A Comparison of Orthographic and Perspective Projections in the Generation of Textures for Billboards

Carlos Saraiva
IST/INESC-ID
Av. Rovisco Pais, 1000 Lisboa
carlos.saraiva@ist.utl.pt

João Fradinho Oliveira
IST/INESC-ID
Av. Rovisco Pais, 1000 Lisboa
Joao.Oliveira@vimmi.inesc-id.pt

João Madeiras Pereira
IST/INESC-ID
Av. Rovisco Pais, 1000 Lisboa
jap@inesc.pt

Bruno Rodrigues de Araújo
IST/INESC-ID
Av. Rovisco Pais, 1000 Lisboa
brar@vimmi.inesc-id.pt

Abstract

Image-based rendering techniques take the approach of rendering new images based on existing ones, effectively separating the rendering complexity from the geometric complexity of the scene they represent. With the advance of hardware capabilities, these techniques have regained interest. However, the creation of images previously mentioned is error-prone, so computation must be done in order to correct them. This paper presents a comparison between the uncorrected errors apparent in billboards when the applied texture is generated with a perspective or with orthographic projection.

Keywords

Image-based rendering, orthographic, perspective, projection, billboard, distortion, comparison

1. INTRODUCTION

Despite the fast advance of hardware capabilities, the complexity of the scenes to render has also grown extremely fast. Even with today's modern hardware, complex scenes cannot be rendered in real-time by brute-force methods.

This introduces the need of simplifying the scene to be rendered, reducing its inherent complexity to levels that can be rendered faster, with as little loss in visual quality as possible.

For this purpose, several techniques have been invented. These techniques can fit in two categories: reduction of the rendering space and reduction of the complexity of the scene.

Techniques in the first category denominated as visibility culling algorithms discard invisible parts of the scene from the rendering pipeline, usually aided by a subdivision of the scene into a hierarchy of volumes, such as a BSP [Fuchs80], octrees [Jackins80] or kd-trees [Bentley75]. These hierarchical structures allow for great portions of the scene to be quickly discarded, as if one of the nodes is invisible, it is guaranteed that its descendant nodes are also invisible.

Techniques in the second category try to reduce the complexity of a portion of geometry in the scene, in order to speed up the rendering. These can range from simplifying the geometry using level of detail mechanisms [Clark76] (taking advantage of the reduced projection and perceived detail of the geometry as the distance increases) to alternate representations that use nearly no geometry at all in representing the object.

These alternate representations can range from converting the geometry into a surface defined by an equation that can be solved reasonably fast, to image-based rendering approaches. This last alternate representation method strives to render the scene from a certain viewpoint given renders of the scene from other viewpoints. From these other renders, information of the scene can be extracted and reprojected into the new viewpoint.

This approach is being considered for use in the development of VIZIR, an FCT (Fundação para a Ciência e Tecnologia) project which aims to develop a framework for interactive visualization of massive data sets using a cluster of machines built of off-the-shelf hardware, with support for multi-screen environments, such as the LEMe Wall at IST/TagusPark [Araújo05] or the CAVE at Lousal [Costa07].

2. IMAGE-BASED RENDERING

In the category of image-based rendering techniques, many methods can be considered, ranging from those that use no geometry at all (like resampling a set of images given the viewing parameters [McMillan95a]) to more

hybrid approaches that use both geometry and images to represent the underlying scene [Jeschke02].

It is also important to note that image-based rendering can be more useful when dealing with certain types of scenes which aren't handled so well with geometric level of detail techniques. An example of such a scenario is the representation of a model constituted by "rough" curves (like a fractal): too much triangle reduction and what is supposed to be a curve can become noticeably linear; too small of a reduction can result in a great number of triangles to draw. Most image-based techniques do not suffer from this problem, as they are independent of the scene.

The techniques can also be classified by how the images are created: some use a predetermined set of possible viewing positions, creating images that represent more distant zones in the scene [Sajadi09]; others create images of objects or subvolumes of the scene and render the scene by representing each of the objects or subvolumes [Schaufler98].

This last category has the advantage that the scene can be represented from any viewpoint and not only a predetermined set. One possible approach useful for viewing models from an outside point of view is to render the model from a series of viewpoints and use those renders as textures placed into billboards, as seen in [Aliaga99].

Without correction, this technique can introduce distortion, since the projection of the scene into a plane done during the texture generation can differ from the projection used while viewing the scene with billboards.

