Computer Graphics-Based Target Detection for Synthetic Soldiers

Christian J. Darken
Brian E. Jones, LT USN
Naval Postgraduate School
Monterey, CA 93943
831-656-2095
cjdarken At nps Dot edu

ABSTRACT: *The information provided to a software agent fundamentally affects its behavior. It is a trivial observation that an agent cannot respond to an environmental stimulus of which it is not aware. In a similar vein, providing an agent with information that a human participant in the simulation would not be aware of in the same circumstances may result in inaccurate agent behavior. In 3D virtual simulations, the most basic information provided to an agent concerns what battlefield entities that they can see. The standard approach used in 3D simulations with high visual fidelity, such as video games, is to use a line-of-sight (LOS) trace between entities to determine if they can see each other. LOS is arguably a very poor model of target detection, particularly in its failure to take camouflage, smoke, and darkness into account. In this paper, we describe a new approach that is an adaptation of a standard target acquisition model, to the domain of high visual fidelity simulations rendered on conventional graphics hardware. The new approach avoids the problems of LOS. We describe two variants of the approach, compare their predictions to human performance, and characterize their remaining deficiencies.*

1. Introduction

The information provided to a software agent fundamentally affects its behavior. It is a trivial observation that an agent cannot respond to an environmental stimulus of which it is not aware. In a similar vein, providing an agent with information that a human participant in the simulation would not be aware of in the same circumstances may result in inaccurate agent behavior. In 3D virtual simulations, the most basic information provided to an agent concerns what battlefield entities that they can see.

In this work, we define and analyze two adaptations of a standard target acquisition model (ACQUIRE) to the domain of "video game-quality" high visual fidelity 3D simulations. Furthermore, both adaptations are tested against human performance data. This work represents the first time that we are aware of in an ACQUIRE variant has been tested with comparably accurate size and brightness data, which would be very difficult to do with real world data. An additional point of interest is that the computational burden of producing this data is on the GPU (Graphics Processing Unit), instead of the CPU, as with the most closely related previous work, described below.
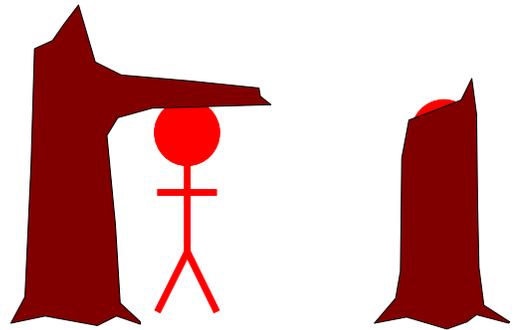
Figure 1: To LOS, the figure at left is invisible since the line to the top of the figure's head is blocked, while the figure at right, completely obscured except for the top of the head, is visible.

2. Background and Related Work

The standard approach used in 3D simulations with high visual fidelity, such as video games, is to use a line-of-sight (LOS) trace between entities to determine if they can see each other. This line is traced from the eye of the agent to the top of the target. If the line is unobstructed, it is assumed that the agent sees the target. As illustrated in Figure 1, LOS-driven target detection can be very different to that of humans. Additionally, LOS easily detects camouflaged, smoke-

obscured, and shadowed targets that humans find impossible to see. In a training or game context, LOS results in agents that either fail to engage obvious targets or engage targets that are invisible to the human participants in the simulation.

Darken (2004) describes three alternatives to simple Line-Of-Sight of varying degrees of accuracy and computational complexity: a multi-trace version of line-of-sight, a raster algorithm, and an exact calculation of visible surface.

The game industry and military sims traditionally use a line of sight to determine visibility (Richbourg, 1996) (Liden, 2002).

