Computer Graphics-Based Models Of Target Detection: Algorithms, Comparison To Human Performance, And Failure Modes

Christian J. Darken
Naval Postgraduate School
Monterey, CA 93943
cjdarken@nps. edu

Target detection is one of the fundamental phenomena that must be modeled in military simulations. When the target detection model fails, entities that should not be mutually aware engage, and entities that should fight ignore one another. The potential negative consequences for training and analysis are obvious. We describe three closely related computer graphics-based detection models for virtual simulation that can avoid some of the limitations of previous approaches. These models incorporate a standard target detection model, but feed it with more accurate target exposure and contrast data than has been done previously. Two variants of the base model attempt to improve the target contrast calculation and add color sensitivity. We compare the predictions of these models to human performance to show that the model variants have their intended effect. The performance of even the best models can deviate drastically from the performance of the human eye under some circumstances represented in our experiment. We lump these into categories as an aid to understanding the state of the art and to motivate future research.

Keywords: target detection, synthetic vision

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. When an agent does not see hostile entities that it should be aware of, it will ignore them. If it does see hostile entities that should actually not be detectable, it may engage them. Both types of behavior can be highly disconcerting to human participants when they occur in a training situation, and can add systematic biases to analytic simulations. We believe that target detection models need a good deal of improvement to be adequate for current and near-term military training and analysis needs.

Our approach in this work is to use human performance in detecting targets while seated in front of a computer monitor to evaluate and improve target detection models. For “puckster automation” applications, where the goal is to replace human players in a simulation by software agents, this type of data is the gold standard. For other applications, where modeling of battlefield perception is the goal, it represents a starting point that can then be refined by human performance data from the field.

In this work, we define and analyze three adaptations of a standard target acquisition model (ACQUIRE) to the domain of “video game quality” high visual fidelity 3D simulations as are increasingly applied to modeling ground combat for purposes of training or analysis. One of these adaptations is the first to attempt to add color sensitivity to ACQUIRE. All three adaptations are tested against human performance data. This work represents the first time that we are aware of that 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.

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. As illustrated in Figure 1, LOS-driven target detection can be very different to that of humans. LOS traces a line 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 regardless of how little besides the top of the head is visible. Additionally, LOS easily detects camouflaged, fog-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. The game industry and military simulations traditionally use a line of sight to determine visibility [1] [2].

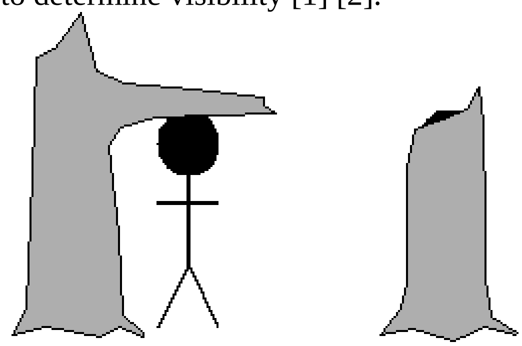

Figure 1: To LOS, the figure at left is invisible since the line to the top of the figure's head is blocked, while the figure at right, completely obscured except for the top of the head, is visible.

Darken [3] 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.

ACQUIRE is a standard target acquisition model [4]. 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 officially recommended as a model of optical target detection, it has been used for that purpose many times in military simulations, to the point where it seems reasonable to consider it a de facto standard. 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), developed for SIMNET but which has impacted many simulations, 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 [5]. 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 [4]. The Reece and Wirthlin model extended ACQUIRE to take target motion into account, which is outside the scope of this paper. However, they did not utilize some aspects of the base ACQUIRE model, 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 [6] [7] [8]. A description of the experiment and a preliminary analysis of the results appeared in the latest of these theses [8]. A prior conference paper described two of the algorithms presented in this paper, but not the third [9].

3. Target Detection Models

In this section, the three target detection models that we discuss in this work will be defined, namely LOS (Line-Of-Sight), and FBA and GBBA, our two adapted versions of ACQUIRE (see Table 1).

