USE OF LoD TECHNIQUES IN LUMINANCE CHANGES ON VIDEO GAMES

Iñigo Barona, Ramón Mollá, Francisco López and Mª Carmen Juan
Department of Information Systems and Computation
Technical University of Valencia
Camino de Vera, s/n - 46022, Valencia,
Spain
inigo@barona.es

September 12, 2008

KEYWORDS
eye adaption, LoD, video game, illumination change

Abstract

The human eye is able to adapt itself to a wide range of luminances from very bright daylight to very low night light. The human visual system modifies its sensitivity when an increment or a decrement of illumination occurs. During this transition, known as eye adaptation, our system experiments a temporary blindness. Illumination changes commonly happen in video games (driven through a tunnel, entering in a dark house or walking in a cave under torchlight) and rendering techniques must take into account theses situations. In this paper, we present a system for accelerating interactive rendering, based on using polygonal simplifications in luminance variations on the scene. We discuss the influence of the preadaptation light, the duration of the transition and the responses to different intensities changes.

INTRODUCTION

Nowadays, 3D scenes are becoming more and more complex so that current graphic hardware is scarcely powerful enough to render some games in real time and the ones that are coming soon. In spite of the huge progress on graphic hardware, interactive rendering is still one of the biggest challenges in video games, specially for very low graphic power portable devices such as portable video consoles, smart phones and so on.

Although many calculations are been predated by GPUs from the CPUs, there are still many of them that still remain in generic CPUs using multithreading and multicore CPUs. What is more, non graphical CPU loads (simulation, AI, physics, HCI,...) are increasing every time.

So, a reduction in the graphical payload on the CPU is mandatory in order to improve graphic performance (next gen video games) [1] [2], reduce power consumption (mobile games) [3] or to increase other video games aspects like behavior (AI), realistic appearance (physics, real time fluids simulation) [4], smoother simulation (increasing sampling frequency) or improving HCI through more complex interactions [5].

Polygonal simplification algorithms and LoD techniques are one of these techniques used to reduce rendering cost keeping a visual fidelity grounded on a geometric criteria. They are based on the fact that mostly of the geometry of a detailed 3D model is unnecessary depending on the visual circumstances.

The quality of an image made by a computer depends unavoidably on the visual perception of the viewer that is sat in front of the computer 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 influences on the perception quality of a computer generated image.

LoD techniques have been normally used to change the amount of triangles in an object mesh depending on a distance-to-the-camera criterion. In this paper we propose to associate LoD techniques to the eye drop of perception when a sharp change in the illumination of a scene happens during the performance of a video game. In this case, from very bright scenes to very dark ones.

In point 2 of this paper, we present a brief background of LoD techniques and a brief analysis of the Human Visual System (HVS) perception and we introduce a critical analysis that justifies our new approximation to the problem. Point 3 presents the way we designed the tests to verify our hypothesis: the models used, the luminosity changes, model resolutions, criteria for the resolution changes and test organization. Point 4 present the results of our tests on human players. Point 5 and beyond finally resume the conclusions and future works of our research.

BACKGROUND AND PREVIOUS WORK

Level of detail

Level of Detail (LoD) offers a means of improving the performance of a video game under certain circumstances by trading visual detail for speed [ASTHE1994]. This is done by storing a number of representations of an object, each varying in complexity (e.g. polygon count), and then selecting an appropriate model to use at each frame of the simulation. A more complex representation will appear more detailed, but it will consume more computing time.

The most common selection criterion used to determine an object's LoD is its distance from the viewpoint, i.e. lower detail models are employed as the object becomes more distant. Indeed, it is this case which most people associate with the term LoD. There are two principal ways of implementing distance LoD: either by calculating the distance of the object from the viewpoint, or by determining the area of the object's bounding volume on the screen.

Our aim in this paper is to use any discrete LoD technique to reduce objects complexities, not to present just another LoD algorithm.

