



A Virtual Reality Tool for Teleoperation Research

Nancy Rodriguez, Jean Pierre Jessel, Patrice Torguet

► To cite this version:

Nancy Rodriguez, Jean Pierre Jessel, Patrice Torguet. A Virtual Reality Tool for Teleoperation Research. *Virtual Reality*, 2002, 6 (2), pp.57-62. 10.1007/s100550200006 . hal-00338968

HAL Id: hal-00338968

<https://hal.science/hal-00338968>

Submitted on 26 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Virtual Reality Tool for Teleoperation Research

Nancy RODRIGUEZ
rodri@irit.fr

Jean-Pierre JESSEL
jessel@irit.fr

Patrice TORGUET
torguet@irit.fr

IRIT – Institut de Recherche en Informatique de Toulouse
Equipe Synthèse d'Images et Réalité Virtuelle
Université Paul Sabatier
118 Route de Narbonne
31062 Toulouse, France

A Virtual Reality Tool for Teleoperation Research

Nancy RODRIGUEZ, Jean-Pierre JESSEL, Patrice TORGUET
IRIT – Institut de Recherche en Informatique de Toulouse

Abstract

Efficiency in teleoperation systems is strongly affected by the relationship between the operator and the remote environment. If the operator has access to easily comprehensible information, his capability for opportune decision-making increases. Virtual reality has showed to be an effective means of displaying information to the human operator. It allows creating three-dimensional and interactive environments, which can be explored and controlled in an intuitive way. In our system, called ASSET, we have applied virtual reality techniques in design and implementation of teleoperation systems human-machine interface. ASSET's objective was to create an object-oriented software framework for teleoperation systems development. This tool allows flexible customizing and can be used as a testbed for evaluating interaction techniques, devices, simulation models and for research about autonomous agents.

Keywords: teleoperation; virtual reality; autonomous agents; testbed; robotics; distributed environment

1. Introduction

Gradecki has defined Virtual Reality as a “technology that allows a user to view a virtual environment from any point and angle, and interact with objects that make up that environment” [1]. Thus, Virtual Reality enhances communication between a system and its users by making intuitive examination and manipulation possible. The use of virtual environments in teleoperation context can improve task performance, thanks to

the efficient display and visualization of complex data [2]. Teleoperation is used to allow manipulation in dangerous or unreachable work sites. A 3D simulation in the controller site provides an environment to the operator for experimenting naturally with the machines and objects of the remote work environment. Also, he can practice independently without concern of damages could be induced in the real equipment under impropriated use. Interfaces based on virtual reality are able to compensate for large time delays since abstract control is generally less sensitive to temporal feedback delay than direct control [3]. This is why we have designed and implemented the *Architecture for Systems of Simulation and Training in Teleoperation*, ASSET that will be described later in this paper.

2. Related Work

One axis of research in virtual environments is the design and construction of more flexible virtual reality systems, able to create prototypes quickly and easily to explore different design alternatives. The flexibility of a VR system allows its use as a higher-level tool, as in the work described by Balcisoy et al [4]. This work presents a framework for testing the design of objects in an augmented reality context. The definition of modelling object geometry is extended with modelling object behaviour to allow users to experiment with a large set of possibilities without having extensive knowledge of the underlying simulation system. This approach shows that users can decrease the time spent on prototype evaluation and have a realistic testing environment. Like in the system described by Jung et al. [5], a VRML/Java-based virtual environment - populated with heterogeneous articulated agents - which serves as testbed for exploring principles for the design of autonomous agents. One of the system explicit goals was that diverse kinds of articulated agents should be integrated into the

virtual environment. That is done by supplying VRML models of the agents' visual appearance and agent control clients that follow a predefined communication protocol. By using (free) portable standard technologies – i.e. VRML97 and Java – the system can be installed on a variety of platforms and even runs on a standard PC without any special graphics hardware.