3.1 Line-Of-Sight (LOS)

The LOS model simply traces a line segment from the eye of the agent to the top of the target. If this line segment 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). The general architecture of FBA (also valid for the two variants described below) is pictured in Figure 4.

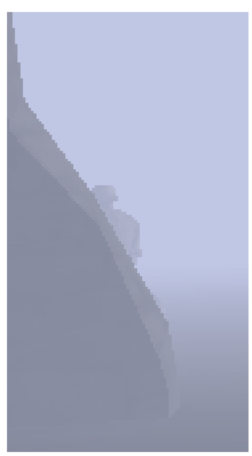

Figure 2: **Wide shot of target partially hidden behind a pile of rubble**.

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.

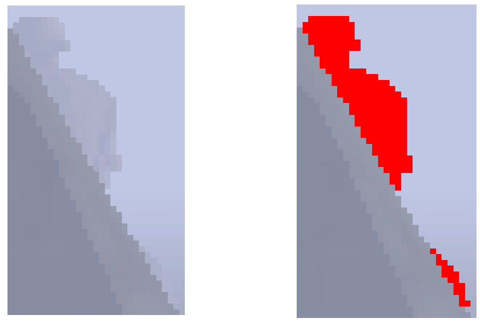

Figure 3: Normal color mini-render (left), false color mini-render (right).

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. Note that the application of texture (bitmaps) to the 3D models is not strictly necessary for the false color mini-render and may be omitted to save computation.

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 and Wirthlin [4], we use the square root of the number of pixels in all cases, regardless of target shape.

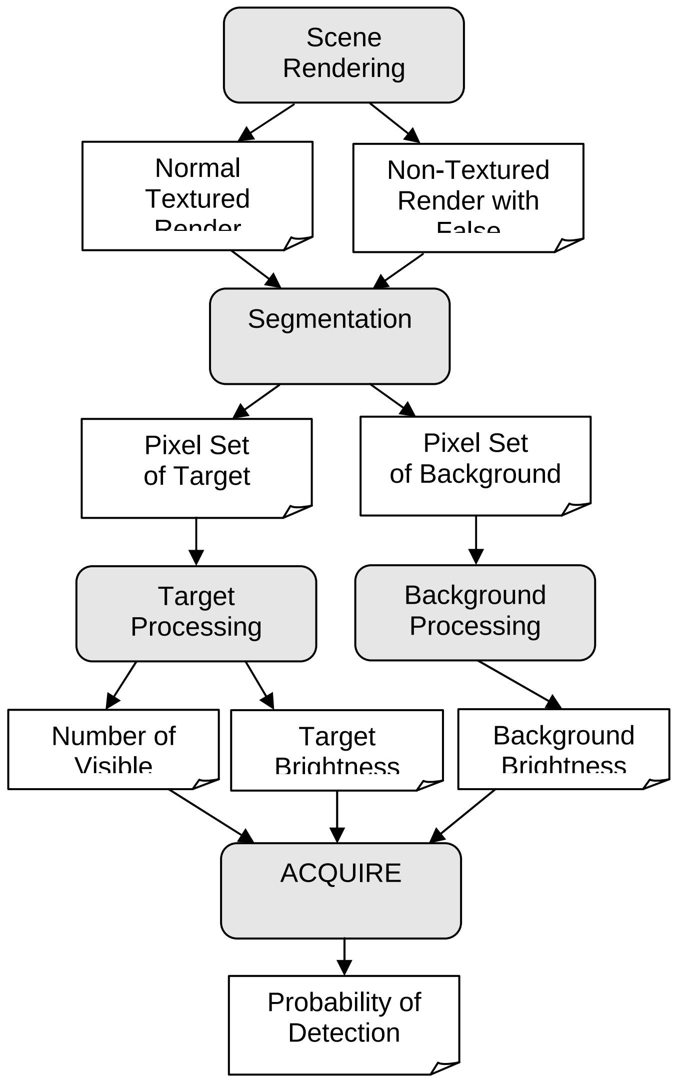

Figure 4: Detection algorithm data flow

ACQUIRE requires the brightness (irradiance) of both target and background in Watts/m2. 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/m2. 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 squared pixel color values. Since ACQUIRE depends only on ratios of irradiance values, we can therefore plug in sum-of-squared color values wherever an irradiance value is called for. Details of the model are provided in Appendix 1.

