communication bandwidth) coupled with increased robot autonomy proved the solution for overcoming the time delay problem during the first teleoperation of a space robot from ground in 1993 [42]. Tasks such as assembly, or tracking and capture of an object floating in zero gravity were successfully executed despite almost seven seconds of round trip time delays. Key to this impressive demonstration was the use of virtual models of the robot geometry, as well as of its sensory data. This overcame the problem of poor registration of the simulated and real worlds. The tele-sensory-programming system provided the real robot with *relative* positions between the gripper and its environment, thus compensating for absolute position inaccuracies. This solution is presently extended by Jet Propulsion Laboratory, using a collaboration of a ground station operator (large delays) with a Space Station-based operator (no time delays) for the control of space robots [43].

IV. USE OF ROBOTICS IN VIRTUAL REALITY

The above discussion outlined some of the ways VR simulations can help robotics and manufacturing. In all truth robots can in turn be beneficial to VR simulations by acting as force feedback interfaces. Present commercial VR simulations typically ignore the forces that occur during interaction with the synthetic world. Thus the user cannot feel the hardness, compliance or weight of virtual objects he is manipulating. This lack of adequate sensorial feedback diminishes the simulation realism, which in turn reduces the usefulness of the VR system. A recent National Research Council report on the technological challenges of VR states that:

Being able to touch, feel, and manipulate objects in an environment, in addition to seeing (and hearing) them, provides a sense of immersion in the environment that is otherwise not possible. It is quite likely that much greater immersion in a VE can be achieved by the synchronous operation of even a simple haptic interface with a visual and auditory display, than by large improvements in, say, the fidelity of the visual display alone." [44].

While the benefits of haptics for VR are clear, developments have been slow due to limitations in present actuator technology needed to provide the feedback forces [7]. The poor power/weight and power/volume ratio of present actuators lead to bulky and heavy interface devices that are tiring for the user and may pose a safety risk. McNeely at Boeing Co. proposed "robotic graphics," or the use of robots as an unencumbering way to add the haptic sensorial modality to VR simulations [45]. A robot carrying an specially designed turret would track the user's motions and provide "just-in-time" force feedback. The turret can have various shapes or combinations of shapes that replicate the surfaces of virtual objects. When the user reaches to touch or push a virtual object, the robot moves and orients the turret in a corresponding location mapped to that of the virtual object. When the object is hard and immovable (such as a virtual wall) then the robot locks its brakes resisting any motion of the user into that object.

Tachi and his colleagues at the University of Tokyo [46] implemented McNeely's concept in what they called an "active environment display." This was a six DOF pantographic link mechanism supporting a "shape approximation device." As illustrated in Fig. 10, the user wore a passive sensorized exoskeleton on his arm with a hardened finger attachement. Information on the fingertip location was then used by a PC controller to position and orient the turret to display edges, surfaces or vertices where contact is made with the virtual object. Since the pantograph was impedance controlled the interface can reproduce not just shape but also the object's inertia, viscosity or compliance. The drawback of this system is that only a limited number of shapes can be displayed, and a single point contact can be simulated at a time. Later Hirota and Hirose at the University of Tokyo [47] proposed the use of a pin array to substitute the turret in order to have more flexibility in the kinds of shapes that can be displayed. Their prototype had a 4 × 4 lattice of feedback rods with a stroke of ±25 mm. A soft foam sheet covering the array allowed multiple finger interactions, which was another improvement over Tachi's system.

In all robotic graphics systems the robot/turret combination has to anticipate the user's motion and respond accurately, based on fast and precise tracking information. If the communication between the tracker and robot controller has long time delays or the overall system calibration is poor, the simulation realism is lost. Gruenbaum and his colleagues at Boeing Co. recently developed a virtual control panel simulation in which

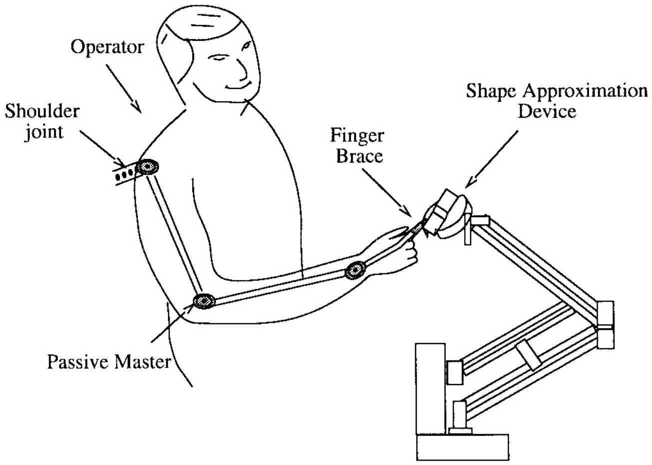


Fig. 10. Active Environment display with "shape approximation device" (adapted from [46]). © 1994 IEEE. Reprinted by permission.

