Optimal Information Placement in an Interactive 3D Environment
[Paper vgsp_0020]

Priyesh N. Dixit and G. Michael Youngblood
The University of North Carolina at Charlotte
College of Computing and Informatics, Department of Computer Science
9201 University City Blvd, Charlotte, NC 28223-0001
{pndixit, youngbld}@uncc.edu

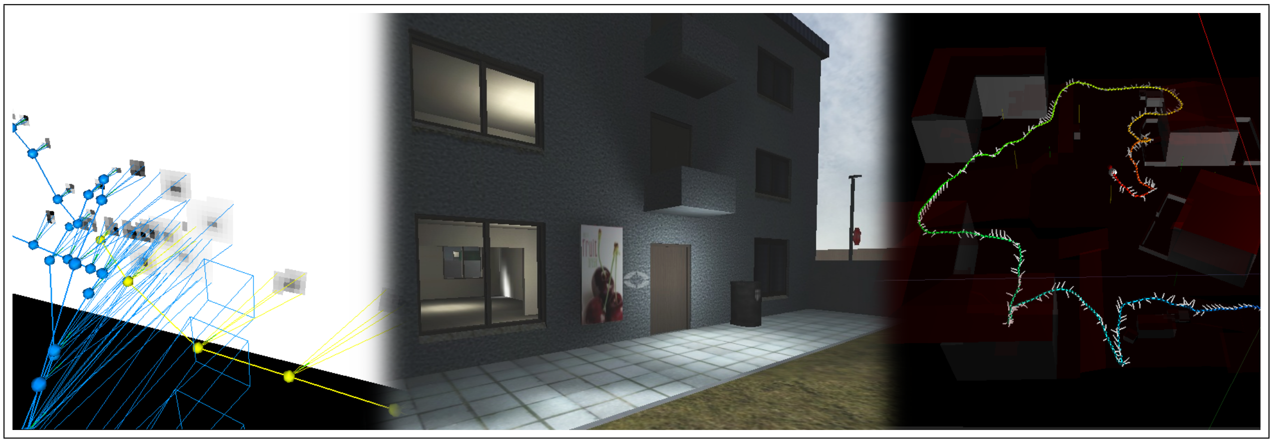


Figure 1: *The phases of information placement from left to right. First a corpus of existing player traces from play testing is used to calculate surface information value and local densities, then information artifacts are placed in areas of desired information value, and finally new players interact in the new environments observing and utilizing new information elements to the desired effect.*

Abstract

The correct placement of important artifacts and information in interactive three-dimensional (3D) environments is important to ensure that those key artifacts and information are seen and absorbed by the immersed user. This can include training information, advertisements, clues, interaction points, and other information that needs to be conveyed to or manipulated by the user. We propose a novel algorithm for calculating the optimal positioning of such artifacts and information based on a corpus of prior play testers, which are used to determine distance-weighted and radially focused observation densities on surfaces of interactive 3D environments. We have developed a tool called HIIVVE (Highly Interactive Information Value Visualization and Evaluation) which allows for interactive evaluation as well as offline processing of the information value surfaces. A user study involving information placement using the calculated information value surfaces and observation densities shows that higher valued locations do yield improved user observation by as much as 58.3%.

CR Categories: I.6.3 [Simulation and Modeling]: Applications; I.6.6 [Simulation and Modeling]: Simulation Output Analysis;

Keywords: information value, data visualization, interactive data mining, game environments

1 Introduction

The proper location of key artifacts and information in interactive environments is important to increase the likelihood that the user will see this information. Specifically in a very large world, it is imperative to put information in a place where users will be able to find it easily. Usually this is done subjectively by the designer of the world. But this is not always the optimal solution since different users will explore the environment in many different ways. We propose an objective algorithm for calculating the optimal positioning of such artifacts and information.

This work can benefit several fields including education, advertising, art asset generation and level design. For example, if a company is developing a high-budget game and is spending heavily on art assets they want to ensure that all of those expensive assets are viewed by the players. This algorithm will help direct effort into key areas of these 3D interactive environments and reduce unnecessary effort in areas not normally seen. Used in playtesting for prototypes, this work may assist in developing these interactive environments by allowing the developers to correlate their intention with actual play.

In-game advertising is another key application of this technology. As discussed in [Hong 2005], in-game advertising is a growing field which is expected to be very important to the video game industry in the future. However, currently the advertisements are being placed purely subjectively by the level designers. This algorithm can be used for placing advertisements where players are more likely to see them, maximizing the impact of advertising and even creating distinction between premium, regular, and budget advertising space.