ACQUIRE is a standard target acquisition model (Reece, 1996). It is capable of producing the probability of detecting a target in a given amount of observation time. ACQUIRE computes the detection probability as a function of the brightness (irradiance in watts per square meter) of the target, the brightness of the background of the target, and the subjective size of the target, in terms of its "number of resolvable cycles". While not recommended as a model of optical target detection, it has been used for that purpose many times in military simulations. Constructive simulations, lacking detailed 3D models of combatants and terrain, obviously must use nominal brightness values, for example using a constant brightness for the background depending on geographic location and time of day, and looking up target brightnesses by target type from a table.

The Compact Terrain Database library (libctdb) describes a method of computing one of the key inputs to ACQUIRE, i.e. how much of a target is visible. It involves approximating the target and possible obstructing objects as 2D rectangles perpendicular to the field of view of the observer. By rasterizing the rectangles, the amount of the target that is visible can be estimated (Smith, 1993). In this work, we will use the actual shape of the object, terrain, and possible occluding objects as represented by a detailed 3D model.

Reece and Wirthlin adapted ACQUIRE for use as a model of the visual target acquisition of individual combatants (Reece, 1996). The Reece and Wirthlin model extended ACQUIRE to take target motion into account, but neglected other aspects of ACQUIRE, such as its ability to take target contrast into account, i.e. fixed brightness values were used corresponding to optimal viewing conditions. In this work, we will calculate a brightness value based on the same quality of image that is presented to a human viewer, taking lighting, camouflage, and smoke into account in a relatively accurate way.

The adaptations of ACQUIRE described in this work have not previously appeared in the form of a published paper, though they have been described in three Master's theses (Pursel 2004) (Correia, 2005) (Jones 2006). A description of the experiment and a preliminary analysis of the results appeared in the latest of these theses (Jones 2006).

3. Target Detection Models

In this section, the three target detection models that we will discuss in this work will be defined, namely LOS (Line-Of-Sight), and FBA and GBBA, our two adapted versions of ACQUIRE.

3.1 Line-Of-Sight (LOS)

The LOS model simply traces a ray from the eye of the agent to the top of the target. If this ray is unobstructed, i.e. it does not intersect any polygons other than those belonging to the target, then there is a 100% certainty that the agent detects the target. If the ray is obstructed, the agent has a 0% chance to detect the target, i.e. it never detects the target.

3.2 Framebuffer-Based Acquire (FBA)

FBA is an application of ACQUIRE to simulations that are being rendered on conventional graphics cards. It is applicable to any simulation that has full control of the visual rendering of the models and access to the framebuffer (the data object corresponding to the pixels displayed on screen).
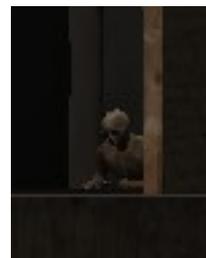
Figure 2: Wide shot of target in a building and partially visible through a window.

FBA's computation is based on a rendering of the target from the agent's point of view that is generated in the same manner as the rendering of the environment that a human user of the simulation sees. See Figure 2 for an example. The angles and colors of lights and the textures (bitmaps) applied to the surface are all taken

into account, together with any occluding objects, smoke, and fog. For each agent/target pair, a rendering of the target from the agent's point of view is produced. Since only a tight view of the target is required, we call this image a "mini-render". This image does not ever need to be shown on a screen; its sole purpose is to provide the data that will feed FBA's algorithm.

Figure 3: Normal color mini-render (left), false color mini-render (right).

The mini-render contains complete information on the appearance of the target and background, but ACQUIRE requires us to separate the two. Segmentation of images into objects as practiced in computer vision is a computationally expensive and error prone operation. Fortunately, in this context we can get pixel-perfect segmentation at the cost of a second mini-render. For the second render, we use our control of the rendering to color the figure bright red. This false-color mini-render is generated for the same scene conditions as the first, and so is in perfect registration with it. We use the false-color mini-render to tighten the view of the target to the minimum rectangular window that includes all target pixels. To include a modest amount of background, we then expand this minimum rectangle by 5% in all directions. The normal color mini-render is cropped identically to maintain registration between the normal color and false color images. The images in Figure 3 have both been cropped according to this procedure.

