To appear in Proc. of 1998 IEEE Industrial Electronics Conference (IECON98), Aachen Germany

Virtual Reality Visualization of Distributed Tele-Experiments

Armin Hopp Dirk Schulz Wolfram Burgard Armin B. Cremers Dieter Fellner
Department of Computer Science III, University of Bonn, 53117 Bonn, Germany

Abstract

The increased costs of laboratory environments such as mobile robots and the specialization of research groups increase the demand for collaboration between different research groups. Although the Internet can be regarded as the most important medium for cooperation over large distances, it does not provide the necessary bandwidth for the transmission of video-streams required for example during a tele-experiment with a mobile robot. In this paper we present a combination of a tele-experimentation environment for autonomous mobile robots (RTL) with a minimal rendering tool-kit for virtual reality (MRT-VR). The RTL-system includes means for bridging transmission gaps by using simulation techniques to predict the robots actions and supports automatic viewpoint selection for virtual cameras in the MRT-VR system. The MRT-VR allows the on-line visualization of experiments in a virtual environment. MRT-VR additionally supports natural navigation of the user through the 3D virtual environment in various aspects. It includes a collision detection to avoid that users get inside of objects such as walls or cupboards and supports climbing of stairs. Additionally users can select different viewpoints or even automatically follow the robot during its operation. Finally it supports the synchronization of the viewpoints for distributed observers. This paper demonstrates, that by combining both techniques an improved visualization quality with respect to the precision of the simulation and the ease of operation of tele-operated systems is obtained.

1. Introduction

The visualization of joint experiments between distributed research groups is an important means for supporting collaboration over the Internet. Due to the increased costs of laboratory environment such as mobile robots and due to the increased specialization of research groups on the other hand, there is a growing need for cooperation between such research groups. Unfortunately, the Internet as one of todays major communication media only has a limited bandwidth so that no video streams of joint experiments can be transmitted.

In this paper we describe, how such experiments can be supported by a combination of the tele-experimentation environment RTL and the minimal rendering tool MRT-VR. The RTL-system is integrated into a robot control system and uses simulation techniques to predict the behavior of the robot in the case of transmission gaps. This way it provides the necessary means for smooth animations of ongoing experiments. The MRT-VR system is a multi user virtual environment and is designed to support distributed education and design. It is able to perform real-time rendering as well as ray-tracing and radiosity. Furthermore, it supports the navigation through the scene by avoiding collisions with virtual objects thus preventing the user, for example, from getting lost inside of an object. The RTL-system controls the MRT-VR system by transmitting the current pose of the robot and by automatically selecting viewpoints in the virtual scene.

Computer graphics visualizations are widely used in tele-operation interfaces for mobile robots, especially during space missions [9, 1], where the transmission delays are extremely large (up to several minutes) and only a low bandwidth is available. The visualization interfaces used are specialized for the remote control of a robot under these conditions. In contrast to this, we employ a general purpose virtual environment system, as we are mainly interested in the observation of the robot's autonomous performance during an experimental operation in a distant laboratory by researchers working in different locations.

Automatic camera control in virtual environments gained increasing attention over the last few years in the Computer Graphics and AI communities, mainly with the focus on the intelligent selection of a viewpoint according to heuristic rules taken from cinematography. Drucker and Zeltzer [5] introduce a technique to determine optimal camera positions for individual camera shots in virtual environments. Camera positions are subject to constraints which are derived from cinematographic rules. They are determined using constrained optimization techniques. Because of the complexity of this task the camera positions have to be computed off-line. He et al. [8] present a system that automatically switches between camera positions in a virtual environment. A sequence of actions in the environment is partitioned into a sequence of shots where each shot is assigned a camera which might also perform camera movements.

While these techniques are mainly motivated from an esthetic point of view, this approach is often not appropriate in the visualization of distributed tele-experiments. Here, the viewpoints generally have to be chosen such that they provide the maximum of information to the viewer. In our current system we allow the user to specify static viewpoints allowing the user to monitor specific actions of the robot such as complicated navigation operations or even manipulation processes. The MRT-VR system furthermore provides a special tracking mode allowing the user to automatically follow the robot while it is moving through the scene.

2. The MRT-VR System

The MRT-VR system is a multi user virtual environment implemented on top of the MRT library (**M**inimal **R**endering **T**oolkit, MRT-VR = MRT-"virtual reality"). The MRT library implements real-time rendering (OpenGL, XGL and Direct3D) as well as ray-tracing and radiosity. The different renderers are all based on one object-oriented scene graph.

