Virtual Environments for Developing Strategies for Interdicting Terrorists Carrying Dirty Bombs

Min-Hao Matt Wu
California Institute of Technology
mattwu@caltech.edu

Annie Hsin-Wen Liu
California Institute of Technology
aliu@cs.caltech.edu

K. Mani Chandy
California Institute of Technology
mani@cs.caltech.edu

ABSTRACT

Strategies for detecting terrorists carrying radioactive material can be evaluated in virtual environments more easily than they can be in the real world. Real scenarios expose personnel to radiation and concomitant dangers. The execution of multiple real-world scenarios – such as catching terrorists in factories, houses and open spaces – is expensive. This paper describes virtual environments for interdicting terrorists carrying radioactive material. The virtual environments are constructed by incorporating the physics of radiation into virtual-world platforms. We explore the relative advantages of a gaming engine (*Half-Life 2*), a 3D online virtual world (*Second Life*) and a robot simulator platform (*Stage/Player*) for developing strategies for interdicting dirty bombers. Preliminary results on implementations of these virtual environments are presented.

Keywords

Detection, interdiction, radioactive, terrorist, dirty bomb, online game, Poisson process, ray tracing, virtual worlds

INTRODUCTION

An explosion of a dirty bomb is a crisis. Federal, state and local governments of many nations cooperate in intercepting radioactive material; however, the possibility that terrorists will succeed in acquiring small amounts of radioactive material cannot be ruled out completely. Therefore, governments are studying procedures to interdict sources of radiation in buildings and in open areas such as arenas where political rallies and sports events are held.

This paper describes an ongoing project to build virtual environments to help develop strategies to interdict terrorists on foot, carrying radioactive material in backpacks or bags, within buildings or in the open. A secondary goal is to evaluate procedures that alert scientists and medical personnel accidentally moving radioactive material from appropriate to inappropriate locations.

Terrorists carrying dirty bombs may follow several strategies including shielding radioactive material with lead, hiding behind metal walls in buildings, and staying near benign radioactive sources such as medical patients. Our project builds virtual environments in which teams playing roles of terrorists and security personnel compete against each other. An initial evaluation of strategies is presented in Chandy, Pilotto and McLean 2007.

TECHNICAL CHALLENGES

Multiple Goals: Virtual environments that aid in dirty-bomb detection have three main goals:

1. Reflect reality with high fidelity for people playing security officers and terrorists to help them develop strategies and intuition based on realistic environments.
2. Enable collaboration among teams of humans, software agents and robots. Security personnel located in different places and possibly different countries should be able to collaborate in developing strategies.
3. Provide scientific visualization to help scientists develop strategies using mathematical models and computer programs to represent humans, robots and software agents.

We call these goals the *realism, collaboration* and *math goals* respectively. Next we identify requirements for virtual environments to meet these goals.

Realistic Representation of the Environment: The realism goal requires accurate representation of the environment including audio and video depiction of features such as buildings, doors, and backpacks. For example, the posture and gait of a person in the virtual world carrying an extraordinarily heavy backpack should reflect its weight. The math goal doesn't require the same degree of fidelity; for example, doors do not need to squeak when they are opened to achieve this goal. All three goals do require accurate representation of gross environmental features such as fences and baggage screening areas, as well as sources of background radiation such as brick walls and medical patients who have ingested radioactive compounds.

Representation of Sources of Radiation, Sensors and Agents: Terrorists may have access to different elements and isotopes such as Cesium-137 and Cobalt-60. The amounts and types of radioactive material that can cause damage vary over a wide range. The intensity of radiation visible to sensors can be reduced by wrapping radioactive material in lead or other material. Photons may get absorbed or refracted by walls and other obstacles. A wide range of sensors for detecting radiation, including prototypes for intelligent personal radioactive locators (IPRL) have been developed; these sensors vary in accuracy, directionality and ability to distinguish isotopes.

Resources for interdicting terrorists include security officers on foot; sensors in vehicles such as police cars; mobile sensors in UAVs and other unmanned vehicles; static sensors attached to street lamps, doors and fences; and surveillance systems based on cameras and motion detectors. Terrorists may use a variety of mechanisms for transporting material in an open area. All three goals require that the virtual environment represent a range of strategies that terrorists and security personnel may employ. The math goal also requires representation, analysis and support for designs of devices that don't exist today but may exist in the future.

