Theses and Dissertations   1. Thesis and Dissertation Collection, all items

2006-09

A computer graphics based target detection model

Jones, Brian Edward.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/2628

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A COMPUTER GRAPHICS-BASED TARGET DETECTION MODEL

by

Brian Edward Jones

September 2006

Thesis Advisor: Christian J Darken
Second Reader: J Jeffery Crowson

Approved for public release; distribution unlimited

This Thesis Done in Cooperation with the MOVES Institute

THIS PAGE INTENTIONALLY LEFT BLANK

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2006	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE A Computer Graphics-Based Target Detection Model		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Brian Edward Jones			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES: The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution unlimited		**12b. DISTRIBUTION CODE** A	

13. ABSTRACT (maximum 200 words)

Modeling of visual perception for computer-generated forces and intelligent software agents is usually fairly feeble in computer games and military simulations. Most of the time, tricks or shortcuts are employed in the perceptual model. Under certain conditions, these shortcuts cause unrealistic behavior and detract from military training and user immersion into the simulated environment. Many computer games and simulations trace a ray between the target and observer to determine if the observer can see the target. More complex models are sometime used in military simulations. One of these models used in Army simulations is the ACQUIRE model. This model still may produce debatable results. The ACQUIRE visual perception model uses a single value for the target's contrast with its background. This can cause unrealistic results in certain conditions, allowing computer-generated forces to see targets that should not be seen and not see targets that should. Testing these more complex models needs to be completed to determine the conditions under which the model gives questionable results. Testing ACQUIRE against human subjects helped determine when ACQUIRE behaves reasonably. The study consisted of multiple scenes with a target in many positions, multiple postures, and many different lighting and fog conditions. Now that testing and analysis is complete, modifications can be made to the visual perception model allowing it to give better results in more varied conditions, such as: low light, excessive fog conditions, and partially hidden targets.

14. SUBJECT TERMS Target Detection, Artificial Intelligence, Synthetic Players, Perceptual Modeling			**15. NUMBER OF PAGES** 105
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

i

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution unlimited

A COMPUTER GRAPHICS BASED TARGET DETECTION MODEL

Brian Edward Jones
Lieutenant, United States Navy
B.S., United States Naval Academy, 1999

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS AND
SIMULATIONS**

from the

**NAVAL POSTGRADUATE SCHOOL
September 2006**

Author: Lieutenant Brian E. Jones

Approved by: Christian J Darken
                 Thesis Advisor

                 J Jeffery Crowson
                 Second Reader

                 Dr. Rudolph P. Darken
                 Chair, MOVES Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Modeling of visual perception for computer-generated forces and intelligent software agents is usually fairly feeble in computer games and military simulations. Most of the time, tricks or shortcuts are employed in the perceptual model. Under certain conditions, these shortcuts cause unrealistic behavior and detract from military training and user immersion into the simulated environment. Many computer games and simulations trace a ray between the target and observer to determine if the observer can see the target. More complex models are sometime used in military simulations. One of these models used in Army simulations is the ACQUIRE model. This model still may produce debatable results. The ACQUIRE visual perception model uses a single value for the target's contrast with its background. This can cause unrealistic results in certain conditions, allowing computer-generated forces to see targets that should not be seen and not see targets that should. Testing these more complex models needs to be completed to determine the conditions under which the model gives questionable results. Testing ACQUIRE against human subjects helped determine when ACQUIRE behaves reasonably. The study consisted of multiple scenes with a target in many positions, multiple postures, and many different lighting and fog conditions. Now that testing and analysis is complete, modifications can be made to the visual perception model allowing it to give better results in more varied conditions, such as: low light, excessive fog conditions and partially hidden targets.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

THIS PAGE INTENTIONALLY LEFT BLANK

I.   INTRODUCTION

A.   PROBLEM AND THESIS STATEMENT

Visual perception modeling for agents (synthetic players) in computer games and computer-generated forces in military training simulations is weak. This results in lower immersion in the games and training simulations and begs to ask the question whether or not an analytical military training simulation is valid.

In this work, we introduce a new model to more accurately reflect human visual perception by determining what an agent sees using a method similar to that used to generate the human's view of the simulated 3-D environment. Specifically, we will model one player's ability to see another in a complex 3-D environment. A program, built on Delta3D, http://www.delta3d.org/, an open source game engine developed by the MOVES institute, has been developed to demonstrate and test the visual perception model.

B.   MOTIVATION

Current computer games and military simulations employ shortcuts to model visual perception. These tricks usually result in unrealistic agent behavior, that is, behavior that does not accurately model human behavior.   This unreasonable behavior can create questions regarding the legitimacy of military simulations.

The most common technique used to determine if an agent can see a target is by casting a ray from the observer to the target. For human targets and observers, that line-of-sight ray is cast from the observer's head to the top of the target's head. If that ray is unbroken, by intersection with another object, the target is visible to the agent. If that ray intersects another object, even a small one like a tree branch, the target is hidden from the observer.

Consider a sniper in a simulation. A sniper, the target, might be in a place where an observer would have an incredibly hard time finding him, such as in a

cluttered area or through a small window. The observer will most likely be able to see the sniper using a line-of-sight model as the target's head almost must be in the open in order to see the observer to make a shot count. The sniper would be well hidden to human perception, given a well-chosen hiding place and the use of clutter or walls for concealment.

There are more intricate models of visual perception used in some military simulations, the U.S. Army's ACQUIRE model for instance. The inputs to the ACQUIRE model still produce questionable results for simulation agents. One of the inputs is a contrast value, which is the contrast between the target and its background. According to ACQUIRE, the contrast between a target standing on a hill, backlit by the sky, may be the same as a target hidden in a wooded or urban environment. Clearly, a backlit target is very visible to a human and therefore should be especially to agents as well.

C.   SCOPE OF THE THESIS

This thesis will test the AQUIRE model to determine if its results coincide with human behavior. The software used to test ACQUIRE was based upon the work of J. Steve Correia, LT, USN and Dr. Christian Darken using the Delta3D simulation engine.

D.   THESIS ORGANIZATION

The remainder of this thesis is organized into four chapters:

1.   Chapter II – Research and Background

Describes the ACQUIRE model for target detection.

2.   Chapter III – Experiment Setup and Results

Describes the experiment run to test ACQUIRE and begins the discussion of the results.

3.   Chapter IV – Results and Analysis

Describes the results of the study and begins the analysis of those results.

4.   Chapter V – Conclusions and Future Work

Describes the conclusions reached by the study and other work that can be done to test ACQUIRE more thoroughly. Also describes improvements that should allow ACQUIRE to behave more realistically.

THIS PAGE INTENTIONALLY LEFT BLANK

II. RESEARCH AND BACKGROUND

A. INTRODUCTION

A small amount of work has been conducted related to visual perception for artificial intelligence agents in software simulations. This section will describe the U.S. Army's Night Vision and Electronic Sensors Directorate (NVESD) detection model used in some military simulations, the ACQUIRE algorithm. This section will also describe some other alternative perception models.

B. NVESD ACQUIRE

Initially, the ACQUIRE algorithm was developed to predict the performance of imaging systems in the following spectral bands; visual, near visual and infrared. ACQUIRE accepts the inputs that are suggested in AI Programming Wisdom; distance to the target and current visibility levels (lighting and fog conditions). Another input that ACQUIRE employs is a determination of line-of-sight, ensuring that at least part of the object is able to be seen by the observer.

There are two modes of operation that ACQUIRE works in, target spot detection and target discrimination. Target spot detection calculations are based on signal-to-noise ratio. Target discrimination is based on Johnson cycle criteria methodology. (Correia05)

Target spot detection methodology applies to cases where the target is viewed against a uniform background. Detection will occur when the signal-to-noise ratio on the display that subtends the target exceeds that of the background. In layman's terms, detection occurs when there is enough energy from the target that reaches the sensor to create a "hot-spot" on the display. (NVESD)

The target discrimination methodology applies to situations where the target is in an environment with a non-uniform background. This methodology

can be used to predict the probability of detection as well as the prediction of greater levels of target discrimination, such as classification, recognition and identification. (NVESD)

Since its inception, a few recent military simulations have incorporated ACQUIRE. Some of these are: the Combined Arms and Support Task Force Evaluation Model (CASTFOREM), the U.S. Marine's Team Tactical Engagement System (TTES) and the Janus Training Simulation (Pursel04).

The following is a very brief introduction to the inputs and outputs of NVESD's ACQUIRE algorithm. For a more in-depth explanation, J. Steve Corriea's Thesis (Corriea05) has an excellent description.

1.   ACQUIRE Inputs
ACQUIRE uses five inputs to determine the detection probability of the target in the scene; target intensity, background intensity, field of view, fog and lighting conditions.

a.   *Target Intensity*
The target intensity is the pixel intensity value, averaged over the entire target visible to the system and the ACQUIRE algorithm.

b.   *Background Intensity*
Just like the target intensity input, the background intensity pixel value is the average value of the background pixels surrounding the target in the mini-render.

c.   *Field of View*
The field of view takes care of the relative size of the target on the screen and compensates for that size. ACQUIRE estimates the field of view by using the square root of the number of visible target pixels.

d. Fog and Lighting Conditions

ACQUIRE does take into account the visibility conditions of the scene at the time the calculation are made. The amount of fog and the time of day are both inputs to the system.

2. ACQUIRE Outputs

ACQUIRE uses those inputs to determine a probability of detection, in both infinite and finite time.

The infinite time detection probability is given when assuming there are no time constraints on the target detection in the scene. This is the output we will be using for the study.

The finite time probability of detection is used to determine the likelihood of detection given a time limit for the search. The time limit for this output is arbitrary, that is, it can be changed by the user. Search time is an additional input for the finite time ACQUIRE algorithm.

C. OTHER PERCEPTUAL MODELS

In order to study the intelligent software agent's use of cover and concealment, three perceptual models were described by C. Darken in a paper in 2004. Those models described are; the object space approach, raster approach and the multiple ray casting approach. (Darken2004)

1. Object Space Approach

This approach assumes an unlit scene with a light source originating at the observer's eyes. Any object in the scene that is lit at this point is considered visible by the agent. This will allow the agents to find the shadows the scene that will give cover.

2. Raster Approach

This approach begins by a beginning rendering of the scene from the viewpoint of the observer, just large enough to include the entire target. The polygons rendered are then pared down to just those that could be part of the target entity. Those target pixels are placed in a buffer as visible surfaces of the target.

3. Multiple Ray Casting

This approach is similar to the raster approach, but renders the target in multiple sections, such as feet, torso and head. Once the sections are rendered, rays are cast from the viewpoint of the observer to the sections to determine if the rays intersect any of the sections of the target. If any of the rays intersect the target polygons, the target is visible to the observer.

4. Reece's Approach

The approach developed by Reece models a human's center of vision as well as a human's peripheral vision. They modeled the peripheral vision to be sensitive to light and motion. Any movement or lighting found in the peripheral vision is determined to be detected immediately. The center of vision is modeled with a simulated search out to 30 degrees from center. Detection in this portion of the model is not immediate, to simulate the search pattern that humans use while searching from targets, an acquisition time is calculated for detecting objects.

The model takes into account acuity to compare the target's size and motion in determination of a probability of detection. However, color and contrast are not considered in the model; and environmental lighting does not change. (Reece96)

III.   EXPERIMENT DESIGN

A.   SIMULATION REFINEMENT AND EXPERIMENT DESIGN

The simulation used to test the ACQUIRE-NPS algorithm was an adaptation of LT Corriea's work implementing ACQUIRE, initially developed by NVESD.

B.   HOW ACQUIRE-NPS WORKS

ACQUIRE-NPS has some complex workings; the following gives some detail on how the algorithm calculates the detection probability for a target on the screen. ACQUIRE-NPS follows the following steps for the calculation.

1.   Renders the Scene Twice

ACQUIRE-NPS starts by rendering the scene twice. The first render is in normal color with the second render using a false color for the target.

The false color used by AQUIRE-NPS is not important to the calculations. We arbitrarily chose to falsely color the target red for this stage in the process. Figures 1 and 2 show the difference between the normal and false colored scenes,

Figure 1.   Normally Colored Scene

Figure 2.   Falsely Colored Scene

2.   Line-of-Sight Calculations

After both scenes are rendered, AQUIRE-NPS then completes a line-of-sight calculation on the target. The system will look at the falsely colored render

of the scene to determine if any of the false color is visible. If so, the calculations will continue to determine the probability of detection.   If not, the system determines that the target is completely hidden from sight and stops calculations, giving a detection probability of zero.