The main design goal for the MRT-VR environment is to support distributed education and design. MRT-VR is developed as an extension to the MBone-Tools [10], a widely used framework for video and audio conferencing over the Internet. The MBone-Tools use multicasting [4] an extension to the Internet protocol that provides bandwidth efficient group communication. As demonstrated in [11] multicasting is an ideal communication medium for networked virtual environments with a large number of participants.

MRT-VR adds extra capabilities for cooperative work over the Internet to this environment. Like most of todays virtual reality systems MRT-VR uses VRML as the scene description language, but in addition to navigation facilities it offers special techniques for the communication between distributed users, for example highlighting objects of interest or directing the viewpoint of other session members to instructive perspectives.

2.1. MRT-VR Communication

The network communication component of MRT-VR is designed to allow for the manipulation and deletion of any existing object in the scene description as well as for the creation of new objects. The data transmitted between distributed MRT-VR session members is organized as a stream of scene manipulation messages. To suit the needs of the different application and local network environments in which MRT-VR might be used, this stream can be sent over a variety of different data transport protocols. Among them are MBone as the most advanced communication channel for cooperatively working over the Internet, and TCX [6] which is a special communication protocol for robot control systems (see Figure 1).

All incoming messages are handled by the object-oriented data replication layer of MRT-VR. A dispatcher decodes the messages determining the action to be taken, e. g. insertion, deletion, and modification of an object. The messages are then forwarded to the class of the object, which in turn either forwards it to the object involved (modify message) or composes or destructs a new object (insert, delete messages).

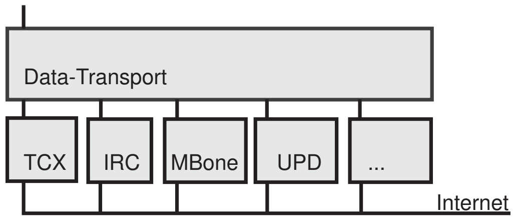

Figure 1. The MRT-VR Data-Transportation layer and some of the supported communication protocols

2.2. Navigation

While navigating through 3D spaces users often loose their orientation [7], either by stepping into an object, e. g. diving into the ground or ending up between two sides of a wall, or by performing navigation actions, that usually are not carried out in the real world, such as viewing the scene upside down.

MRT-VR supports a viewer who is navigating through the scene in several ways. It contains different levels of collision detection which directly affect the camera path and meet the user's intuitive expectation of solid objects. MRT-VR furthermore controls the user's camera path by controlling the distance of the virtual camera to relevant objects such as the floor and simultaneously avoiding collisions with other objects. This way, for example, it achieves the consistent impression of "walking" on the ground by keeping the distance to the ground constant. It also supports stepping stairs and jumping on a podium. Additionally, MRT-VR provides methods for following moving objects through the scene. In such a situation MRT-VR chooses the viewpoint depending on more complex constraints given by the distance to the observed object, visibility and viewing direction (see Figure 2). The efficient implementation of these distance based navigation techniques is tremendously eased by the ray-tracing facilities supplied by the MRT library.

These features of MRT-VR are highly important, for example, in the context of the visualization of mobile robot experiments, because it allows the viewers to follow the robot without the need to navigate through the scene by hand.

3. The RTL System

The robotic tele-laboratory system RTL is designed as an experimentation platform which permits distributed researchers to carry out experiments with an autonomous mobile robot over the Internet. Because of the varying and sometimes very low bandwidth of the Internet, RTL relies on 3D graphics visualizations of the robot and the laboratory environment instead of video transmissions. In addition to lower bandwidth requirements graphics visualization offer more flexible inspection possibilities than video transmission. Experimenters can choose arbitrary viewpoints in the virtual scene, while they are restricted to the viewpoints

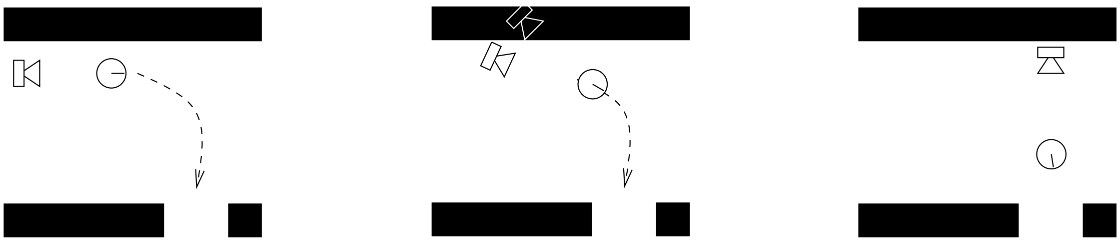

Figure 2. A virtual camera following the robot

Figure 3. The RWI B21 robot RHINO.