This correction is performed by reprojecting an image generated from known viewing conditions into another viewing condition. This reprojection is a function that maps a texel to a number of pixels in the final image. However, several texels can be mapped onto a single pixel, so the texel that has the closest projection is used. Some techniques employ a z-buffer to this end, others are based on analysis of the image [McMillan95b].

Some methods of correction are CPU-based, where each texel in the original image is simply reprojected into the new viewpoint [McMillan97]. While highly parallel in nature, this kind of technique is not trivial to reproduce in GPU using current graphical APIs, such as OpenGL or DirectX.

In order to exploit the capabilities of modern GPUs, the reverse is usually attempted: finding which source element is the one used in the destination element and sampling it. This problem is not trivial to solve since there is the possibility of the existence of multiple solutions and therefore a search must be performed for accurate calculation of the source element to use.

These GPU-based methods change the sampled texture coordinate based on a structure that contains the displacement relative to the billboard's normal, typically a texture called a height map (like seen in [Kaneko01]).

Some techniques compensate for this displacement by applying an offset to the texture coordinate based on the view angle relative to the billboard and the height of the billboard at that point [Kaneko01]. While fast, this technique is not accurate and can therefore present incorrect results.

Other techniques strive for absolute correctness, at the cost of performance. Such techniques typically perform a short-distance raycast on the height map, considering points by an increasing order of distance to the camera. If no solutions are skipped due to a large step size, the first solution found is the correct one [Tatarchuk06].

However, there is still an underlying question: determining what projection should be used during the billboard texture generation, so that the distortion is the smallest in a full range of views, and thus requiring less correction. With that question in mind, this paper makes a comparison using textures generated with perspective projections and orthographic projections. The technique presented, however, does not strive to be a complete solution to be used by itself, in every case: the disk space needed can be too large if the technique is applied without care, it may not work well with all scenarios (interpenetrating objects, dynamic lighting/shadows, etc), pre-processing time might be too long for the intended application, etc. However, when complemented with other techniques, for the purpose of viewing a static scenario, it can become a viable choice. Furthermore, the focus of the paper is the comparison of the two projections for the generation of billboards and does not present a thorough analysis of the problems which are intrinsic with a billboard approach.

3. TEST METHODOLOGY

An implementation was made in OpenGL that creates billboards from the several viewing angles using predetermined lighting conditions. These angles around the bounding sphere of the object are given by the center of the triangles of a 4 times subdivided icosahedrons to create 5120 billboard images with orientation steps of less than 5.5 degrees in any direction [Oliveira06]. The geometry of these billboards is set on a plane that always intersects the center of the sphere. To save disk space, these images are cropped to fit the content. An example can be seen in Figure 24 in the appendix.

For the comparison, these billboards are rendered without lighting and are compared to the geometric version (original triangles) used as a correct reference point. This geometric version is rendered in the same lighting conditions used during the billboard texture generation. This ensures that the lighting conditions do not affect the test results. It should be noted that lighting can be somewhat made dynamic: in addition to the diffuse color, one can store the normals of the surface. However, lighting will present problems with shadows and non-directional lights, since the surface being used to represent the scene is a plane with incorrect depth and the sampled texture is not corrected for perspective distortion.

Since the scale of the scanned model can vary widely, it becomes important that the test be insensitive to that factor.

To do so, the bounding sphere of the scene was calculated. The purpose of the sphere is to give a base measure of distance to be used in perspective transformations. At this base distance (a function of the radius of the sphere and the field of view of the camera), the sphere tightly fits into the viewing frustum. Relative distances are a multiplier placed on this base distance.

This base distance can be calculated by the formula:

$$b_{distance} = radius \div sin(\alpha)$$

where α is half the field of view angle.

To test the average distortion, each model was tested from 14 angles, given by the subdivision of a sphere into 6 meridional lines and 2 zonal lines, plus the two poles.

The relative distances used were 2^n with n varying between -1 and 6. At greater distances, the model became too small to be considered (smaller than a pixel).

With the two renders complete, they are compared by calculating an error for each pixel of the image, p_{dif}, as defined in [Oliveira08], where:

$$p_{dif} = |P1_R - P2_R| + |P1_G - P2_G| + |P1_B - P2_B|$$

The sum of all p_{dif} results in the total error of the image, but another useful measure is the average error of the image. This is done by dividing the total error by the amount of pixels that are in the area of interest. This area of interest is defined by the tightest axis-oriented rectangle that contains every pixel that is not part of the background from both renders.

