Interactive Simulation of the NIST USAR Arenas

Jijun Wang
Department of Information Sciences
University of Pittsburgh
Pittsburgh, USA
jw1@pitt.edu

Michael Lewis
Department of Information Sciences
University of Pittsburgh
Pittsburgh, USA
ml@sis.pitt.edu

Jeffrey S. Gennari
Department of Information Sciences
University of Pittsburgh
Pittsburgh, USA
jgennari@sis.pitt.edu

Abstract - *We are developing interactive simulations of the National Institute of Standards and Technology (NIST) Reference Test Facility for Autonomous Mobile Robots (Urban Search and Rescue). The NIST USAR Test Facility is a standardized disaster environment consisting of three scenarios of progressive difficulty: Yellow, Orange, and Red arenas. The USAR task focuses on robot behaviors, and physical interaction with standardized but disorderly rubble filled environments. The simulation will be used to test and evaluate designs for teleoperation interfaces and robot sensing and cooperation that will subsequently be incorporated into experimental robots. This paper describes our novel simulation approach using an inexpensive game engine to rapidly construct a visually and dynamically accurate simulation for both individual robots and robot teams.*

Keywords: Teleoperation, Robotics, Urban Search and Rescue, Simulation, Reference Tasks.

1 Introduction

Following city-wide disasters, such as the Kobe earthquake in Japan, urban search and rescue (USAR) has generated increasing interest in the field of mobile robotics. USAR tasks range from the physical domain such as navigation and locomotion to perceptual one such as target recognition and the design of teleoperation strategies. Researchers at NIST responded to this trend by developing a standardized disaster environment to serve as a reference domain for common tasks associated with USAR [5]. NIST's USAR facility consists of three disaster zones of progressive difficulty. These regions were designed to be easily replicable at competitions and research facilities across the globe. However, they still require time, money, space, and some building expertise to construct. In addition, fabricating the robots to traverse the arenas is a daunting task requiring time and expertise which may yield ill results when placed in a USAR environment. To bypass these limitations we have developed a simulation of one of the disaster zones known as the Orange Arena. The simulation will serve as a tool to allow USAR researchers to evaluate their designs before implementing them. Using Epic Games Unreal Tournament technology as the backbone of our simulation we have produced a virtual environment complete with realistic dynamics and high quality graphics in a remarkably short amount of time, with little cost, and with no specialized equipment.

In this paper we describe our simulation of the National Institute of Standards and Technology's Orange Arena created using the Unreal Tournament computer game. We will first introduce NIST's reference arenas that serve as testbeds to evaluate robots designed for USAR environments. Next, we will discuss the advantages of using commercial video game technology for simulations as well as the challenges and results of implementing this technology for our simulation. We will then explore the processes involved with simulating robots to traverse the simulated arena. Finally, we will discuss initial findings from research conducted to explore various control strategies for robotic teleoperation.

2 The NIST Arenas

Unveiled at the 2000 AAAI Mobile Robot Competition, the USAR facility consists of three arenas of progressive difficulty: the yellow arena, the orange arena, and the red arena.

2.1 Yellow arena

The yellow arena is the simplest of the arenas. It is composed of a large flat floor with perpendicular walls and moderately difficult obstacles. The challenges in the yellow arena are largely perceptual and designed to disrupt an operator's perception and a robot's sensors. For instance, mirrors, transparent Lucite walls, venetian blinds, and large areas completely darkened by tarps (Figure 1 rear) are present throughout the arena.

2.2 Orange arena

The orange arena is a bi-level arena with more challenging physical obstacles such as stairs and a debris littered ramp. The floor is covered with debris including paper, pipes, and cinder blocks. As in the yellow arena, the walls are textured with confusing patterns such as stripes and mirrors. Finally, there are large gaps in the flooring on the second level that are difficult to detect (Figure 1 front).

2.3 Red arena

As the most challenging of the arenas, the red arena, presents fewer perceptual difficulties but places maximal demand on locomotion. The design mimics an actual rubble pile with mounds of cement blocks and slabs, chicken wire, and other debris.

