Volume xx (200y), Number z, pp. 1–7

Exploiting human visual peripheral characteristics on 3d real-time computer applications

F. Lopez, R. Molla, I. Barona

Technical University of Valencia, Valencia, Spain

Abstract

There are several limitations on the Human Visual System that can be exploited when doing real-time computer graphics applications. Our visual acuity continually decreases in the periphery of the eye, allowing reductions in resolution, color and size of the objects that are away from our focus. Inattentional blideness is the failure to notice a fully-visible, but unexpected object because attention was engaged on another task, event, or object.

Using a 3D video game we have measured the users perception to LoD changes in the periphery of the visual system when the user is involved in a task which is highly dynamic and interactive. In one preliminary result the users were only capable of detecting less then 20% of the LoD changes and we show that with some adjustments this figure can be decreased further.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism; I.4.8 [Computing Methodologies]: Image processing and computer vision–Scene Analysis

---

1. Introduction

The human visual system (HVS) has lots of limitations that can be exploited when developing real-time 3d applications like computer games or virtual worlds. Our visual acuity continually decreases in the periphery of the eye, allowing reductions in resolution, color and size of the objects that are away from our focus. Lower resolution changes in our visual system, like movements or coarse image changes are detected instantly by our peripheral vision, but in order to be able to identify the stimulus we must turn our vision in order to be able to use our central vision to focus on the point of interest and to identify identify the object or stimulus.

Nowadays, in order to create more realistic effects, images with higher definition whilst sustaining frame rates that maintain the user experience at the standard level, 3d real-time applications are demanding more and more computing power. In addition to that, there also exists non-graphical loads of computing, like simulation, AI, physics, HCI, etc. that are migrating from CPU to GPU and overloading the graphics processing unit. As a result, there is always a need to reduce the graphical payload on CPU and GPU in order to enhance the application performance, reduce the power consumption (for mobile devices) and to improve other characteristics of video games like physics simulation, behaviour, smoother simulations and HCI interaction [Mar08, Ols08]. Traditionally, one of the most common techniques used for reducing graphic rendering costs, whilst keeping a visual fidelity restricted by a geometric criteria has been the use of Level of Detail (LoD) techniques [Cla76]. The quality of an image made by a computer depends unavoidably on the visual perception of the viewer that is positioned in front of a screen. The eyes collaborate with the brain to perceive an object, so visual fidelity is even more connected to a perception criteria rather than to a geometrical criteria. Obviously, geometry quality has an influence on the perception quality of a computer generated image. LoD techniques have normally been used to change the amount of triangles in an object mesh depending on a distance-to-the-camera criterion. Even more, if there is a specific task to accomplish by the user on the application there is another concurrent phenomenon that can occur. Inattentional blideness is the failure to notice a fully-visible, but unexpected object because attention was engaged on another task, event, or object. We have asked 20 people to play a 3d computer game that uses peripheral perception based criterion for LoD changes on 3d models. The game immerses the user in highly dynamic interaction that proposes a challenge to them that requires their full attention and physical coordination. In the first test, we found that only 20% of the total changes in LoD were detected by users. On the second test, we end up with a function that defines a safe-zone for making LoD changes without being noticed, and in the last test we made a comparison of LoD changes using diferent numbers of intermediate models.

In section 2 we present a brief background on LoD techniques and the Human Visual System (HVS) perception characteristics, and highlight points that justify the experiments that we have carried out. Section 3 introduces the developed application and explains the nature of the tests, the LoD scheme and the LoD switching technique. This section also explains the criterion for LoD selection based on peripheral distance from the fovea. In section 4 we present the results of our tests on 20 subjects and finally section 5 resumes the conclusions of our research.

2. Background and previous work

2.1. Human Visual System

There are several limitations on the Human Visual System that can be exploited when doing real-time computer graphics applications.

In the 1960s researchers developed some ideas about our ability to perceive details and tried to find the limits of our HVS. They ended up with a function called Contrast Sensitivity Function (CSF)[MS74]. CSF tells us how sensitive we are to the various frequencies of visual stimuli. If the frequency of visual stimuli is too high we will not be able to recognize the stimuli pattern any more. We can think of an image made of vertical black and white stripes, if the stripes are very thin (i.e. a few thousand per millimeter) we will see just a gray image and not the individual stripes. This limit on resolution is directly related to the number of photoreceptors that our eye has. Contrast sensitivity varies between individuals, reaching a maximum at approximately 20 years of age. Daly [Dal] provides a version of this CSF that adds a speed variable to the model. This model applies well to static images and when we are receiving the stimulus with our central vision (fovea). Figure 1 shows the common form of a CSF.