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. A monotonic nonlinear transformation is applied to the depth values as they are generated in the graphics card to provide enhanced depth value comparisons for near objects.

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.

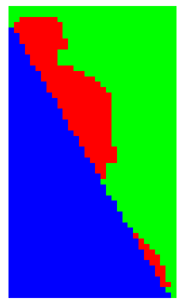

Figure 5: **False color mini-render including depth buffer information. The pixels on the left and bottom of the image are closer to the camera than the target**.

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 labels them depending on whether how far they are relative to the closest target pixel. In the example presented as Figure 5, the background pixels that are closer than the target, representing a pile of rubble the target is hiding behind, are colored blue.

3.4 Computer Graphics-Based Acquire (CGBA)

Computer Graphics-Based Acquire is based on GBBA, but with enhancements to make it sensitive to color. The basic ACQUIRE algorithm only cares about differences in brightness between the target and background. This makes sense given ACQUIRE's origins as a model of target detection using the earliest night vision equipment, which presumably had only monochrome displays. The unfortunate consequence is that ACQUIRE predicts a red target on a gray background to be undetectable if the intensity of the red matches the intensity of the gray. CGBA attempts to remedy this problem in a simple, pragmatic way by computing target/background contrast separately in the red, green, and blue color channels of the computer images and then averaging the results to produce an aggregated contrast value that is used to determine target detectability. In terms of the detailed model in Appendix 1, the brightness B_T and contrast C are computed for each color channel (e.g. red) as if the other color channels (e.g. blue and green) have value zero. The final contrast is computed as an average of the three color-specific contrast values. The rest of the model remains the same.

	Failure due to rectangular approximation of objects	Failure when contrast is with foreground object	Missed detections when there is contrast in color but not brightness
libctdb	x	x	x
FBA		x	x
GBBA			x
CGBA			

Table 1: Table motivating the various target detection models discussed above in terms of the problems they are intended to mitigate**.**

4. Experiment Design

We see little need for an experiment to compare the new algorithms 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 line tracing to determine target detectability. Computation of visible area via mini-renders solves these problems. However, the relative performance of the new models and how each compares with human performance requires study.

We designed an experiment to measure how accurately the algorithms designed above model human target detection performance. It is interesting to note that if our ultimate goal 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.

The system described by Correia [7] 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. It 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 could not detect any target by pressing the space bar.