NIST designed these arenas to be portable, modular, and replicable in an attempt to make them simple to reconstruct by researchers and organizers of robotic competitions around the globe. However, the resources and space required to erect the arenas make them unfeasible for many researchers. In order to overcome these pitfalls we are developing computer-based simulations of the obstacles found in the USAR environment, and the kinematics of robots that traverse it.

3 Game-Based Simulations using Unreal Tournament 2003

Real time "out the window" or "through the camera" simulations have classically been difficult, time consuming, and expensive to build requiring specialized hardware and personnel. Computer game engines, the underlying simulations on which computer games are built, offer an inexpensive and high quality solution to these problems [7]. Aside from affordability, today's game engines offer advanced graphical displays, realistic environmental, physical dynamics, and dramatically reduced development times.

In the past a great deal of energy has focused on robotic sensing, navigation, map building, and field experience. However, field experimentation [1,8] suggests that perceptual issues, such as situational awareness, may be equally if not more important. The simulation we are developing can be used to address both of these issues, but thus far we are focusing on the perceptual problems associated with direct human teleoperation including effectively relaying spatial information to operators, such as robot's attitude, etc. Other areas that can be addressed with our simulation include studying strategies for autonomous robot behavior and intermediate levels of adjustable autonomy.

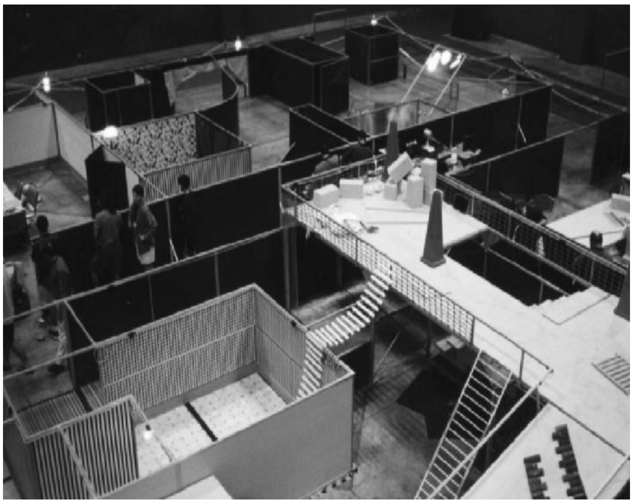

Figure 1 Orange Arena (Front) and Yellow Arena (back)

Our simulation of NIST's Orange Arena is built using Epic Game's Unreal Tournament 2003 game engine. Unreal Tournament 2003 (UT2003), the game we run to access it's engine, is a networked first person (egocentric) shooter. UT2003's client/server model allows many clients to interact in a common environment via a central server. Clients are represented in the game as programmatically controlled entities known as "bots". Users can control a bot's behavior remotely through the Gamebots API which provides a TCP/IP socket interface from the server to a bot [6]. The Gamebots API contains a set of procedures and commands that allow a bot to be remotely controlled and allow access to data associated with the bot such as location information. In addition, UT2003 allows a client to attach a remote camera to a bot as a spectator. A spectating client can see the simulated environment through the viewpoint of the bot. By fusing data returned from the bot via Gamebots, such as location and attitude data, with the video feed of a spectating client, a rich simulation platform is obtained.

4 Simulating the Orange Arena

In November of 2002 a team of ten researchers visited the NIST USAR facility to collect data needed to construct the simulation and identify design requirements necessary for robots to operate effectively in these environments. The data included measurements of physical dimensions, light levels, first hand observations of obstacles, and wall textures. The information gathered was then used in conjunction with digital photographs taken on site and CAD models provided by NIST to tease out design requirements for robots and to develop a simulation. The CAD models, originally created using ProEngineer software, were transformed into a simplified format suitable for animation using Nugraf and exported to 3D Studio Max for editing and detailing. The simplified images were then re-exported into Unreal Tournament format and used to create the arena. In addition, a series of digital photos had been taken at the NIST facility. These photos were entered into Unreal's image editor and used to texture the walls and obstacles in the simulation. Lighting levels in the simulations were adjusted to generate a disparity between open areas and shadows with a ratio of roughly 15:1. Finally, special features such as rubble, pipes, orange netting, mirrors, and victims were created and added to the simulation.