Global Virtual Environment: The interdiction of terrorists is carried out by collaborations of many agencies across the globe. Worldwide collaboration is often represented in a simplistic way, for example by assuming that the global environment determines a threat level (e.g. red or green) that influences resources used in a locale such as a sports field. The virtual world should enable collaboration across countries and reflect the global environment as worldwide collaboration among security agencies and also among terrorist cells becomes increasingly significant.

VIRTUAL ENVIRONMENT PLATFORMS

We are evaluating three, quite distinct, platforms for building virtual environments:

1. *Half-Life 2* (see Half-Life 2 SDK),
2. *Second Life* (see Second Life Reference) and
3. *Player/Stage* (Gerkey, Vaughan and Howard 2003)

Overview of Platforms

The *Half-Life 2* game engine is a platform for action games including *Half-Life 2* and *Counter-Strike*. Each player purchases a license and downloads the game engine onto the computers on which it is played.

Second Life is an Internet-based virtual world that allows users to interact with each other and with 3D environments through avatars that can move, see and communicate. Companies and experimenters can purchase land in the virtual world and build 3D environments on their land. Anybody with Internet access can become a *Second Life* participant without cost. Thus, people across the globe can collaborate easily and explore regions to which they have access.

Player is a server that interfaces to a variety of robots that may have different sensor and actuator functions. *Stage* simulates multiple robots moving in a 2D bitmapped environment controlled through *Player*. In addition, the USC Robotics Group has developed *Gazebo* for high-fidelity rendering and simulation of 3D environments.

Primary Strengths and Weaknesses of Platforms for Developing Strategies for Radiation Detection

Our initial experiments suggest that the *Half-Life 2* game engine offers the most realistic representation of the physics of radiation as well as audio and video. *Second Life* is best for collaboration and representation of global virtual worlds. *Player/Stage* is an ideal platform for developing mathematically based models and strategies for interdiction. No platform seems to dominate the others across all goals.

Most of our current experiments use *Half-Life 2*. The game is written in C++, and the source code can easily be modified to give desired functionalities. Furthermore, since the game is designed for high performance graphics, it fully utilizes graphics cards hardware acceleration and can generate millions of ray traces per second, making simulation of individual photons and simulating the photons' refraction/reflection a possible.

Second Life allows people to purchase 3D environments and avatars. The market for environments and avatars allows in *Second Life* enables rapid development of large-scale environments. For instance, security agencies in different countries could acquire and develop different regions in the 3D virtual world and then allow collaboration across regions free of charge. Aside from ease of access and collaboration, Second Life's limited API, however, makes our first goal – realism – difficult. For example, we have not been able to model photon absorption accurately in *Second Life* using their API. Second Life is satisfactory in an open field environment where absorption by non-metallic objects, such as trees, is less critical.

Player/Stage is the ideal platform for specifying movement and search strategies for agents programmatically. *Player/Stage* provides simulation drivers for a variety of sensors, making it the most realistic environment to encode strategies for autonomous agents. One limitation of *Stage* is that it provides only a 2D environment; however, we could use *Gazebo* to simulate in 3D. Also, direct human control over the robots is difficult in *Player/Stage*.

Detailed Analyses

Next, we discuss technical issues in incorporating radiation physics into the platforms.

Photons are generated by a radiation source in a random (Poisson) manner. Photons are emitted in every direction with equal probability. The time between successive photons emitted by the source depends on the type and amount of source material. The rate at which photons are generated is proportional to the amount of radioactive material. Thousands of photons may be generated per second by radioactive material in a backpack.

Photon Detection: A sensor is a slab of crystal connected to electronics. When a photon strikes the crystal the photon is recorded. The rate at which the sensor detects photons depends upon the orientation of the crystal slab and the distance of the crystal from the source. The crystals in sensors are of the order of centimeters whereas sensors are usually several meters or even hundreds of meters away from radioactive sources. Let μ be the rate at which the source generates photons and λ the rate at which photons are recorded by a sensor of area A meter-squared located r meters away from the source, and let θ be the angle of orientation between the (normal to the) sensor and the line joining the center of the sensor and the source. As a first approximation: $\lambda = \mu. A. \cos(\theta) / 4.\pi.r^2$ provided A is much smaller than r because the area of the surface of a sphere with radius r is $4.\pi.r^2$ and $A. \cos(\theta)$ is (approximately) the fraction of this surface occupied by the sensor. At each time step, we determine θ and r from the API provided by the virtual environment platform, and then we compute λ given (constant) source rate μ and sensor area A. We then calculate the number of photons to hit the sensor during that interval using Poisson's formula. This number is then registered on the sensor either as sound – as a number of ticks as in a Geiger counter – or visually on a graphical device. This calculation overestimates the photon-detection rate when the sensor is close to the source, but this error is acceptable because it occurs when the source has effectively been detected.