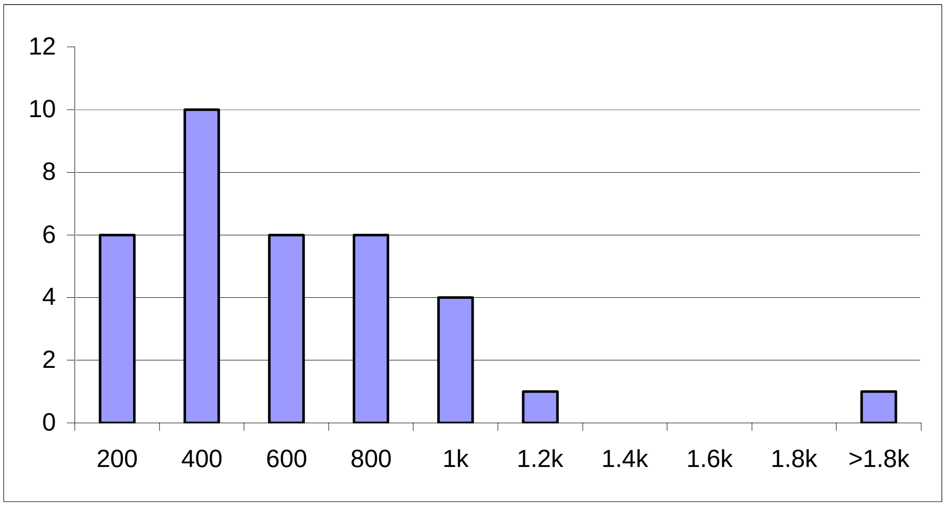

Figure 6: Histogram of figure sizes for the set of scenes included in the experiment, measured in pixels.

A set of 36 scenes was 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 rather 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. They were also intended to be challenging for human subjects, so the targets are typically small and the scenes foggy, dark, or both. The range of sizes of the figures in the scene and their expected detection difficulty are given as Figures 7 and 8. 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).

Subjects used a computer mouse to identify the figure in the scene indicating they found the hidden figure. If the target could not be seen, the subject was instructed to press a key on the keyboard. The subjects were instructed to favor accuracy over speed. No time limit was enforced, 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.

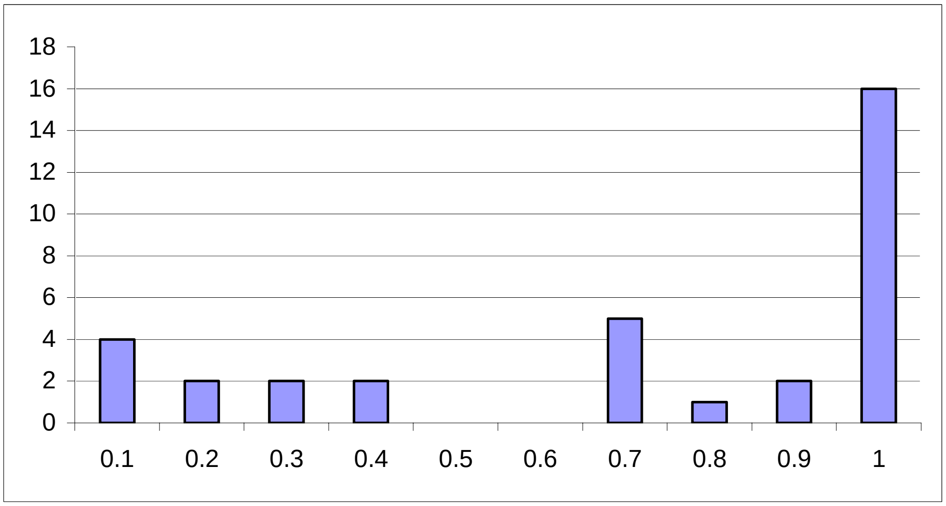

Figure 7: Histogram of GBBA detection probabilities over the set of scenes.

5. Results

Twenty-two subjects were run through the experiment. Whether or not they found each target and the time they took to respond was recorded. Twenty subjects were males while two were female. Seventeen were military members and five were civilians. 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.

To analyze the performance of the algorithms, we took an adversarial approach, whereby the correctness of each algorithm's predictions are treated as a null hypothesis that we attempt to reject. For each scene, the subjects' responses are independent Bernoulli trials, and are therefore binomially distributed. The significance value (p) for each scene was computed using a two-tailed test. Good performance by an algorithm corresponds to the inability to reject the null hypothesis with high confidence (low p), i.e. larger p means better algorithm performance. Results are displayed as Figure 8. 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 and CGBA perform adequately on ten scenes versus five for FBA, which is quite an improvement.

