An Immersive 3D User Interface for Mobile Robot Control

N. Aucoin, O. Sandbekkhaug and M. Jenkin
Department of Computer Science,
York University, North York, Ontario, Canada

Key Words: mobile robotics, user interface, immersive displays.

Abstract

Very few autonomous mobile robots are really fully autonomous, and most require some sort of operator intervention. For example, in the ARK (Autonomous Robot for a Known environment) project, the ARK robot requires an operator to provide high-level mission commands. These high-level commands can be provided via a $2\frac{1}{2}$D map-based user interface as well as through an immersive 3D interface. The 3D interface provides the operator with a virtual reality-like control interface. It allows the operator to move through a simulation of the robot's environment, to examine the environment through an immersive display, and provides access to high-level mission commands in a more informative and natural way than is possible with the standard $2\frac{1}{2}$D map-based user interface.

1 Introduction

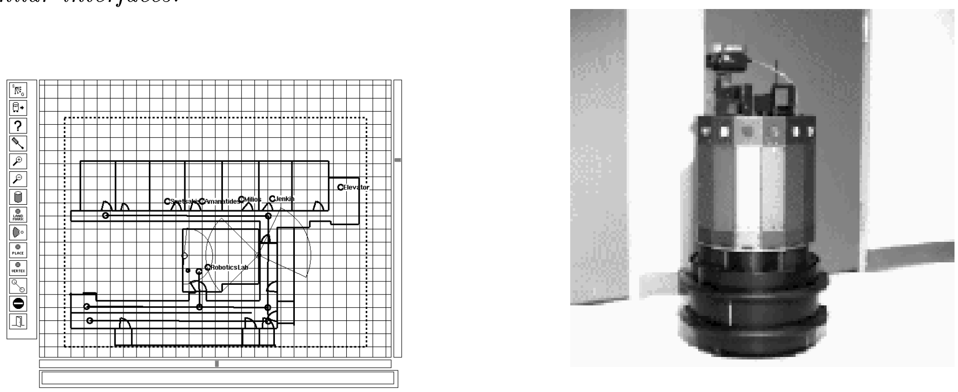
The ARK (Autonomous Robot for a Known Environment) Project is a precompetitive research project involving Ontario Hydro, the University of Toronto, York University, Atomic Energy of Canada Ltd., and the National Research Council of Canada. The main technical objective of this four year project is to develop a sensor-based mobile robot that can navigate autonomously in an industrial environment using pre-mapped visual landmarks to aid pose maintenance.

The ARK project has built three different robots; ARK-1, a proof of concept vehicle, ARK-2 an industrial prototype, and ARK-lite, an ongoing research vehicle. ARK-1 and ARK-2 are based on the Cybermotion Navmaster platform, while ARK-lite is built on the Nomadic Technologies Nomad 200 platform (see Figure 1(b)). The ARK-lite uses on-board sonar, infrared and bumper sensors to monitor its local environment and utilises a reactive control architecture to avoid unmodeled and moving objects. Utilising a video camera mounted upon a computer controlled pan and tilt unit the robot uses a set of previously mapped visual landmarks to correct odometry errors as it moves. Additional technical information on the ARK robots can be found in [1].

The standard user interface for the ARK robots is based on a $2\frac{1}{2}$D map similar to the ARK-lite user interface shown in Figure 1(a). The map is primarily 2D but does contain some height information. Through this $2\frac{1}{2}$D interface an operator can command the robot to plan a path from its current location to a specified goal location, execute a pre-planned path, and perform other high level tasks. The user indicates operations using a set of buttons and indicates locations in the robot's environment by pointing with a mouse on the displayed map. Although this user interface was found to be sufficient for a number of tasks, it was found that the operator experienced difficulties when trying to make fine motions of the robot and had difficulty visualising the robot's operating environment.

The problem of presenting an effective integrated user interface for a remotely operated device has been considered in a number of different mobile robotics and teleoperational systems. Various solutions have been proposed, and each has it's own strengths and weaknesses. In tightly defined and restrictive applications it may be possible to provide a highly task-based interface. For example, [2] describes a mobile robot system for hazardous waste drum storage inspection. In this system, the drums themselves define the system tasks, and thus a 2D map based user interface, based primarily on the drums, is possible. Note that even within a domain with well specified tasks, anomalous conditions may require a higher level of operator control, and thus [2] also provides a joystick interface to the device. Driving a remote device with a joystick or some other similar input mechanism can be very difficult. Unless extensive sensor measurements of the environment are available and presented to the operator in an effective and timely fashion, it can be very easy for the operator to become disoriented with respect to the remote environment, which can lead to operator error. For example, Marino [3] describes JPL's predictive

ARK-lite robot and it's $2\frac{1}{2}D$ user interface which displays a 2D map of the robot's environment. ARK-1 and the ARK-2 have similar interfaces.