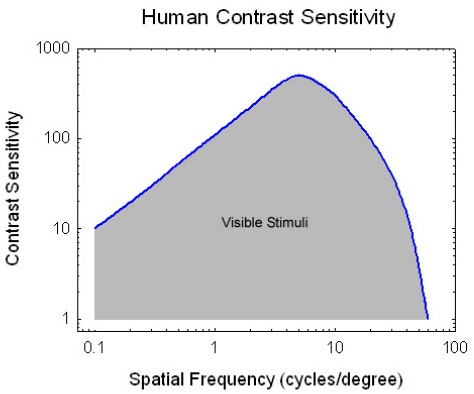

Figure 1:

The above does not apply when talking about peripheral vision. The main functions of peripheral vision are, recognition of well-known structures and forms with no need to focus by the foveal line of sight, the identification of similar forms and movements (Gestalt psychology laws) and the delivery of sensations which form the background of detailed visual perception. Our visual acuity decreases in the periphery of the fovea, meaning that it becomes harder to distinguish high resolution details, colors and the size of objects. Thibos et al. [TSB96] analyzes the CSF models on the peripheral vision and exposes significant differences with the traditional foveal CSF because of how the visual process works in the periphery of the eye.

2.2. Visual Perception

Several works have been done with respect to human visual perception and computer generated graphics, in [OHM*04] there is a complete survey on the topic and it is still a good starting point to analyze the problem and search for open threads. We will focus on Interactive Graphics, there are at least two ways of exploiting the lack of resolution on our vision in the periphery at the moment of rendering graphics.

One is to combine the presentation device with another device that captures (like an eye-tracking system) the users gaze and estimates where the user is looking on the projected image. With that information, the system can simplify the models (in geometry space) or the rendered image, decreasing the computing power needed for the application, always taking into account that even if the user perceives some loss of quality in the display, they are still able to do the task without the changes affecting the performance.

[LMYM01] used an eye-tracking device in their experiments to produce high-resolution images only in the areas in which the user is focusing. They proved that they can filter images with a radius of 4.1 degrees and that the user will not be able to distinguish between the original and filtered image. In [LW07] there is an analysis of how late can you update a gaze-contingent display without user detection. They state that it can be a value between 5 ms and 80 ms depending on the activity and the level of degradation between two consecutive frames. Focusing on video-games, a big limitation in such systems is the need for a precise eye-tracker system with high temporal frequency, which is still not available in the market for mass consumers. There are several studies related to this approach, for more details the reader can refer to the survey mentioned above.

Another approach different to a gaze-contingent one, is to use the implicit information defined by the specific task that the user has to perform in the application. For example, in a FPS video-game, when the user is shooting at a moving target for a couple of seconds its highly probable that he will focus all his attention on this particular place of the screen. Perceptual blindness is the phenomenon of not being able to see things that are actually there, caused by the mental focus or attention which cause mental distractions.

Cater et al. [CCW03] performed an experiment on still images, that consisted on counting particular objects on a computer generated image at different resolutions. They found that user was no able, in most of the cases, to tell about the differences between the same images at different resolutions. For completeness, they also used an eye-tracker system analyze the pattern of movement of the user and to confirm that the task imposed completely defined this pattern.

We focused this study on trying to understand how vision is degraded in the periphery when the user is performing a dynamic task with physical interaction like playing a video-game or navigating in a virtual world. As we are not making any assumption on the possible degradation function we did not use the well known CFSs as a criterion for simplification, but a simpler two alternative forced choice on several experiments trying to measure and register what the users did perceive in a real application. Also we found a motivation for this tests because of the emerging techonologies on large displays and projection systems that are rapidly entering in the final consumers market.

3. Experiment implementation

Most of the studies on visual perception mentioned above have used static images, with little or null interaction by the user, and without setting an additional challenge that would distract the user from the experiment itself.

We believe that current models of visual perception based on CSFs for the central vision, peripheral vision and movement, model well the physiological limits of the HVS, but we also believe that there exists other factors that can be taken into consideration to try to exploit even more the loss of visual acuity in specific circumstances.

What is more, with the continuos increment in the size of TVs and Monitors, and the decreasing prices on home projection systems, the users FOV will keep increasing, this is what justifies the need to do more research on peripheral visual perception on interactive graphics applications.