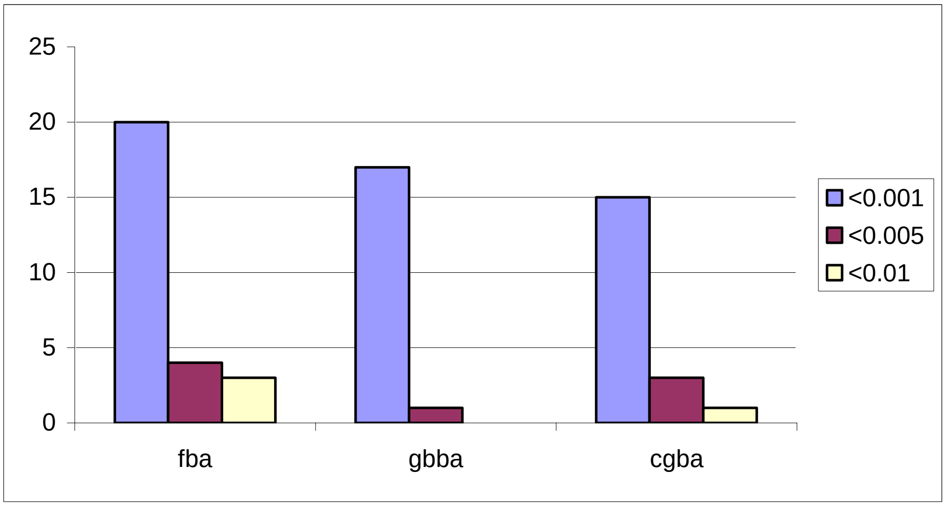

Figure 8: Histogram of test power over all 34 scenes with a visible target (the target was not visible in two scenes) for all models. Larger p (test power) values correspond to better model performance.

In our opinion, the histogram of test powers does not represent the final word in the analysis of these models. The models predict detection probabilities close to one or zero in many cases. Such predictions leave little or no room for error, and result in p values that are close to or equal to 1.0. 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 and CGBA. A histogram of L1 values for the individual scenes is given as Figure 9.

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%). CGBA performs comparably, with a better test power than FBA in 20 out of 32 scenes (62%), and a better L1 score in 22 out of 32 scenes (69%)

On the two scenes with a red target, FBA failed on both. Since the intensity of the red target was selected to match the target's background as defined by FBA, this is the expected result. GBBA succeeded on one, since the difference in how the background is defined provided some amount of intensity difference in the one case. Only CGBA, which is sensitive to color differences between target and background even in the absence of intensity difference, succeeded on both. Details are provided in Table 2.

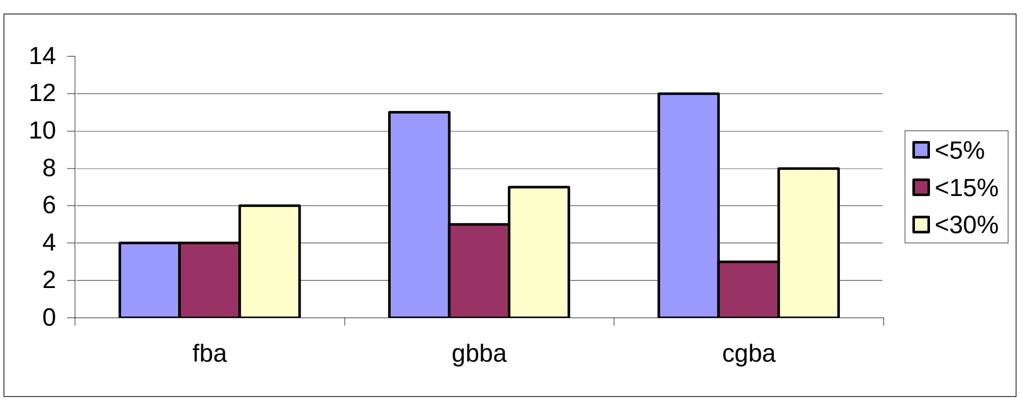

Figure 9: Histogram of L1 values for model differences from the experimental data (smaller is better). For example, the 15-30% bin contains those scenes for which the predicted detection rate was off by between 15 and 30 percentage points.