(a) $2\frac{1}{2}D$ user interface *(b) ARK-lite mobile robot*

Figure 1: ARK-lite user interface and platform

3D graphics display for the remote control of a teleoperational vehicle. In their system the operator uses a control joystick while viewing a bank of monitors showing real and simulated views of the robot. Their system can present a rendered 3D graphics overlay with live video obtained from the remote vehicle. Although [3] presents only a 'robot perspective' view of the environment, they report that "Advanced Teleoperations provides an enrichment of operation remote-control dexterity, 3-D viewing and remote information display"[1].

In order to construct an advanced teleoperational interface for a mobile robot, it is necessary that the interface be consistent, integrated, and natural to use. Mechanisms which rely on a large bank of monitors, with complex user interactions cannot be expected to provide a natural input mechanism. One technology which can be exploited to provide a more natural interaction mechanism is an immersive display or virtual reality technology. Immersive displays have been used to assist in teaching driving[4], treating fear of heights[5], and to provide advanced scientific visualisation tools[6], among others. By providing the operator of these interfaces with the sense of presence of being located in some other remote environment, immersive displays provide a more engaging display technology than is possible with banks of monitors. A system that uses the concepts of a system similar to [3] but which utilises immersive display technology may provide a more effective user interface for mobile robot control than is possible by either a single screen or non-immersive teleoperational console.

[1] Marino:93 page 6.

2 Presenting information on an immersive display

For an immersive display to provide an effective mechanism for control of a mobile robot, the interface must do at least two things; it must provide the operator with a useful representation of the robot's operational environment, and it must provide suitable interaction mechanisms for robotic or teleoperational control. For the immersive environment to provide a useful representation of the robot's operational environment, the operator should be able to view, and navigate through, the environment. For the entire interface to provide interaction, some mechanism for operator input beyond that required for the immersive display must be provided.

The ARK-lite immersive interface is based around a head mounted display (HMD) and a six degree of freedom joystick. Video is displayed on a Liquid Image HMD which also provides stereo sound to the operator. Six degree of freedom (DOF) head tracking is accomplished via a Flock of Birds head tracker. The operator is also equipped with a six DOF Cyberman three button joystick to provide additional input control. Video is generated by an R8000 SGI Indigo^2 workstation with the Extreme graphics option.

A fundamental question in the design of an immersive interface for a mobile robot is how to manage the display of both the immersive visual display as well as any visual tokens which must be displayed as part of the interaction mechanism. The display portion of a head mounted display can be considered as a simple flat display surface, but interaction mechanisms which are appropriate on "flat" monitors are unlikely to be well suited for head mounted displays.

Although the display surface of the Liquid Image

HMD does subtend a relatively large visual angle, its actual display surface is quite small. With a visual field 640x480 pixels in size, there is not much physical screen real estate to reserve for any graphics required for interaction. In addition, due to the magnification optics built into the HMD, it is only possible to read the center of the screen without strain.

Given the need for graphical displays not related to the immersive display, limited screen real estate, and the fact that the best view is in the center of the screen, a user interface is required that is in some sense foveal. Thus the ARK-lite immersive display introduces a *fish bowl* metaphor for the control and manipulation of graphical objects.

Menu, Map and Video interaction objects are displayed over the immersive background.

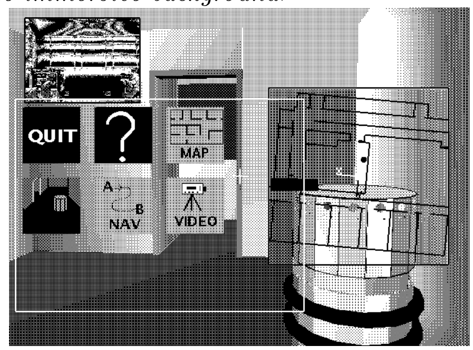


Figure 2: Interaction Objects in the Display

Fish Bowl Metaphor The fish bowl metaphor is an extension of the desk-top metaphor common in 2D graphical user interfaces. Imagine being a fish in a fish bowl. Looking out through the walls of the fish bowl you can view the environment within which your bowl sits. The external world outside the fish bowl projects onto the bowl's exterior surface. The interior surface of the bowl completely surrounds the operator providing 360^o of desktop surface. Semi-transparent and opaque 2D graphical objects can be placed on the surface of the bowl. Interaction mechanisms are provided so that the operator can:

- Translate the operator and the bowl through the external environment.
- Rotate inside the bowl to view out through different portions of the bowl. This is known as the pan model of operation.
- Rotate with the bowl so that the objects on the surface of the bowl obscure different regions of the external environment. This is known as the fixed model of operation.
- Select objects on the surface of the bowl and
  - Move them to other locations on the surface of the bowl, including placing them on top of other objects on the surface of the bowl.
  - Dispose of them.
  - Resize them.

As the operator's field of view is limited, only a portion of the fish bowl is visible at any one time.