Differing from TV, in which once a channel is selected requiring no further initiative from the user (whilst keeping they intellectually active and stimulating their imagination), video-games represent a continuous challenge, as there is the need observe and analyze the environment, they must acquire and store new information, do inductive and deductive reasoning, build and apply cognitive strategies in an organized way and develop fine-tuned psychomotricity abilities (coordination, speed, sensitivity, etc). Meaning that the player is continously obliged to interact, making decisions and using their psychomotricity abilities.

Although we did not use an eye-tracking system, we believe that putting a moving target on the screen and asking to the user to shoot at it in a limited time with a limited number of bullets we can recognize at the end of the task if the user was focused on it. Taking that into account, at the end of the trials, we only kept the results that showed an acceptable performance in time and shots to the POI.

Even more, to be able to adapt the game difficulty to different levels of experience on gaming, we considered an adaptation of the impact zone size on the POI in order to make it easy to play for non-gamers and not too boring for gamers that showed up at the tests.

3.1. The application

For the purpose of this work, we developed a simple 3d video-game to measure the users perception of changes in resolution of 3d models in the periphery of his vision. The application was developed using OpenSceneGraph, OpenAL and free content for academic purpose from the web.

During the game, the user has to drive a vehicle (a tank) through a big city following its instrumental (a Head-Up display) to specific points on the city. When the user gets to each special mark, a scene is presented to the user, showing a 3d Model (LoD) similar to a statue, and a villan character (represented by a flying Darth Vader) that is shooting at the statue. The villan is the Point of Interest (POI) for us, and this represents a big simplification of the problem comparing with other systems with eye-tracking support. The user has to protect the statue, and shoot at the villan (POI) that moves around the screen. The villan will start to move at random speeds and in random directions and the user has to use the keyboard (arrow keys) to aim the cannon of the tank and shoot repeteadly until the villan leaves the scene. Whilst the user follows the POI with his eyes and shoots at it, the 3d statue will start changing as a function of the distance and other criteria that is explained in the next section. Before starting the game, users are instructed that they should not look at the 3d Model but to the POI all the time and if they sense anything on their peripheral vision they must press a key to register the event. During this sequence, the only objects that move on the screen are the POI that is flying and the 3d model that is changing its resolution using several static LoDs.

Figure2 shows a screenshot of the application, with the bunny model at the left part of the screen and the POI flying around. The villane has an energy bar that decreases each time the user hits it, and the user has a bullet counter and timer.

Figure 2:

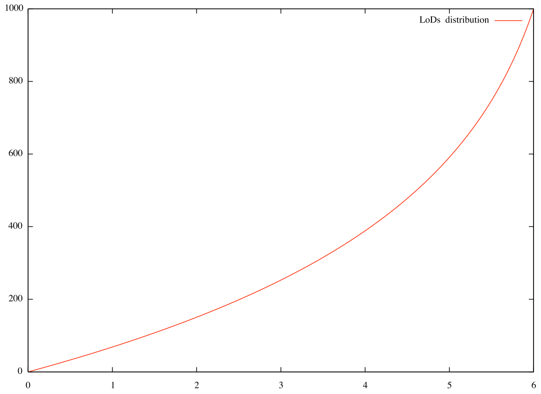

Figure 3: *Mapping function for LoD changes*

3.2. Participants and tests

We ran the tests on 20 people, at the Universidad Nacional del Comahue, in Argentina. The tests were taken in an isolated room with normal illumination and a 15.4 monitor at resolution 1024x768, all the participants presented normal uncorrected vision. The subjects were positioned at 50 cm of the screen and we used an adjustable chair to adjust height differences.

We made different tests to count the detections of LoD changes and to find “safe zones” for LoD changes for each model and resolution.

As a simple approach, the first test consisted on a mapping of distance and LoD levels, in function of the distance of the POI to the model, using the function of the figure 3, which maps each static model to an interval of distances from de POI. The function of the figure is arbitrary and experimental, trying to map more models with high quality near the central vision and less in the periphery. The model moves freely over all positions of the screen, and the user has to shoot at it and register every LoD change perceived with his peripheral vision. The function of figure 3 is as follows:

$$\frac{(1.0-log_{10}(1.0+(scale*(numLods-x)/numLods))*maxDistance}{log_{10}(scale+1)))}$$

A second test was carried out with the intent to aproximate a function for each model that could define like “safe-zones” from a practical point of view of implementation of static LoD in an interactive 3d application. In order to do that, now at the begining of the interaction, the flying villane rests at a close distance of the model, and when the user tries to hit it, it will start moving in a circular-like fashion around the model, in order to preserve an interval of distance in 2d space with respect to the model. While the POI moves, the LoD changes in the model will start to happen and continuosly decrease the quality (one step of degradation each 2 seconds) regarding the distance, until the user detects a change and presses the key for the detection. After a user detection, the last quality is restablished, the character flies away a small distance and the process starts again but using as first quality the same as the previous position that the user did not detect. The application gives the user 2 seconds of delay to press the detection key, and then ignores key detections, and the models change resolution after 2 seconds. So, for example, if we use 6 models, considering that the first model is already there, after 10 seconds of no detections at all, the application will register that in that particular distance the user did not detect any LoD change, and the model will end up with the coarsest quality.