3.   Mini-Renders are Created

Once ACQUIRE-NPS determines that line-of-sight to the target does exist, the system will create a mini-render of the target and the closely surrounding background pixels.   Figures 3 and 4 are samples of the mini-renders created during the study.

Figure 3.   Normal Color Mini-Render

Figure 4.   False Color Mini-Render

The mini-render allows ACQUIRE-NPS to use the background pixels closest to the target instead of the entire scene's background. The background closest to the target is the most important background in determining the contrast between the target and its background. As you can see in Figure 1, there are many different colors and contrasts in the scene. In order to calculate the contrast the target has with its background, the darker walls in the lower left corner of the scene have no bearing on the contrast between the target and background. The lighter foggy area directly around the target matter most.

The pixels closest to the target are the most important to finding the contrast in a non-uniform environment. Figure 5, below, will be used for the explanation. It is clear that the foggy sky at the top of the scene and the dark brown at the bottom of the scene are not important while determining the contrast between the target and its background. The background that is most important is the foggy area behind the target and wall directly in front of the target.

Figure 5.   Explanation for use of the Mini-Render

The background pixels from the rest of the scene are discarded due to the use of the mini-render. The rest of the background only adds noise to the environment which can confuse the human user with clutter. This will usually draw the eyes to different areas of the screen, sometimes to completely different areas than where the target is, making the detection of the target harder for a human subject.

4. ACQUIRE Inputs

ACQUIRE-NPS uses five inputs to determine the detection probability of the target in the scene; target intensity, background intensity, field of view, fog and lighting conditions.

a. Target Intensity

The target intensity is the pixel intensity value, averaged over the entire target visible to the system and the ACQUIRE-NPS algorithm.

b. Background Intensity

Just like the target intensity input, the background intensity pixel value is the average value of the background pixels surrounding the target in the mini-render.

c. Field of View

The field of view takes care of the relative size of the target on the screen and compensates for that size. ACQUIRE-NPS estimates the field of view by using the square root of the number of visible target pixels.

d. Fog and Lighting Conditions

ACQUIRE-NPS does take into account the visibility conditions of the scene at the time the calculations are made. The amount of fog and the time of day are both inputs to the system.

In the system used to test ACQUIRE-NPS, the scene's fog can be controlled using the '+' and '-' keys on the keyboard's number pad. Fog is incremented and decremented in increments of 2 with a value of 1 being completely clear to 100 being completely fogged in.

ACQUIRE-NPS allows for changing the lighting conditions of the scene by changing the time of day displayed. Table 1 shows the number key and its corresponding time of day displayed.

Key Press	Time of Day
0	0200
1	0400
2	0600
3	0800
4	1000
5	1200
6	1400
7	1600
8	1800
9	2000

Table 1. Number Key and Corresponding Time of Day

5.   ACQUIRE Outputs

ACQUIRE-NPS gives the same outputs as the original ACQUIRE algorithm.

C.   SOFTWARE IMPLEMENTATION

LT Corriea's original system was used as the backbone for the system to test the ACQUIRE-NPS algorithm.

His system was modified to determine whether or not the mouse pointer was on the figure when the mouse button was pressed. This was done using a pick node method.

The system was also changed to allow the user to indicate that he or she could not find the figure in the scene by pressing the space bar.

D. MODEL AND SOFTWARE DESIGN

Some other modifications had to be made to LT Correia's system in order to create the interesting scenes needed for the study.

1. Initial Modifications

In order to move the target figure around the environment in a realistic way, a walk motion model was added to the simulation. When activated, this allowed a user to walk the figure around to capture a variety of character placements for the study.

Once the character could walk, it was necessary to allow the camera to move as well. A fly motion model was added to the simulation for this functionality.

These additions allowed us to create an assortment of interesting scenes in which to conduct the study.

2. Creating the Scenes

Once the fly and walk motion models were in place, the scenes were able to be created.

The figure was walked around the 3D environment as the camera was moved into different positions looking searching for interesting scenes to use. Most of the time scenes that were chose had the figure partly hidden by portions of the scenes, such as walls. The desert camouflage clad figure was in one of four poses; standing, crouching, kneeling and lying prone.

Once the figure and the camera positions were in place, the lighting and fog conditions were adjusted in an effort to find a probability of detection given by ACQUIRE-NPS that were unintuitive.

E.   EXPERIMENT SETUP

1.   Equipment Used

The simulation was run on an Alienware Area-51M 7700 laptop housing a Pentium 4 3.6GHz processor. The system is configured with 1GB Random Access Memory and NVIDIA GeForce Go 6800 graphics card with 256MB of video memory.

A Dell DLP projector was used, in lieu of the laptop's 17 inch SXGA screen, in order to keep the screen as constant as possible over all subjects. Using the laptop's screen the person's height and the angle the screen was currently at affected how screen was seen. The scenes were projected on the wall in Watkins Hall, room 382. The projector was set to a resolution of 1280 by 1024.

2.   Experiment Setup

This experiment was run in an office on the Naval Postgraduate School campus in the Watkins Hall Annex, room 382. The room was setup using dark sheets over the windows and shades to block as much sunlight as possible.

Office furniture was adjusted to allow the subject to sit across the room with the projected screen on the opposite wall. The subject searched through a collection of 36 scenes for the figure of a character hidden in the scene.   The office setup is shown in Figure 6.

Figure 6.   Office Setup.

Subjects were asked to use a computer mouse to identify the figure in the scene, indicating they found the hidden figure. If the character was not seen, the subject was instructed to press the keyboard. Appendix A is a more in-depth explanation of the experiment and instructions for the subject.

No time limit was given for each scene, allowing the subject to take sufficient to locate the figure. We did not want to influence the subject to guess about the figure's whereabouts without due time to thoroughly look though the scene.

To try to keep people from guessing too much, we created two scenes where the character is not in the scene. This was done to persuade the subjects not to guess the location of the figure, to make sure they had a very good idea of where the figure was before making a decision on the figure's location.

The experiment was setup using a large variation of environmental conditions as well as varying target positions and placements. Figures 7 and 8 show the scene variations used.

Figure 7.   Scene Collection 1

Figure 8.   Scene Collection 2

The entire selection of scenes selection used in the experiment is available in both normal coloring and false coloring in Appendix B.

3.   Subject Population

During the experiment, we had a wide variety of subjects tested. 20 subjects were males while two were female. We also had 17 military members and five civilians in the study. Eight were trained in aviation with three civilian pilots, four military pilots and a military flight officer. Four subjects were foreign citizens and 18 were U.S. citizens.   Two subjects were ground combat trained U.S. Marine Corps trained members plus two subjects who were U.S. Marine Corps pilots with training in ground combat during the Marine Corps Basic School in Quantico, VA.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. RESULTS AND ANALYSIS

After 22 subjects, we have seen some problems with ACQUIRE-NPS's probability of detection for the character. Out of 36 scenes, we've determined that nine scenes are consistent with the probability of detection provided by ACQUIRE-NPS. On the other hand, the data collected for 25 of the scenes is not consistent with ACQUIRE-NPS's detection probability.

A. RESULTS

As mentioned above, the data collected by the experiment agrees with the probability of detection in nine out of 36 scenes. The p-value was calculated for each scene in order to determine whether or not the data collected during the study coincided with the detection probability as determined by ACQUIRE-NPS. The p-value is the determination of whether or not the probability of detection matches the data collected during the study.

The algorithm used to calculate the p-value takes as inputs the following; the number of people who correctly located the figure in the scene, the number of people who tried to locate the figure (22) and the detection probability given by ACQUIRE-NPS. Figure 9 is the histogram of the results.
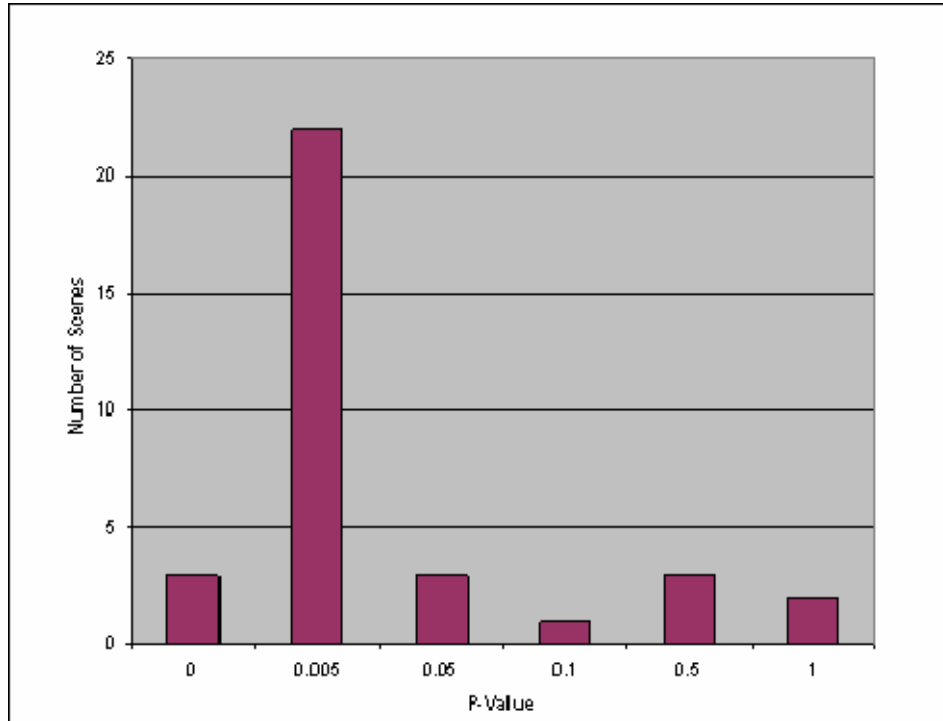
Figure 9.   Histogram of P-Values

We are giving ACQUIRE-NPS the benefit of the doubt in determining whether or not the data collected in the study matches what we collected from ACQUIRE-NPS for the scenes. We determined any p-value of .05 or above as being consistent with ACQUIRE-NPS's probability of detection.

Using that criteria, ACQUIRE-NPS was correct for 9 scenes. Two scenes were removed from the analysis of ACQUIRE-NPS's accuracy since the target figure was not present in them. In 25 scenes ACQUIRE-NPS was wrong, even with the great benefit of the doubt given.

1.   Determining an Offset

The conclusion was made that an offset may be required to compensate for there not being zero light for an entirely black pixel. It was believed that this offset would allow for a greater benefit of the doubt given to ACQUIRE-NPS. Starting with the original contrast equation:

22

$$\left| \frac{P_t^2 - P_b^2}{P_b^2} \right|$$

Where:

P_t is pixel intensity of the target

P_b is the pixel intensity of the background

An offset was added to each of the terms giving:

$$\left| \frac{(P_t + f)^2 - (P_b + f)^2}{\left(P_b + f\right)^2} \right|$$

Where:

f is the added offset

The completed equation to implement an offset in ACQUIRE-NPS is:

$$\left| \frac{P_t^2 - P_b^2 + 2f(P_t - P_b)}{P_b^2 + 2P_b f + f^2} \right|$$

Using the data from the experiment and the offset ACQUIRE-NPS equation, a "best-guess" offset was determined. The sum of the differences of ACQUIRE-NPS's probability of detection and the actual percentage of subjects who correctly identified the target in the scene was determined.

$$\sum_{scene=1}^{34} \left| P_{det} - Actual\% \right|$$

Where:

23

P_{det} is ACQUIRE-NPS probability of detection with the offset

Actual% is the percentage of subject who correctly detected target

Summation to 34 to exclude scenes where no targets are present

For the first run, the offset was changed in increments of 10 from -1000 to 1000. Figure 10 shows the results of that run of the algorithm.
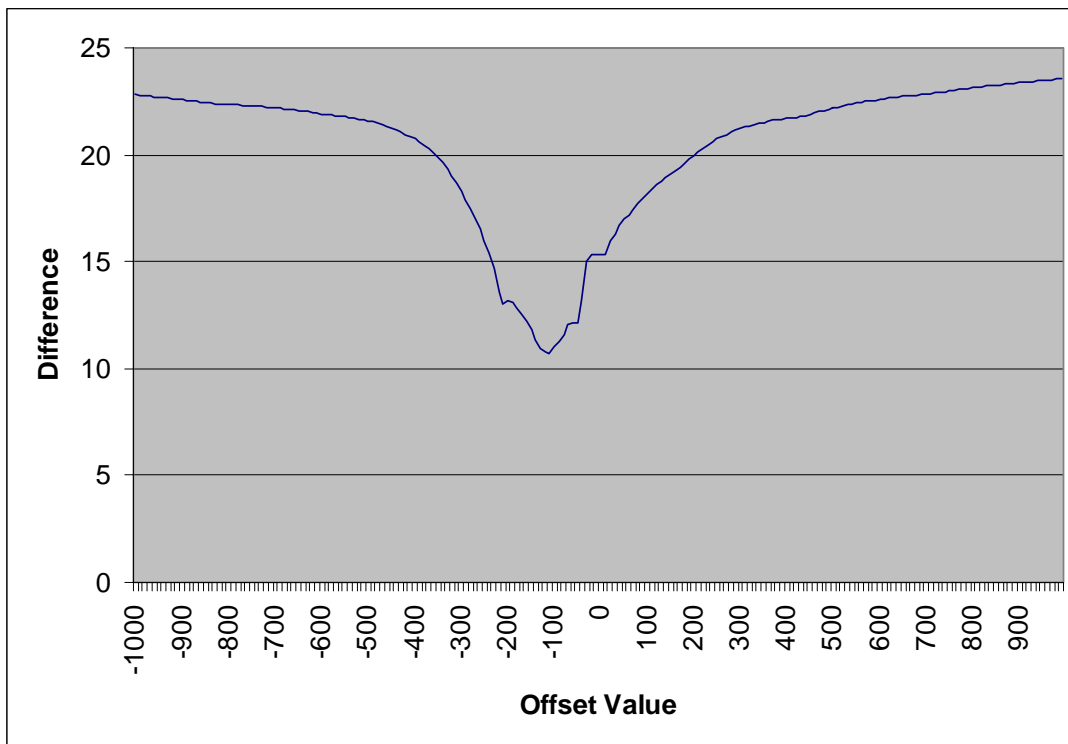
Figure 10.   Offset Value

Looking for the smallest difference between ACQUIRE-NPS's probability of detection and actual detection, that data was used to narrow down the search. For the next run of the algorithm, the offset was incremented by 1 from -350 to 170. This range was used to ensure we had the actual minimum of difference not just a local minimum. Figure 11 shows those results.
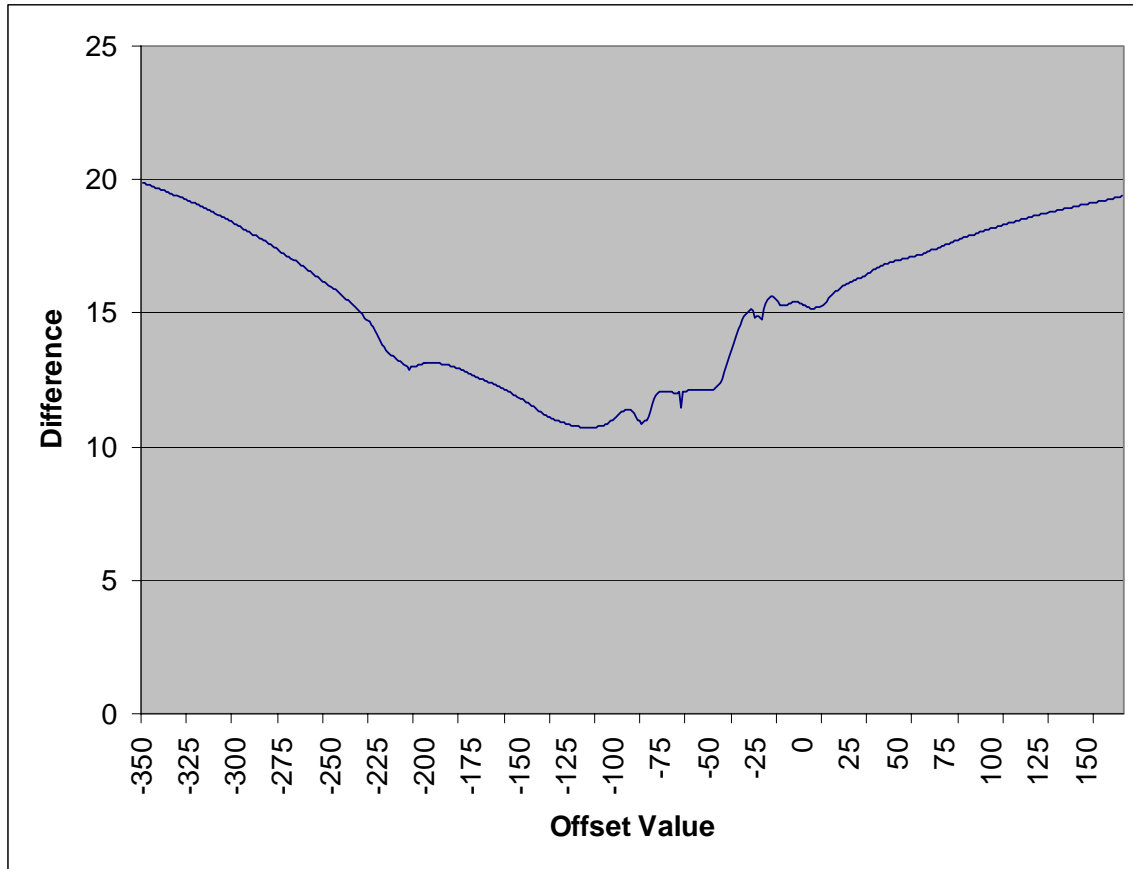
Figure 11.   Offset Results

The range of offset values was reduced once more from -119 to -105 with the offset incremented by .1. Figure 12 shows the results of that run.
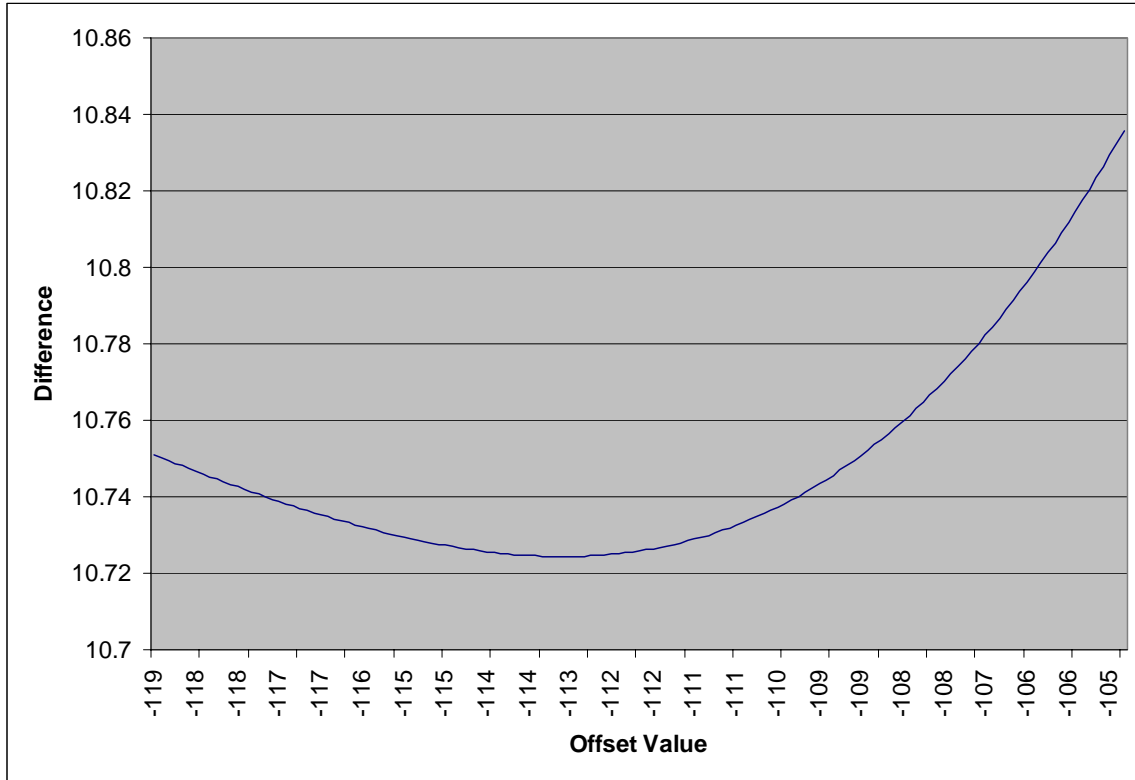
Figure 12.   Best Offset

The "best-guess" offset value was determined to be -113.1.   The determination not to narrow down the offset any more was due to the small variations of difference when the offset was incremented by .1 as shown by Table 2.

Offset	Difference
-113.4	10.72447
-113.3	10.72442
-113.2	10.72439
-113.1	10.72438
-113	10.72439
-112.9	10.72443
-112.8	10.72449

Table 2.   Offset Values around Best Offset

The ACQUIRE-NPS algorithm used for the experiment had an offset of -113.1 placed in the contrast algorithm. Once the offset was set, the ACQUIRE-

NPS algorithm was run again on the scenes used in the experiment. Figure 13 shows the p-values for ACQUIRE-NPS's probability of detection with the offset in place.

Histogram After Offset Applied
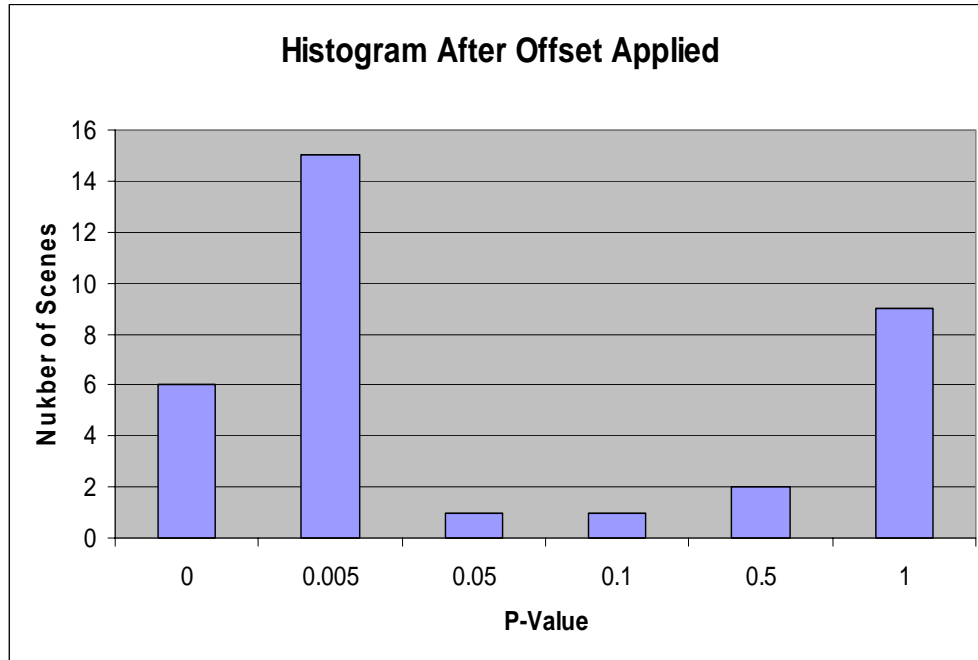
Figure 13.   Histogram of P-Values after Offset Applied

After the offset was applied and ACQUIRE-NPS run against the scenes again, it was determined that ACQUIRE-NPS was correct in 13 scenes and wrong in 21. Again the two scenes without the target figure were disregarded.

2.   Disregarding Occlusions

Initial research has begun to explore the use of the Depth Buffer to disregard the portions of the background that are in front of the target. Having contrast between a target and an object in front of that of target does not help a human in detecting a target. However, the contrast between the target and portions of the background that is actually further away than the target is very use3ful in detecting that target. This is because a target against a background

further away will create a silhouette showing the target's contours and contrast. Anything in front of the target will show the object's contrast and contours, not necessarily those of the target.

ACQUIRE-NPS has been modified to use the Depth Buffer to determine if any of the background pixels are closer than the closest target pixel. If so, they can be ignored when calculating the contrast value. Figure 14 is a mini-render using the z buffer to discriminate between the target and background pixels that in front of the target and background pixels that are behind the target.

In Figure 14, the green pixels are in front of the target, the red pixels are behind the target. The blue pixels are the target.

Figure 14.   Mini-Render using the Depth Buffer

Using this version of ACQUIRE-NPS, the scenes from the experiment were tested again. Table 3 shows the results.