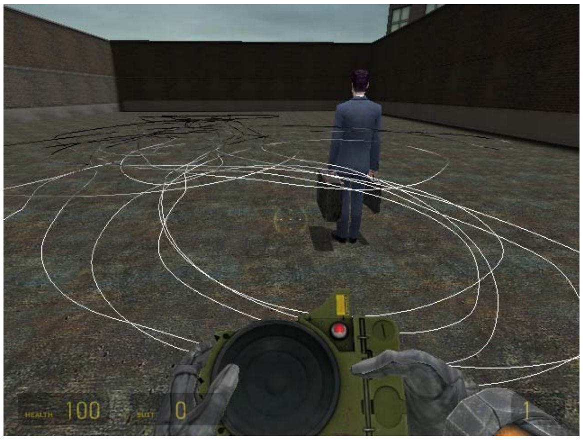
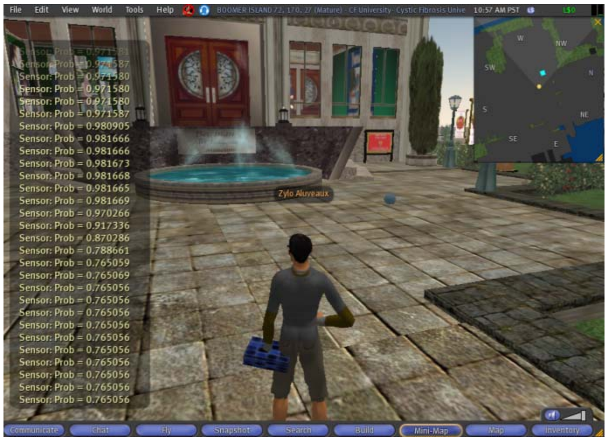


Figure 1. A *Half-Life 2* screenshot demonstrating the probability of receiving radiation decaying over distance. The detector traces of shades white to black correspond to high to low probability of receiving radiation at that segment by the detector.



```
Sensor: Prob = 0.971581
Sensor: Prob = 0.971587
Sensor: Prob = 0.971580
Sensor: Prob = 0.971580
Sensor: Prob = 0.971580
Sensor: Prob = 0.971587
Sensor: Prob = 0.980905
Sensor: Prob = 0.981666
Sensor: Prob = 0.981666
Sensor: Prob = 0.981673
Sensor: Prob = 0.981668
Sensor: Prob = 0.981665
Sensor: Prob = 0.981669
Sensor: Prob = 0.970266
Sensor: Prob = 0.917336
Sensor: Prob = 0.870286
Sensor: Prob = 0.788661
Sensor: Prob = 0.765059
Sensor: Prob = 0.765069
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
Sensor: Prob = 0.765056
```

Figure 2. A *Second Life* screenshot showing the decrease in hit probability as the detector (blue box) moves away from the radiation source (blue ball). The console on the left logs past probabilities of the detector receiving radiation.

Absorption of Photons by Walls: The probability that a photon will pass through a wall decreases exponentially at a rate that depends on wall thickness and wall material. Our system executes ray traces from the source to the center of the sensor, treating the radiation source as a point. The platform API is used to determine the material and thickness of obstacles along the ray. Our algorithm then calculates the probability of absorption. Thus, in our implementation using the *Half-Life 2* game engine, a terrorist is less "visible" to security officers with sensors on the opposite side of thick metal or even concrete walls. We had difficulty using the API in *Second Life* to determine the material and thickness of obstacles along rays; the API indicated obstacles but didn't provide enough information to distinguish material and thickness of obstacles. Thus, photon absorption is modeled less accurately in our implementations using *Second Life* than in our implementations based on *Half-Life*. Since the environment in *Player/Stage* is specified by the programmers it is straightforward to determine exactly what lies along any ray.