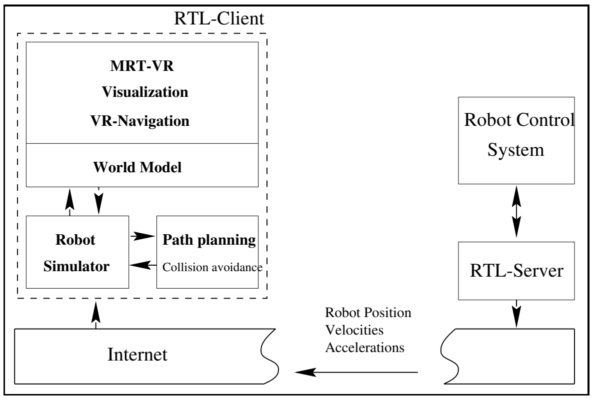

Figure 4. Information flow in the RTL tele-operation environment.

of few statically mounted cameras when using video transmission.

RTL employs MRT-VR as the user interface for the tele-experimenters. On the one hand, RTL benefits from the advanced virtual reality navigation and inspection mechanisms of MRT-VR, instead of implementing it's own visualization component. On the other hand, RTL adds an additional inspection technique, based on automated viewpoint switching, which eases the observation of the robot in it's environment. Furthermore MRT-VR is enhanced by RTL with a simulation based dead reckoning component enabling it to perform smooth animations of software controlled 3D objects even when large Internet transmission delays occur.

RTL has a client-server architecture [3] (see Figure 4) and has been implemented for the robot RHINO [2], an RWI B21 (see Figure 5). The robot control system of RHINO consists of several software modules, each performing a distinct part of the robot control task, like collision avoidance, robot localization, path-planning and task-planning [12]. The RHINO project focuses on the development of a flexible service robot platform which can be used for example as a delivery robot in office environments as well as a mobile information agent.

The server of the RTL system is a module of the RHINO system and a member of an MRT-VR session at the same time. As a module of the RHINO system, it receives the current position of the robot as well as the current plan of the robot's future actions from the responsible modules of the RHINO system. According to the robot's, the RTL system as an MRT-VR session member animates the robot's avatar in the virtual scene and decides automatically which viewpoint has to be taken inside the scene.

A client of the RTL system is an MRT-VR client enhanced with a special robot simulation component. The RTL client receives messages containing the robot's position and speed as well as it's next target position from the server. A simulator predicts the behavior of the real robot when transmission delays occur and produces the MRT-VR update messages which control the 3D visualization. It simulates the odometry and the proximity sensors of the robot, and employs a replication of the robot's path planning and collision avoidance facilities to achieve a reliable prediction of the real robot's actions. The accuracy of RHINO's localization module in combination with the reliability of the predictive simulation of RTL ensures, that RTL always presents a nearly exact representation of the real situation in the laboratory (see Figure 5).

4. Automatic Camera Selection in RTL

As part of the MRT-VR system the RTL-client is able to define it's own camera position. RTL exploits this ability to automatically change the experimenters viewpoint during an experiment.

The need for such a mechanism arises quiet naturally during the design phase of a tele-experiment. In most cases

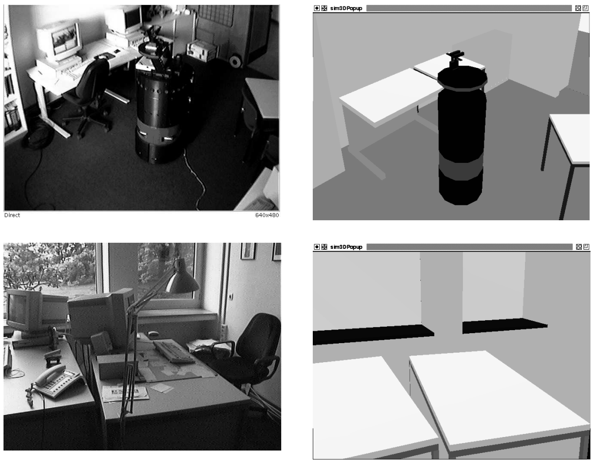

Figure 5. RTL permits 3D visualizations of real scenarios. The first row shows a real and a synthetic picture of RHINO in an office, whereas the second row shows a real and a computed view through RHINO's camera.

the experimenter has a clear idea from which viewpoint he wants to observe a certain phase of an experiment. RTL implements a simple automatic switching mechanism, which can easily be specified during the experiment design phase. The approach is based on regions of interest, which are linked to specific static viewpoints. A region of interest is a cube in (x, y, Θ)-space, the space of robot positions (x, y) including the heading Θ.