Figure 2 Simulated Orange Arena in UT2003

In order to create a believable simulation we had to recreate not only the physical appearances of objects within the arena, but also the behavior of those objects in the simulation. For example, human remains are placed in the actual arena as targets for a robot to discover. These victims may produce subtle and obvious movements to serve as beacons for a robot to follow (Figure 3). In the simulation victims are built from skeletal models provided by the game. Based on the victims found in the actual arena, the simulated victims (Figure 4) boast realistic physical appearances and dynamics. In the future we hope to extend the reconfigurability of the arena to allow experimenters to quickly move walls, redistribute the victims, and generate new and more challenging obstacles and scenarios.

Dynamic objects such as victims possess different properties than a static object such as a wall or ramp. For this reason objects within the simulation are separated into three classes based the interactions they have with the a robot and it's sensors. The classes of objects we developed include:

- Static geometric objects: Objects that are part of the core simulation, immutable, and unmovable, including wall and floor textures. Static geometric objects may affect a robot, for instance, by reducing wheel traction thus hindering locomotion.

- Dynamic geometric objects: Objects that can move and change states including bricks, pipes, and victims. In addition, these objects can have more complex interactions with a robot such as changing a robot's attitude.

- Environmental objects: Objects that are intangible parts of the arena including lighting and sound effects. Though environmental objects do not have a direct effect on a robot, they may hinder an operator's ability to identify objects of interest.

To create a useful simulation we also had to duplicate the laws of physics and robot dynamics. Fortunately, Unreal Tournament 2003 incorporates MathEngine's Karma Physics Engine which simulates rigid multibody dynamics. Animation level speed is maintained without sacrificing physical fidelity through algorithms that exclude non-interacting objects. As in the real world, each simulated object has its own physical properties that determine how that object interacts with its environment. These properties include mass, inertia, and friction. In addition, the simulated world itself has properties, such as gravity, that dictate how objects interact [11].

Figure 3 Victim in Actual NIST Arena

Figure 4 Victim in Simulated NIST Arena

5 Simulating Robots

The simulated robots used to traverse the arena were created by extending Unreal Tournament's vehicle class. A vehicle is essentially a chassis with one or more wheels attached (Figure 5) that posses physical properties that govern how the robot interacts with it's environment.

Figure 5 Simulated Robot

These properties include mass, a center of gravity, engine power, and wheel friction and are configurable by adjusting values assigned during development. The simulated robot modeled an actual robot being constructed by members of our team for from Carnegie Mellon University. The simulated robot's velocity, acceleration, and steering are all based on observations and measurements taken from actual robot as it was being constructed. Finally, the frame rate of the simulated video feed has been set to a sluggish one and a half frames/second to match the actual robots camera speed.

In addition, an array of sensors will soon be available to researchers using the simulation. The Unreal engine has the capability of imitating many different types of sensors by corrupting ranging data available to "bots" to reflect the characteristics of the sensor being modeled. Potential sensors can include proximity sensors such as sonar, laser based sensors such as laser radar (LADAR), and thermal sensors such as forward looking infrared (FLIR). Initial work has been done to determine the feasibility and effort needed to simulate each type of sensor [11], but the work is still preliminary and a more extensive exploration is needed. The final vision for our simulation is to provide a tool for rapid prototyping where robot schematics can be used to easily fabricate a simulated robot with comparable dynamics and physical dimensions without ever having to solder a wire or purchase any physical equipment.

This vision is within our grasp thanks to the modular design of the Unreal Tournament game engine. New levels, characters, textures, and vehicles can be rapidly developed and implemented. In fact, our simulation of the orange arena was completed in less than two months and has already been used to evaluate teleoperation strategies and appropriate levels of robot autonomy in a USAR context. By harnessing UT2003's video feed we have been able to create a realistic teleoperation interface (Figure 6) that is the same interface we are using to control an actual robot. The simulation incorporates the same controls, FOV, and camera control. Features found in the simulated interface include headlight controls, low-bandwidth video (1.5 frame/second), an attitude indicator, a battery level indicator, and a communication link level indicator. Synchronization between the interface and the state of the simulation is assured because feedback from the game engine is transmitted in the order of one packet per tenth of a second, but even this is adjustable to accommodate limited bandwidth of communication channels.