Virtual reality allows operators to perceive and control complex systems in a natural way. The Virtual Environment Vehicle Interface (VEVI), a system developed by the NASA Ames Research Center's Intelligent Mechanism Group, exploits this ability. VEEI uses real-time, interactive, 3D graphics and position/orientation sensors to control different vehicles in a variety of diverse environments [6]. VEEI was designed and implemented to be modular, distributed and easily operated. VEEI system was followed by VIZ, the latest version of the Autonomy and Robotics Area's developments of Virtual Environment for Vehicle Interfaces [7]. VIZ is designed to be a generic modular and flexible VR-based interface that allows easy reconfiguration and rapid prototyping. VEEI and VIZ have shown that virtual reality interfaces can improve the operator task knowledge and provide valuable tools to understand and analyse the remote environment. The 3D simulation enables telerobot control even when communication rates are slow. Aditya et al, have used Java 3D in their telerobotic system for creating and manipulating a Mentor robot [8]. Robot simulation is chosen to substitute a real image from a camera for reducing communication latency and system traffic. This system is a sample of the efforts to applying teleoperation systems to low bandwidth networks. In the interest of web-based telemanipulation, Ghiasi et al. have proposed a general framework of Java-based components that allows researchers to focus on their particular applications instead of building the infrastructure for Web interaction themselves [9]. These components only require researchers to build an interface to their

framework, instead of implementing a complete end-to-end solution. This framework is designed to enable the manipulation of simple to medium complexity devices via the World Wide Web. Example application domains include small robotic vehicles and robotic arms.

As we see, researchers look for maximum flexibility from their hardware and software, to meet the specific needs of new projects, and because general reusable frameworks avoid redundant or unnecessary developments. On the other side, combination of virtual reality and simulation allows development of more flexible and useful systems. Virtual Reality provides a level of realism that is key for training people in real control procedures. Behaviour and attributes can be assigned to objects to simulate their response and function within the real world. The simulation engine handles the computational tasks required to add physical properties to the objects within the virtual environment [10, 11].

Following these ideas, we are developing a system, based on virtual environments, portable and extensible, intended to facilitate the construction of new teleoperation systems. In ASSET, we have identified a number of core requirements present in all teleoperation applications. These mechanisms are provided as a set of reusable components, highly configurable, to which it is possible to add or change modules in a dynamic way. This allows us to use the system as a testbed for evaluation of 3D scenes, artificial intelligence behaviours, new devices or interaction techniques. An additional objective in ASSET is to have a low cost, light weight, object-oriented system developed with open source products, which does not require special hardware. For this reason, we have chosen Java and Java3D to develop our system, so that it can run on any platform without further changes. Also, even if we can use high-end displays, we are using a conventional computer monitor to display the visual world. Our system can

thus be regarded as a low cost VR solution that can be used for many applications in telerobotics research.

3. System Design

ASSET is a tool for helping development of teleoperation systems, easily adaptable to different contexts. The ASSET system provides an object-oriented component-based architecture and a modular design to easily make extensions and modifications.

The architecture of teleoperation systems implemented by ASSET is shown in Figure 1.

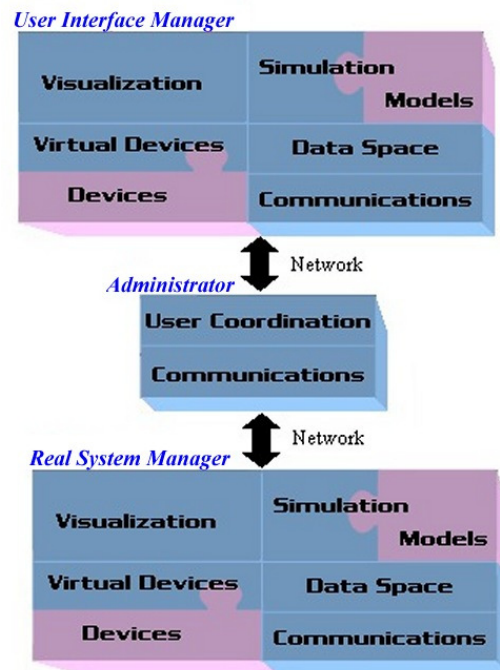


Figure 1. ASSET architecture

ASSET architecture consists of three modules: the User Interface Manager, the Real System Manager and the Administrator. The User Interface Manager is responsible for the communication between the system and the user, it handles the local simulation and the interaction devices. The Real System Manager controls the real system. It executes commands and manages coherence between the real state and the simulated state in

order to update the user simulation. The Administrator transmits commands from users to effectors, and the information from the real system to the users. Conflicts raised by orders of different users are solved by its Coordination component.

Communication between ASSET modules is managed by their Communications and Events component. All interactions between components (simulation, devices, visualization) are defined by message passing and synchronized by means of the data space and event handling. Conversion between generic information and specific information is made only by components – such as physic devices controllers or behavioural units - supplied for a particular application. This allows reusing already built components and facilitates the integration phase because the functional interface of each component remains constant.