Visual perception

The HVS is capable of functioning over a huge range of light levels. The amount of light available on a bright day at the beach ($10^8 cd/m^2$) is 10,000 times greater than the light available in a dimly lit room [6] and much more than the low limit the eye can perceive ($10^{-6} cd/m^2$). The visual system can manage this huge range using a process known as *eye adaptation*. The adaptation is not instantaneous and the process can take from 20 to 30 minutes and it is different for rods and cones [7]. For example, when entering into a dark room, like a cinema hall, it is not easy to find a seat in the darkness. However, after a few minutes, many details can be seen that were barely perceptible before.

As the eye adapts to the darkness, pure scotopic vision starts from $0.034 cd/m^2$ down, its sensitivity improves. On the other hand, as light increases, eye sensitivity is lost allowing to see high levels of luminosity (pure photopic vision starts from $3.4 cd/m^2$ up). Nevertheless, the eye dynamic range is higher in photopic conditions [8].

Graph 1 shows the time necessary to dark adaptation for cones and rods. The red line corresponds to rods adaptation and blue line to cones adaptation. The result of both is the curve shown with block dots [9]. We observe that cones reach their maximum sensibility before the rods. After 8 minutes approximately, rods are more sensitives than cones and during this transition (Mesopic vision) images have a very strong appearance of blue-ish colors until a completely loss of the colors in scotopic conditions. This is called the Purkinje effect [10, 11].

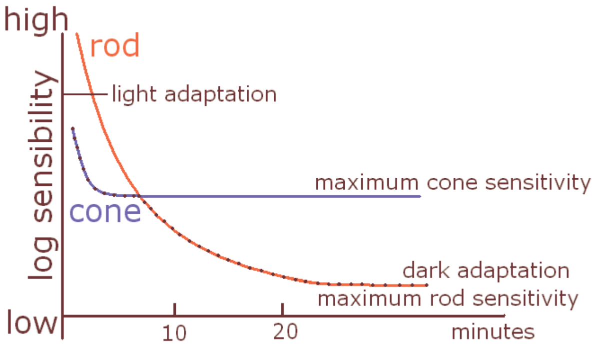

Figure 1: Dark adaptation curve.

Critic to the state of the art

In order to modulate the amount of geometry a LoD technique may discard, there have been some approaches in the past such as relating LoD to an object's motion [GIACOMO2007], peripheral location using the gaze-contingent rendering of the HVS [MURPHY2007] or to automatically balance the system's computational load in an attempt to maintain a fixed frame rate [FUNK1993].

Although the eye adaptation process is known a lot of time ago, it has not been used in computer graphics in order to simplify scenes according to light changes during the execution of a real time graphical environments such as video games or VR applications.

Our aim, in this paper, is to evaluate the feasibility to modulate LoD techniques associated to lighting changes. For this reason we did a perception test using LoD techniques and changing the lighting of the scene between some given values. We wanted to determine the amount of time a video game can reduce its geometry quality without being noticed by a player. This amount of time collected by the game in this transition times can be used to improve other parts of the game: AI, physics, HCI,... or to allow the video game run at higher frame rates or in lower power consoles.

IMPLEMENTATION

This section describes the tests made to collect reaction time results. Each person was invited to pass two tests. Each probe was individual and sequential. The first part consisted on a reaction test with the aim of measuring the time elapsed between a visual stimulus and hitting any key as fast as the player could. The second part measured the time of adaptation when the viewer change from a bright environment to a darker environment. Finally, the true adaptation time was calculated subtracting the time of reaction from the time of adaptation.

Reaction test

This main purpose of this test is to know how long it takes to the viewer to react after a stimulus. The test produces stimulus of maximum contrast, it flips between a white and a black screen. The viewer has to react as soon as possible pressing any key. During the test, there are 5 luminosity changes separated between different time intervals in order to avoid synchronizations. It measures all the times of reaction and it calculates the average time of reaction. Times intervals are shown on table 1. They are always the same sequence and in the same order. Times were calculated randomly.

Stimulus	Time (ms)
1	4000
2	2000
3	5000
4	500
5	1500

Table 1: Times intervals between stimulus.