The renders were made at the same resolution used when generating textures. The textures were generated with orthographic projection and with perspective projection, with fields of view of 40, 60 and 90 degrees. The same angles were used for viewing the billboards with perspective projection.

4. OBTAINED DATA

This section contains the data, separated in quantitative and qualitative results, used for the analysis in the next section.

4.1 Quantitative Results

Table 1 presents the total error and average error for the Stanford Bunny model, with billboards viewed with a perspective projection with a field of view of 40 degrees, according to the method used to generate the billboard texture. Tables 2 and 3 do the same, but for billboards viewed with a field of view of 60 and 90 degrees.

Cases where the field of view during runtime is the same as used for the generation of the textures are marked with an asterisk in the top of the relevant column.

The Tables corresponding charts are also shown (Figures 1-6), with the horizontal axis measuring relative distance and the vertical axis measuring error. In all the charts, the horizontal axis is presented in a logarithmic scale. Due to the large range of values of the charts showing the total error, these charts also present a logarithmic-scaled vertical axis (Figures 1, 3 and 5).

4.2 Qualitative Results

All renders displayed are shown from an angle used to generate a texture (so the comparison is performed in the ideal situation). Other angles were used to obtain an average value of total and average error (with the greatest difference being less than 5.5 degrees, as mentioned).

Figures 7-12 show the Stanford Bunny model when rendered at two relative distances (more specifically, at 1 and 2 units of relative distance) with the difference (calculated as described in Section 3) being shown for ease of comparison. The difference is shown for each colour channel (since Bunny and David are in greyscale, their difference is also in greyscale).

Comparisons are also shown at a relative distance of 1 for the Batalha cathedral model (about 1 million triangles) and for the Stanford scanned David model (the 2mm scan resolution version, with approximately 8 million triangles) in Figures 13-18. We note that the results from a relative distance of one are rather close to the model. We were somewhat surprised to find the rendering quality which can be observed in the accompanying video to still be useful at that distance, whereas other perspective image based rendering techniques [Sajadi09] only use image impostors at a great distance.

Since the Batalha renders occupy less screen-space, for easier comparison all three images of the renders presented in this paper have been equally cropped (about 20% in each side).

In addition, the appendix at the end contains some more renders (Figures 19-22) of the Bunny model at a relative distance of 1, with textures generated using various perspective projections along with the respective difference to the correct triangle representation displayed.

5. ANALYSIS OF THE DATA

In the average error charts, an interesting phenomenon occurs: the average error starts decreasing with distance and then increases again. This increase of error accompanying the increase in distance can seem strange, as increased distance would supposedly make the projective rays closer and closer to a perspective projection with a low field of view (or even an orthographic projection), so one could be tempted to say that as the distance from the object grew, the error would converge to zero (becoming zero when the object became smaller than a pixel). The lowest average error can be seen marked in bold in Tables 1-3, with the value of least error always being at a relative distance of 4.

This effect, however, can be explained: as it can be seen in the comparative Figures 7-12, the greatest error lies on the border of the model. At any distance, the maximum possible distortion increases with the distance to the center of the billboard. Therefore, the border of the model, which is farther away from the center of the billboard, has a greater distortion. Comparing Figures 9 and 12, one can see that the interior part of the projection of the model decreases in error, but the borders continue with a great error. As the distances increases, this fact maintains, and