Scene Number	Normal P(det)	Occlusion added P(det)	% Correct
1	0.613325	0.00645355	0.272727273
2	0.155516	0.207612	0.045454545
3	0.0233719	0.797814	1
4	0	0	0.954545455
5	0.997602	0.385906	0.5
6	0.999951	1	0.136363636
7	0.000100634	0.982789	1
8	0.836703	0.836703	1
9	1	1	0.954545455
10	0.955698	0.985929	0.272727273
11	0.00122026	0.221222	0.136363636
12	1	1	0.727272727
13	0	0	0.818181818
14	0.992668	0.00444001	0.954545455
15	0.676234	0.923238	1
16	0.999979	0.999993	0.772727273
17	0.0305938	0.994748	1
18	0.99875	0.999964	0.954545455
19	0.847204	0.986573	0.954545455
20	0.000378397	0.127457	1
21	0.78488	0.592278	0.227272727
22	0.999949	0.999999	0.318181818
23	1.13E-05	0.999867	0.863636364
24	0.0653888	0.867673	0.727272727
25	0.927046	0.988133	1
26	1	1	1
27	0.146964	0.725147	0.909090909
28	0.785458	0.999905	0.954545455
29	0.299228	0.299228	1
30	0.000171252	0.897773	0.727272727
31	0.832327	0.882711	0
32	0.999996	0.986587	0.727272727
33	0.170295	0.78736	1
34	0.0538681	0.651942	1
35	0.961959	0.996706	0.409090909
36	0.553112	0.911748	1

Table 3. Normal and Occlusion Added Probability of Detection Along with Actual Percentage of Subjects that Detected the Target

Using the Depth Buffer Discrimination shows a great difference in those scenes where partial occlusion of the target has occurred. ACQUIRE-NPS gives most of the scenes where the target is partially hidden by portions of the

background a probability of detection that is closer to the actual percentage of subjects that detected the target during the study.

B.   PROBLEMS WITH ACQUIRE

We believe that ACQUIRE-NPS has problems determining detection probability because of the following problems; color not considered in calculations, scene clutter not considered, the shape of the figure not considered and only one intensity number for the target and background pixel value.

1.   Color Not Considered

The current version of ACQUIRE-NPS does not take into consideration the color of the target or the background surrounding the target. In order to show that color is important, we developed two scenes where the target's color was changed from the desert camouflage to a bright red, where the intensity of the red is so similar to the intensity of the background that ACQUIRE-NPS gives a detection probability of near zero for one scene and zero for the second.

As common sense dictates, the red character was immediately detected and indicated by the subjects. All subjects in the study found the character in both scenes, even though they should have been invisible according to ACQUIRE-NPS. It was such an oddity to see I received many different comments regarding the red figure in the scene. Some of those comments were:

"You have a bug in the simulation, the guy is already red."

"Why is the character already red?"

"What happened? He's already red."

Figure 15, seen below, is one of the scenes we used to demonstrate that color does count and should be taken into consideration when calculating the probability of detection.

Figure 15. Initially a Falsely Colored Target

2. Clutter is a Problem

Some of the scenes we used in the study had the character around some clutter of differing size and shape. The scene below shows one of those scenes. As one might think, the clutter caused confusion in some people. Others took longer to look for the character. Others claimed that the character wasn't in the scene. Clutter in the scene does cause the probability of detection to decrease in human subjects and that behavior should be shown in ACQUIRE-NPS. Figure 16 below shows how a target can get lost in a cluttered scene.
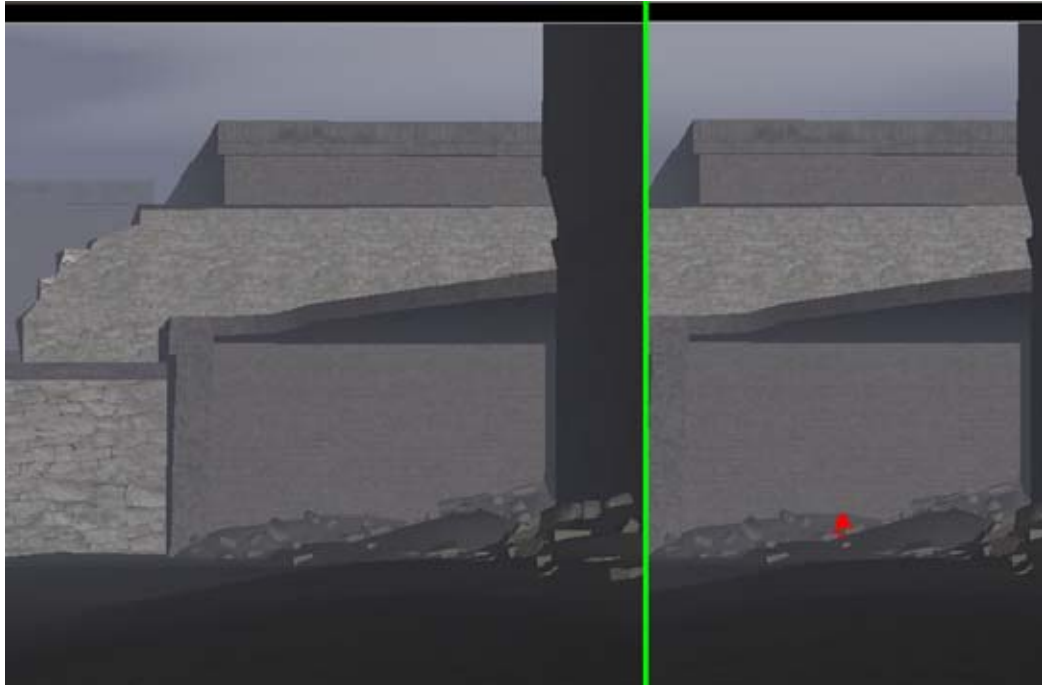
Figure 16.   Cluttered Scene with Normal and False Coloring

3.   Shape Not Considered

The shape of the figure is also another issue that is not taken into consideration. In some of the scenes, the background image was a cityscape of straight lines, as in Figure 17. The roundness of the character's shoulders and head drew people's eyes to the target, that statement was said to me by multiple subjects. Figure 17 is portrayed in false colors to easily show the target's contours.

Figure 17.   False Colored Scene Showing Figure Contours

4.   Single Intensity Number

ACQUIRE-NPS uses a single number for the background intensity as well as the target's intensity on the screen. This characteristic of ACQUIRE-NPS is the cause of what we call the background averaging problem. Figure 18 is used to illustrate the problem with using a single number for these values.

Figure 18.   Single Number Intensity Example

It is very easy to see in the above figure that an average of the black and white background could calculate out to be the same intensity as the grey character. The character shown is obviously very visible and contrasts greatly with both the white and black portions of the background. However, ACQUIRE-NPS will use a single intensity number averaged over both, which could result in a prediction that the figure is not visible or barely visible.

5. Variations between Runs

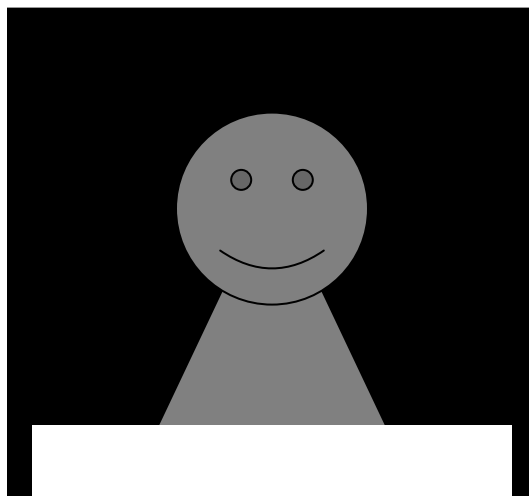Before, during and after the experiment, ACQUIRE-NPS gives slightly different contrast values and different number of visible target pixels for each time the algorithm is run on a typical scene. Because the contrast values and numbers of visible pixels change, the probability of detection provided by ACQUIRE-NPS is slightly different each time the algorithm is run.

The variations that occur are dependant upon the machine that the environment is run on. On one machine, there is no variation at all. Another machine in the MOVES department has variations of up to 20%. However, on the machine that my experiment was run on, the variations were very small, but they did exist. The standard deviation of the variations for both with and without an offset calculated in the probability of detection was less than .01. Tables 4 and 5 show some of the variations. For the full tables of all scenes, please see Appendix C.

Scene Number	P(det)	P(det)	P(det)	P(det)	P(det)	Standard Deviation
1	0.614644	0.614644	0.613325	0.614644	0.614644	0.000589875
2	0.140689	0.155815	0.156713	0.154621	0.14494	0.007262697
3	0.023084	0.023084	0.0233719	0.023084	0.023084	0.000128753
4	0	0	0	0	0	0
5	0.997602	0.997601	0.997602	0.997601	0.997602	5.47723E-07
6	0.999951	0.99995	0.999951	0.99995	0.999951	5.47723E-07
7	0.000100634	0.000100634	0.000100634	0.000100634	0.000100634	1.51522E-20
8	0.836703	0.836703	0.836703	0.836703	0.836703	0
9	1	1	1	1	1	0
10	0.955698	0.955698	0.955698	0.955698	0.955698	1.24127E-16

Table 4. Showing Variation Before an Offset was Applied

Scene Number	P(det)	P(det)	P(det)	P(det)	P(det)	Standard Deviation
1	0.0159053	0.0159053	0.015843	0.015843	0.0159053	3.41231E-05
2	0.00288597	0.00371093	0.00259835	0.00322949	0.00336772	0.000430706
3	0.317275	0.316621	0.319839	0.319839	0.317275	0.001546979
4	0	0	0	0	0	0
5	1	1	1	1	1	0
6	0.162172	0.161739	0.161626	0.161626	0.161344	0.000300825
7	0.000491963	0.000723809	0.000723809	0.000723809	0.000723809	0.000103685
8	1	1	1	1	1	0
9	0.997635	0.997635	0.997635	0.997635	0.997639	1.78885E-06
10	1	1	1	1	1	0

Table 5.   Showing Variation After an Offset was Applied

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS

ACQUIRE-NPS does a fairly good job modeling human behavior is some situations. In others, though, some more work needs to be done in order to get ACQUIRE-NPS to model human behavior more closely.

1. ACQUIRE Improves on Line-of-Sight

ACQUIRE-NPS completely avoids the line-of-sight ray tracing used in many modern games. By avoiding this algorithm all together, the problems associated with it are solved.

ACQUIRE-NPS does not suffer from the unrealistic predictions regarding the visibility of the target that line-of-sight does. A tiny piece of the target (top of the head) being visible or not is not the only determinant used in detecting the figure. In that respect, ACQUIRE-NPS does behave more realistically than casting a single ray.

As discussed earlier, the study conducted in testing ACQUIRE-NPS consisted of 22 subjects and 36 scenes. A line-of-sight algorithm will say that the target is completely visible and obvious in 29 scenes, not visible in seven scenes, including two scenes where the target is absent from the scene. Contrasted with the results of the experiment, it is clear that a line-of-sight algorithm would be too optimistic for scenes used in the study.

2. ACQUIRE is Missing Important Factors

ACQUIRE-NPS does not take color, shape, texture and clutter into consideration, as shown in Figure 16. As Reece mentioned, "the fovea is used for shape, pattern, and color discrimination, and thus the primary means of identifying targets." (Reece96) These factors are extremely important in target detection and need to be considered.

3. ACQUIRE's Representation of Intensity is a Problem

As we saw earlier, using a single number for the intensity of the target and the background is problematic and should be addressed. Splitting the background and target into multiple pieces can be an avenue of exploration to address the problem of having a single intensity value for each.

B. FUTURE WORK

As we have seen, there are some improvements that should be done to make the ACQUIRE-NPS algorithm behave more realistically. Additionally, there are some other areas that need work done.

1. Urban ACQUIRE

As the ACQUIRE algorithm was originally intended to model night vision devices in open terrain, several Army organizations, including the TRADOC Analysis Center – Monterey (TRAC-Monterey) are interested in developing an algorithm to use instead of ACQUIRE which could model target acquisitions in high resolution close quarters simulations, such as in urban environments. Such work could make use of an in-depth functional analysis of how acquisitions are made in an urban environment.

Associated areas of research could involve the topic of false positive detections, where an object other than a target is selected as a target. Few simulations account for such occurrences and the experimental techniques in this thesis could be modified to capture pertinent data. Research topics to explore could include determining the factors that affect the probability of making false positive target detections and determining how, once a false positive detection has occurred, people go about resolving them.

There is also an opportunity to explore how visual noise interrupts the process of detecting targets. For example, how does a civilian population moving through an environment disturb the target acquisition process? Such

research would be highly relevant to current needs for analysis given the kinds of combat missions typical in the current Global War on Terror.