Using the same approach of the previuos test, we repeated it but using a subset of the LoD models used in the previous test, to see if under bigger resolution changes the user detections improve because of the coarse resolution changes. We let meshes A,B and C be ordered from higher to lower resolution, as we wanted to see if a given LoD change that ends with the mesh C, say from mesh A to B, and then to C, is detected at similar distances as a change from mesh A directly to C. The results were sound with the intuition, in the sense that we can put a coarse mesh in a specific position passing through a series of intermediate meshes to avoid the user detection.

3.3. LOD implementation

We used two models provided at Stanford 3D Repository, the bunny and the dragon. The scheme of LoD used is static, different LoDs for each model were constructed using a Quadric Edge Collapse Decimation filter with the application MeshLab [CCR08]. Six LoDs were used for each model, for the bunny ranging from 3000 to 80 triangles in a log_2 downscale fashion, and for the dragon from 7k to 1.5k reducing by a factor of 30% each step.

For the transition between LoDs we implemented the technique suggested at [GW07] using alpha blending without exposing hidden faces, to avoid popping effects on LoD switching. The time used to switch between any two LoDs is 300 ms, and during that interval of time two meshes are rendered simultaneously.

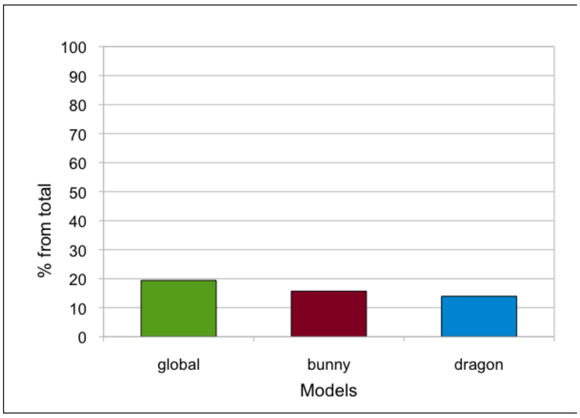

Figure 4: *Percentage of detections of LoD changes per user from a mean of 42 changes per user*

4. Results

In the first test, we used a fixed mapping function to assign more models in the neighborhood of the central vision and less models in the periphery. Even without a proper calibration, the results were quite successful. Figure 4 shows that users were able to detect less than 20% of the LoD changes, from a mean of 42 changes per user test. In the mapping function described in section 3.2, the scaling factor (for positive values) *"scale"* can be used to easily adjust the mapping curve and lower even more these detections to a more conservative LoD management with no detections at all.

The second test, worked as an easy way to adjust the LoDs for each distance, and to find a safe-zone of changes that should not be perceived by the user performing the assigned task. Results are showed in figures 5,6 for the bunny and the dragon respectively, and its remarkable that far from a visual angle of 13 degrees (or aproximately 400 pixels in this configuration) any user did perceive changes at all even on the bunny at 80 triangles, or the dragon at 1500.

The last test consisted on measuring the change detections when the LoD steps are bigger. Figure 7 shows the results, and we can see that when we use big steps for LoD changing the users perceive the changes at closer distances.

5. Conclusions

The results obtained from the tests showed us that peripheral vision perception can be altered by the task that the user has to accomplish, and there are several things that must be taken into consideration when implementing LoD techniques at the periphery. It is clear that a simplified criterion doesn't adjust to all

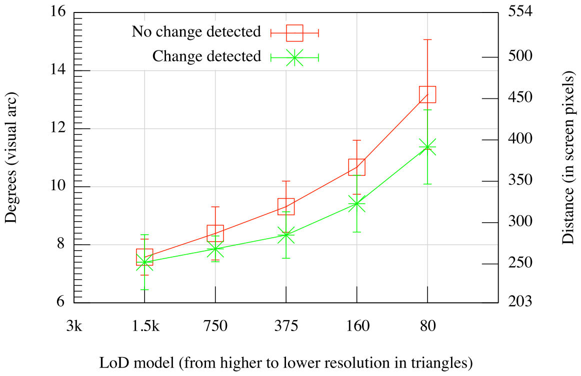

Figure 5: *Mean number of detections and non-detections of LoD changes in the periphery for the bunny model*