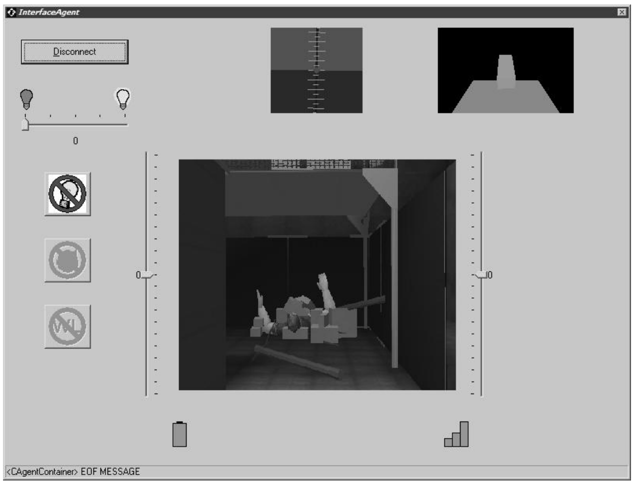

Figure 6 Simulated Interface for Teleoperation

Using this simulated interface USAR researchers were able to evaluate the advantages and pitfalls of the interface's design before the completion and testing of the actual robot.

While our initial simulation was limited to a single teleoperated robot the foundation for experimenting with adjustable levels of autonomy and multiple robots has been laid by making use of the flexible client-server architecture of UT2003 in conjunction with intelligent agent technology. The control chain of operator-RETSINA agent-Gamebots-video feedback (Figure 7) replicates the control chain of operator-RETSINA (on Intel Staytem board)-robot-video feedback of the experimental robots the Carnegie Mellon team is developing.

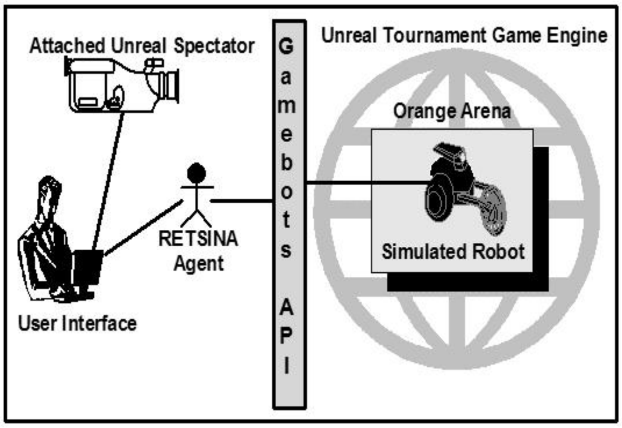

Figure 7 Architecture of the current USAR Simulation

6 Initial Findings from Using the Simulation

Experimentation using the USAR simulation will initially be focused on control and aiding strategies affecting teleoperation of mobile robots. Telerobotic control has continued to prove a difficult task for a variety of reasons. For instance, limiting an operator's field of view can cause objects of interest to go unnoticed [1,3,10] yet when given full control over five degrees of freedom (2 position, 3 view) operators can become disoriented and survey knowledge may suffer [3]. Perceptual problems such as these are complicated when combined with rugged and deceiving terrains frequently encountered in USAR environments. Operators can easily become lost [8], disoriented, or perceptually confused [1]. In fact, one of Murphy's conclusions from experience of operating robots at the World Trade Center site was that the greatest difficulty in USAR involved in perception and identifying victims rather than locomotion or navigation [9]. Our initial studies will concentrate on these perceptual problems with a pilot experiment already in progress comparing a suite of commercially available 5 DOF controllers at a treasure hunt [4] task. In subsequent studies we plan to investigate strategies for aiding through conditional viewpoint/motion coupling, scripted viewpoint control [10] for dealing with negative obstacles (holes), inclines, abrupt changes in direction, and pattern recognition driven attention direction for locating victims.