2.   Color Aware ACQUIRE

As we saw from the results, color is important in the detection of a target. Refer to Figure 15 to see why color is important.   Remember, according to ACQUIRE-NPS predicted a 7% probability of detection as seen with a falsely colored red figure. However, as you can easily imagine, 100% of the subjects in my experiment easily detected that target.

Currently, an algorithm is present in the software used to implement ACQUIRE-NPS for this experiment.   In this algorithm, color is taken into consideration, computing contrast for each color (red, green and blue). I believe this rudimentary algorithm can be developed into a mature algorithm that can be added to the original ACQUIRE-NPS algorithm for more realistic results.

3.   Depth Buffer Discrimination

As shown in Figure 14, more research is needed to determine how to best use the Depth Buffer to modify ACQUIRE-NPS so that its results are closer to the results of the human subjects in the study.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A

Reason for Experiment Form

Thank you for your participation in this experiment. Dr. Christian Darken and I, LT Brian Jones, are interested in determining the validity of an ACQUIRE-Like probability of detection model designed and built by Dr. Darken and LT J. Steve Correia, USN. The ACQUIRE Model is currently used in multiple Military simulations, including Janus Training Simulation, Combined Arms and Support Task Force Evaluation Model (CASTFOREM) and the U.S. Marine Corps' Team Tactical Engagement System (TTES).

After a brief demonstration showing the various positions the humanoid character can be in, you will be given a series of scenes using varying lighting and fog conditions. If you cannot detect the character, or believe the character is not present in the scene, press the spacebar on the keyboard. If you can detect the humanoid character, place the mouse pointer over the character and press the left mouse button, indicating where you believe humanoid character to be.

This experiment does not take time into consideration. Please be as accurate as you can, be sure you see or don't see the character before taking action.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B

A.   EXPERIMENT SCENES

   The following show the scenes used in this study. The top picture will show the normal colored scene and the bottom will show the false coloring of the target. For the two scenes where there was no target available, there will only be one picture available.
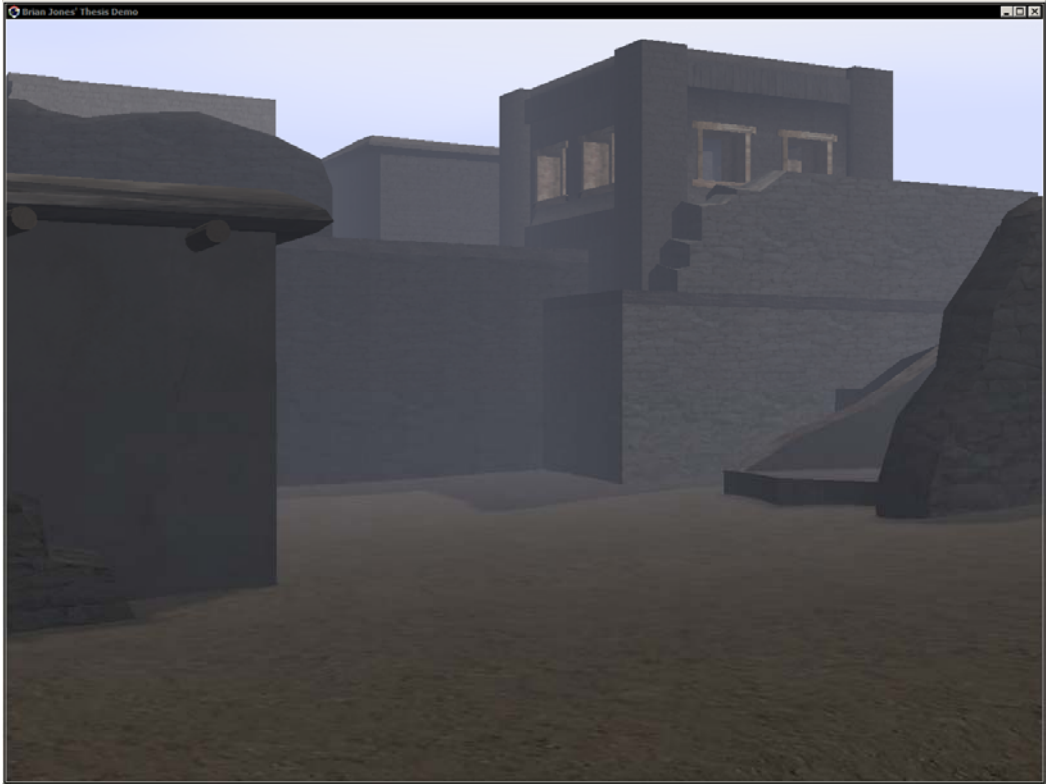
1. Scene 1

2. Scene 2

3.   Scene 3

4.   Scene 4

5. **Scene 5**

6. Scene 6
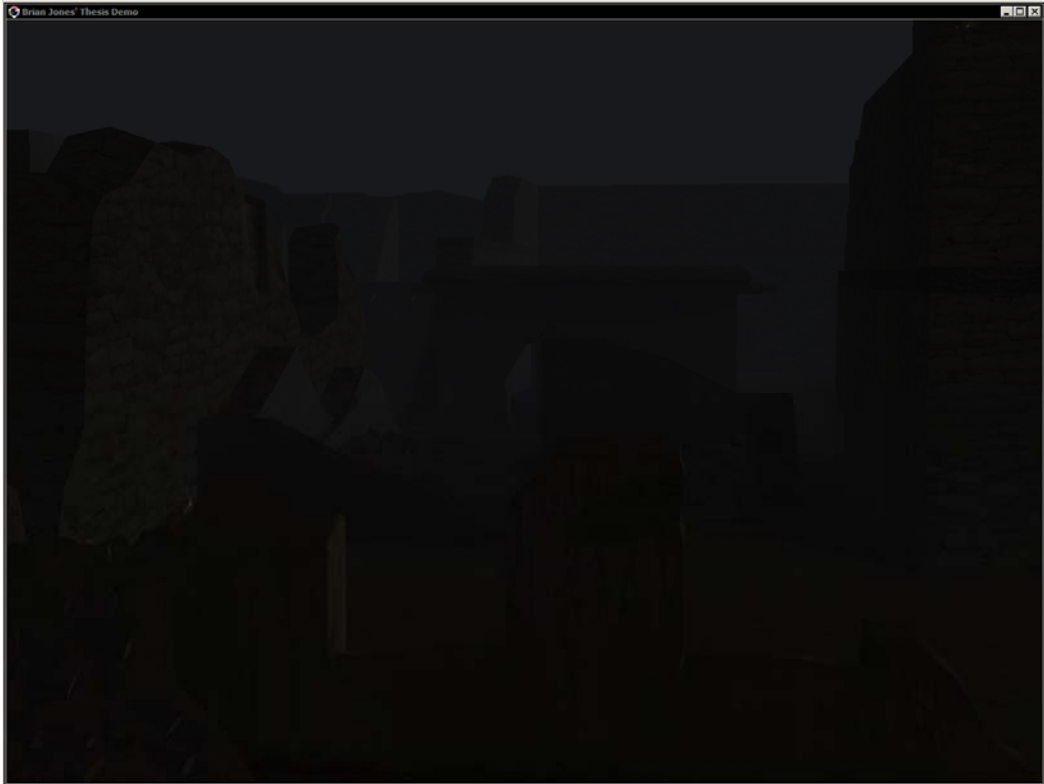
7. Scene 7
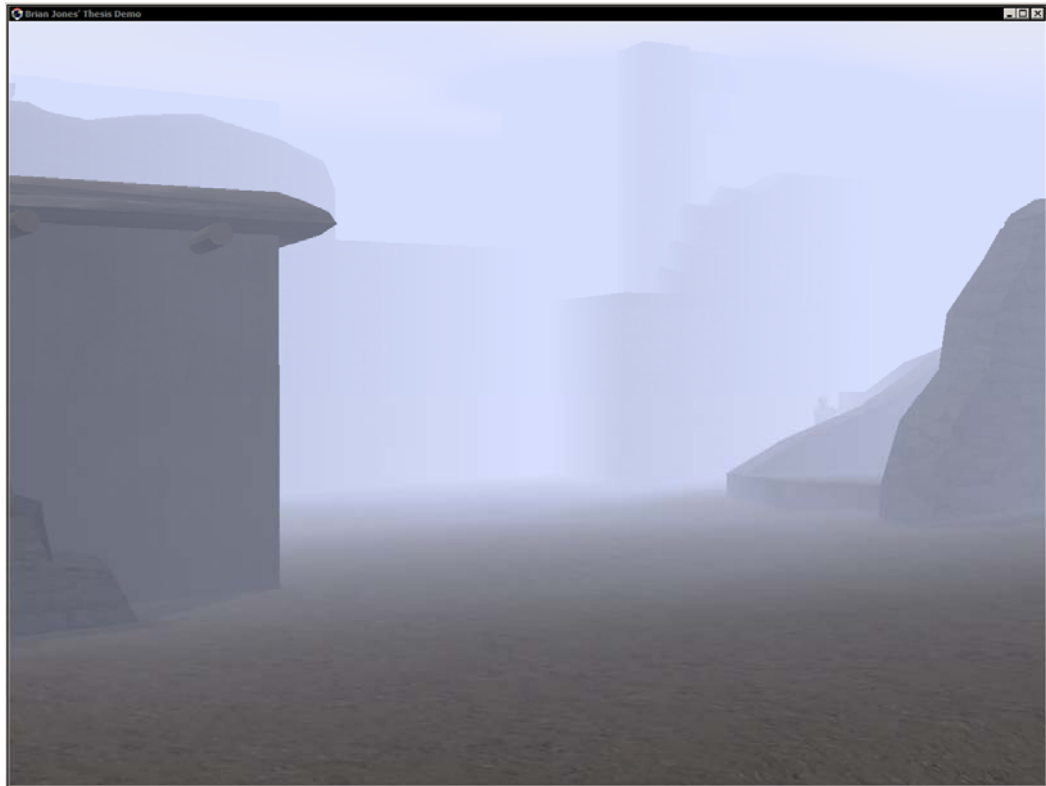
8.   Scene 8
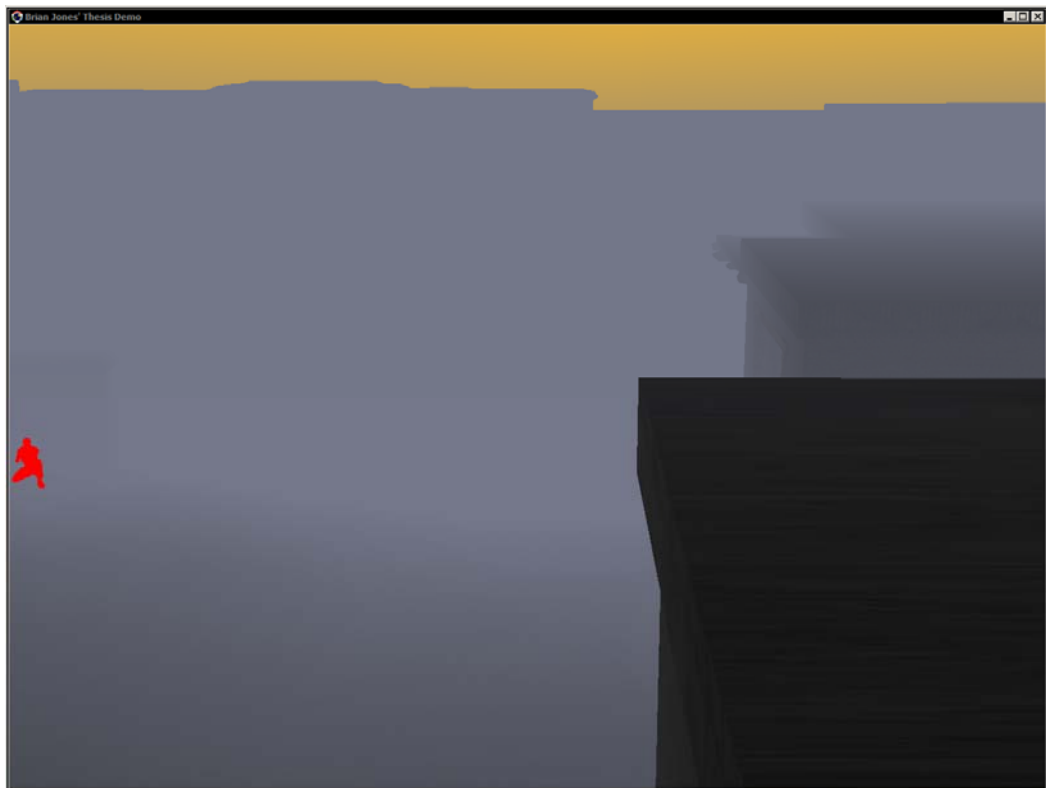
9. Scene 9
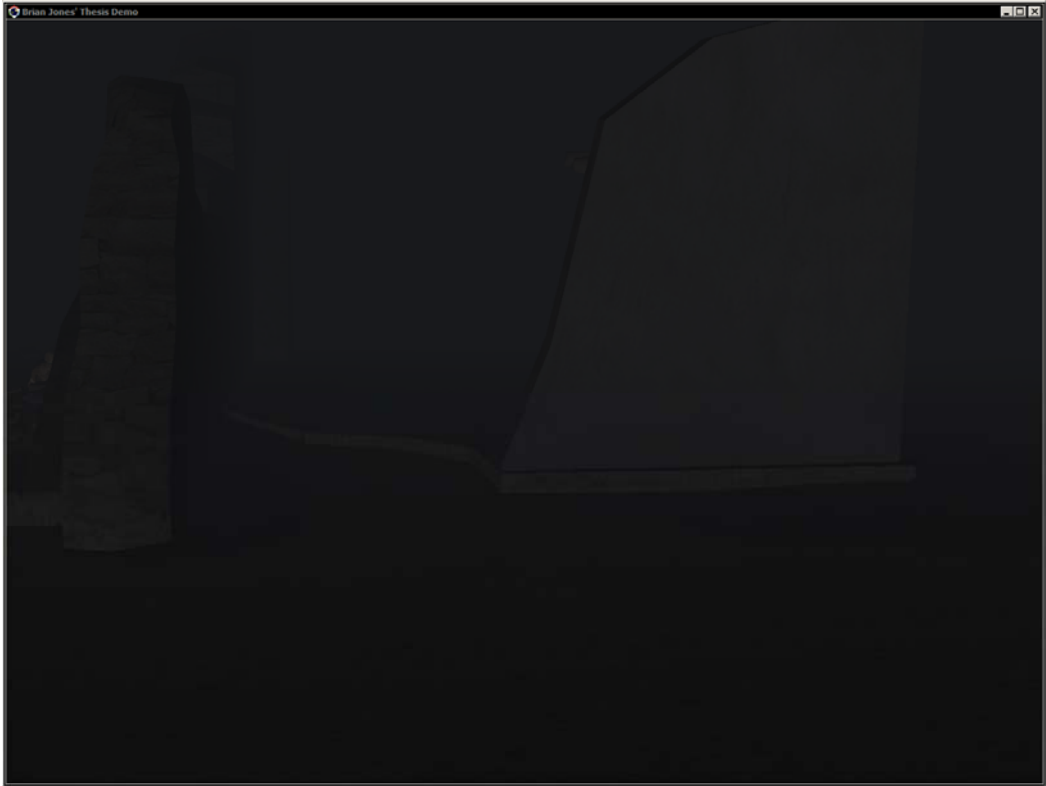
10.   Scene 10