RTL switches to a new viewpoint whenever the robot leaves a region of interest according to a selection scheme specified by the user.

Figure 6 displays an example environment covered by 5 regions of interest. The graph in Figure 7 represents the transition scheme for this environment. For the sake of clarity we do not consider the dependency on the robot's heading. In this example the regions of interest DW-1 and DW-2 completely overlap and denote the same (x, y, Θ)-cube (see Figure 6). However, they are linked to different viewpoints. RTL keeps the viewpoint of camera 2 when the robot leaves ROOM-1 as long as it moves on the doorway DW-1 and it keeps the viewpoint of camera 3 when it leaves ROOM-2 as long as it moves on DW-2.

The user is not forced to provide a complete and unambiguous selection scheme. To deal with possible ambiguities or missing transitions RTL applies the following heuristic rules:

1. If the selection scheme specifies several adjacent regions of interest for a region of interest and a robot position, RTL nondeterministically chooses one of them.

2. If no adjacent region of interest is specified for a robot position, RTL nondeterministically chooses one the regions, the robot's current position lies in.

3. If no region of interest exists at the robot's current position, RTL keeps it's viewpoint.

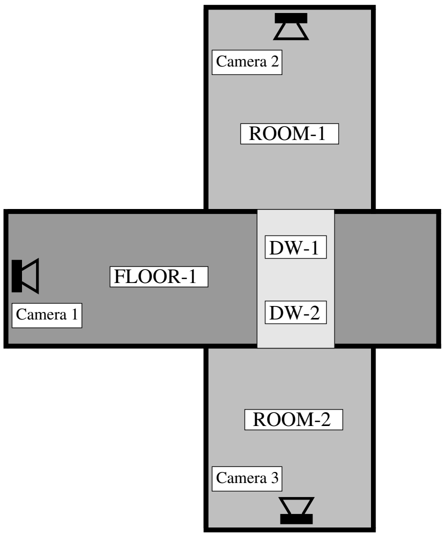

Figure 6. Example environment covered by 5 regions of interest

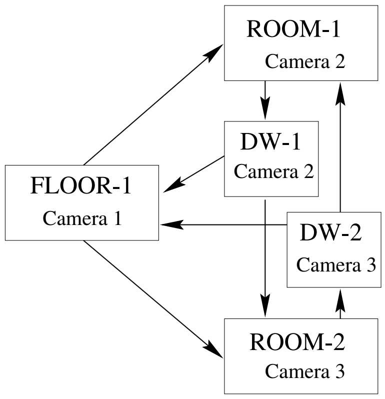

Figure 7. Transition scheme for the example in Figure 6

4. Initially a region of interest is chosen according to rules 2 and 3.

In principal, this technique allows to approximate any viewpoint control which is solely based on the robot's position. Even camera movements depending on the robot's trajectory can be simulated using a large number of small regions of interest covering the trajectory and linked in sequence.

5. Experimental results

In the current state of implementation, RTL supports the accurate and smooth visualization of navigation experiments. These experiments can be observed by a group of experimenters which are distributed over the Internet. Figure 8 illustrates the capabilities of the tele-experimentation system, by showing two sequences of still images of the 3D

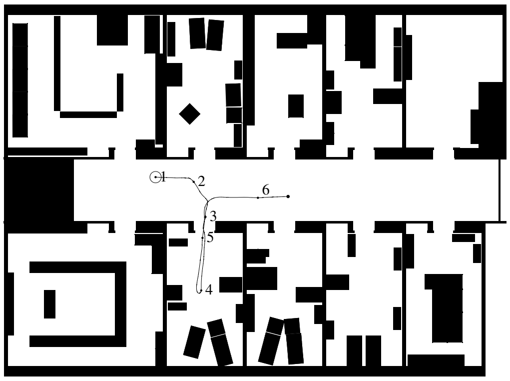

Figure 9. Path of the robot during the experiment

visualization generated during an experiment in our office environment. Figure 4 shows the robot's trajectory during this experiment. The numbers indicate the positions where the images were generated.