Perception test

This test tried to measure the HVS time of adaptation for different luminosity changes. The idea was to present an image with a certain luminosity during a time of adaptation (approximately 10 seconds). This image showed an object non-simplified (NS object). Afterward, the viewer pressed a key and it occurred a decrement of luminosity on the image. At this moment, the test flipped a image of the NS object and one of a simplified model (S object) using the same darkness conditions. The flipping was not done at the screen refresh rate because of the persistence effect of the eye. The images were interlaced at a rate of 10fps, 5 for the NS image (odd frames) and 5 for the S image (even images). The viewer had to press a key as soon as possible when he realized any slight vibration or any movement on the object. When the key was pressed, it meant that the viewer HVS were adapted to the new luminosity conditions and he noticed the biggest polygonal reduction. After the first detection, the test increased the resolution of the S object slightly to the next predefined step using a discrete LoD technique. This new resolution image was interlaced again with the NS object using the same darkness conditions. The viewer had to go on pressing a key when he detected any change. The resolution of the NS Object increased again during 3 loops more until the highest resolution was achieved.

Models choice
We chose two different models to perform the test. The first one was the Stanford Bunny. We chose this model due to its popularity in computer graphics and because it is a relatively simple model. It is so smooth that it contains a lot of redundant information in the original triangles collection. Simplification and compression algorithms produce very nice results. Since this tests is focused on interactive rendering acceleration, the number of triangles in the original model is far away too much (69,451 triangles) from real time applications. The test uses a basic bunny simplified up to 5,000 triangles (figure 2) with *Quadric Edges Collapse Decimation.* To simplify the model we used the tool *Meshlab* [12].

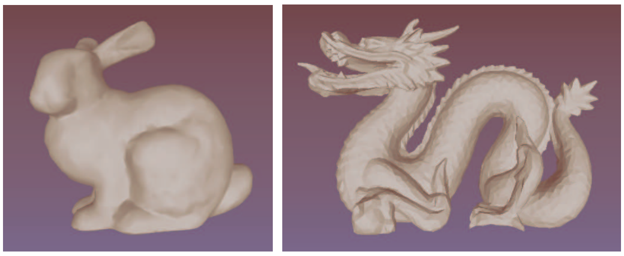

Figure 2: Basic bunny(5,000 polygons) and basic dragon (10,000 polygons)

On the other hand, the second model is more complex: the Stanford Dragon. It is extremely rough because of its flakes. In the same way as the original bunny, the dragon was also simplified to a basic dragon with 10,000 polygons (figure 2) without any loss of quality for the distance and size of the image produced for the tests.

Luminosity changes
Luminosity change is the difference of luminance between the initial image and the dark ones, in other words, is the difference of the Y component in the YIQ [13] model between both images. The Y component has value between 0 and 1: 0 is a completely black image and 1 a completely white one. In this test we consider that $Y = 2/3$. This value is the mean of every pixel on the screen Y value. This luminosity provides the brightest image without burnt areas. To calculate the other Y values in the rest of the images used for the test, the maximum luminosity was divided by 2 in every step. This is the same as to move one stop down of the f number of a conventional camera. Table 2 shows all different luminosity values used.
The minimum luminosity considered was 0.021 since lower values were scarcely visible on the screen and did no valuable results.

Luminosity changes depend on the preadaptation image and the next one. It is the difference between both of them the luminosity change than the viewer is perceiving. The maximum luminosity change occurs when it flips from the maximum value to the minimum value. Table 3 describes the Y value of the adaptation image

Luminosity (Y)	Value
2/3	0.666 / 27.030 cd
1/3	0.333 / 13,515 cd
1/6	0.166 / 6.737 cd
1/12	0.083 / 3,369 cd
1/24	0.042 / 1,705 cd
1/48	0.021 / 0.852 cd

Table 2: Different luminosity values used(Y).

(A image), the Y value of the image after the luminosity change (LC image) and the difference of luminosity. Y value translated into candelas is also shown taking into account the maximum bright of the screen (300cd/m2).