11. Scene 11

12. Scene 12

13. Scene 13
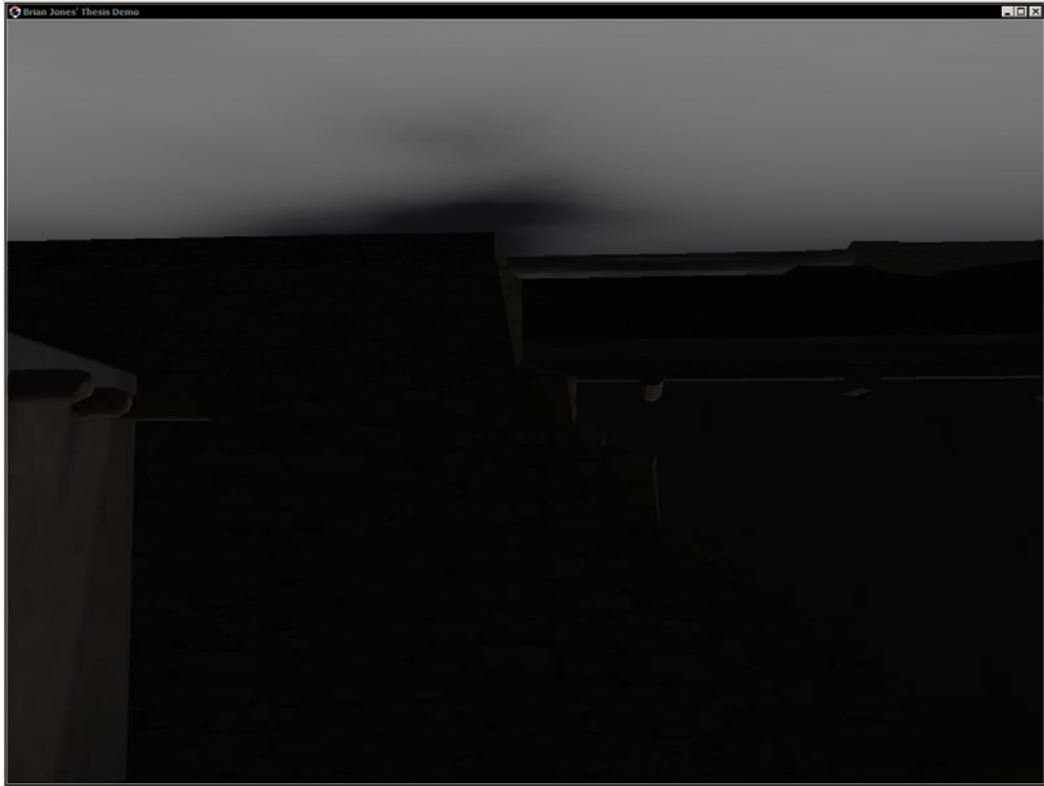
14.   Scene 14
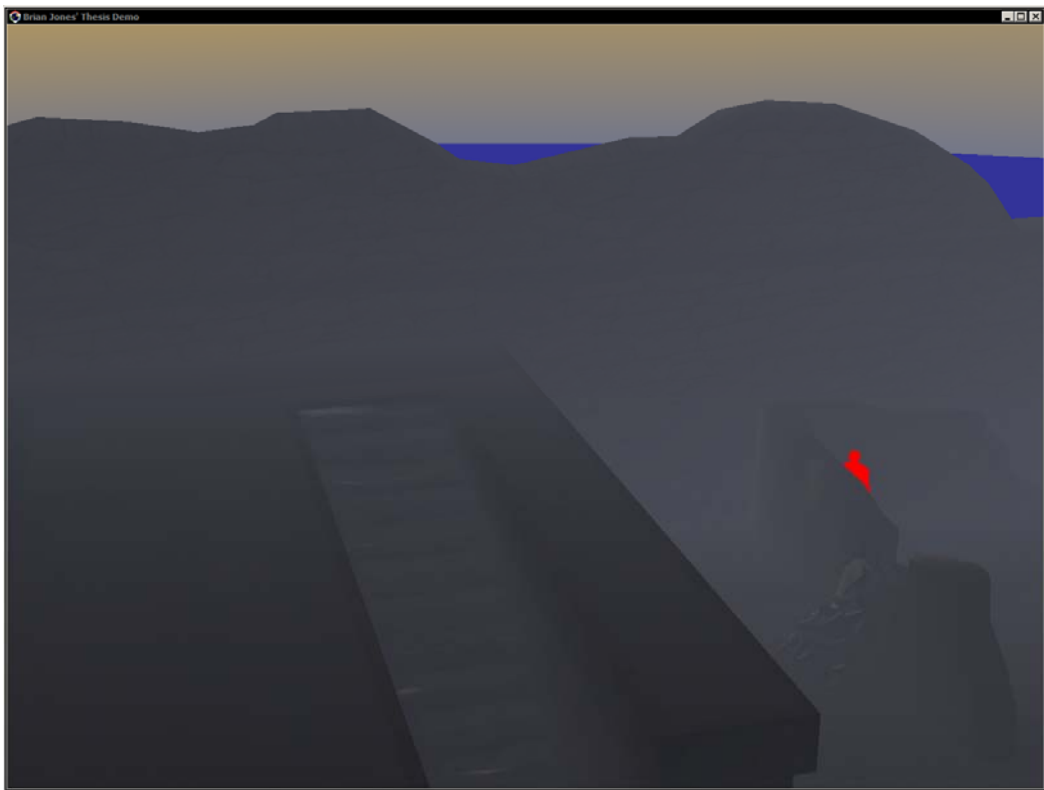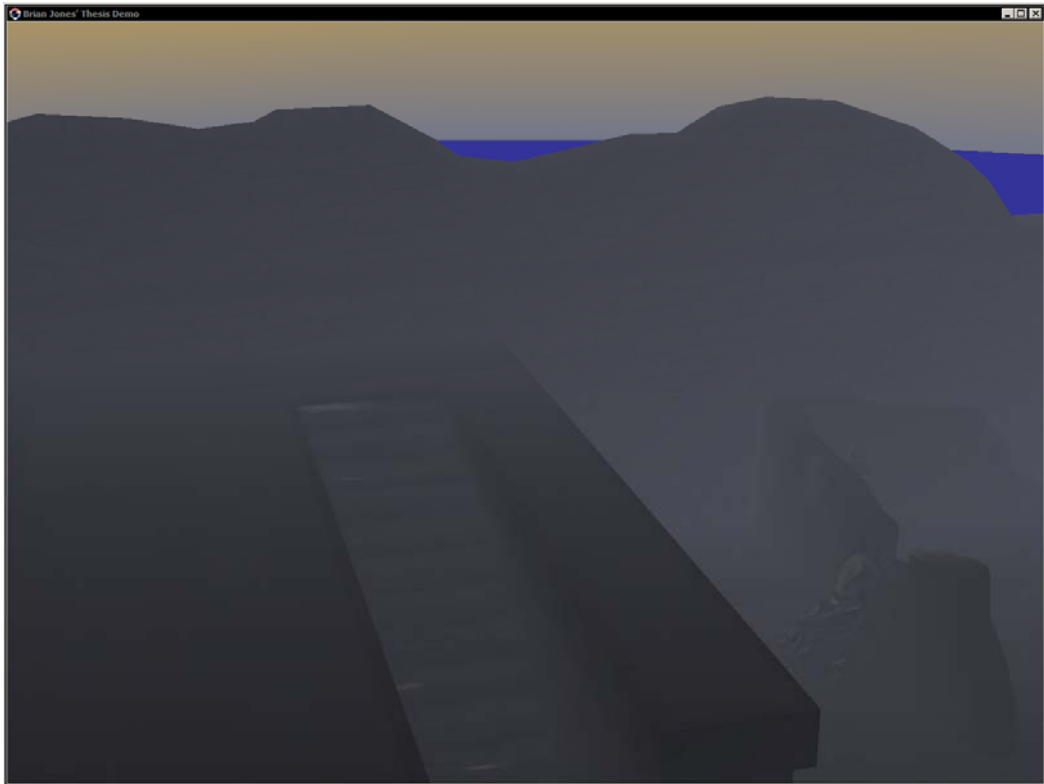
15.   Scene 15