Scene ID	FBA	GBBA	CGBA	Experimental
17	0.03	1	1	1
20	0	0.13	1	1

Table 2: Performance on the scenes with a red target. Values are predicted and actual fractions of subjects that detect the target.

6. Analysis

The results show that, while CGBA and GBBA seem better than FBA, all versions of the model 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 CGBA had trouble with, and created the following list of issues.

6.1 Averaging

Figure 10: 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.

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.

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 in Figure 10. If the highlight is very bright, it may result in raising the average intensity of the target to a significant extent.

6.2 Shape and Texture

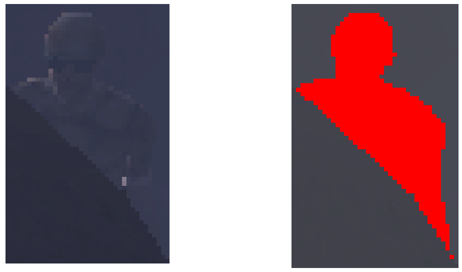

Figure 11: Contrast-enhanced normal color image at left and corresponding false-color at right. The rim lighting exposed the shape of the target, making it easy to pick out despite the lack of contrast in the original image**.**

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.

6.3 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**.**

6.4 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**.**

7. Conclusions and Future Work

All three models 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. CGBA performs about as well as GBBA, but is sensitive to color differences between the target and background that GBBA would miss. 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 CGBA is the superior algorithm, and to isolate remaining deficiencies in CGBA that can be addressed in future work.

We believe that CGBA 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 texture, could potentially be solved with improved processing of the mini-renders. 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

The author wishes to thank Doug Reece for very helpful discussions.

Partial support for this work was provided by grants from the Naval Modeling and Simulation Office and by the U.S. Army TRADOC Analysis Center, Monterey.

9. References

[1] Richbourg, R., and Olson, W., "A Hybrid Expert System that Combines Technologies to Address the Problem of Military Terrain Analysis." Expert Systems with Applications, 11 (1996): 207-225.

[2] Liden, L., "Strategic and Tactical Reasoning with Waypoints," in AI Game Programming Wisdom, edited by S. Rabin, 211-220, Boston: Charles River Media, 2002.
[3] Darken, C. "Visibility and Concealment Algorithms for 3D Simulations," in Proceedings of Behavior Representation in Modeling and Simulation 2004, http://www.sisostds.org/index.php?tg=articles&idx=More&article=477&topics=127.
[4] Reece, D. and Wirthlin, R. "Detection Models for Computer Generated Individual Combatants," in Proceedings of the 6th Conference on Computer Generated Forces and Behavioral Representation, 1996.
[5] Smith, J.E. Compact Terrain Database Library User Manual and Report. Programmer's Reference Guide. ModSAF Documentation, 1993.
[6] Pursel, E. R. "Synthetic Vision: Visual Perception for Computer Generated Forces Using the Programmable Graphics Pipeline." Master's thesis, Naval Postgraduate School, 2004.
[7] Correia, S. "Agent-Based Target Detection in 3-Dimensional Environments." Master's thesis, Naval Postgraduate School, 2005.
[8] Jones, B. "A Computer Graphics Based Target Detection Model." Master's thesis, Naval Postgraduate School, 2006.
[9] Darken, C. "Computer Graphics-Based Target Detection for Synthetic Soldiers," in Proceedings of Behavior Representation in Modeling and Simulation 2007, http://www.sisostds.org/index.php?tg=fileman&idx=list&id=2&gr=Y&path=CGF-BR%2F2007+BRIMS%2F2007+BRIMS+Full+Papers+and+Presentations.

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

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

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

$$E = 2.7 + 0.7CN / N50$$

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 Biography

Christian J. Darken, PhD, is an Associate Professor of Computer Science at the Naval Postgraduate

School, where he is an affiliate of 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, arguably 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.