Using a 3D interactive visualization tool we created, called HIIVVE (Highly Interactive Information Value Visualization and Evaluation), and interaction data from any 3D first or third-person interactive experience, we can calculate the observation density and overall surface information value for all of the surfaces in the environment. We start with a set of player traces, which is collected interaction data that includes position, orientation, and elapsed time. This information is represented in a 3D visualization along with all of the static geometry for that environment (e.g., buildings, walls, and so forth). Using these player traces from various players in the environment, we can project a distance-weighted view frustum from the dynamic viewpoint of the player to the static surfaces of the world geometry. Using a distance squared weight, the view frustum projection upon each surface can be collected in an additive fashion. This creates an observation density map for each surface. A summation over the entire surface yields the overall information value of that surface. Ordering the surfaces by the most viewed would then reveal the optimal surfaces for information placement from the surface information value.

Context is key [Coutaz et al. 2005] and we are tied to the context of our data. The original player traces used to calculate the information value surfaces were obtained using the Urban Combat Testbed given a specific start and goal location plus world geometry (together which provide a specific context for the UCT game). The location of the start and the goal greatly influence the player search task. Therefore, the *optimality* of a surface is within the context of the scenario for which it was calculated.

Once we acquired the optimal surfaces for an environment we conducted a study to determine whether the surfaces are indeed the optimal ones. We created five scenarios with three posters placed in varying levels of information value. We had human players play a single level and then tell us which posters they remembered seeing in the environment. We then compare the average information retention between these five test groups.

This paper first discusses relevant related work, establishes the problem of where to place key information artifacts in an interactive 3D environment, and explains the method of calculating the information value for the surfaces. We then explain the methodology and algorithm in great detail. Once we have explained our methodology, we discuss the results of our user study.

2 Related Work

Locations of *seen points* based on the field-of-view of users interacting in 3D virtual environments taking into account distance, time of gaze, and radial focus drop-off have previously been studied [Chittaro et al. 2006]. However, these evaluations were conducted in 2D and did not take into account full calculations for object and surface occlusion. In general, their work is more focused on user spatial flow as the traversed and interacted in the environment. Our work takes into account time, distance, and radial focus but explores and includes full 3D fidelity, environments of complex geometry, and full object and surface occlusion.

In a study that was conducted to find the optimal navigation technique in a complex 3D environment [Suma et al. 2007], one of the metrics measured was whether the subjects remembered the objects that were in the environment. This evaluation is similar to ours, but the focus was not on the placement of the objects but rather how the navigation technique used affected the user's memory of objects. The study asked users to navigate the environment using different navigation techniques (real walking, move where pointing and move where looking) and take a few tests afterwards. The tests asked the users to identify the objects they saw from a list, write down the objects they saw from memory and also to place the objects from a list onto a map. The Suma *et al.* study was based on evaluating control modalities and not information placement, but the mental evaluation was similar. In that study it was observed that the various navigational techniques did not aid in remembering objects. If the information value of the surfaces in the environment had been calculated *a priori* from previous trials, the objects could have been placed in areas that were most visible to the users. In doing so, they could emphasize the differences in object memory based on the navigation techniques themselves knowing that the objects were placed in the areas of the highest probability of observation.

Various metrics for exploring way-finding in an interactive 3D virtual environment, including player movement and orientation, involving field-of-view perception and memory have been studied [Ruddle and Lessels 2006]. Ruddle and Lessels conducted a study to evaluate how users navigate a virtual environment. The users were given the task of finding targets placed in the environment. The field-of-view of the subjects was used to classify the errors made while searching for the targets. They found that most of the missed targets were in the users field-of-view but were just not noticed—even with a wide field-of-view. We have also observed the same phenomenon, supporting that users in 3D interactive environments may not fully observe their surroundings.

3 Problem Statement

The major problem that we are trying to overcome answers the following question: *How do we know where to place information artifacts in games?* The most obvious solution is to let the level designers subjectively place these artifacts where they *think* the player might see them. This is not always the best solution however. Since no two players will explore an environment in exactly the same way it is important to place these artifacts objectively, especially if they could be placed more effectively by some proven, repeatable method. For example, in *Tomb Raider Legend* (published by *Eidos Interactive*) the clues and key interaction points are often placed in locations that are very hard to find for the average player. If they could somehow analyze the data from various play testers to see where they explored the most, it would make it apparent what the easiest to find location for the artifacts would be. However, the game designer may not always want to put the artifacts in the easiest to find location. They might also want to put it somewhere in the median value areas that are neither the easiest nor the hardest to find.