ACQUIRE produces a probability of detection that is based on the brightness and size of the target, as well as the brightness of the target's background. Adapting ACQUIRE to this type of simulation requires specifying exactly what values will be provided to ACQUIRE in all circumstances.

ACQUIRE's requirement for target size information is a bit unusual. The subjective size of the target must be specified to ACQUIRE in terms of its "number of resolvable cycles". Imagine painting the target with alternating black and white bars and asking the observer to report the number of bars. Obviously, as the bars become very fine, the observer will eventually see them as a solid gray and be unable to count them.

The number of resolvable cycles of a target is based on the maximum number of bars that the observer can count before they become too fine. This number is then divided by two to represent the number of complete cycles, one cycle including both a white bar and its neighboring black bar.

Consider a target that is represented on the computer screen as a single row of pixels. If we assume that a human observer is placed sufficiently close to the screen, stripes that are one-pixel wide should be resolvable. We estimate the number of resolvable cycles as twice the number of pixels in the target, giving full benefit of the doubt to the observer's eye.

Most targets are not a single pixel wide, of course. For a square target, the number of pixels on an edge is the appropriate number, i.e. the square root of the number of pixels. Following Reece (1996), we use the square root of the number of pixels in all cases, regardless of target shape.

ACQUIRE requires the brightness (irradiance) of both target and background in Watts/m^2. This is problematic for two reasons. First, while we assume we have access to the framebuffer, and thus to the color values of every pixel of an image that is actually or potentially displayed to the screen, those color values are not in units of Watts/m^2. Secondly, while we assume access to all of the pixel color information contained in the framebuffer, ACQUIRE wants only one number to characterize the brightness of the target and background.

Using camera light meters, we have empirically determined that the irradiance of a pixel in Watts/m^2 is well approximated to within a multiplicative constant by the sum of the pixel color values squared. Since ACQUIRE depends only on ratios of irradiance values, we can therefore plug in squared sum-of-color values wherever an irradiance value is called for.

For FBA, we simply take the average sum of squares of the pixel color values over the all target and background pixels in the mini-render as the input to acquire.

3.3 Graphics Buffer-Based Acquire (GBBA)

GBBA makes use of a second buffer on the graphics card in addition to the framebuffer, namely the depth buffer. During the normal process of rendering a graphical image based on 3D models, a graphics card produces a matrix of depth values, one per pixel, that indicate how far away from the camera the part of the model that is imaged in the pixel is. The depth values

range between 0 and 1, where 1 represents the far clipping plane of the image, beyond which nothing will be rendered.

The insight behind GBBA is that not all contrast between target and background is equally valuable. Where the "background" is actually closer to the observer than the target, high contrast means that we see the contour of the background well, but not necessarily the target.

GBBA uses the depth buffer to exclude any part of the background that is in front of the target from consideration. It does this by first making a pass over all the target pixels (identified using the false-color mini-render) and finding the one that is closest to the camera. Then a pass over the non-target pixels in the false-color mini-render colors them depending on whether how far they are relative to the closest target pixel. In the example presented as Figure 4, the background pixels that are closer than the target, the edge of the window to the right and below the target, are colored blue.
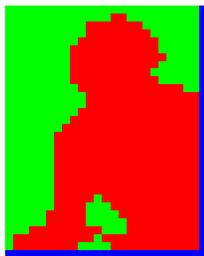
Figure 4: False color mini-render including depth buffer information. The blue pixels on the right and bottom of the image are closer to the camera than the target.

4. Experiment Design

There is no need for an experiment to compare FBA and GBBA to LOS with regard to problems such as the ones discussed in and around Figure 1. The main problems with LOS come from the fact that it relies upon rays to determine target detectability. Computation of visible area via mini-renders solves these problems. However, it remains to be seen whether FBA or GBBA is the superior choice and how either compare with human performance.