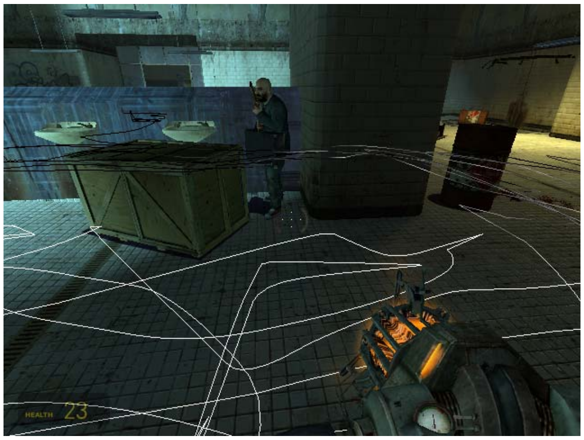


Figure 3. A *Half-Life* screenshot demonstrating the effects of absorption of photons by different materials. On the left side a wooden box blocks the source from the detector, on the right side a thick concrete wall, and in the back a lead wall. The effects of absorption due to different materials are demonstrated by the difference in intensities received behind different materials.

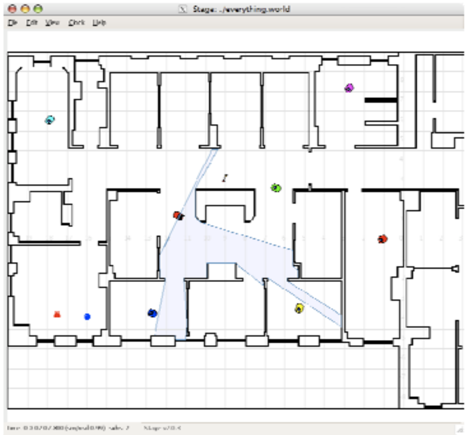


Figure 4. A *Player/Stage* screenshot of a roaming agent trying to find radiation sources behind walls.

Background Radiation: We estimate background radiation as follows. We generate rays from the sensor in different directions. We trace the rays to determine what material they contact. Let p_m be the fraction of rays from the sensor that strike material m first and let g_m be the rate per square meter of photons generated by a wall of material m. Then the total background radiation is: $\Sigma_m p_m.g_m.A$ where A is the area of the sensor. We generate rays from the sensor in proportion to the cosine of the angle from the normal to the sensor; thus, most rays are directed directly ahead of the sensor and none are directed parallel to the sensor. The formula follows by assuming that the intensity of photons striking a sensor r meters from a radiation source is inversely proportional to the square of r.

Medical Radiation: Radiation from a medical patient is modeled in the same way as radiation from a terrorist with a lower level of intensity. More accurate models, that we plan to implement later, will simulate using detectors that can distinguish the energy spectrum from isotopes used for medicine and those used in dirty bombs.

Agent Movement: The current implementations using the *Half-Life* game engine and *Second Life* allows agents to be moved by human players manipulating an input/output device such as a computer mouse. In *Half-Life*, agents are programmed to hill climb the radiation intensity map to locate radiation sources. We expect that programming agents in *Player/Stage* will be particularly easy though human manipulation may be difficult or impossible.

FURTHER WORK AND RELATED WORK

Our analysis and preliminary implementations show that an ideal platform for a radiation-interdiction virtual environment should offer features from all the *Half-Life 2* game engine, *Second Life* virtual world and *Player-Stage* simulation environment. Such a platform does not seem to exist. Therefore our plans are to continue exploration of different platforms for different purposes, and include the following steps. (1) Integrate stationary sensors on street lamps, sensors on mobile robots moving on a plane, sensors carried by security officers, and aerial robotic sensors into virtual worlds using *Half-Life 2* and *Player/Stage*. (2) Develop virtual environments representing different types of real environments including open spaces such as parks as well as commercial areas such as ports on different