Our approach provides an objective solution to this problem. Given the time-varying playtest data from various users, the algorithm will provide the designer with a list of all of the surfaces in the environment in order of information value. So, if they wanted to place the artifact on or near the highest information value surfaces they can look at the top of the list, for a median value location they can look in the middle of the list, and so forth.

This work seeks to prove the following hypothesis:

Using prior observation under a specific context to determine surface observation densities and derived information values of all game surfaces, this data can be used to guide the placement of new information artifacts in accordance with desired observation results under the same context.

4 Discovering the Value of Information

The value of information refers to the likelihood that the user will observe a piece of information that is placed on a particular surface. Discovering the value of information requires that we have interaction data for the environment in question. The interaction data must include position and orientation of the user over time. We also need data from as many users as possible in order to get a good representation of an average user exploring the environment. The following sections will explain the process in more detail.

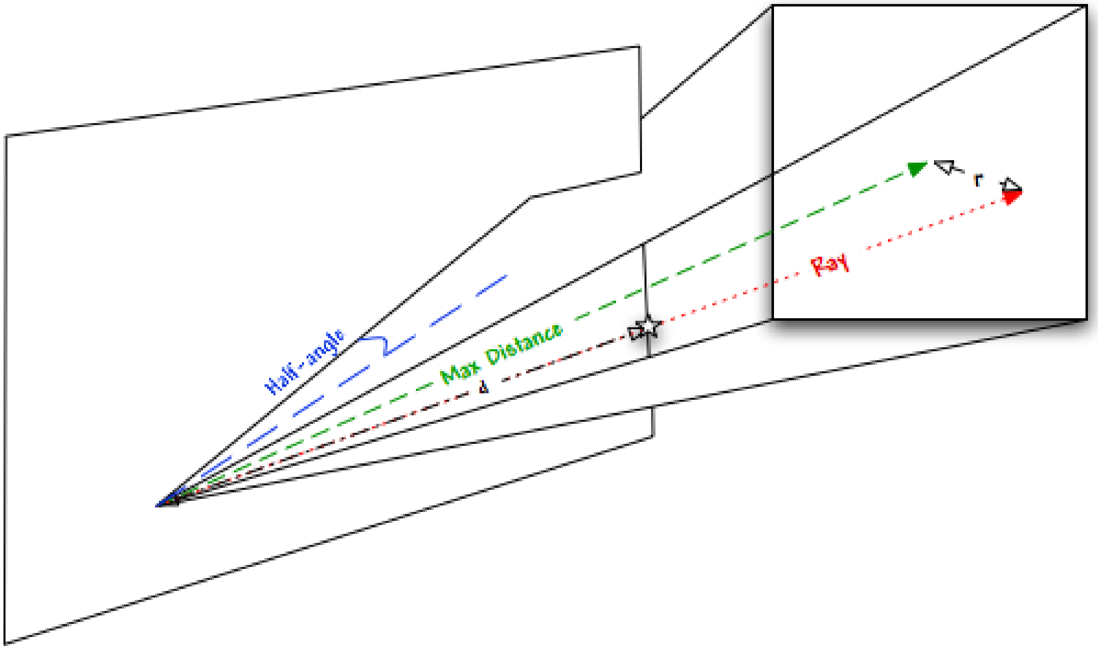


Figure 2: *An example view frustum showing one ray intersection.*

4.1 Information from Observation

Using a collection of interaction data for an environment we can calculate the observation density for each and every surface in the environment, as shown in Algorithm 2. This is done by calculating the intersection between the user's view frustums (the view-extended area of sight in a 3D environment, geometrically shaped like a square-based pyramid) and the surface, and updating the values on the surface where it is intersected. Once we have determined this value we know that the surface with the highest observation density is the optimal surface for information placement in the environment based on the source user population. We also know the exact location on the surface that has the highest observation density by coloring the texture on the surface wherever there is an intersection. Each surface in the environment is assigned a 2D texture of its own, the size of which is determined by a minimal *bounding quad* around the surface.