The ASSET system offers to programmers the services related to teleoperation missions: 3D simulation, devices management, communications and VR-based graphical user interface. For building a specific application, these basic services are integrated with the particular application resources. For example, for behaviours testing, the developer only needs to provide a geometric model and a control class for his entity. In the same way, for testing a new device, the developer only needs to implement the services defined in the virtual device. For helping development and evaluation tasks, ASSET can be used in simulation only (useful in training and experimentation) or in direct control, i.e. changes to the simulated robot are sent in real-time to the actual device. In this case, commands are previewed and simulated before sending them to the real robot to verify command validity and to reduce real system errors.

ASSET uses virtual reality interaction techniques – such as 3D visualization and devices integration - and simulation to allows researchers, designers and users to construct and work with machines and environments in a quick and inexpensive way,

compared to real systems costs. Also, our generic framework provides for easy extensibility and reusability because processes are separated of specific data. This allows ASSET to be used in fast construction of prototypes and test programs for new devices or simulation models.

The ASSET architecture has been implemented using Java. For visualization purposes, we are using Java3D that offers a high-level API for designing 3D systems. These components were chosen specifically because they are free. In order to achieve our goal of a generic, highly configurable system, we use an object-oriented approach. As each module and component is an object accessed only by its functional interface, it can be maintained and changed independently of each other. Users modules are loaded from a configuration file, read at initialisation. This configuration file allows the full customisation of a particular application needs: communication protocol, simulation objects (geometry and behaviour), interaction devices, actuators and commands validity. For analysing different environments configurations, the user can change this file, rather than modify the source code and re-compile. The configuration file approach has permitted rapid prototyping and implementation of various environments of which the most significant is described in the next section.

4. Evaluation

In order to help the user to accomplish a task, teleoperation systems have to integrate different tools such as visualization, several interaction devices, planning tools, etc.... The interface must be able to give complete information of the real world and the operator, using a distributed platform, can be helped by other users as well as by autonomous robots. The objective of our project, called EVIPRO (*Virtual Environment for Robotics*), is to combine teleoperation, virtual reality and adaptive systems to

improve the control of teleoperation missions [12]. The overall system consists of a reactive system (ASSET) that understand the events of the dynamic environment, and a behavioural simulation system used by autonomous robots (A3) whose mission is to help the user. Figure 2 shows EVIPRO configuration:

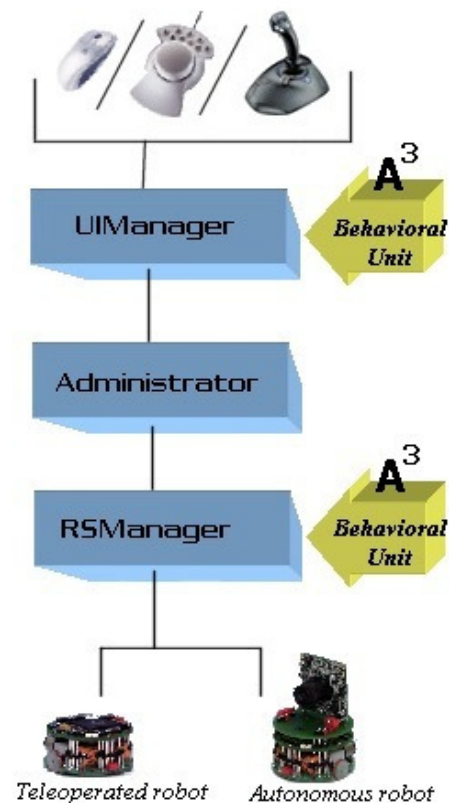


Figure 2. EVIPRO configuration

EVIPRO system is used for controlling two Khepera robots [13]. One of the robots is teleoperated and a behavioural unit called A3 controls the other one. The user addresses commands to his robot by means of an interaction device – spacemouse and joystick controllers have been implemented - and the user interface. These commands are sent to the local simulator to produce user feedback. If the result of his action is a valid simulation state, the commands are sent to the control module of the real system by the way of the Administrator. Real System updates the simulation allowing the behavioural unit to know the state of each simulation object, and to react as well as possible to the

new environment. This unit filters information to obtain the needed data for making a choice of behaviour and update his state. Finally, the two Khepera robots execute their commands. The Khepera autonomous robot is guided by the behavioural unit associated with the remote simulator, allowing the A3 system to only work with valid global data. The unit in user simulation just produce visual feedback. The strong object-oriented model of the ASSET system allows for easy reconfiguration of the robots. Furthermore, because this model separates the geometric robot model from its intermediate control layers, we can develop and evaluate the different application components separately.