Change Nº	A Image	LC Image	Difference
1	0.666	0.021	-0.645/26.2 cd
2	0.666	0.042	-0.624/25.3 cd
3	0.666	0.083	-0.583/23.7 cd
4	0.333	0.021	-0.312/12.7 cd
5	0.333	0.042	-0.291/11.8 cd
6	0.333	0.083	-0.250/10.1 cd
7	0.166	0.021	-0.145/5.9 cd
8	0.166	0.042	-0.124/5.0 cd
9	0.166	0.083	-0.083/3.4 cd
10	0.083	0.021	-0.062/2.5 cd

Table 3: Description of the luminosity changes.

Models resolutions

The object of this work is to discover if it is possible to simplify a mesh during a luminosity change without affecting the visual quality the observer notices. In this case, the different mesh resolutions for the two models have been chosen following two different criteria: the Hausdorff distance and the RMS error. First of all, it has been defined the *LoD* for the base model (Bunny model). The resolutions were: 5000, 4000, 3000, 2000 y 1000 polygons.

Hausdorff distance We measured the mean Hausdorff distance in one direction (base model to simplified model) for each resolution of the bunny. This distance has been calculated with the tool *Metro* [14] which allows to compare pairs of meshes. Next, we found the corresponding *LoD* of the dragon with the same Hausdorff distance to the base dragon model with 10000 polygons. In table 4, different distances and resolutions of the dragon are shown. From now on, Dragon Hausdorff refers to the Stanford dragon model with the *LoD* indicated on table 4. We needed a metric to unify and compare the test made using different models (dragon and bunny) and resolutions. Although these models have different resolutions, they have the same Hausdorff distance in the same step of the test.. For example, the bunny *LoD* with 4000 polygons has a Hausdorff distance equal to 0.047 respect to the base model with 5000 polygons. The dragon *LoD* with 7100 polygons has exactly the same distance respect to the base model of 10000 polygons.

Bunny [1]	H. Distance (10^{-3})	Dragon H. [1]
5000	0	10000
4000	0.047	7100
3000	0.092	5200
2000	0.153	3700
1000	0.291	1970

Table 4: Hausdorff distance to the base model.

Error Root-Mean-Square (RMS) In this case, we followed the same method than before but we me sure the RMS error of the rendered images of each *LoD* of the bunny. The RMS error has been calculated with a function in *Matlab*. Next, we found the number of polygons necessary to have images of the dragon that produce the same RMS error than the bunny. In table 5, the RMS error and the different resolutions are shown. Dragon RMS refers to the different *LoDs* with the resolution of the table.

Bunny [1]	RMS error	Dragon RMS [1]
5000	0	10000
4000	0.3376	9500
3000	0.3394	7000
2000	0.3428	4300
1000	0.3496	3400

Table 5: RMS error to the base model.

Tests organization

First we passed the reaction test and later the perception test. The second one was divided into three parts: one for the bunny and two for the dragon resolution (Hausdorff and RMS). Each part had 10 luminosity changes. Table 6 shows the tests order with test number, the luminosity change and the model used.

---

[1] polygons

	Test number		
Change Nº	Hausdorff	Bunny	RMS
1	1	11	21
2	2	12	22
3	3	13	23
4	4	14	24
5	5	15	25
6	6	16	26
7	7	17	27
8	8	18	28
9	9	19	29
10	10	20	30

Table 6: Tests order.

In order to have reliable tests, they were all done in completely darkness so the person was not affected by other environmental factors. The tests were done with the same monitor (21.6" TFT 300cd/m2) for everybody and sat at the same distance (70 cm). So the monitor gave a global luminosity of 40.585cd using its maximum intensity and contrast. The luminosity (cd) associated to every change of the Y component of the whole image was calculated taking into account these numbers. Those values can be seen in Table 3 and 2.

Both tests were passed to 25 volunteers and each one took around 15 minutes to complete the test. 15 persons were males and 10 were females. Once all the test were done, we extracted the timing results. We subtracted the results of the reaction test to the results of the perception test. We obtained the timing results that each person took to realize the simplifications. Those final times are the ones that have been used to study the results.