16.   Scene 16

17.   Scene 17

18.   Scene 18

19.   Scene 19

20.   Scene 20

21.   Scene 21

22.   Scene 22
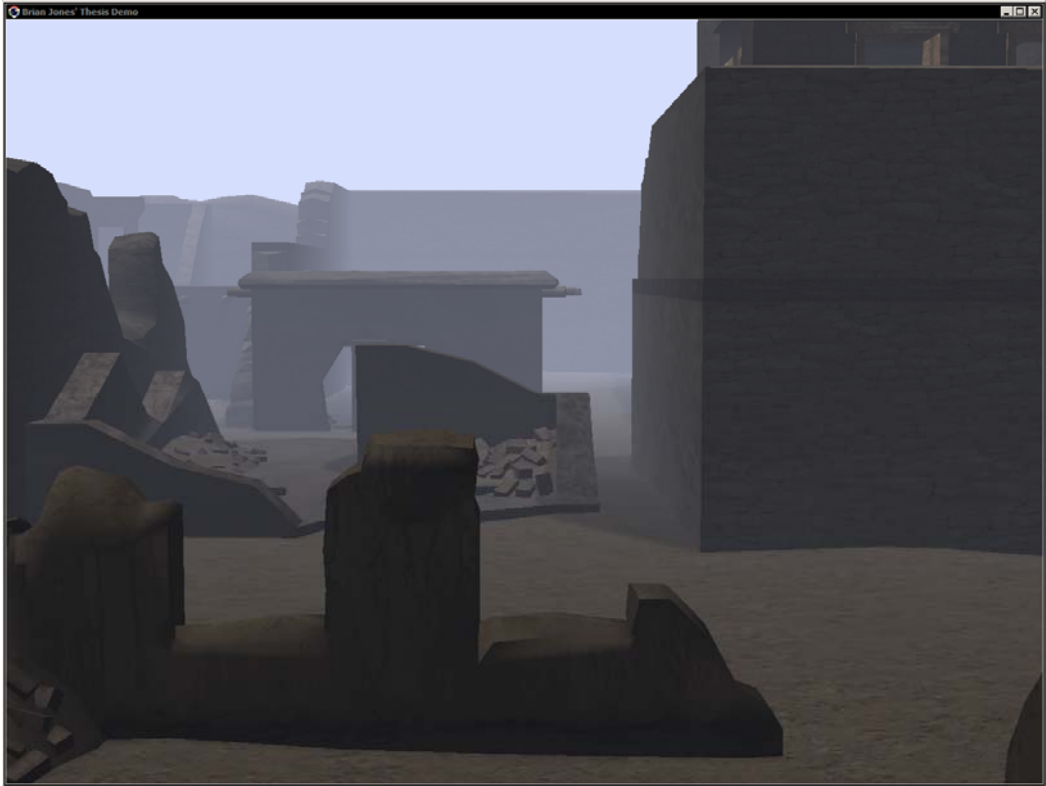
23. Scene 23

24.   Scene 24

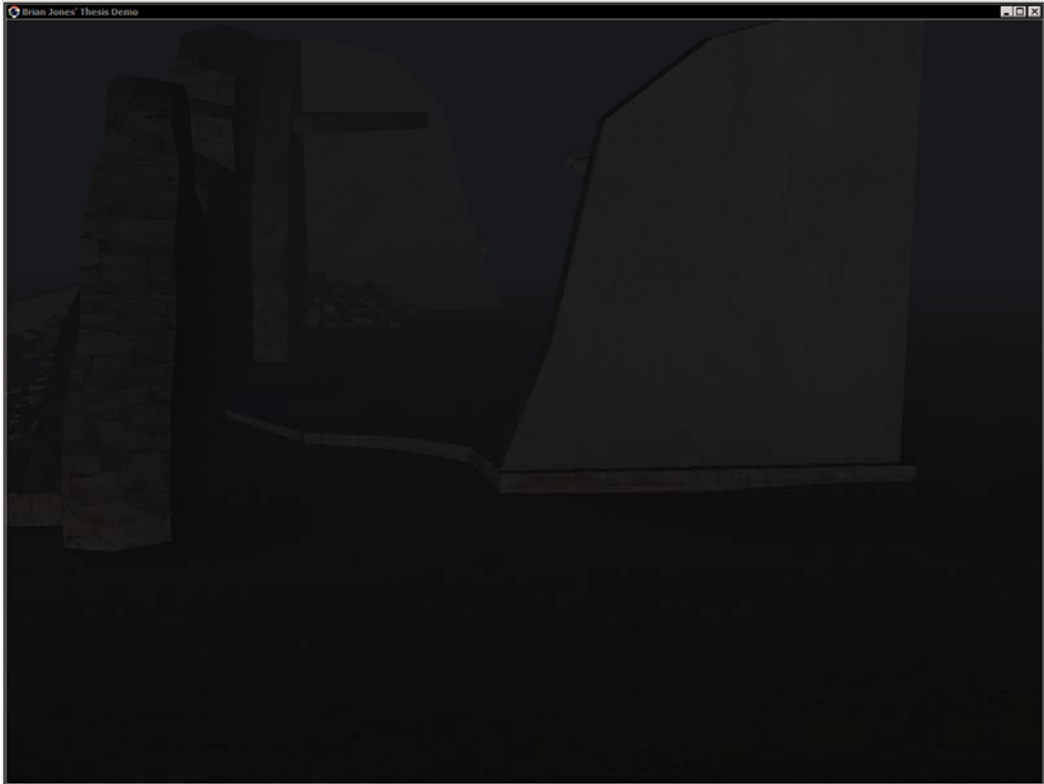## 25. Scene 25

26.   Scene 26
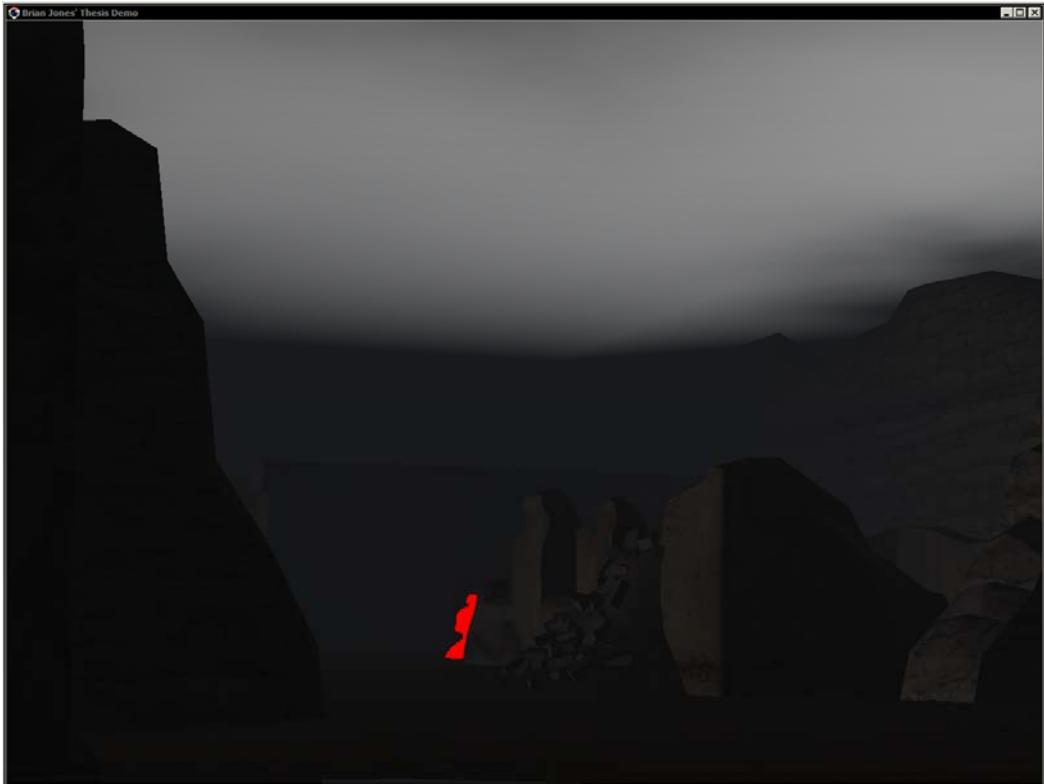
27.   Scene 27

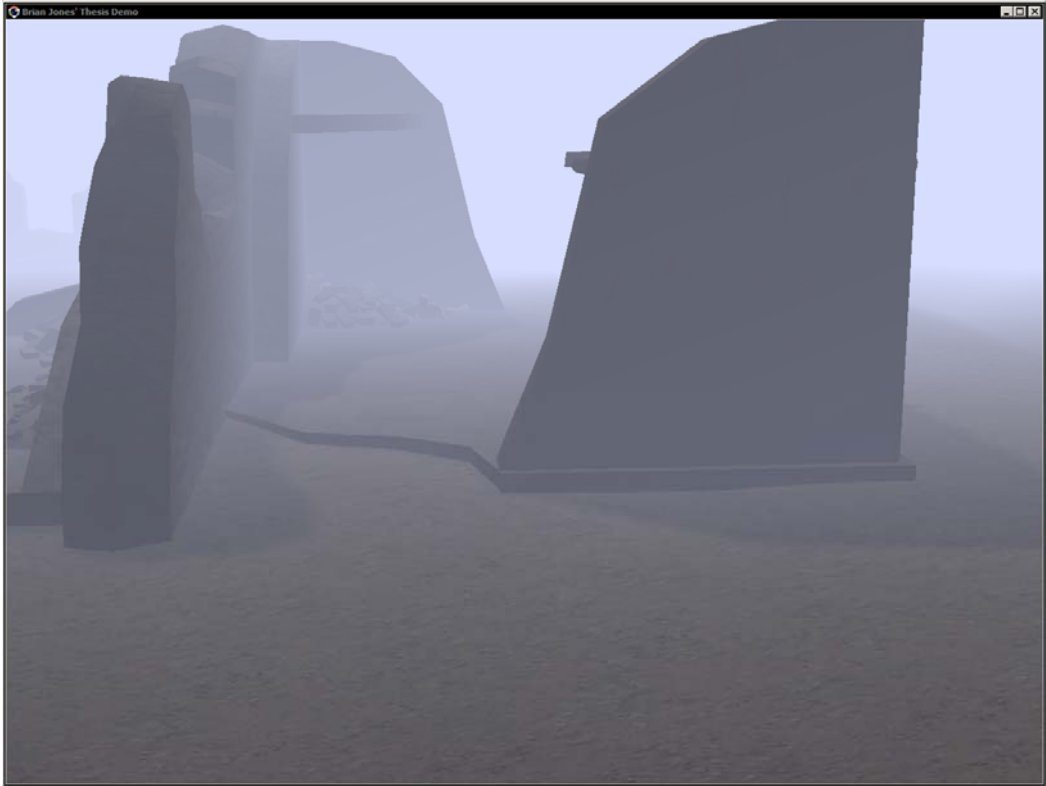## 28.   Scene 28
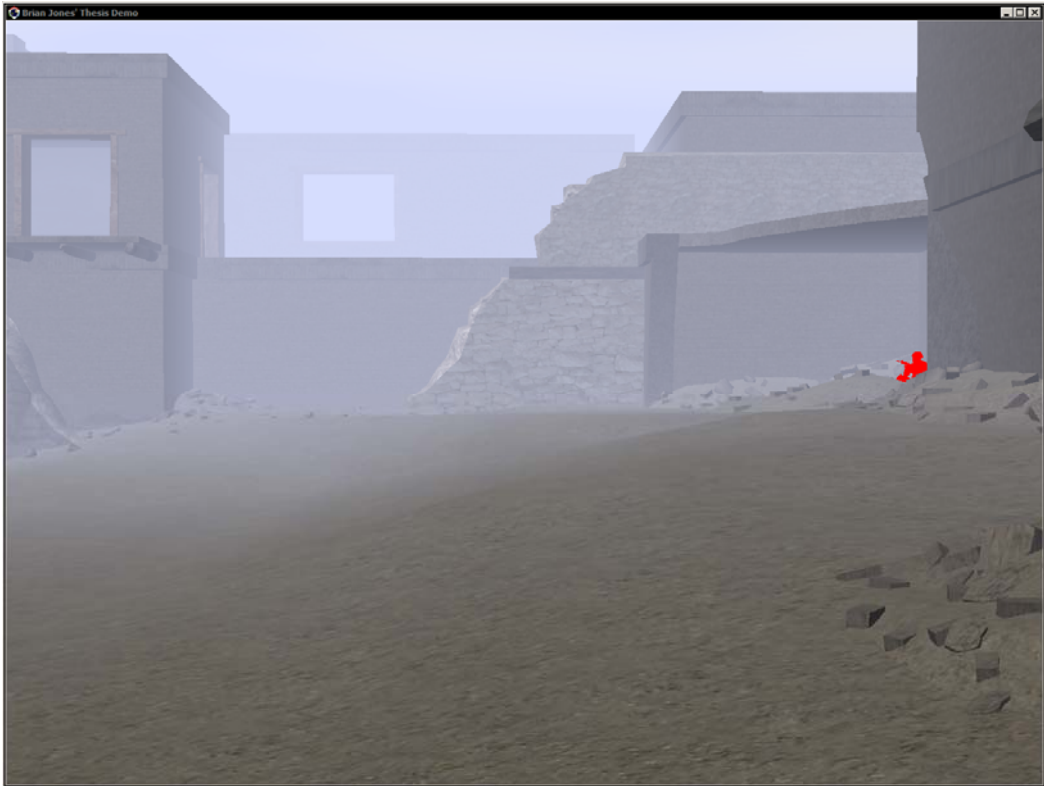
29.   Scene 29