We designed an experiment to measure how accurately the algorithms designed above model human target detection performance. It is interesting to note that if we were trying to model battlefield perception, our technique of using computer graphics imagery as the basis for the experiment would require defense. Our actual goal, however, is to eventually construct synthetic soldiers that will replace human participants in the simulation. For the pursuit of this goal, human perception in the virtual world of the simulation is the gold standard.

Steve Corriea's original system (Correia, 2005) was used as the backbone for the system to test the algorithms. This system provided the basic infrastructure for camera and figure placement and manipulation of lighting and fog. 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 believed the figure was not in the scene or was too hard to see by pressing the space bar.
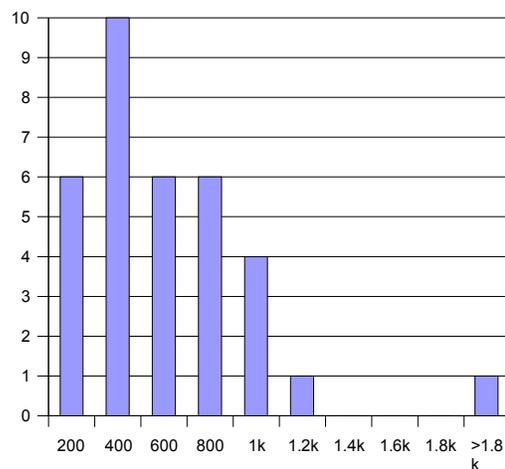
Figure 5: Histogram of figure sizes for the set of scenes included in the experiment, measured in pixels.

A set of 36 scenes were created by varying a desert camouflage-clad character's posture amongst standing, crouching and prone, and varying the lighting and amount of fog. The character was not visible in two of the scenes. A projector was used to present the images to the subjects in the hopes to minimize differences in image quality due to slight variations in subject position. Scenes were not selected randomly, but were selected because they seemed interesting or because we found the predictions of the FBA algorithm, the only one implemented at the time, on that particular scene to be unintuitive. Two scenes were added specifically to investigate ACQUIRE's handling of colored targets. In these scenes, the targets were colored bright red and presented to the subjects that way (not to be confused with false coloring simply to produce values to feed the ACQUIRE algorithm). The range of sizes of the figures in the scene and their expected detection difficulty are given as Figures 7 and 8.

Subjects used a computer mouse to identify the figure in the scene indicating they found the hidden figure. If the character cannot be seen, the subject was instructed to press a key on the keyboard, indicating that the figure was not detectable to the subject. No time limit was given for each scene, allowing the subject to take sufficient time to locate the figure. To try to discourage subjects from guessing, we included two scenes containing no figure at all.
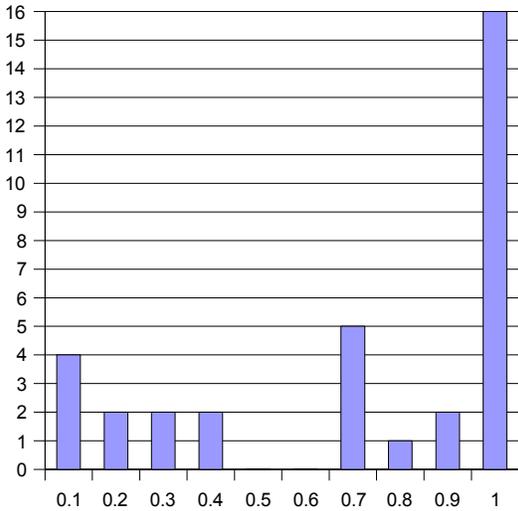
Figure 6: Histogram of GBBA detection probabilities over the set of scenes.

5. Results

22 subjects were run through the experiment. Whether or not they found each target and the time they took to respond was recorded. 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.