an Adept One manipulator carried a custom panel with various knobs and dials [48]. The researchers discovered that when an electromagnetic 3-D tracker was used to measure the user's fingertip position the misalignment was found to be as high as 15 cm. The misalignment was due to the interference of the robot metallic structure with the tracker measurements. This led to the use of a video tracking system (light bulb and cameras) which still had 1.1 cm error in the direction normal to the display panel. Another concern was user safety which lead to the installation of a Plexiglas panel with holes interposed between the robot and the user. While this increased safety, it also limited the user's freedom of motion, again reducing the simulation realism.

Another approach to the above safety issue in robotic graphics is to replace older position-controlled robots with newer special-purpose manipulators designed from the onset as haptic interfaces. An example is the PHANToM arm developed by Massie and Salisbury at the Massachusetts Institute of Technology, and illustrated in Fig. 11 [15]. Depending on the particular model, the manipulator has a work envelope accommodating the user's wrist up to full shoulder motion. Furthermore, it has optical position encoders to read the gimbal position, as well as three dc actuators to provide translating forces to the user's fingertip. As opposed to position-controlled manipulators the PHANToM is fully back-drivable, such that the user will not feel any forces as long as there is no interaction in the virtual world. The low inertia and friction of this gravity-compensated arm result in a very crisp, high-quality haptic feedback. The high mechanical bandwidth of the interface (800 Hz) allows it to feed back small vibrations such as those associated with contact with rough surfaces. Thus it is possible to map surface mechanical textures over the various virtual objects being manipulated, and then distinguish them based on how they feel (hard-soft, smooth-rough, sticky-slippery, etc.).

Another approach to increased safety in robotic graphics is the use of passive robots that lack external sources of energy. Peshkin and his colleagues at Northwestern University proposed a "programmable constraint machine" using nonholonomic elements, where the entire kinetic energy is provided by the user [49]. The computer controls some areas of the workspace which become programmable constraints, such that motions in those areas lose degrees of freedom. However, it is still the user who has to push the constraint machine to generate a motion. The added benefit of such a passive interface is assured system stability during physical interaction with the user.

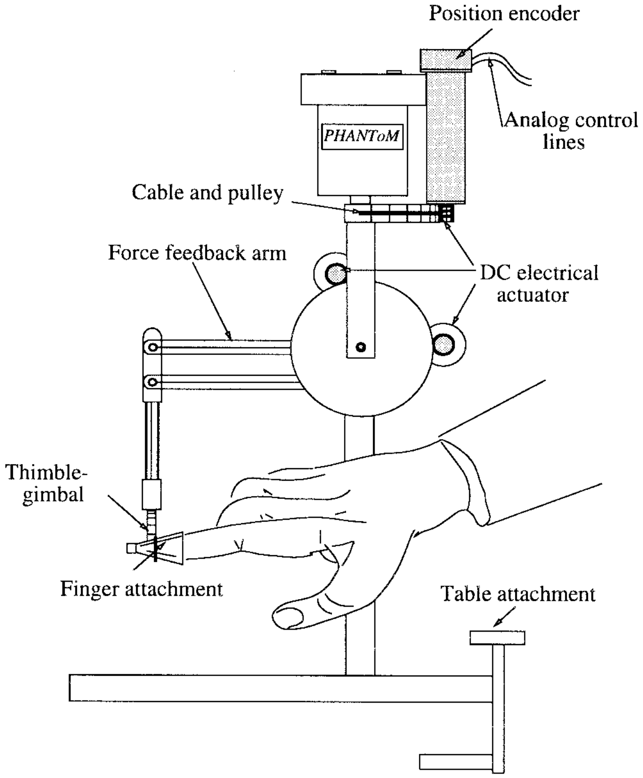


Fig. 11. The PHANToM Master (adapted from [15]). © ASME. Reprinted by permission.

V. CONCLUSION

By now the reader should be able to see the great potential and *mutual* benefits that Robotics and Virtual Reality offer each other. The research projects that have been reviewed show great promise for VR use as tool for CAD, robot programming, plant process simulation, supervisory and collaborative teleoperation, especially when poor visual feedback and time delays exist. Conversely, robotics is beneficial to VR in general by providing haptic interfaces and human factors know-how. The synergy between VR and Robotics is summarized in Table I [50]. Space limitations precluded other interesting application areas, such as in medical robotics and in microrobotics, from being reviewed here.