ANALYSIS OF THE RESULTS

Our hypothesis was that the time of adaptation depended on the difference of lighting between the initial image and the last one. We assumed that no difference would be noticed when changing both initial and final lighting. In order to avoid very long tests that could be boring to the people tested and could discard the validity of the test, we decided to start from different starting intensities dividing the higher one by power of two. So, the starting Y luminance was 0.666, 0.333, 0.166 and 0.083. Since there were ten steps to change this maximum starting Y, we decided to jump to the three lower intensities: 0.083, 0.042 and 0.021. The combination of luminosity jumps can be seen in Table 3.

Models

In this part, there are drawn three graphs that correspond to the three models. Graph 3 shows the mean time of detection for each luminosity change and for each resolution (1000, 2000, 3000, 4000) for the bunny. Graph 4 represents the same for the dragon Hausdorff model and graph 5 for the dragon RMS model.

Comparing the models Bunny and Dragon Hausdorff, both models have a similar behavior for the first and second resolutions. Resolutions 3 and 4 present a time of detection higher for the bunny model. When comparing the bunny and the dragon RMS model, they also behave in the same way for the first and second resolutions. Time of the 3rd resolution is also higher for the bunny than for the RMS. However, using the 4th resolution, the time detection is considerably higher for RMS. Finally, comparing Hausdorff and RMS models, similar results for resolutions 1, 2 and 3 are obtained but RMS produces times much higher than Hausdorff for the 4th resolution.

Taking into account than the geometry generated using the Hausdorff metric is lower than the one generated using RMS, it is recommended to use Hausdorff metric to simplify a model when using these geometries in real time graphical applications.

We have to take into account that the differences between LoD models are more difficult to be found on the Bunny that on the Dragon because of their shapes.

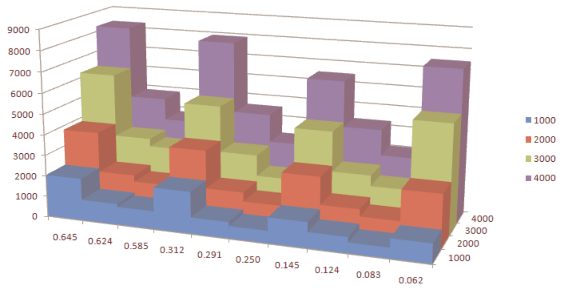

Figure 3: Detection time for bunny model.

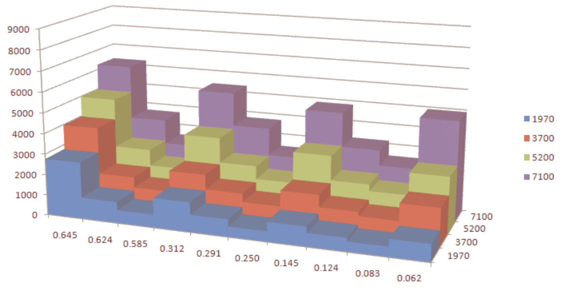

Figure 4: Detection time for Hausdorff dragon model.

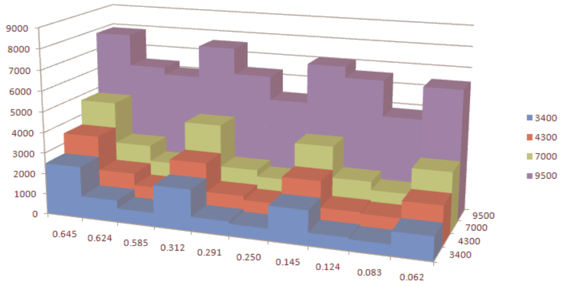

Figure 5: Detection time for RMS dragon model.