Rel. Distance	Ortho		FOV 40*		FOV 60		FOV 90	
	Total	*Average*	*Total*	*Average*	*Total*	*Average*	*Total*	*Average*
0.5	3.44×10^{7}	$1{,}45\times10^{2}$	$3{,}39\times10^{7}$	$1{,}44\times10^{2}$	$4{,}00\times10^{7}$	$1{,}70\times10^{2}$	$5{,}10\times10^{7}$	$2{,}18\times10^{2}$
1	7.15×10^{6}	$9{,}69\times10^{1}$	$7{,}42\times10^{6}$	$1{,}03\times10^{2}$	$9{,}47\times10^{6}$	$1{,}31\times10^{2}$	$1{,}31\times10^{7}$	$1{,}81\times10^{2}$
2	1.11×10^{6}	$6{,}49\times10^{1}$	$1{,}40\times10^{6}$	$\mathbf{8{,}13\times10^{1}}$	$1{,}99\times10^{6}$	$\mathbf{1{,}15\times10^{2}}$	$2{,}95\times10^{6}$	$1{,}71\times10^{2}$
4	2.20×10^{5}	$\mathbf{5{,}27\times10^{1}}$	$3{,}55\times10^{5}$	$8{,}43\times10^{1}$	$4{,}86\times10^{5}$	$\mathbf{1{,}15\times10^{2}}$	$7{,}14\times10^{5}$	$\mathbf{1{,}70\times10^{2}}$
8	6.76×10^{4}	$6{,}82\times10^{1}$	$1{,}01\times10^{5}$	$1{,}01\times10^{2}$	$1{,}31\times10^{5}$	$1{,}31\times10^{2}$	$1{,}80\times10^{5}$	$1{,}80\times10^{2}$
16	2.86×10^{4}	$1{,}22\times10^{2}$	$3{,}31\times10^{4}$	$1{,}42\times10^{2}$	$3{,}87\times10^{4}$	$1{,}66\times10^{2}$	$4{,}95\times10^{4}$	$2{,}13\times10^{2}$
32	9.56×10^{3}	$1{,}99\times10^{2}$	$1{,}09\times10^{4}$	$2{,}26\times10^{2}$	$1{,}12\times10^{4}$	$2{,}31\times10^{2}$	$1{,}38\times10^{4}$	$2{,}88\times10^{2}$
64	$\mathbf{3.83\times10^{3}}$	$5{,}26\times10^{2}$	$\mathbf{3{,}87\times10^{3}}$	$5{,}30\times10^{2}$	$\mathbf{3{,}94\times10^{3}}$	$5{,}40\times10^{2}$	$\mathbf{4{,}18\times10^{3}}$	$5{,}75\times10^{2}$

Table 1. Error for the Stanford Bunny when viewed with a perspective projection with a field of view of 40 degrees, with the smallest errors in each column marked in bold.

Rel. Distance	Ortho		FOV 40		FOV 60*		FOV 90	
	Total	*Average*	*Total*	*Average*	*Total*	*Average*	*Total*	*Average*
0.5	$4{,}22\times10^{7}$	$1{,}80\times10^{2}$	$4{,}20\times10^{7}$	$1{,}83\times10^{2}$	$4{,}67\times10^{7}$	$2{,}03\times10^{2}$	$5{,}57\times10^{7}$	$2{,}42\times10^{2}$
1	$8{,}01\times10^{6}$	$1{,}19\times10^{2}$	$7{,}88\times10^{6}$	$1{,}21\times10^{2}$	$9{,}50\times10^{6}$	$1{,}46\times10^{2}$	$1{,}24\times10^{7}$	$1{,}90\times10^{2}$
2	$1{,}24\times10^{6}$	$8{,}30\times10^{1}$	$1{,}36\times10^{6}$	$9{,}22\times10^{1}$	$1{,}83\times10^{6}$	$1{,}23\times10^{2}$	$2{,}66\times10^{6}$	$1{,}79\times10^{2}$
4	$2{,}18\times10^{5}$	$\mathbf{6{,}20\times10^{1}}$	$3{,}11\times10^{5}$	$\mathbf{8{,}74\times10^{1}}$	$4{,}30\times10^{5}$	$\mathbf{1{,}21\times10^{2}}$	$6{,}27\times10^{5}$	$\mathbf{1{,}76\times10^{2}}$
8	$6{,}27\times10^{4}$	$7{,}37\times10^{1}$	$8{,}48\times10^{4}$	$9{,}88\times10^{1}$	$1{,}12\times10^{5}$	$1{,}31\times10^{2}$	$1{,}58\times10^{5}$	$1{,}85\times10^{2}$
16	$2{,}47\times10^{4}$	$1{,}25\times10^{2}$	$2{,}83\times10^{4}$	$1{,}43\times10^{2}$	$3{,}34\times10^{4}$	$1{,}68\times10^{2}$	$4{,}46\times10^{4}$	$2{,}24\times10^{2}$
32	$9{,}19\times10^{3}$	$2{,}31\times10^{2}$	$9{,}39\times10^{3}$	$2{,}36\times10^{2}$	$1{,}01\times10^{4}$	$2{,}54\times10^{2}$	$1{,}28\times10^{4}$	$3{,}23\times10^{2}$
64	$\mathbf{3{,}27\times10^{3}}$	$5{,}18\times10^{2}$	$\mathbf{3{,}29\times10^{3}}$	$5{,}20\times10^{2}$	$\mathbf{3{,}34\times10^{3}}$	$5{,}29\times10^{2}$	$\mathbf{3{,}68\times10^{3}}$	$5{,}83\times10^{2}$

Table 2. Error for the same model when viewed with a field of view of 60 degrees instead, with the same marking for the smallest errors.