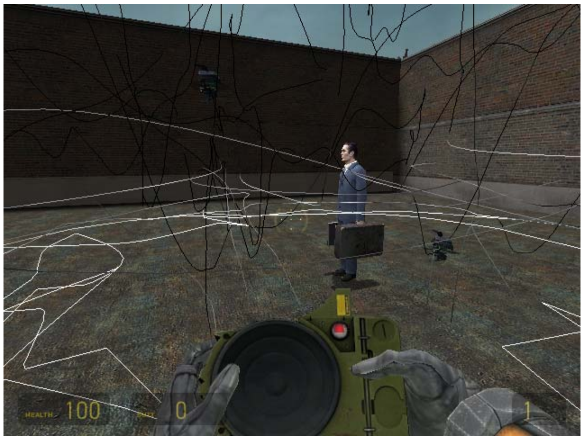


Figure 5. A *Half-Life 2* screenshot demonstrating resulting intensity traces of autonomous helicopter agents carrying radiation detectors in the 3D space.

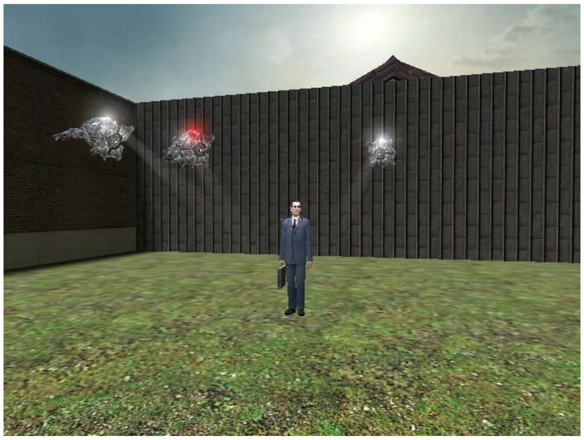


Figure 6. A *Half-Life 2* screenshot showing autonomous agents having identified the radiation source and taking photographs of the suspect.

platforms. This will help in evaluating the person-months required to develop virtual worlds that security agencies would like to use, and will also provide insight into strategies appropriate for different spaces. (3) Develop and prove optimum strategies for coordinating mobile and stationary sensors for simple situations such as stationary radioactive sources. This will help in understanding what can be proved mathematically for this type of problem. A great deal of work has been done on search strategies since World War II, and much of that deals with detecting submarines; (see McGee and Hedrick 2006, Santana, et. al 2005). Our work seeks to extend these strategies to deal with the specific threat of dirty bombers.

Counter-Strike: *Counter-Strike* is a First-Person-Shooter game that is loosely related to our problem. The game has two forces, the terrorists and the counter-terrorists. In one scenario of the game, the terrorists plant a timed bomb in a few possible locations. The objectives for the counter-terrorists are to eliminate the terrorists and defuse the bomb before it detonates. The difference between this game and our simulation is that the bomb in *Counter-Strike* is not radioactive, so they need to be detected by vision. (See Introduction to Counter-Strike)

ACKNOWLEDGMENTS

Suggestions and advice from Ryan Mclean, Concetta Pilotto, Joel Burdick, Jeremy Ma, Timothy Chung and Paul Upchurch are gratefully acknowledged. This research is supported by the California Institute of Technology Information Science and Technology Center and AFOSR MURI FA9550-06-1-0303.

REFERENCES

1. Chandy, K. M., Pilotto, C., and McLean, R. (2007) “Networked Sensing Systems for Detecting People Carrying Radioactive Material,” California Institute of Technology, Computer Science Technical Report and submitted Fifth International Conference on Networked Sensing Systems, 2008.
2. Gerkey, B. T., Vaughan, R. T., and Howard, A. (2003) The Player-Stage Project: Tools for Multi-Robot and Distributed Sensor Systems, *Proc. 11th International Conference on Robotics*.
3. Half-Life 2 SDK. (2005) http://developer.valvesoftware.com/wiki/SDK_Docs
4. ”Introduction to Counter-Strike.” (2006) http://whisper.ausgamers.com/wiki/index.php/Introduction_to_Counter-Strike#de_
5. McGee, T. G., and Hedrick, J. K. (2006) “Guaranteed Strategies to Search for Mobile Evaders in the Plane,” *Proc. 2006 American Control Conference*.
6. Santana, P., Barata J.,Cruz H., Mestre A., Lisboz J. and Flores L. (2005) A multi-robot system for landmine detection, *Proc. 10th IEEE International Conference on Emerging Technologies and Factory Automation.*
7. “Second Life Reference” (2003) http://oz.slinked.net/history/Detailed_Reference.pdf