We have a few main constants that are defined before the algorithm is executed as shown in Figure 2,. The max distance is the distance between the viewpoint of the player and the centroid on the frustum's base. The half-angle of the frustum (i.e., the frustum angle from the surface-orthogonal center of the pyramid to the outer edge as shown in Figure 2) determines the size of the projection of the frustum onto the surface under evaluation. Note that currently we only use square-based frustums for simplification. These two variables are constant for each execution of the algorithm. In order to find the intersection between the view frustum and the surfaces we project a ray from the viewpoint to every point on the base of the view frustum at a specified interval. When there is an intersection with a ray and a surface, we calculate the texture coordinate at that point and update the value accordingly.

$$w_i = \frac{\frac{V_{max}}{d_i^2}}{r} \tag{1}$$

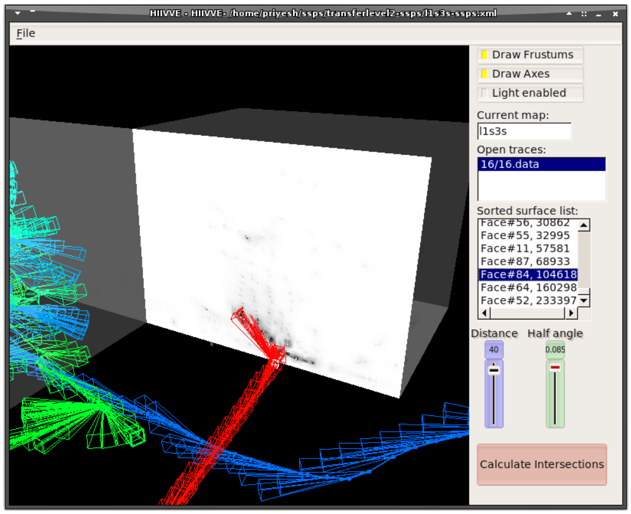
We show the equation for calculating the observation density for a single intersection point i in Equation 1. Where V_{max} is the maximum total value for the point, d is the distance of the point from the user and r is the distance from the center of the view frustum projection on the surface to the point on the base where the ray is being projected. Refer to Figure 2 for an image that shows both d and r in context.

Figure 3: *Urban Combat Testbed.*

4.2 Targeted Environments

For this research, we utilize the player traces [Youngblood 2002] collected from the Urban Combat Testbed (UCT) project [Holder et al. 2007; Youngblood and Holder 2004; Cook et al. 2007; Youngblood et al. 2006]. The Urban Combat Testbed is a first-person shooter game based on the Quake 3 engine. In Figure 3 you can see a screenshot of the Urban Combat Testbed. The primary objective for the player in UCT is to find and defuse an Improvised Explosive Device (IED) placed somewhere in the environment. UCT records the player as they play the map and try to find the goal. We have over 400 player traces from a previous study that can be used to calculate the observation density. This provides us with an initial dataset for testing our algorithm.

Although UCT was used in this case, this algorithm is applicable to any 3D first or third-person interactive environment. The only requirement is that it must be able to save the interaction data for the user and it must generate an XML file for the geometry which contains all of the relevant static surfaces in the environment. This XML file can generally be produced from any polyhedron-based 3D environment by the UCT Common Toolset [Holder et al. 2007; Cook et al. 2007; Youngblood et al. 2006]. Both of these files are in a well defined format which makes it relatively easy to output them from any game even if not using the UCT Common Toolset. UCT and the UCT Common Toolset will be publicly released on www.urban-combat.net in the Summer of 2007.


HIIVVE - HIIVVE- /home/priyesh/ssps/transferlevel2-ssps/l1s3s-ssps.xml
File
Draw Frustums
Draw Axes
Light enabled
Calculate Intersections

Current map: l1s3s

Open traces: 16/16.data

Sorted surface list:
Face#56, 30862
Face#55, 32995
Face#11, 57581
Face#87, 68933
Face#84, 104618
Face#64, 160298
Face#52, 233397

Distance 40

Half angle 0.085

Figure 4: *HIIVVE tool interaction interface with player trace loaded and surface information value densities illustrated.*

4.3 HIIVVE

The Highly Interactive Information Value Visualization and Evaluation tool (HIIVVE) is built using C++. The graphics rendering is done using OpenGL and the user interface is created in FLTK (Fast and Light ToolKit). A screenshot of HIIVVE is shown in Figure 4. The user can adjust the max distance and the half-angle of the view frustums with a slider bar. The sorted surfaces list-box provides a list of all of the surfaces in order of information value once the *Calculate* button has been pressed. Rendering the lighting, XYZ axes, and the view frustums can be toggled with buttons. Due to the libraries used by the tool it is cross-platform and can run on Mac OS X, Linux, and Windows. We plan to also make this tool publicly available in the Summer of 2007.