Rel. Distance	Ortho		FOV 40		FOV 60		FOV 90*	
	Total	*Average*	*Total*	*Average*	*Total*	*Average*	*Total*	*Average*
0.5	$5{,}31\times10^{7}$	$2{,}34\times10^{2}$	$5{,}26\times10^{7}$	$2{,}36\times10^{2}$	$5{,}56\times10^{7}$	$2{,}50\times10^{2}$	$6{,}14\times10^{7}$	$2{,}76\times10^{2}$
1	$7{,}19\times10^{6}$	$1{,}41\times10^{2}$	$6{,}99\times10^{6}$	$1{,}43\times10^{2}$	$8{,}01\times10^{6}$	$1{,}63\times10^{2}$	$9{,}70\times10^{6}$	$1{,}99\times10^{2}$
2	$1{,}04\times10^{6}$	$1{,}00\times10^{2}$	$1{,}08\times10^{6}$	$1{,}07\times10^{2}$	$1{,}38\times10^{6}$	$1{,}36\times10^{2}$	$1{,}88\times10^{6}$	$1{,}85\times10^{2}$
4	$1{,}77\times10^{5}$	$\mathbf{7{,}50\times10^{1}}$	$2{,}23\times10^{5}$	$\mathbf{9{,}43\times10^{1}}$	$3{,}01\times10^{5}$	$\mathbf{1{,}26\times10^{2}}$	$4{,}27\times10^{5}$	$\mathbf{1{,}79\times10^{2}}$
8	$5{,}26\times10^{4}$	$9{,}59\times10^{1}$	$6{,}74\times10^{4}$	$1{,}21\times10^{2}$	$8{,}33\times10^{4}$	$1{,}47\times10^{2}$	$1{,}09\times10^{5}$	$1{,}93\times10^{2}$
16	$1{,}84\times10^{4}$	$1{,}51\times10^{2}$	$2{,}25\times10^{4}$	$1{,}84\times10^{2}$	$2{,}60\times10^{4}$	$2{,}10\times10^{2}$	$3{,}00\times10^{4}$	$2{,}42\times10^{2}$
32	$7{,}11\times10^{3}$	$3{,}09\times10^{2}$	$7{,}67\times10^{3}$	$3{,}29\times10^{2}$	$8{,}38\times10^{3}$	$3{,}54\times10^{2}$	$9{,}13\times10^{3}$	$3{,}86\times10^{2}$
64	$\mathbf{2{,}42\times10^{3}}$	$6{,}75\times10^{2}$	$\mathbf{2{,}54\times10^{3}}$	$7{,}13\times10^{2}$	$\mathbf{2{,}65\times10^{3}}$	$7{,}35\times10^{2}$	$\mathbf{2{,}73\times10^{3}}$	$7{,}48\times10^{2}$

Table 3. Error for the same model when viewed with a field of view of 90 degrees (smallest errors marked in bold).

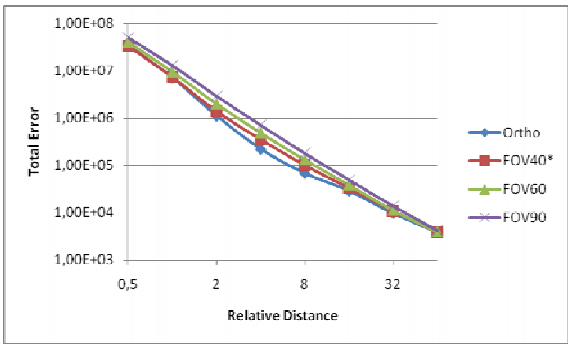

Figure 1. Total error with a FOV of 40 degrees.