Comparing different luminosity changes

As can be seen, the three graphs follow the same tendency. For changes 1, 4, 7 and 10 (-0.645, -0.312, -0.145 and -0.062) the time detection is significantly longer than for the others changes. All the changes were made from a bright image in the pure photopic vision (26.2, 12.7, 5.9 and 3.4 cd) to one in the mesotopic vision window (0.85, 1.7 and 3,4 cd). Since cases 1, 4, 7 and 10 jump to 0.85 cd, nearer the pure scotopic vision (0.034cd), these cases are more influenced by the scotopic time of adaptation. That is the reason for higher time of adaptation showed.
All three graphs follow the same tendency:

1. If resolution of a mesh increases, the time of detection increases. It does not depend on the luminosity jump and LoD criteria (Hausdorff/RMS) or model (bunny/dragon)
2. Starting from the same luminosity, as the difference of luminosity increases, the time of adaptation increases. That is the reason why cases 1,2 and 3 show a descendant tendency. It is the same for cases 4, 5 and 6 and for cases 7, 8 and 9. This is specially visible when the luminosity of final images is closer to the scotopic vision: cases 1, 4, 7 and 10.
3. Arriving to the same luminosity, as the difference of luminosity increases, the time of adaptation increases. That is the reason why cases 2, 5 and 8 show a descendant tendency. It is the same for cases 3, 6 and 9. This is specially visible when the luminosity of final images is closer to the scotopic vision: cases 1, 4, 7 and 10.

We must enhance that even a black image on a monitor emits light, so probably the volunteers did not reach scotopic condition. However, a tunnel effect occurs and it is responsible of the adaptation time.

Non-detected cases
When a player detected a vibration or flickering in the image, he pressed a key. In some cases, the volunteers did not press any key. This means that they did not find any difference between the base model and the simplified model under some given luminosity conditions. Those cases are called non-detected.
Graph 6 shows the number of non-detected for every luminosity change and resolution for the bunny model. For example, in the first change, 12 people did not find any difference between the base model (5000 polygons) and the model with (4000 polygons). Graph 7 shows the number of non-detected for the Hausdorff dragon model and graph 8 for the RMS dragon model.

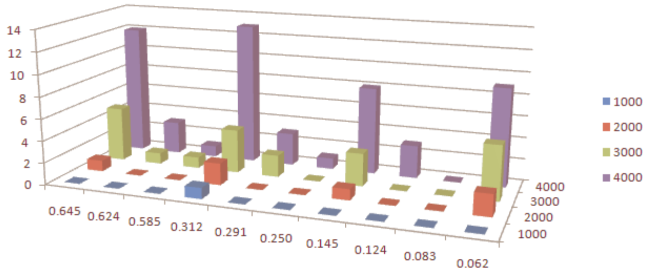

Figure 6: Number of non-detected for the bunny model.

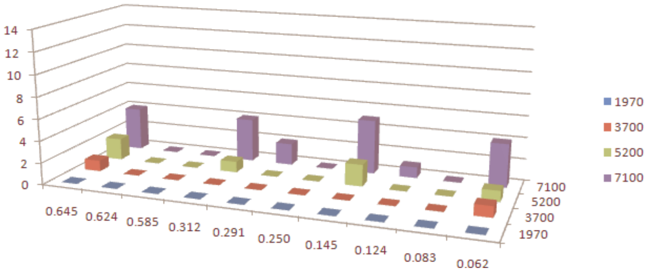

Figure 7: Number of non-detected for the Hausdorff dragon model.

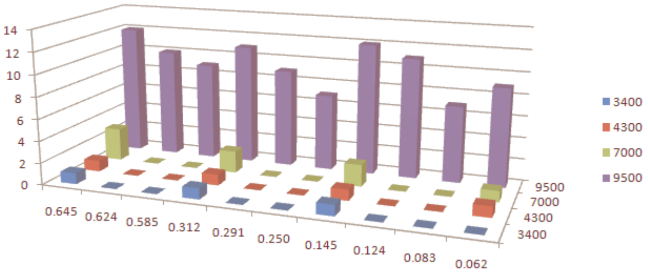

Figure 8: Number of non-detected for the RMS dragon model.