Figure 3. User interface of ASSET system

The construction of this system allows us to verify the architecture and the ideas stated in ASSET. Thanks to the Reflection package of Java, it is possible to dynamically load users classes and components to the system, to add new functionalities without modifying core modules. Nevertheless, some difficulties have arisen, the most

important being collision detection. In fact, in Java3D it is possible to know when collision is produced but it is very difficult to determine collision information. For the moment we are using a simple algorithm of collision detection based on bounding spheres while we work in the integration of a new collision detector.

5. Conclusions and Future Work

We have developed ASSET, a support environment that serves as a testbed for teleoperation systems development. In this experimental environment, the developer can test new simulation models, behaviours and devices. ASSET is built only with commonly available tools such as Java and Java3D and has not high-cost hardware requirements. We have used our system in the building of a cooperative robotics application, and we have been able to demonstrate the benefits of its flexibility, in particular for the dynamic integration of behaviours in the simulation and for the construction of specific devices independent applications.

Future work will cover several aspects. At the current time, we are working in the following directions: dynamic devices and environment configuration, integration of devices specific capacities (plug-ins), improvements on the collision detection engine and multi-user coordination. Another valuable feature to be integrated will be task planning in order to allow the operator to use the result of his experiences in simulation and to execute a task in the real system without permanent control.

References

1. Gradecki, J. The Virtual Reality Programmer's Kit. John Wiley & Sons: New York. 1994
2. Stanney KM, Mourant RR, Kennedy RS. Human Factors Issues in Virtual Environments: A Review of the Literature. Presence 1998; 4: 327-352

3. Pook P, Ballard, D. Remote Teleassistance. IEEE International Conference on Robotics and Automation, Nagoya, Japan, May 1995, 944-949
4. Balcisoy S, Kallmann M, Fua P, Thalmann D. A framework for rapid evaluation of prototypes with Augmented Reality. In: Proceedings of ACM Symposium on Virtual Reality Software and Technology (VRST 2000) , Seoul, Korea, October 2000. Available in: <http://ligwww.epfl.ch/~thalmann/publi.html>
5. Jung B, Milde JT.. An Open Virtual Environment for Autonomous Agents Using VRML and Java. 4th Symposium on Virtual Reality Modeling Language, Paderborn, Germany, February 1999, 7-11
6. Piguet L, Fong T, Hine B, Hontalas P, Nygren E. VEVI: A Virtual Reality Tool for Robotic Planetary Exploration. Virtual Reality World Conference, Munich, Germany, February 1995, 263-274
7. Nguyen L, Bualat M, Edwards L, Flueckiger L, Neveu C, Schwehr K, Wagner MD, Zbinden E. Virtual Reality Interfaces for Visualization and Control of Remote Vehicles. Autonomous Robots 2001, 1:9-18
8. Aditya A, Riyanto B. Implementation of Java 3D Simulation for Internet Telerobotic System. In: Proceedings of IASTED International Conference Modelling and Simulation (MS'2000) . Pittsburgh, USA, May 2000

9. Ghiasi S, Seidl M, Zorn B. A Generic Web-based Teleoperations Architecture: Details and Experience. SPIE Telemanipulator and Telepresence Technologies VII. Boston, USA, September 1999, 234-247
10. Anderson J. A Generic Simulation System for Intelligent Agent Designs. Applied Artificial Intelligence 1995; 5:527-562
11. Robb RA, Cameron B. Virtual Reality Assisted Surgery Program. Biomedical Imaging Resource, Mayo Foundation, Rochester, Minnesota
<http://www.mayo.edu/bir/reprints/VRASP1.html>
12. Heguy O, Rodriguez N, Jessel JP, Duthen Y, Luga H. Virtual Environment for Cooperative Assistance in Teleoperation. Proceedings of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG'2001), Plzen, Czech Republic, February 2001, 3:II9-II12
13. K-team. Khepera, Miniature Mobile Robot. K-team. <http://www.k-team.com/robots/khepera/index.html>
14. Fong TW, Thorpe C, Baur C. Collaborative Control: A Robot-Centric Model for Vehicle Teleoperation. In: Agents with Adjustable Autonomy, Papers from the AAAI 1999 Spring Symposium, Technical Report SS-99-06, The AAAI Press: Menlo Park, California. 1999
15. Smith N, Egert C, Cuddihy E, Walters D. Implementing Virtual Robots in Java3D using a Subsumption Architecture. Proceedings from the Association for the Advancement of Computing in Education, Honolulu, USA, 1999, 975-980