In order to test the FBA and GBBA algorithms, we took the subjects' responses on any single scene to be independent Bernoulli trials. The null hypothesis is that the algorithm's probability of detection given unbounded time is correct. Since the subjects' responses are clearly independent trials, the null hypothesis corresponds to a binomial distribution. We did a two-tailed test. The power of the test was computed. Results are displayed as Figure 7. All of the columns except that at the far right indicate algorithm failure of greater or lesser degree. Both

algorithms fail on most of the scenes. Nonetheless, if we take the inability to reject the null hypothesis at 0.05 to be adequate performance, GBBA performs adequately on ten scenes versus five for FBA, which is quite an improvement.
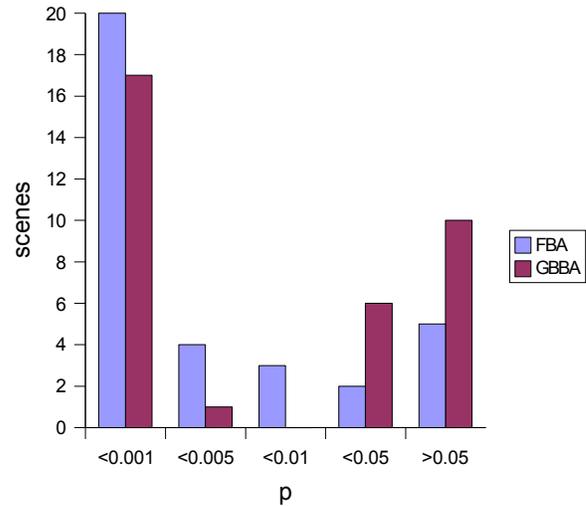
Figure 7: Histogram of test power over all 34 scenes with a visible target (the target was not visible in two scenes) for both the FBA and GBBA models. Larger p (test power) values correspond to better model performance.
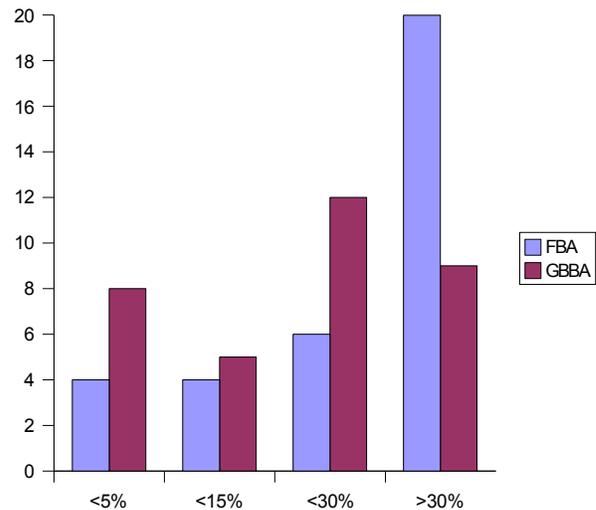
Figure 8: Histogram of L1 values for FBA and GBBA differences from the experimental data (smaller is better). For example, the 15-30% bin contains those scenes for which the predicted detection rate was off by between 15 and 30 percentage points.

In our opinion, the histogram of test powers does not represent the final word in the analysis of these models. The models predict values close to one or zero in many

cases. Such predictions leave little or no room for error, and result in p values that are small or zero. A less strict measure of model performance is the L1 distance from the model prediction to the experimental data for each scene, i.e. the absolute value of the error in detection probabilities expressed as percentages between 0% and 100%. The sum of the L1 distances over all scenes has a value of 17.1 for FBA and 12.1 for GBBA, again showing considerably better performance by GBBA. A histogram of L1 values for the individual scenes is given as Figure 8.

Comparing scene by scene for all scenes with target, GBBA has a better test power than FBA in 19 out of 32 scenes (59%) , and a better L1 score in 20 out of 32 scenes (62%).

6. Analysis

The results show that, while GBBA seems better than FBA, both perform poorly on many of the scenes in the experiment. What is the cause of the remaining problems? We made a careful pass through the scenes GBBA had trouble with, and created the following list of issues.