All three graphs follow the same tendency. The amount of non-detected changes is:

1. Lower as the luminosity jump decreases and the luminosity of the final images move apart from the scotopic vision. Cases 2, 5 and 8 present lower amount of non-detected changes than cases 1, 4, 7 and 10. Cases 3, 6 and 9 have lower amount than cases 2, 5 and 8.

2. Higher as the resolution of the mesh is closer to the resolution of the original object. This is specially observable figure 10 where the number of non-detected grows as the resolution grows as well and in figure 8 when using 9500 polygons (case 4). This is specially important when the final image luminosity is closer to scotopic vision. It means the darkest image (cases 1, 4, 7 and 10).

3. Higher depending on the resolution and also on the final luminosity as can be seen in figure 11.

It is not clearly visible if the preadaptation luminosity influences to the amount of non-detected changes. Cases 1, 4, sometimes are higher and sometime are lower than cases 7 and 10.
Finally, graph 10 shows the mean number of non-detected for the same start luminosity and graph 11 for the same final luminosity.

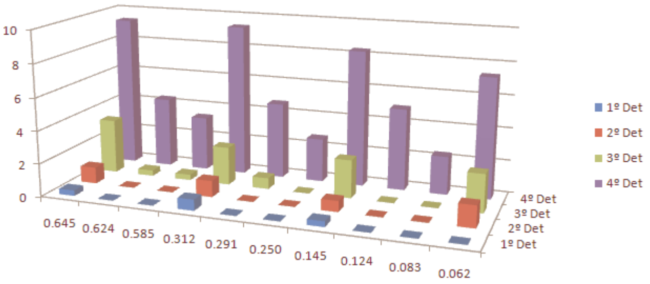

Figure 9: Mean of non-detected for all models.

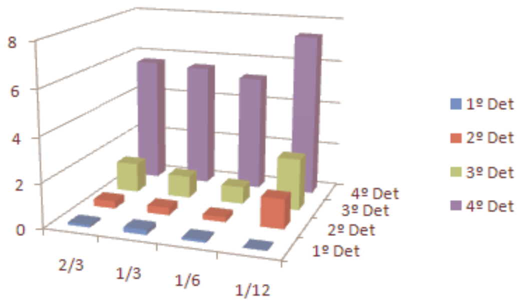

Figure 10: Number of non-detected for the same start luminosity.

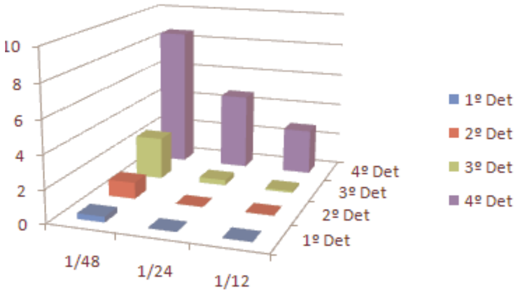

Figure 11: Number of non-detected for the same final luminosity.

CONCLUSIONS

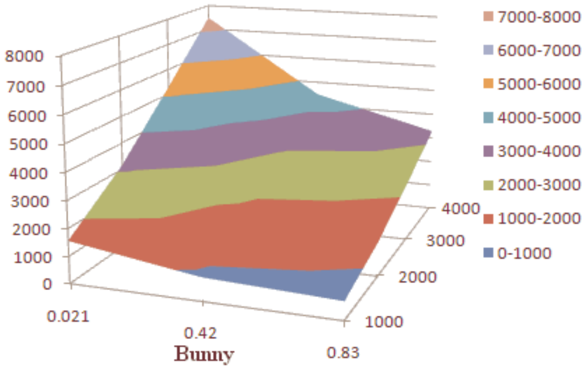

Figure 12: Bunny Spline.

After the analysis of the results we can extract some conclusions:

1. The duration of the time of adaptation is principally determined by the level of luminosity reached: the conditions that the image is observed after the luminosity change.

2. The luminosity of the image of pre-adaptation also influence this duration.

3. The results indicates that exist a few seconds of adaptation where the observer can not distinguish some details that will find afterward.