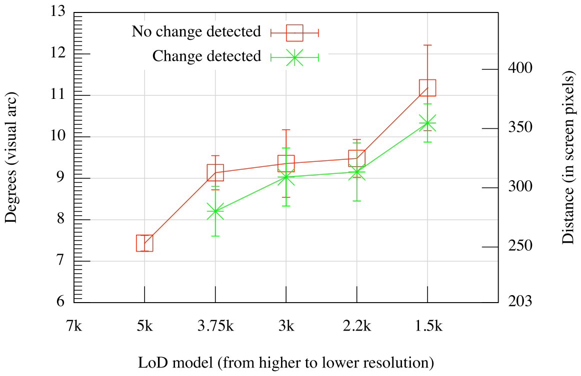

Figure 6: *Mean number of detections and non-detections of LoD changes in the periphery for the dragon model*

the needs, even for one application. The level of difficulty of the game or the application plays an important role in the definition of the parameters used at the moment of simplification. It is necessary to define more factors that can influence in the loss of the users visual perception and carry out more tests. Within those factors, we are considering the applications semantics, such as the level of difficulty, the speed of the game, etc, and other factors such as the size of the screen (from mobile devices to virtual reality caves).

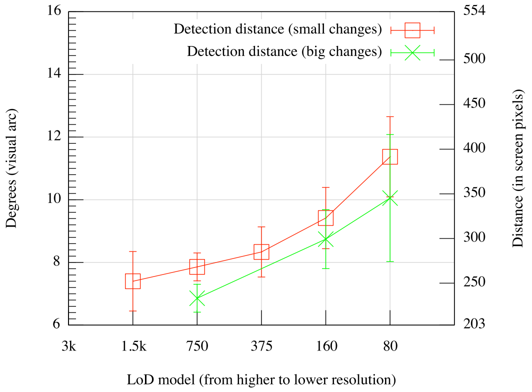

Figure 7: *Comparison of the mean distance of LoD changes detections when switching at small steps vs big steps. With big steps, switches are in the sequence (3k,750,160,80)*

References

[CCR08] Cignoni P., Corsini M., Ranzuglia G.: Meshlab: an open-source 3d mesh processing system. *ERCIM News 73* (2008), 45–46.

[CCW03] Cater K., Chalmers A., Ward G.: Detail to attention: Exploiting visual tasks for selective rendering. In *Proceedings of the 14th Eurographics workshop on Rendering* (2003), Eurographics Association, p. 280.

[Cla76] Clark J.: Hierarchical geometric models for visible surface algorithms.

[Dal] Daly S.: Engineering observations from spatiovelocity and spatiotemporal visual models. *Vision Models and Applications to Image and Video Processing*, 179–200.

[GW07] Giegl M., Wimmer M.: Unpopping: Solving the image-space blend problem for smooth discrete lod transitions. In *Computer Graphics Forum* (2007), vol. 26, Amsterdam: North Holland, 1982-, pp. 46–49.

[LMYM01] Loschky L., McConkie G., Yang J., Miller M.: Perceptual effects of a gaze-contingent multi-resolution display based on a model of visual sensitivity. In *the ARL Federated Laboratory 5th Annual Symposium-ADID Consortium Proceedings* (2001), pp. 53–58.

[LW07] Loschky L., Wolverton G.: How late can you update gaze-contingent multiresolutional displays without detection? *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 3*, 4 (2007), 7.

[Mar08] Mark W.: Future graphics architectures.

[MS74] Mannos J., Sakrison D.: The effects of a visual fidelity criterion of the encoding of images. *IEEE Transactions on Information Theory 20*, 4 (1974), 525–536.

[OHM*04] O'Sullivan C., Howlett S., Morvan Y., McDonnell R., O'Conor K.: Perceptually adaptive graphics. *Eurographics State of the Art Reports 4* (2004).

[Ols08] Olson T.: Hardware 3D graphics acceleration for mobile devices. In *IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008* (2008), pp. 5344–5347.

[TSB96] Thibos L., Still D., Bradley A.: Characterization of spatial aliasing and contrast sensitivity in peripheral vision. *Vision Research 36*, 2 (1996), 249–258.