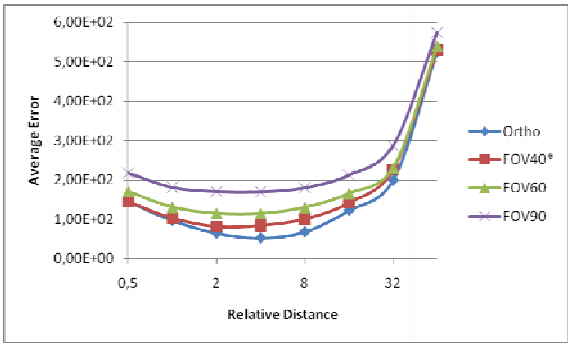

Figure 2. Average error with a FOV of 40 degrees

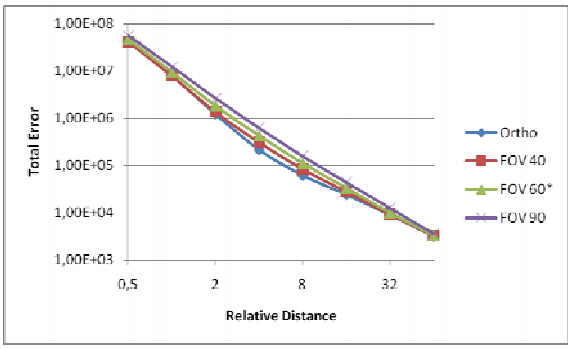

Figure 3. Total error with a FOV of 60 degrees.

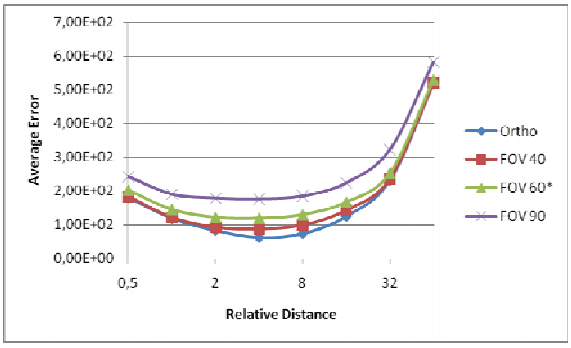

Figure 4. Average error with a FOV of 60 degrees.

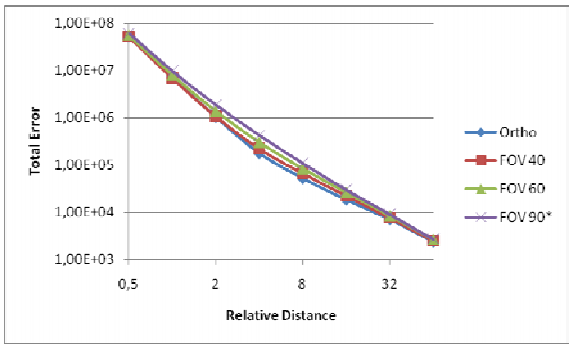

Figure 5. Total error with a FOV of 90 degrees.

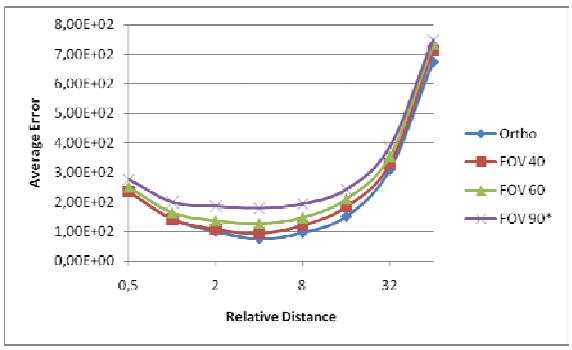

Figure 6. Average error with a FOV of 90 degrees.

eventually, the border error becomes dominant over the interior part with low error. Figure 23, in the appendix, shows how the border difference is still great, even at large distances.

Also, as the distance increases, another factor comes into account: the sampling of the texture. If no mipmapping is used, the sampled texel can easily miss an important feature of the texture. If mipmapping is used, another problem arises in its construction by the GLU library (at least in the implementation used): when downscaling the image, the transparency (alpha) values are not taken into account when determining the downscaled colour value.

This results in the background colour "bleeding" [Purnomo04] into neighbouring texels. The only way to guarantee that this effect is not present in the scene is to discard pixels for which the alpha value is less than one. This can result in otherwise valid pixels being discarded only because they suffered from this mixture with the background colour. If the alpha test is not made, pixels are not discarded, but instead a colour error is introduced. Since at larger distances the border becomes a more prominent feature, this results in an increase in average pixel error with the increase in distance.