4.4 Methodology

The algorithm for calculating the intersection with one frustum is shown in Algorithm 1. The information value for a surface is found by calculating the sum of all observation densities for the entire surface. The observation density metric is determined by calculating the intersection of a view frustum and the surface polygon. A ray is projected from the eye point to each point on the base of the frustum projection on the surface. We then check to see if any polygons are intersecting that ray. We only update the closest intersection point to the viewpoint to cull out occluded surface intersections.

Algorithm 1 Compute Individual Frustrum Effect

```
Find $baseSize$ of frustum base using the $frustumAngle$
Find the $centroid$ of the base
Find the $topLeft$ and $bottomRight$ corner points of the base
for $i = 0$ to $baseSize$ do
    for $j = 0$ to $baseSize$ do
        Find the current $basePoint$ to shoot a ray to using $i$ and $j$
        Reset $bestPoint$ to NULL
        Calculate $ray$ from $eyePoint$ to $basePoint$
        for all $surfaces$ in the environment do
            Calculate $intersectionPoint$ on $surface$ by the $ray$
            if $intersectionPoint$ is closer than the $bestPoint$ then
                if $intersectionPoint$ has not yet been updated then
                    $bestPoint = intersectionPoint$
                end if
            end if
        end for
        if intersection found then
            $d$ = distance of $bestPoint$ from $eyePoint$
            $r$ = distance between $centroid$ of base and $basePoint$
            $value$ = $(V_{max}$ / $d * d)$ / $r$
            Update texture value at the $bestPoint$ by $value$
        end if
    end for
end for
```

This algorithm has an efficiency of $O(n \cdot m)$ where n is the number of viewpoints and m is the number of surfaces. Although we are culling the viewpoints that are inside buildings, it can still take a long time to run the algorithm. Figure 5 shows a graph of the runtimes for the algorithm with a varying number of viewpoints. The number of surfaces was a constant 258 since we used the same map.

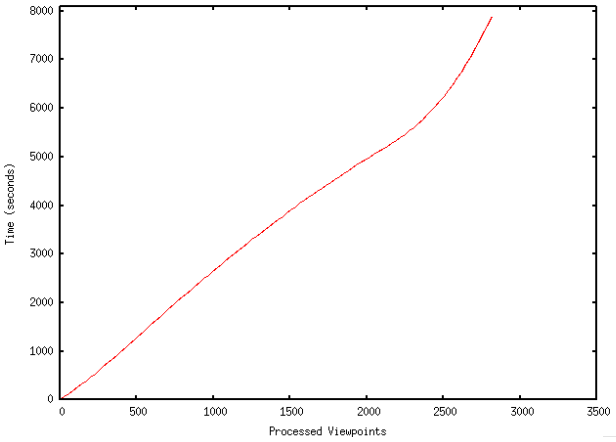


Figure 5: *Running time for the algorithm with varying number of viewpoints*

As seen in Algorithm 2, we are processing each player's viewpoints as a view frustum and calculating the intersection with each of the surfaces. We are also storing an *updated* flag for each point on the surface's texture to make sure that we don't update the same texture more than once per frustum.

Algorithm 2 Cycle through player sample population

```
for all $players$ do
  for all $viewpoints$ in the $player's$ path do
    if $viewpoint$ is inside an object then
      skip this viewpoint
    end if
    Calculate Individual Frustum Effect for $viewpoint$
    Clear the update status of all surfaces
  end for
end for
Clear intersected surface list
for all $surfaces$ in environment do
  insert the $surface$ into list, sorted by total value
end for
```

5 Experimentation and Results

In order to verify our algorithm, we conducted a study using one of the levels from the Urban Combat Testbed (Level 1 Scenario 3). We selected three posters to place in the UCT environment and conducted an IRB approved human trial study to test whether the users observed the posters in the environment.

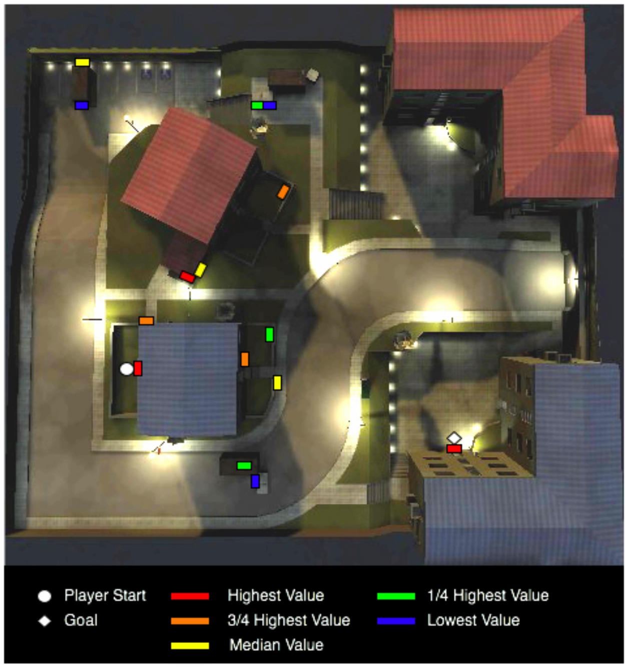


Figure 6: *Location of information elements placed within the UCT test environment based upon previous player traces and calculated surface information value.*