Since VR is a younger technology than robotics, it will take some time until its benefits are recognized, and until some existing technical limitations are solved. Full implementation in robotics, manufacturing and other areas will require more powerful computers than presently exist, faster communication links, better modeling (especially physical modeling) and better calibration techniques (see also [51] for a "State of

TABLE I
VR and Robotics Synergy (Adapted from [50]).
© John Wiley & Sons. Reprinted by Permission

Application	Traditional Approach	Virtual Reality
CAD design	concepts penciled, no haptics	hand gesture interaction, haptics integrated
Robot Programming	Tedious, knowledge of specific robotic language	User-friendly, high-level programming
Teleoperation	Poor sensorial feedback, Impossible with large time delays, Single-user	Enhanced sensorial feedback, Possible with large time delays, Multiplexed
Haptic feedback	Adequate in some cases, Expensive Human-factors expertise	Special-purpose interfaces, Safety issues, Improved simulation realism

VR" review). Once better technology is available, usability, ergonomics and other human-factors studies need to be done, in order to gauge the effectiveness of the VR-based systems. The important thing to understand now is the true synergy between the fields of Virtual Reality and Robotics. Their respective research and development communities need to learn more about each field strengths and weeknesses and collaborate more to overcome them. To that end this issue of the Transactions represents a significant step in the right direction.

Note: The bibliography is limited by necessity. Readers interested in further VR and Haptics references should consult the author's books [1] and [7]. A good VR *on-line* resource is the University of Washington web site (http://WWW.hitl.washington.edu/projects/knowledge_base /virtual-worlds/). For up-to-date research papers on haptics the reader should also consult the *Proceedings of the ASME Haptics Symposiums* published by the ASME Dynamics Systems and Controls (DSC) Division.

References

[1] G. Burdea, and P. Coiffet, *Virtual Reality Technology*. New York: Wiley, 1994.

[2] J. Krieg, "Motion tracking: Polhemus technology," *Virtual Reality Syst.*, vol. 1, no. 1, pp. 32–36, Mar. 1993.

[3] J. Kramer, P. Lindener, and W. George, "Communication system for deaf, deaf-blind, or nonvocal individuals using instrumented glove," U.S. Patent 5 047 952, 1991.

[4] B. Roehl, "The virtual i/o 'i-glasses!' HMD," *VR World*, pp. 66–67, May/June, 1995.

[5] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: The design and implementation of the CAVE," in *Proc. ACM Comput. Graphics, SIGGRAPH 93 Conf.*, Anaheim, CA, 1993, pp. 135–142.

[6] E. Wenzel, "Localization in virtual acoustic display," *Presence–Teleoperators and Virtual Environments*. Cambridge, MA: MIT Press, Mar. 1992, vol. 1, no. 1, pp. 80–107.

[7] G. Burdea, *Force and Touch Feedback for Virtual Reality*. New York: Wiley, 1996.

[8] *WorldToolKit Reference Manual Version 2.1b*. Sausalito CA: Sense8 Co., 1995 (see also http://WWW.sense8.com).

[9] Viewpoint Datalabs, *Dataset Catalog*, Orem, UT, vol. II, p. 186, 1993 (see also http://WWW.datalabs.com).

[10] P. Richard, G. Burdea, G. Birebent, D. Gomez, N. Langrana, and P. Coiffet, "Effect of frame rate and force feedback on virtual object manipulation," *Presence–Teleop. Virtual Environ.*, vol. 5, no. 1, pp. 1–14, 1996.

[11] W. Mark, M. Finch, S. Randolph, and J. VanVerth, "UNC-CH force feedback library," UNC-CH, TR. 96-012, p. 47, 1996 [Online]. Available: http://WWW.cs.unc.edu/Research/tech-report.html.

[12] J. Kraftcheck, T. Dani, and R. Gadh, "State of the art in virtual design and manufacturing," *VR News*, pp. 16–20, May 1997.

[13] C.-C. Chu, T. Dani, and R. Gadh, "Multimodal interface for a virtual reality based computer aided design system," in *Proc. 1997 IEEE Int. Conf. Robot. Automat.*, Albuquerque NM, Apr. 1997, pp. 1329–1334.