6.1 Averaging

The fact that ACQUIRE uses only a single number to describe the intensity of the target and background is obviously a great simplification. Both target and background generally extend over many pixels, and have a great degree of variation in brightness in some scenes. Consider an image where a gray figure stands before a background that is half white and half black. Averaging the background intensity to produce a single number may result in ACQUIRE returning a zero probability of detection, even though the figure is glaringly obvious.

Figure 9: Contrast-enhanced normal-color image at left showing a highlight on the helmet that raises the average brightness of the target enough to confuse the algorithm. Corresponding false-color image at right.

While somewhat less intuitive, averaging can also result in detection probability estimates that are too high. Imagine a gray soldier perfectly blended with his

gray background, except for an oddly-shaped highlight, as if Figure 9. If the highlight is very bright, it may result in raising the average intensity of the target to a significant extent.

6.2 Color

It is intuitively obvious and verified by experiment that color can play a role in target detection (see Figure 10). Color is ignored by ACQUIRE.

Figure 10: Although the color of the target is a striking contrast to the background, its intensity is similar, leading ACQUIRE to predict a nearly zero (.0004) detection probability. Note that this is a detail of the actual image presented to the subjects and not a false color image!

6.3 Shape and Texture

Shape and texture are also not considered by ACQUIRE. In some of the scenes, even though the overall contrast of the target was poor, the target displayed a lot of characteristic shape or texture, for example by being slightly rim-lit, as in Figure 11.
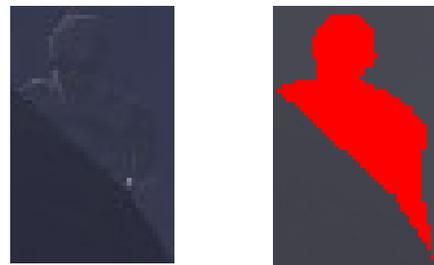
Figure 11: Contrast-enhanced normal color image at left and corresponding false-color at right. The rim lighting exposed the shape of the target, making it easy to pick out despite the lack of contrast in the original image.

6.4 Clutter

ACQUIRE depends only on the target and its background. Human vision can be greatly influenced by distractor objects in the field of view, i.e. "clutter". Many of the scenes contained a great deal of clutter. The overall rate of "false positive" detections was slightly over 10%, but had a maximum value of 36% on one scene. An example of a cluttered scene is given as Figure 12.

Figure 12: The clutter at the bottom of this image distracted subjects from the figure. Normal color at left, and false color at right.

6.5 Pose and Framing

In many instances where the ACQUIRE prediction was high, the difficulty seemed to be that the pose or framing of the target was unusual. The worst offenders in this regard were scenes where the target was prone and only the legs or feet were showing, as in Figure 13.

Figure 13 The unusual framing of the prone subject's legs by the doorway made this target difficult for the experiment's subjects to detect. Normal color at left, false color at right.

6.6 Dimness

Despite the fact that ACQUIRE is fundamentally based on contrast, and so this should be the strong point of the approach, we felt that ACQUIRE overestimated the detection probability when the targets were very dim.

7. Conclusions and Future Work

Both FBA and GBBA presented clearly avoid the most extreme problems of line-of-sight detection. This is obvious simply from the algorithm definitions; no experiment is required. The results of the study indicate that GBBA, which defines as background only those pixels that are further away than the target, is superior to FBA, which takes all pixels around the target as the background. Note that because the scenes selected for the experiment were intended to be challenging for the algorithms, the algorithms probably would perform much better on "typical" scenes from the course of a simulation run. The experiment was valuable primarily because it provides an indication that GBBA is the superior algorithm, and to isolate remaining deficiencies in GBBA that can be addressed in future work.

We believe that GBBA can be extended to solve some of its difficulties outlined in the section above. The averaging problem, as well as the lack of sensitivity to color and texture, could potentially be solved with small changes to the algorithm. Clutter, on the other hand, clearly requires larger changes. No amount of improvement in the processing of our mini-renders will account for the presence or absence of clutter outside the mini-render.