We ran the value of information algorithm on the level to determine the total observation density on all of the surfaces based on 32 previously recorded player traces for that level. When doing the calculation of information value, the view frustum distance was set to 175, the half-angle of the frustum was set to 0.085 radians, and V_{max} was set to $100,000$. We then determined the surfaces in the low, low quarter, median, high quarter and high range for information value. Five UCT scenarios were then created, each with the same three posters placed on the appropriate surfaces. Figure 6 shows the placement of the three posters for each of the five levels. Users of varying experience with first person shooters were asked to play one level in the Urban Combat Testbed. Each participant was assigned one of the five levels at random and told that they must find and defuse the IED and that they have a time limit of five minutes. Once they finished the level, they were given a short post-test. The post-test asked them to circle the images they remember seeing in the environment, or if they did not see any they can circle "NONE". A screenshot of some of the posters in the UCT environment is shown in Figure 7. Note that users were not instructed to look for specific information elements in the environments—their focused task was to defuse the IED.

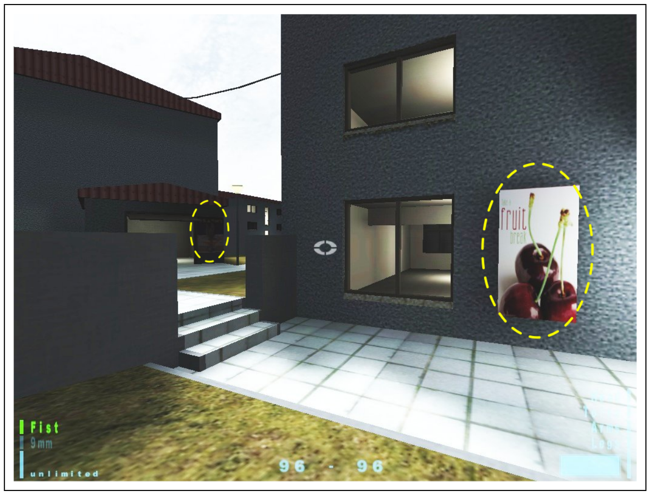


Figure 7: *Wall-mounted posters in the UCT environment. Fruit poster in the foreground and Army poster in the background, both are circled.*

The number of posters that were seen by the users in each of the five levels is shown in Figure 8. This was measured by loading the recorded player traces of the participants and determining if the posters were ever within their view frustum. If we look at the number of posters that were visible to the participants based on their recorded player trace, we can see a pattern based on the level played. For instance, the players that played in the high value level always saw at least two of the posters, and the players in the lower-quarter value level saw almost none of the posters. In the median value level, there was an even distribution between the number of posters seen. However, there is an uneven distribution for the low value level because of the current low sample size[1], which is eight participants per scenario for a total of 40 participants. According to the current study results, 75.0% of the information was seen in the high value scenario with a standard deviation of 15.4%. In the high-quarter scenario, 37.5% of the information was viewed with a standard deviation of 21.3%. The median level also had 37.5% of the information viewed and the standard deviation was 33.0%. The low-quarter level had the lowest amount of information viewed at 16.67%, with a standard deviation of 17.8%. Finally, the low level had 33.3% of the information viewed with a standard deviation of 35.6%. The higher curve for the low level is due to an outlier as seen in Figure 10. Subject 32 uncharacteristically was drawn to the posters and led a path through the environment that led them to view all three posters. However, despite clearly observing the posters they only remembered seeing one of them.

[1] This study is still being continued as the paper is written with a target population of 150 participants—30 in each scenario.