The first sequence shows the visualization obtained by the automatic robot tracking mechanism of MRT-VR. It should be noted that, for the second image of this sequence, MRT-VR had to correct the relative viewpoint of the viewer as illustrated in Figure 2 since the viewer would have been inside the wall of the corridor, otherwise.

The images of the second sequence have been obtained by the automatic camera switching component of RTL. The regions of interest and the virtual viewpoints in this example have been chosen to put a special focus on the doorway. On each side of the door a virtual camera is installed which provide full views of the robot while it is handling the door and while it enters or leaves the room. This way we get a detailed illustration of the robot's actions during the situations of interest of this experiment and we automatically switch to long views during the less interesting phases of the experiment and watch the robot move.

It is should be emphasized, that the users are not bound to one visualization method for a complete experiment. They can also manually switch between viewpoints at any point in time.

6. Conclusions

In this paper we presented an integration of the robotic tele laboratory environment RTL with the virtual reality minimal rendering tool MRT-VR. This integrated system allows distributed researchers to observe joint experiments visualized in a virtual environment over the Internet. This approach has several advantages. First, it does not require high bandwidth connections to transmit video streams. Second, the users can adopt their viewpoints to focus on relevant aspects of the ongoing experiment. Third, the virtual reality environment supports the documentation of experiments, since appropriate animations can be computed off-line after the experiment has been finished. The system supports experimenters in choosing appropriate viewpoints during the experiment. It provides a navigation mechanism that avoids collisions with objects in the virtual scene. Furthermore it includes a tracking mechanism that can be used to automatically follow any moving object. Finally, the user can define static viewpoints that are chosen automatically according to the current position and orientation of the robot.

Future work on this environment will address the integration of information obtained with the robot's sensors into the scene. For example, whenever the robot passes a door and measures its opening angle, the state of this door should also be updated in the virtual environment. Additionally the system should also support manipulation tasks in which objects are moved by the robot. Furthermore, it is important to provide animated visualization of participants and their viewpoints by individually selectable avatars.

References

[1] B. Brunner, K. Landzettel, B. M. Steinmetz, and G. Hirzinger. Tele-sensor-programming – a task-directed programming approach for sensor-based space robots. In *Proc. of the Int. Conf. on Advanced Robotics (ICAR)*, 1995.

[2] J. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos, and S. Thrun. The mobile robot Rhino. *AI Magazine*, 16(2):31–38, Summer 1995.

[3] A. Cremers, W. Burgard, and D. Schulz. Architecture of the Robotic Tele Lab. In *Proc. of the 1997 Annual Conference on Advances in Multimedia and Simulation*, Bochum, Germany, 1998.

[4] S. Deering. Host extensions for IP multicasting. In *Request for Comments (RFC) 1112, Internet Engineering Task Force (IETF)*, August 1989.

[5] S. M. Drucker and D. Zeltzer. CamDroid: A system for implementing intelligent camera control. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, 1996.

[6] C. Fedor. TCX. An interprocess communication system for building robotic architectures. Programmers guide to version 10.xx. Carnegie Mellon University, Pittsburgh, PA, December 1993.

[7] D. Fellner and O. Jucknath. MRTSpace - multi user environments using VRML. In *Proceedings of WebNet96*. H. Maurer (ed.), 1996.

[8] L. He, M. F. Cohen, and D. H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In *Computer Graphics (SIGGRAPH '96 Proceedings)*, 1996.

[9] B. Hine, P. Hontalas, T. Fong, L. Piguet, E. Nygren, and A. Kline. VEVI: A virtual environment teleoperations interface for planetary exploration. In *SAE 25th International Conference on Environmental Systems*, July 1995.

[10] M. Macedonia and D. P. Brutzmann. MBone provides Video and Audio across the Internet. In *IEEE Transactions on Computers*, 1994.

[11] M. R. Macedonia, D. P. Brutzman, M. J. Zyda, D. R. Pratt, P. T. Barham, J. Falby, and J. Locke. NPSNET: A multiplayer 3D virtual environment over the internet. In *Proceedings of the ACM 1995 Symposium on Interactive 3D Graphics*, 1995.