With these results we can not determine a universal function that could determine the simplification of a model during a luminosity change. This duration depends on the given mesh of a model and topology, the amount of mesh simplification and many other factors: textures, shadows, background, scene lighting,etc.
We can conclude that the type of metric to reduce the amount of polygons is significant in order to increase the time of adaptation. Given an amount of polygons to represent an object, a video game developer wants to

increase the time of detection as much as possible. Using the Hausdorff criterion with the 3700 and 7100 polygons dragon and comparing to the 3400 and 7000 polygons RMS dragon, we can see that the time of adaptation is considerably higher in Hausdorff tests than in RMS tests. So we do recommend Hausdorff criterion to reduce polygons in applications that use LoD techniques.
We emphasize that the timing obtained is really a minimum time since all the player's attention is focused on the object. In real video games, the attention is focused on the main actors. All the objects around these actors could have even higher time of adaptation: backgrounds, furniture,...

Criticism

The test were done with statics images and the whole majority of the volunteers recognized that they focused their attention on some critical points of the meshes. A critical point is some area in the mesh that present a big difference between the original model and the simplification. At these points, very small simplifications produces very big geometrical differences that can be seen easily.
The amount of people tested is not statistically significant (25 tests) but it shows a tendency that has been observed unalterably in almost all tests performed. The range of age rises from 18 to 42 years, covering a wide range of current players population.

Future work

In order to be closer to the current applications (videogames, movies) and avoid the detection of static critical points, we suggest to perform the tests moving the meshes and changing the point of view.
We have to test other criteria of simplification, apart from Hausdorff and RMS, in order to increase the time of adaptation using the same amount of polygons.
We would like to increase the amount of players tested and widen the age spectrum, specially for young people that could not be included in this study (6 to 18 years).
Finally we want to enter in the scotopic vision to test this behavior in very dark luminance conditions.

References

[1] William Mark. Future graphics architectures. *ACM Queue*, 6:54–64, March/April 2008.

[2] T.J. Olson. Hardware 3d graphics acceleration for mobile devices. *IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP*, pages 5344–5347, 2008.

[3] Iris; Ohlenburg Jan; Braun Anne-Kathrin; Wetzel Richard Wolfgang; Lindt, Irma; Herbst. Toward next-gen mobile ar games broll. *CG and A*, 28:40–48, July/August 2008.

[4] Vincent; Ghazanfarpour Djamchid; Pueyo-Xavier El Hajjar, Jean-François; Jolivet. A model for real-time on-surface flows. *The Visual Computer, Springer Berlin / Heidelberg*, 2008.

[5] J. Joseph Jr. LaViola. Bringing vr and spatial 3d interaction to the masses through video games. *IEEE CG and A*, 28:10–15, September/October 2008.

[6] Colin Ware. *Information visualization Perception for design*. Elsevier Inc, 2004.

[7] C. Haig S. Hecht and A. M. Chase. The influence of light-adaptation on subsequent dark-adapatiotion of the eye. *Journal of general Physiology*, 20:831–8501, 1937.

[8] H. R. Blackwell. Contrast thresholds of the human eye. *Journal of the Optical Society of America*, 28, 1946.

[9] R. Martinez López. El poder del ojo adaptado a la oscuridad. *Gaceta Óptica*, 403:20–24.

[10] JE Purkinje. *Neure Beitrage zur Kenntniss des Sehens in Subjectiver Hinsicht.* Reimer, 1825.

[11] John P Frisby. *Seeing: Illusion, Brain and Mind.* Oxford University Press, 1980.

[12] P. Cignoni. *MeshLab Visual Computing Lab.* ISTI - CNR, 2007.

[13] S. Feiner J.F. Hughes J.D. Foley, A. van Dam. *Computer Graphics, principles and practice.* Addison-Wesley, second edition, 1997.

[14] C. Rocchiniz P. Cignoniy and R. Scopigno. Metro: measuring error on simplified surfaces. *Computer Graphics Forum, Blackwell Publishers*, 17(2):167–174, 1998.