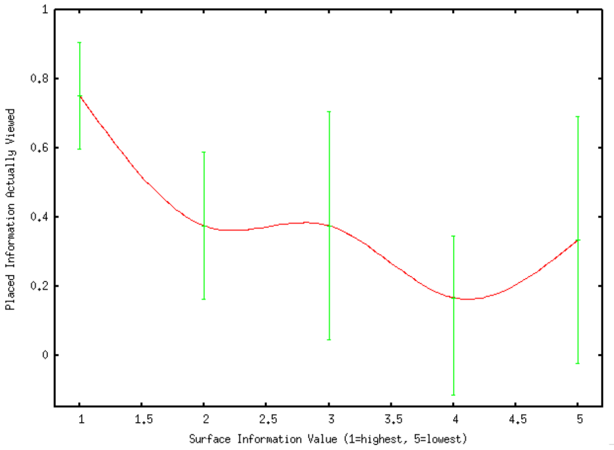


Figure 8: *Actual Viewed Information from Viewport Examination after Interactive Session.*

After they played the level we asked them to fill out a simple post-test that asked which of the posters they remember seeing. Figure 9 shows how many correct posters the participants said they had seen. We found that a majority of the participants did not remember seeing the posters. However, when we analyzed their player trace it was apparent that they did see at least a few of the posters. Although we know that a majority of the participants in the high value level definitely saw at least two of the posters, some of them said they didn't remember seeing any of them. The two posters in this case were placed right in the path to the goal so they could not be missed. This is a psychological aspect that we did not anticipate but had been previously noted in similar work [Ruddle and Lessels 2006]. For the reported views, 25.0% of the information was reported as seen in the high and median levels with a standard deviation of 5.8%. For both the high-quarter and low levels, 12.5% of the information was reported as seen with a standard deviation of 4.4%. The players reported seeing 0% of the information in the low-quarter level.

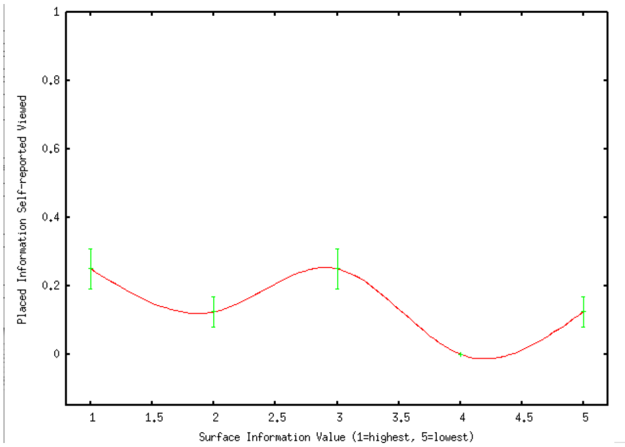


Figure 9: *Self-reported Information Assimilation after Interactive Session.*

If we look at the number of posters that were physically seen compared to how many they remembered, it is clear that the players are not paying attention to the posters when they play the level. The curve in Figure 8 clearly shows that the algorithm was indeed effective in placing the objects within the path of these new players. The largest improvement can be seen between the high value and low quarter value surfaces which yield a 58.3% improvement in user observation.

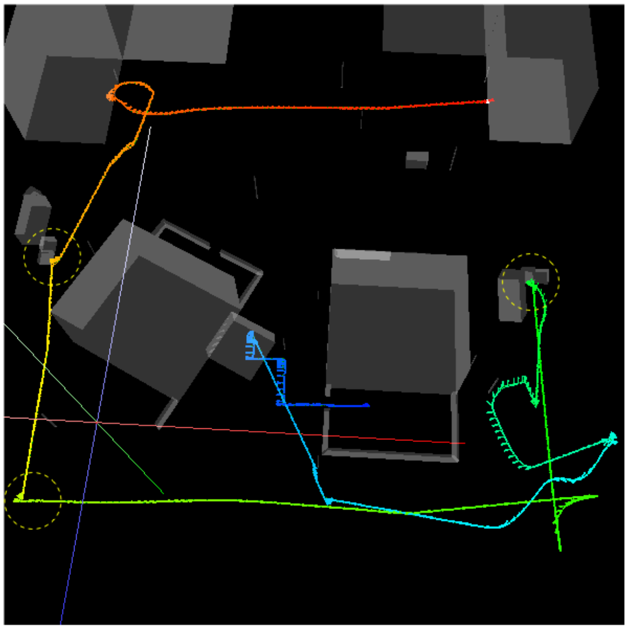


Figure 10: *Subject #32 outlier player trace with the poster locations circled.*

There were several factors that we had not considered when designing the study. The experienced gamers finished the level very quickly and so they did not see most of the posters on the level. They were also accustomed to ignoring in-game advertising or other irrelevant information and were focused on finding the IED—context really does appear to be key in affecting observation. Also, the nature of the task given to them (defusing an explosive device) gives a natural sense of urgency that causes them to finish as fast as possible.

