

Web-Based Mobile Robot Simulator

Dan Stormont

Utah State University
9590 Old Main Hill
Logan UT 84322-9590
stormont@hass.usu.edu

Many roboticists rely on simulation during the early phases of developing navigation algorithms for their autonomous mobile robots. While many commercial robots now come with robust development environments that include visual simulators, these tools aren't available to robotics researchers who are working with a custom-built robot or who have not yet determined which commercial robot will satisfy their requirements. Even researchers using one of the commercial development environments will have difficulty sharing their simulation results with colleagues or others who do not have access to the commercial development tools. Believing that the ability to share simulation results visually with the greatest number of people would be an important capability to have led to the development of the web-based simulation approach described in this paper.

First Generation

The "first generation" of this web-based simulation began with a text file full of waypoints. The text file was being generated by a probability grid-based navigation scheme being investigated for implementation on the mobile robot LOBOTomous at the University of New Mexico. Realizing that this cryptic output file would be incomprehensible to most viewers motivated the initial experimentation with a web-based simulation.

The most important consideration in this initial experiment was making the simulation results accessible to the largest possible audience. This motivated the decision to use the World Wide Web to distribute the visual results and the search for a format that would be platform (and web browser) independent. A number of different visualization tools were evaluated, including Chrome, Live 3D, QuickTime VR, Real-Time Authoring for Virtual Environments (RAVE), and the Virtual Reality Modeling Language (VRML). Of these, only VRML was an open standard that was supported on a large number of platforms with a number of freely distributed VRML browsers and plug-ins. Also, with the release of VRML 2.0, a number of the shortcomings in VRML that had prevented it from

being used for anything more than static visualization of three dimensional objects were addressed. It was now possible to have an animated object move through a virtual environment, while providing the user with a nearly limitless number of viewpoints. Additional flexibility was provided by the ability to run active elements (like Java applets) from the VRML browser.

Having selected the tool to be used, the next step was to build a VRML model of the robot to be simulated, in this case, the robot LOBOTomous. LOBOTomous is a custom built robot with two driving wheels and two castors centered around a central pivot point. The wider base contains the motors and drive circuitry, while the narrower cylindrical body contains the PC-104 control computer, the ultrasonic sensors and their circuitry, in addition to any circuitry added to increase LOBOTomous' capabilities. The VRML model was built to have the minimal detail while still being recognizable as LOBOTomous. A screen shot of the LOBOTomous model is shown in Figure 1.



Figure 1. VRML model of the robot LOBOTomous

Figure 2 shows LOBOtominus in a simulated lab environment. The lab environment is very simplistic, since the furniture is not detailed (the lab benches and other obstacles are nothing more than solid blocks in the model), but the model does allow the visualization of the robot's motion through this environment from starting point to goal.

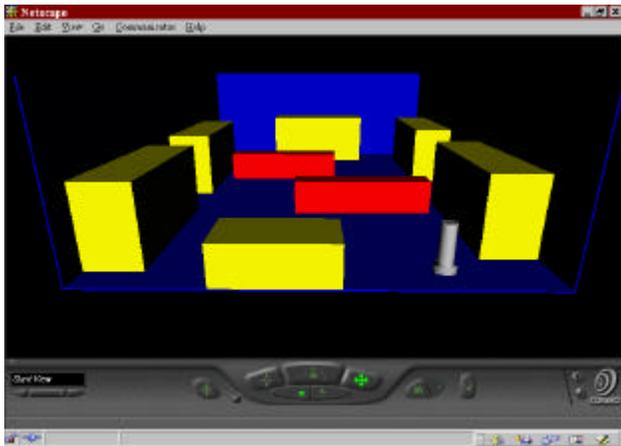


Figure 2. VRML model of lab.

The drawback to the first implementation is that it is not very flexible. The animation of the robot's path through the lab was hand coded in VRML using the text output of a simulated run of the robot through the lab. Changes in the algorithm, environment, or even start and goal positions requires recoding the animation. This lack of flexibility motivated the next generation of the VRML simulation.

Next Generation

As stated earlier, VRML 2.0 greatly enhanced the capabilities of VRML as a modeling tool. Events can be triggered by sensing actions, such as movement of the mouse, a mouse click, or collision with an object. Additionally, the ability to incorporate active code (such as a Java applet) opens the possibility of running the navigation algorithm within the VRML viewer. These untapped capabilities of VRML are the basis for the ongoing work with the VRML simulator.

Specifically, some of the potential enhancements being worked on now will provide the user of the simulation with the following capabilities:

- Use the mouse to select the start and goal locations;
- Navigate through a more realistic environment, with furniture that is recognizable as furniture and possibly varying sensor effects based on the furniture style;
- Select the navigation algorithm to be used from a list of Java applets;
- Select the type of robot simulated, with the appropriate motion model for the drive wheel configuration; and

- Build a map to be navigated through by providing a grid full of cubes and allowing the user to make selected cubes "disappear" by clicking on them with the mouse. As enhancements become available, they will be posted on the Utah State University web server at <http://www.usu.edu/~afrotc/cadre/stormont/vrml.html>. The first generation of simulations are already available on the web site.