Due to the fact that these artifacts occur, the behaviour of the average error at higher distances is hard or impossible to analyse reliably.

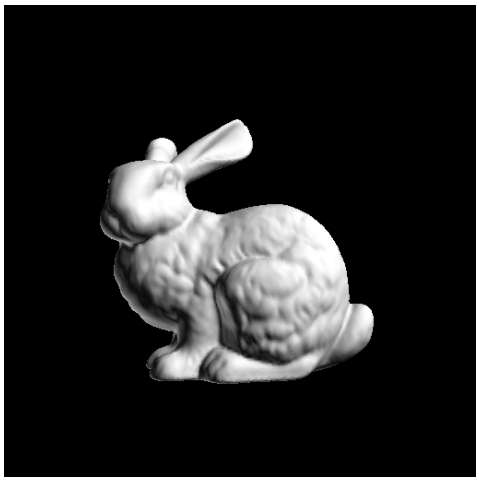

Figure 7. Triangle representation rendering of the Bunny model with a FOV of 40 degrees at a relative distance of 1.

Figure 8. Billboard representation rendering (with orthographic texture) of the model in the same conditions.

Figure 9. The difference between the two images.

Figure 10. Triangle representation rendering of the Bunny model with a FOV of 40 degrees at a relative distance of 2.

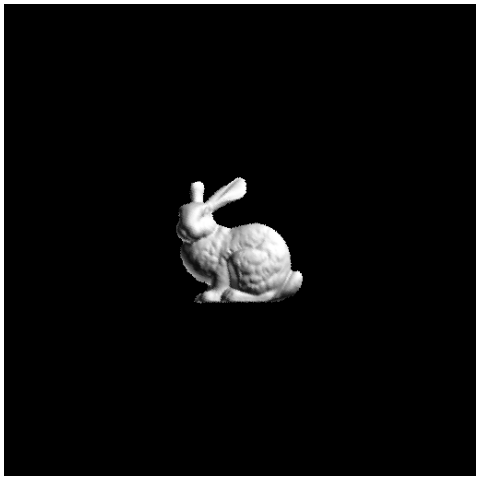

Figure 11. Billboard representation rendering (with orthographic texture) of the model in the same conditions.

Figure 12. The difference between the two images.

However, at these great distances the total error becomes useful. Since the rendering resolution is the same for all the distances, it represents on an equal scale how much error is visible.

The charts detailing total error (with both axes logarithmically-scaled) show that with an exponentially-increasing distance, the error also decreases exponentially, with the lowest total error being at the greatest distance (as one would expect). This can be seen marked with bold in Tables 1-3: the value of least total error is always at the last row and as one looks down the total error columns, the error is always decreasing.

On the total error charts, one can see that there is a consistent protuberance from relative distances 2-16. This is due to the fact that at shorter and longer distances there is either too much or too little error, respectively.

Despite all the artifacts, it can be seen that in all cases the billboards using textures generated with orthographic projections provide either a smaller or approximately equal pixel error when compared to perspective projections.

6. RESULTS

First, it is important to show the advantage of image-based rendering (IBR) in terms of performance, as it's pointless to pay the price of having to render textures and reprojecting them if the performance is not affected by it.

For this purpose, we compared the rendering performance between the IBR approach (2 triangles plus texturing of billboard) versus the complete triangle representation rendering. This was done by measuring the frames per second output when both methods were used, using a program called FRAPS [FRAPS]. These were done at a relative distance of 1, so that the object was fully visible and occupied a reasonable amount of screen space.

This test was performed in an Intel Core2Duo E7200 2.53GHz with 2GB of RAM and a NVIDIA GeForce 8800GT (512MB VRAM). In both cases, the images were rendered by the GPU. The created billboard images as in [Sajadi09] all fit in main memory. "Pauses" in the accompanying video can be explained by the reuse of the same textures, as no better texture was found for that angle. The results can be seen in Table 4.

As it can be seen, the frame rate with the IBR approach is consistently high, regardless of the number of triangles it represents. This is the power of IBR approaches: insensitivity to the complexity of the geometry it represents, which makes it a desirable candidate in research for rendering massive data sets.