Another issue of immediate interest for operations in a rubble filled USAR environment is attitude control, particularly managing a robot's roll. McGovern [8] for instance, reports that all ten "inside-out" teleoperation accidents observed at Sandia National Laboratory involved rollovers (usually on grade). During debriefing, operators expressed surprise and disbelief as they were unaware they were in danger of rolling over. In a USAR environment where normal operating assumptions such as that the surface on which the robot rests is horizontal may frequently be violated, effective methods for displaying attitude will need to be developed. Using our simulation and other environments developed with Unreal Tournament 2003 we have begun experiments to determine effective ways to convey a vehicle's attitude to a remote operator. The Experiments are still in the initial phases, but results are promising. Using Unreal's game engine technology we have been able to quickly develop both outdoor and indoor environments, control mechanisms, and interfaces for these experiments.

7 Conclusion

We have developed a simulation of the NIST's orange arena using commercial video game technology. Using Unreal Tournament's game engine and the Gamebots API we have created a simulation that provides researchers with a high quality affordable testbed to evaluate the design and performance of remotely operated robots in urban search and rescue environments. Though our work has initially been focused on developing control strategies for teleoperation, we hope to extend our research to more complicated issues including cooperation robot teams for USAR and developing strategies for autonomous control. We hope this extendable testbed will allow other researchers to develop, set up, and carry out experiments to more rapidly advance robot design and control for urban search and rescue.

Acknowledgements

This project is supported by NSF grant NSF-ITR-0205526. Katia Sycara and Illah Nourkbaksh of Carnegie Mellon University are co-pi's on the project. Mary Berna, Alexander Gutierrez, Terence Keegan, Kevin Oishi, Binoy Shah, Steven Shamlian, Mark Yong, and Josh Young of Carnegie Mellon University assisted in data collection.

References

[1] Casper, J., Human-Robot Interactions during the Robot-Assisted Urban Search and Rescue Response at the World Trade Center, MS Thesis, Computer Science and Engineering, USF, Apr. 2002.

[2] Gamebots 2003. Network API. http://www.cs.cmu.edu/galk/GameBots/WEB/docapi.html .

[3] Hughes, S. and M. Lewis, Directing Attention in Open Scenes. *Proceedings of the Human Factors and Ergonomics Society's 46th Annual Meeting*, Baltimore MD., 2002.

[4] Jacobson, J. & Lewis, M., An Experimental comparison of three methods for collision handling in virtual environments, in *Proceedings of the 41st Annual Meeting of the Human Factors and Ergonomics Society*, Sept. 22-26, Albuquerque, NM, 1997.

[5] Jacoff, A., Messina, E., Evans, J., Experiences in deploying test arenas for autonomous mobile robots, Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS) Workshop, Mexico City, Mexico, 2001.

[6] Kaminka, G., Veloso, M., Schaffer, S., Sollitto, C., Adobbati, R., Marshall, A., Scholer, A., and Tejada, S., GameBots:A flexible test bed for multiagent team research, Communications of the Association for Computing Machinery(CACM),NY:ACM45(1),43-45, 2002..

[7] Lewis, M. & Jacobson, J., Game Engines in Research. *Communications of the Association for Computing Machinery* (CACM), NY: ACM 45(1), 27-48, 2002.

[8] McGovern, D., Teleoperation of land vehicles. In Stephen Ellis (Ed.) *Pictorial Communications in Virtual and Real Environments*. New York: Taylor and Francis, 182-195, 1991.

[9] Murphy, R., Gaps in Rescue Robotics, presentation to *IEEE Workshop on Safety, Security, and Rescue Robotics*, USF, Tampa, FL, Feb 19, 2003.

[10] Tittle, J., Roesler, A., & Woods, D., The Remote perception problem, *Proceedings of the HFES 46th Annual Meeting*, Baltimore, MD, 260-264, 2002.

[11] Wang, J., Lewis, M., and Gennari, J., A game engine based simulation of the NIST urban search & rescue arenas, to appear in *Proceedings of the 2003 Winter Simulation Conference*, Dec 7-10, New Orleans, LA, 2003.