Input Focus In order to select different graphical objects on the fish bowl for input focus, the operator simply rotates until that object is in the center of view. i.e., the operator simply looks straight at the object of interest. A cross-hair is always displayed in the center of the display to aid the operator determine which interaction object is currently receiving input focus.

Interaction Objects The current implementation of the immersive display supports three different types of interaction objects:

Menu A collection of icons which correspond to particular actions. Due to the difficulty of reading text in the HMD the menu items are limited to large, simple icons. Menus are arranged in a hierarchical structure.

Map A 2D map of the robot's environment.

Video The current view obtained with the robot's onboard video camera.

Figure 2 shows the user interface with each of the interaction objects displayed. The background shows the robot in it's environment, while Menu, Map and Video interaction objects are displayed on the inside of the 'fish bowl'.

Self motion and joystick input The movements of the user, both head rotation and body translation, are used to modify the display of the world. The user may also manipulate a 6 DOF joystick, acting as if it were positioned at the center of the user's head, to add to their physical motion. This allows a wider range of movements through the virtual world than would be possible if the user had to physically move about to change their viewing angle and position.

3 Robot control through the immersive display

The display has two fundamental methods of providing interactions with the robot; robotic and teleoperational.

Robotic Under robotic control, the operator commands the robot to carry out long term, complex plans. In this case the primary function of the immersive interface is to provide the operator mechanisms to examine the execution of plans and to aid the operator choose global operations for the robot. Three different interaction modes are available:

- **walk** In this mode the virtual motion of the operator is limited to the environmental ground plane.
- **ride** In this mode the operator is slaved to the current position of the robot. Rotational changes in the operator's pose permits the operator to view in different directions.
- **fly** In this mode the virtual motion of the operator is unlimited.

Teleoperational This mode lets the user drive the robot. Using a control mechanism similar to that used in the *walk* robotic control, the robot is commanded to move. The operator's virtual viewpoint is slaved to the robot as in the *ride* mode above, so if the operator causes the robot to move forward, the operator will appear to move forward in the immersive display.

4 Conclusion

When designing the icons to use in the menu planes, considerable experimentation was required in order to provide readable and intuitive icons. Certain colour combinations become unreadable when viewed on the head mounted display rather than on a high resolution monitor, and the text must be large enough to be legible on the relatively low resolution display in the helmet. The various commands for navigating through the menus were limited by the finite number of combinations of mouse button presses. Since the user interface requires drag and drop characteristics, as well as requiring input facilities for entering and leaving menu levels, complex mouse button combinations are required for some tasks. Mouse button combinations were kept as simple as possible for the most often used commands.

Various mechanisms were investigated for specifying locations on the map. Initially locations were specified by the current location of the operator in the environment, but this was found to be awkward since positioning yourself virtually is harder than merely looking at where you wish to set the goal. Currently, the goal can be set either at the operator's position, or at where the operator is looking.

This method of controlling mobile robots can be applied to other robotic platforms built to perform various functions. It will allow the operator to have much greater control over the robot since they will be able to visualise more clearly the operational environment.

Acknowledgements
Funding for this work was provided, in part, by the ARK (Autonomous Robot for a Known environment) Project, which receives its funding from PRECARN Associates Inc., the Department of Industry, Science and Technology Canada, the National Research Council of Canada, Technology Ontario, Ontario Hydro, and Atomic Energy of Canada Limited. Financial support from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

References

[1] M. Jenkin, N. Bains, J. Bruce, T. Campbell, B. Down, P. Jasiobedzki, A. Jepson, B. Majarais, E. Milios, J. Service, D. Terzopoulos, J. Tsotsos, and D. Wilkes. ARK: Autonomous mobile robot for an industrial environment. In *IEEE/RA IROS*, Munich, 1994.

[2] E. Byler, W. Chun, W. Hoff, and D. Layne. "Autonomous hazardous waste drum inspection vehicle". *IEEE Robotics and Automation Magazine*, vol. 2(1):pp. 6–17, 1995.

[3] K. Marino. "JPL teleoperations system debuts in cross-country test". *Newsletter of the IEEE Robotics and Automation Society*, vol. 7(4):pp. 5–6, 1993.

[4] J. Kuhl, D. Evans, Y. Papelis, R. Romano, and G. Watson. "The Iowa driving simulator: an immersive research environment". *IEEE Computer*, vol. 28(7):pp. 35–41, 1995.

[5] L. Hodges, R. Kooper, T. Meyer, B. Rothbaum, D. Opdyke, J. de Graaff, J. Williford, and M. North. "Virtual environments for treating the fear of heights". *IEEE Computer*, vol. 28(7):pp. 27–34, 1995.

[6] W. Kruger, C.-A. Bohn, B. Frohlich, H. Schuth, W. Strauss, and G. Wesche. "The Responsive Workbench: A Virtual Work Environment". *IEEE Computer*, vol. 28(7):pp. 42–48, 1995.