[14] R. Gupta, T. Sheridan, and D. Whitney, "Experiments using multimodal virtual environments in design for assembly analysis," *Presence—Teleop. Virtual Environ.*, vol. 6, no. 3, pp. 318–338, 1997.

[15] T. Massie and K. Salisbury, "The PHANToM haptic interface: A device for probing virtual objects," *ASME Winter Annu. Meeting*, 1994, vol. 55-1, pp. 295–300.

[16] J. Wilson, D. Brown, S. Cobb, M. D'Cruz, and R. Eastgate, "Manufacturing operations in virtual environments (MOVE)," *Presence—Teleop. Virtual Environ.*, vol. 4, no. 3, pp. 306–317, 1995.

[17] Y. Yanagihara, T. Kakizaki, K. Arakawa, and A. Umeno, "Task world reality for human and robot system—A multimodal teaching advisor and its implementation," in *Proc. 1996 IEEE Int. Workshop Robot Human Commun.*, Tsukuba, Japan, Nov. 1996, pp. 38–43.

[18] C. Blume and W. Jakob, *Programming Languages for Industrial Robots*. Berlin, Germany: Springer-Verlag, 1983.

[19] J. Rehg, *Introduction to Robotics in CIM Systems*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[20] W. Strommer, J. Neugebauer, and T. Flaig, "Transputer-based virtual reality workstation as implemented for the example of industrial robot control," in *Proc. Interface Real Virtual Worlds Conf.*, Montpellier, France, Mar. 1993, pp. 137–146.

[21] T. Flaig, K. Grefen, and D. Neuber, "Interactive graphical planning and design of spacious logistic environments," in *Proc. Conf. FIVE Working Group*, Scuola Superiore S. Anna, Italy, 1996, pp. 10–17.

[22] Fraunhofer IPA, "VR4—Software tool for dynamic and real-time oriented virtual environments," company brochure, Stuttgart, Germany, 1996 (see also http://WWW.ipa.fhg.de/300/vr/homepage.html).

[23] J.-Y. Jo, Y. Kim, A. Podgurski, and W. Newman, "Virtual testing of agile manufacturing software using 3-D graphical simulation," in *Proc. 1997 IEEE Int. Conf. Robot. Automat.*, Albuquerque NM, Apr. 1997, pp. 1223–1228.

[24] T. Blackmon and L. Stark, "Model-based supervisory control in telerobotics," *Presence—Teleop. Virtual Environ.*, vol. 5, no. 2, pp. 205–223, 1996.

[25] P. Backes, S. Peters, L. Phan, and K. Tso, "Task lines and motion guides," in *Proc. 1996 IEEE Int. Conf. Robot. Automat.*, Minneapolis MN, 1996, pp. 50–57.

[26] A. Kheddar, C. Tzafestas, P. Coiffet, T. Kotoku, S. Kawabata, K. Iwamoto, K. Tanie, I. Mazon, C. Laugier, and R. Chellali, "Parallel multirobot long distance teleoperation," in *Proc. ICAR'97*, Monterey CA, 1997, pp. 1007–1012.

[27] A. Kheddar, K. Tanie, and P. Coiffet, "Detection of discrepancies and sensory-based recovery for virtual reality-based telemanipulation systems," in *Proc. IEEE 1998 Int. Conf. Robot. Automat.*, Leuven, Belgium, vol. 4, pp. 2877–2883, 1998.

[28] M. Bouzit, P. Richard, and P. Coiffet, "LRP dextrous hand master control system," Tech. Rep., Laboratoire de Robotique de Paris, France, p. 21, Jan. 1993 (see also http://WWW.caip.rutgers.edu/bouzit/lrp/glove.html).

[29] D. Cannon and G. Thomas, "Virtual tools for supervisory and collaborative control of robots," *Presence—Teleop. Virtual Environ.*, vol. 6, no. 1, pp. 1–28, 1997.

[30] M. McDonald, D. Small, C. Graves, and D. Cannon, "Virtual collaborative control to improve intelligent robotic system efficiency and quality," in *Proc. IEEE Int. Conf. Robot. Automat.*, Albuquerque NM, Apr. 1997, pp. 418–424.

[31] E. Paulos and J. Canny, "Delivering real reality to the world wide web via telerobotics," in *Proc. IEEE Int. Conf. Robot. Automat.*, Minneapolis MN, Apr. 1996, pp. 1694–1699.

[32] Internet2, Washington DC [Online]. Available: WWW.internet2.edu.

[33] E. Oyama, N. Tsunemoto, S. Tachi, and Y. Inoue, "Experimental study on remote manipulation using virtual reality," *Presence—Teleop. Virtual Environ.*, vol. 2, no. 2, pp. 112–124, 1993.