6 Conclusions

The result from Figure 8 supports our hypothesis, because it clearly show that the surfaces that had the highest information value were seen by the users most often. We also discovered that non-essential information artifacts seemingly are ignored or just not remembered. In order for the users to retain the information they see in these posters, we would probably need to make them more relevant to the game. More work needs to be done on designing in-game advertisements in order for them to be effective or they will be ignored. However, the results are promising and despite the low rate of information retention it is clear that the players are at least exposed to the high information value surfaces in the environment.

This work is currently biased by only 40 participants (eight for each scenario), but this study is still ongoing. This preliminary results clearly and strongly support our hypothesis that using prior observation under a specific context to determine surface observation densities and derived information values of all game surfaces, this data can be used to guide the placement of new information artifacts in accordance with desired observation results under the same context.

7 Future Work

We have found several aspects of the HIIVVE tool and the study that can be improved in the future. We would like to explore ways in which, under the same constraints, we can get the players to notice the information they are seeing. If we used posters that stand out more from the environment, the players may have been able to identify them more easily. For instance, the U.S. Army poster might be blending a little too well with the overall theme of the game and gets overlooked. Also, if we placed the goal in a harder to reach place within the environment it would force the players to explore the environment more and therefore increase the chances of seeing the posters. Additional work will also be focused at developing subjective measures and qualitative guidelines to make sure that information is not placed in a location that is bothersome for the player or seems out of place. It would also be interesting to evaluate subconscious observation of the poster contentmessage—maybe the participants do actually remember some aspect of the poster but not the poster itself.

We would also like to run HIIVVE on more initial player traces to find more detailed information value data. Currently the algorithm is not fast enough to calculate the information value for an extremely large amount of player traces. We hope to speed up the algorithm by doing intelligent interpolation between ray intersections. The core HIIVE algorithms are easy to run in parallel and are well suited for cluster deployment. We had to run 4 parallel processes on a quad-core machine to get the information value for 32 player traces in a reasonable amount of time. In the future we hope to run many more player traces in parallel on a computing cluster and then combine the results.

Acknowledgements

The authors wish to thank the Quakecon 2006 staff and participants who provided the initial player trace data sets for this work. We also wish thank the students at UNC Charlotte who participated in this experiment.

References

CHITTARO, L., RANON, R., AND IERONUTTI, L. 2006. VU-Flow: A Visualization Tool for Analyzing Navigation in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (Nov/Dec), 1475–1485.

COOK, D. J., HOLDER, L. B., AND YOUNGBLOOD, G. M. 2007. Analysis of Human Transfer Learning Using a Real-Time Game Testbed. *IEEE Transactions on Knowledge and Data Engineering*. Under review.

COUTAZ, J., CROWLEY, J., DOBSON, S., AND GARLAN, D. 2005. Context is Key. *Communications of the ACM 48*, 3 (March), 49–53.

HOLDER, L. B., COOK, D. J., AND YOUNGBLOOD, G. M. 2007. A Real-Time Perceptual Testbed for the Evaluation of General AI Systems. In *the Twenty-Second AAAI Conference on Artificial Intelligence*. Under review.

HONG, Q., 2005. Question of the Week Responses: In-Game Advertising?, November. http://www.gamasutra.com/features/20051130/hong_01.shtml.

RUDDLE, R. A., AND LESSELS, S. 2006. Three Levels of Metric for Evaluating Wayfinding. *Presence: Teleoperators and Virtual Environments*, 15, 637–654.

SUMA, E., BABU, S., AND HODGES, L. 2007. Comparison of Travel Techniques in a Complex, Multi-level 3D Environment. *IEEE Symposium on 3D User Interfaces*.

YOUNGBLOOD, G. M., AND HOLDER, L. B. 2004. Agent-based Players for a First-person Entertainment-based Real-time Artificial Environment. In *the 17th International Conferenece of the Florida Artificial Intelligence Research Society (FLAIRS) held in Miami Beach, Florida*.

YOUNGBLOOD, G. M., NOLEN, B., ROSS, M., AND HOLDER, L. 2006. Building Test Beds for AI with the Q3 Mod Base. In *Artificial Intelligence in Interactive Digital Entertainment (AIIDE)*.

YOUNGBLOOD, G. 2002. *Agent-based Simulated Cognitive Intelligence in a Real-time First-person Entertainment-based Artificial Environment*. Master's thesis, The University of Texas at Arlington.