Finally, we note that the methodology of this study, while tuned to constructing perceptual models for software agents, could be adapted to improving ACQUIRE taken as a model of real (versus virtual) battlefield perception. It is interesting to consider using relatively inexpensive experiments in a virtual environment to modify ACQUIRE, afterwards using the minimum number of more expensive experiments with actual targets and terrain to validate the changes.

8. Acknowledgments

9. References

Correia, S. (2005) Agent-Based Target Detection in 3-Dimensional Environments. Master's Thesis, Naval Postgraduate School.

Darken, C. and Paull, G. (2006). Finding Cover in Dynamic Environments. In S. Rabin (Ed.), *Game AI Programming Wisdom 3*. Charles River.

Darken, C. (2004a). Visibility and Concealment Algorithms for 3D Simulations. In *Proceedings of Behavior Representation in Modeling and Simulation*.

Darken, C., Pursel, E., and Correia, J. (2004b) AI on the GPU. Extended abstract for poster presentation at the *ACM Workshop on General Purpose Computing on Graphics Processors* and SIGGRAPH.

Jones, B. (2006) A Computer Graphics Based Target Detection Model. Master's Thesis, Naval Postgraduate School.

Liden, L., "Strategic and Tactical Reasoning with Waypoints." AI Game Programming Wisdom. pp 211-220. Charles River Media, 2002.

Pursel, E. R. (2004) Synthetic Vision: Visual Perception for Computer Generated FOrces Using the Programmable Graphics Pipeline. Master's Thesis, Naval Postgraduate School.

Reece, D. and Wirthlin, R. (1996). Detection Models for Computer Generated Individual Combatants. In *Proceedings of the 6th Conference on Computer Generated Forces and Behavioral Representation*.

Richbourg, R., and Olson, W., "A Hybrid Expert System that Combines Technologies to Address the Problem of Military Terrain Analysis," Expert Systems with Applications, Vol. 11, No. 2, pp. 207—225, 1996.

Smith, J.E. (1993). Compact Terrain Database Library User Manual and Report. Programmer's Reference Guide, ModSAF Documentation.

Appendix 1: Model Details

Let T be the set of pixels comprising the target and B the set of pixels comprising the background. Let n be the number of pixels in T, and for each pixel, let r(p),

g(p) and b(p) be the red, blue, and green color values of that pixel. The results above are based on brightness (irradiance) estimated as the square of the average pixel values, but it is more correct to take the brightness of the target to be

$$B_T = \frac{1}{n_T} \sum_{p \in T} r(p)^2 + g(p)^2 + b(p)^2$$

The brightness of the background, B_B, is computed analogously. Then the target/background contrast is taken to be

$$C = \frac{|B_T - B_B|}{B_B}$$

Then defining

$$N = \sqrt{n_T}$$

and

$$E = 2.7 + 0.7\, C\, N / N50 \quad,$$

where *N50* is a constant scaling perceptual acuity which we take to be 1.0 following Reece (1996). Then the asymptotic probability of detection given an arbitrarily long time to find the target is given by

$$P = \frac{(N / N50)^E}{1 + (N / N50)^E}$$

Author Biographies

CHRISTIAN J. DARKEN, PhD, is an Associate Professor of Computer Science at the Naval Postgraduate School, where he also collaborates intensively with the MOVES Institute. Previously he was Project Manager of the Decision Support Systems project and Senior Member of Technical Staff at Siemens Corporate Research in Princeton, NJ, where he was variously associated with the Learning Systems, Adaptive Information and Signal Processing, and Software Engineering Departments. He was also a programmer on the team that built Meridian 59, the first 3D massively-multiplayer game. He received his Ph.D. in Electrical Engineering from Yale University in 1993, and previously received the M.S. and M. Phil. in Physics from the same institution.

BRIAN E. JONES, LT USN is a recent graduate of the Master's program of the MOVES Institute of the Naval Postgraduate School in Monterey, California.