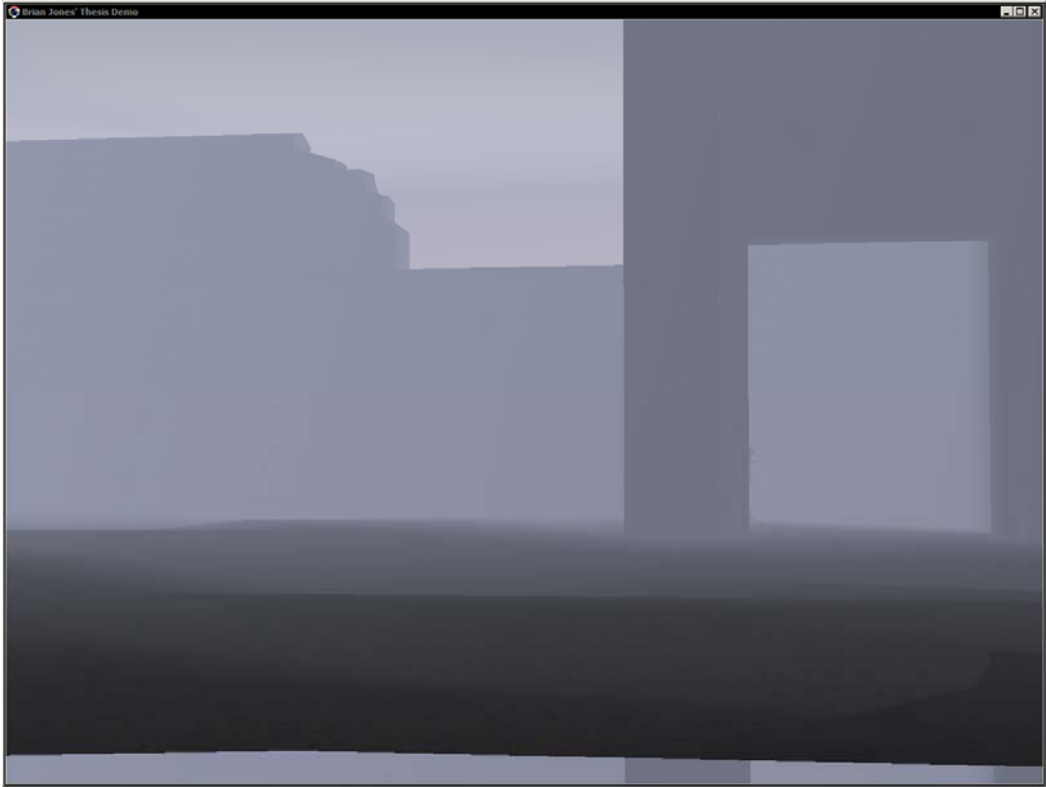## 30.   Scene 30

31.   Scene 31

32. Scene 32

33.   Scene 33

34.   Scene 34

35.   Scene 35

36.   Scene 36

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C

A. VARIATIONS BETWEEN RUNS

The following are the full tables showing the variations in probability of detection (P(det)) between runs.

1. Before Offset Applied

Scene Number	P(det)	P(det)	P(det)	P(det)	P(det)	Standard Deviation
1	0.614644	0.614644	0.613325	0.614644	0.614644	0.000589875
2	0.140689	0.155815	0.156713	0.154621	0.14494	0.007262697
3	0.023084	0.023084	0.0233719	0.023084	0.023084	0.000128753
4	0	0	0	0	0	0
5	0.997602	0.997601	0.997602	0.997601	0.997602	5.47723E-07
6	0.999951	0.99995	0.999951	0.99995	0.999951	5.47723E-07
7	0.000100634	0.000100634	0.000100634	0.000100634	0.000100634	1.51522E-20
8	0.836703	0.836703	0.836703	0.836703	0.836703	0
9	1	1	1	1	1	0
10	0.955698	0.955698	0.955698	0.955698	0.955698	1.24127E-16
11	0.00124682	0.00129148	0.00122026	0.00129148	0.00124682	3.12518E-05
12	1	1	1	1	1	0
13	0	0	0	0	0	0
14	0.992746	0.992441	0.993015	0.993008	0.992427	0.00028899
15	0.676234	0.676298	0.676234	0.676298	0.676234	3.50542E-05
16	0.999979	0.999979	0.999979	0.999979	0.999979	0
17	0.0305427	0.0305938	0.0305938	0.0305938	0.0305427	2.79886E-05
18	0.99875	0.99875	0.99875	0.99875	0.99875	0
19	0.847204	0.847405	0.847204	0.847405	0.847204	0.000110092
20	0.000388496	0.000378397	0.000378397	0.000378397	0.000388496	5.53145E-06
21	0.784245	0.785514	0.78488	0.785514	0.784245	0.0006345
22	0.999949	0.999949	0.999949	0.999949	0.999949	0
23	7.98E-06	7.50E-06	1.13E-05	7.50E-06	7.98E-06	1.60741E-06
24	0.0653888	0.0647576	0.0653888	0.0647576	0.0653888	0.000345722
25	0.927046	0.927046	0.927046	0.927046	0.927046	0
26	1	1	1	1	1	0
27	0.148125	0.14778	0.146964	0.14778	0.148125	0.000474534
28	0.785641	0.78548	0.785458	0.78548	0.785641	9.26364E-05
29	0.298999	0.299388	0.299228	0.299388	0.298999	0.000195111
30	0.000171252	0.000192017	0.000171252	0.000192017	0.000171252	1.13735E-05
31	0.832464	0.832158	0.832327	0.832158	0.832464	0.000153167
32	0.999996	0.999996	0.999996	0.999996	0.999996	0
33	0.170004	0.170295	0.170295	0.170295	0.170004	0.000159387
34	0.0540788	0.05381	0.0538681	0.05381	0.0540788	0.000138664
35	0.961959	0.961778	0.961959	0.961778	0.961959	9.91378E-05
36	0.552963	0.553112	0.553112	0.553112	0.552963	8.16107E-05

2. After Offset Applied

Scene Number	P(det)	P(det)	P(det)	P(det)	P(det)	Standard Deviation
1	0.0159053	0.0159053	0.015843	0.015843	0.0159053	3.41231E-05
2	0.00288597	0.00371093	0.00259835	0.00322949	0.00336772	0.000430706
3	0.317275	0.316621	0.319839	0.319839	0.317275	0.001546979
4	0	0	0	0	0	0
5	1	1	1	1	1	0
6	0.162172	0.161739	0.161626	0.161626	0.161344	0.000300825
7	0.000491963	0.000723809	0.000723809	0.000723809	0.000723809	0.000103685
8	1	1	1	1	1	0
9	0.997635	0.997635	0.997635	0.997635	0.997639	1.78885E-06
10	1	1	1	1	1	0
11	0.021845	0.0218938	0.0214339	0.0214339	0.0222609	0.000349213
12	0.983721	0.98377	0.983755	0.983755	0.983739	1.86548E-05
13	0	0	0	0	0	0
14	0.999861	0.999859	0.999848	0.999861	0.999858	5.41295E-06
15	0.985057	0.985044	0.985044	0.985044	0.985051	5.87367E-06
16	1	1	1	1	1	0
17	0.471962	0.471962	0.472531	0.472531	0.471962	0.000311654
18	1	1	1	1	1	0
19	1	1	1	1	1	0
20	0.00653111	0.0067007	0.00653111	0.00653111	0.00653111	7.5843E-05
21	0.028535	0.028535	0.0286025	0.0286025	0.0286025	3.69713E-05
22	1	1	1	1	1	0
23	2.91E-07	1.51E-07	2.14E-07	2.14E-07	1.16E-07	6.74546E-08
24	0.00303765	0.00303765	0.00303765	0.00303765	0.00303765	0
25	1	1.00E+00	1	1	1	0
26	0.972731	0.972733	0.972733	0.972733	0.972711	9.65401E-06
27	0.00196577	0.00195755	0.00194117	0.00194117	0.00193616	1.26471E-05
28	0.999973	0.999973	0.999973	0.999973	0.999973	0
29	0.948842	0.948891	0.948977	0.948977	0.948953	5.95651E-05
30	5.64E-06	5.14E-06	5.14E-06	5.14E-06	5.14E-06	2.23526E-07
31	0.998823	0.998826	0.998824	0.998824	0.998822	1.48324E-06
32	0.612579	6.13E-01	0.612602	0.612602	0.612712	0.0001213
33	0.985153	0.985122	0.985184	0.985184	0.985153	2.59365E-05
34	1	1	1	1	1	0
35	0.0781587	0.0781642	0.0781642	0.0781642	0.0778739	0.000129233
36	0.996371	0.996361	0.996366	0.996366	0.996371	4.1833E-06

APPENDIX D

A.   SETTING UP NEW SCENES

At the beginning of the demo.cpp source file there is an array of target and camera values that are used to set up scenes in the environment. There are currently 87 sets of target and camera values that can be used to see new scenes.

In order to set up entirely new scenes the following steps should be completed.

1.   Setting up the Fly Motion Model

The fly motion model, used to move the camera around the environment, is currently commented out of the code, keeping the camera from inadvertently from moving around during the experiment. To start it back up, remove the line comment marks in front of the following lines, around line number 240:

FlyMotionModel *fmm=new FlyMotionModel(GetKeyboard(), GetMouse());

fmm->SetTarget(this->GetCamera());

To move the camera around, the keyboard's arrow keys, along with the 'W' and 'S' keys are used. The up arrow key will pitch the camera up. The down arrow will pitch the camera down. The left and right arrow will turn the camera left and right, respectively. The 'W' key moves the camera forward in the scene, while the 'S' key moves the camera backward.

The mouse, while the left button is depressed can also be used to turn and pitch the camera, while the 'W' and 'S' are used as before.

2.   Moving the Character

The walk motion model was left active during the experiment to allow the researcher to give a character demonstration to the subjects. The 'I' key is used

to walk the character forward. The 'J' and 'L' keys turn the character left and right, respectively.

### 3.	Posing the Character

The simulation allows positioning the character in the following poses; standing, crouching, kneeling and laying prone. To cause the character to crouch, the 'C' is used. The 'D' will make the character kneel while the 'X' will lay the character down in the prone position. To cause the character to stand up straight, the 'E' key is used. Originally the 'S' key was used, but with the fly motion model active, the 'S' key moved the camera while making the character change positions.

### 4.	Lighting and Fog

The lighting, the scenes time of day, is tied to a keyboard's number keys. From '0' to '9' goes from 0200 to 2200, in increments of 2 hours.

Fog can be changed using the number pad's '-' and '+' keys. Fog levels are incremented or decremented by 2.

### 5.	Setting Up the Scene

In order to set up interesting scenes, the character must be walked around the environment, while following with the camera. If the character is lost in a wall or somewhere in the scene where he can't be seen, the 'F1' key can be used to position the character and the camera back in the original scene. The 'F1' key, however, does not change the character's pose, time of day or fog levels in the scene.

Once a scene has been determined, pressing the 'F2' and 'F3' buttons will save the camera and target position information to the file camTargPos.txt. Those position information blocks can then be added to the array of target and camera values.

LIST OF REFERENCES

Correia05   Correia, Joaquin Steve. *Agent-Based Target Detection in 3-Dimensional Environments*. Naval Postgraduate School Masters Thesis. 2005.

Darken04   Darken, Christian J. *Visibility and Concealment Algorithms for 3D Simulations*. Behavior Representation in Modeling and Simulation Conference. 2004.

Kane04   Kane, Barbara L., Page, Gary L., Wilson, David L., Bohan, David J. *Cycle Criteria for Detection of Camouflaged Targets*. Proceedings of the RTA-NATO Systems Concepts and Integration Symposium. 2004.

Pursel04   Pursel, Eugene R. *Synthetic Vision: Visual Perception for Computer-generated forces Using the Programmable Graphics Pipeline*. Naval Postgraduate School Master's Thesis. 2004.

Rabin02   Rabin, Steve. AI <u>Game Programming Wisdom</u>. Charles River Media, Inc. 2002.

Reece96   Reece, Douglas A. and Wirthlin, Ralph. *Detection Models for Computer Generated Individual Combatants*. Proceeding of the 6th Conference on Computer-generated forces and Behavioral Representation. 1996.

MSTAMOUT   U.S. Army. *Modeling of Search and Target Acquisition in a MOUT Environment.* Accessed on 22 June 2006 at http://www.moutfact.army.mil/

NVESD   U.S. Army Night Vision and Electronic Sensors Directorate. Standard Category ACQUIRE Approved Standard.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
    Ft. Belvoir, VA

2. Dudley Knox Library
    Naval Postgraduate School
    Monterey, CA

3. Anthony Ciavarelli
    Naval Postgraduate School
    Monterey, CA

4. MAJ John Alt
    TRADOC Analysis Center – Monterey
    Naval Postgraduate School
    Monterey, CA