[12] S. Thrun, A. Bücken, W. Burgard, D. Fox, T. Fröhlinghaus, D. Hennig, T. Hofmann, M. Krell, and T. Schimdt. Map learning and high-speed navigation in RHINO. In D. Kortenkamp, R. Bonasso, and R. Murphy, editors, *AI-based Mobile Robots: Case studies of successful robot systems*. MIT Press, Cambridge, MA, 1998.

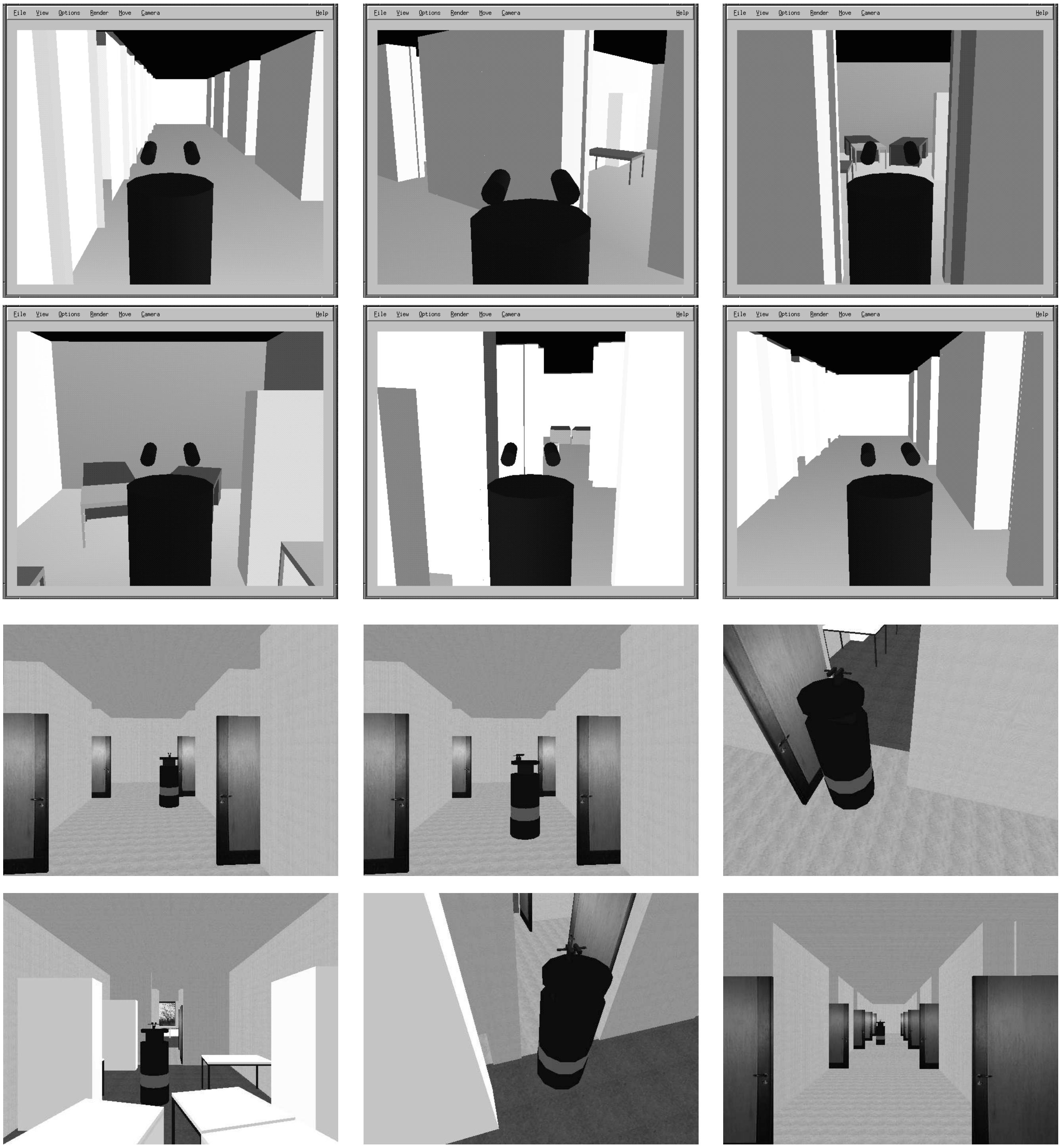

Figure 8. *Top sequence* Automatically following the robot; The experimenter gets the impression of walking behind the robot. *Bottom sequence*: RTL automatically switches viewpoints according to a selection scheme specified by the user