[34] W. Kim, "Virtual reality calibration and preview/predictive display for telerobotics," *Presence—Teleop. Virtual Environ.*, vol. 5, no. 2, pp. 173–190, 1996.

[35] Q. Lin and C. Kuo, "Virtual tele-operation of underwater robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, Albuquerque NM, Apr. 1997, pp. 1022–1027.

[36] R. Taylor, W. Robinett, V. Chi, F. Brooks Jr., W. Wright, S. Williams, and E. Snyder, "The nanomanipulator: A virtual-reality interface for a scanning tunneling microscope," in *Proc. SiGGRAPH'93*, Anaheim, CA, Aug. 1993, pp. 127–134.

[37] A. Bejczy, W. Kim, and S. Venema, "The phantom robot: Predictive display for teleoperation with time delay," in *Proc. IEEE Int. Conf.*

Robot. Automat., Cincinnati OH, 1990, pp. 546–551.

[38] T. Kotoku, "A predictive display with force feedback and its application to remote manipulation system with transmission time delay," in *Proc. 1992 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Raleigh NC, 1992, pp. 239–246.

[39] L. Rosenberg, "The use of virtual fixtures to enhance telemanipulation with time delay," in *Proc. ASME WAM*, vol. 49, pp. 29–36, 1993.

[40] ——, "Virtual fixtures: Perceptual overlays enhance operator performance in telepresence tasks," Ph.D. thesis, Stanford Univ., Stanford, CA, p. 214, Aug. 1994 (see also http://WWW-sul.stanford.edu/depts/eng/theses.html).

[41] C. Sayers and R. Paul, "An operator interface for teleprogramming employing synthetic fixtures," *Presence—Teleop. Virtual Environ.*, vol. 3, no. 4, pp. 309–320, 1994.

[42] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "ROTEX—The first remotely controlled robot in space," in *Proc. 1994 IEEE Int. Conf. Robot. Automat.*, San Diego CA, 1994, pp. 2604–2611.

[43] Jet Propulsion Laboratory, "Ground control station task," Pasadena, CA [Online]. Available: http://robotics.jpl.nasa.gov/tasks/gcs/homepage.html.

[44] N. Durlach and S. Mavor, Eds., *Virtual Reality—Scientific and Technological Challenges*. Washington DC: National Academy Press, 1995.

[45] W. McNeely, "Robotic graphics: A new approach to force feedback for virtual reality," in *Proc. IEEE VRAIS'93 Symp.*, Seattle WA, Sept. 1993, pp. 336–341.

[46] S. Tachi, T. Maeda, R. Hirata, and H. Hoshino, "A construction method of virtual haptic space," in *Proc. ICAT'94*, Tokyo, Japan, July 1994, pp. 131–138.

[47] K. Hirota and M. Hirose, "Providing force feedback in virtual environments," in *Proc. IEEE Comput. Graph. Appl.*, Sept. 1995, pp. 22–30.

[48] P. Gruenbaum, W. McNeely, H. Sowizral, T. Overman, and B. Knutson, "Implementation of dynamic robotic graphics for a virtual control panel," *Presence—Teleop. Virtual Environ.*, vol. 6, no. 1, pp. 118–126, 1997.

[49] M. Peshkin, E. Colgate, and C. Moore, "Passive robots and haptic displays based on nonholonomic elements," in *Proc. IEEE Int. Conf. Robot. Automat.*, Minneapolis MN, Apr. 1996, pp. 551–556.

[50] G. Burdea and P. Coiffet, "Virtual reality and robotics," in *Handbook of Industrial Robotics*, 2nd ed., S. Nof, Ed. New York: Wiley, 1999, ch. 17, pp. 325–333.

[51] L. Rosenblum, G. Burdea, and S. Tachi, "VR reborn," *IEEE Comput. Graph. Appl.*, *Special Issue on VR*, vol. 18, pp. 21–23, Nov. 1998.

Grigore C. Burdea (S'87–M'87–SM'90) received the Ph.D. degree from New York University, New York, NY, in 1987.

He joined Rutgers University, Piscataway, NJ, in 1998 and now is Associate Professor of electrical and computer engineering. He has authored the books *Virtual Reality Technology*, and *Force and Touch Feedback for Virtual Reality*, and co-edited the book *Computer-Aided Surgery*.

Dr. Burdea has served as an Associate Editor on the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION and will Chair the upcoming IEEE Virtual Reality 2000 conference.