It is also important to explain why in the billboard representations the Batalha model has a frame rate that greatly exceeds the other two models. This is due to the nature of the model itself: like previously mentioned, the images in Figures 13-15 were cropped as the model occupied a smaller screen space. This is caused by the existence of some triangles in the far back, resulting from scanning rays hitting tangentially to the floor near the doors at the

	Bunny	**Batalha**	**David (2mm)**
#Triangles	69,451	1,160,186	8,254,150
Texture resolution	512×512	512×512	256×256
Total size of textures	5120MB	5120MB	1280MB
Cropped size of textures	1506MB	612MB	616MB
Billboard (FPS)	~2000	~5000	~2150
Triangles (FPS)	~300	~10	~2

Table 4. Frame rates of the IBR approach versus the rendering of all the triangles

Figure 13. Triangle representation rendering of the Batalha model with a FOV of 40 degrees at a relative distance of 1.

Figure 14. Billboard representation rendering (with orthographic texture) of the model in the same conditions.

Figure 15. The difference between the two images.

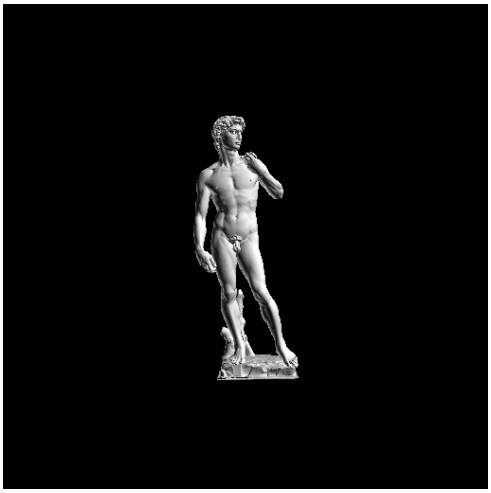

Figure 16. Triangle representation rendering of the David (2mm) model with a FOV of 40 degrees at a relative distance of 1.

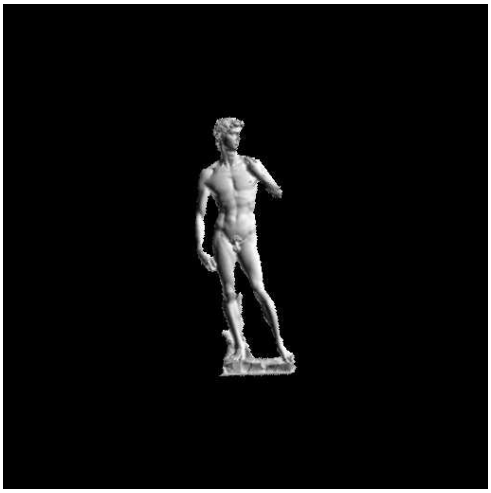

Figure 17. Billboard representation rendering (with orthographic texture) of the model in the same conditions.

Figure 18. The difference between the two images.

entrance of the cathedral (the doors are visible in Figure 13) thus creating long thin triangles in the subsequent mesh. This increases the radius of the bounding sphere so, in effect, the front of the cathedral occupies a relatively small portion of the volume of the sphere which in cause causes the projection to be smaller.

This combined with the fact that the billboard is cropped to only contain relevant portions of the generated image (as mentioned in Section 3), leads to the fact that the geometry used to represent the billboard also occupies less screen space, which leads to the great difference in performance seen.

We can also see from Figures 7 and 8 (and Figures 16 and 17) that, despite the existence of errors in the billboard approach, it can still produce visually pleasing results at short distance. Figure 14 does not present such a good result when compared to Figure 13, but this is due to the fact that it is being shown from an angle where the effect of perspective would be the greatest, and therefore has one of the biggest distortions possible. This effect could only be limited by the restriction of the viewer motion [Gobbetti08].

7. DISCUSSION AND FUTURE WORK

Despite the observed errors, it can be seen that in all the cases, the orthographic projection generally displayed lesser pixel error than perspective projections, with its worst performance being at extremely close distances. At longer distances, despite the increased average error, the image is very small thus almost imperceptible in our empirical study of the results.

We cannot however conclude that the orthographic projection is best for all cases, only that it provides least pixel error when uncorrected. To prove (or disprove) the superiority of orthographic projections for billboard generation, other errors will have to be measured, e.g. the error in texture coordinates instead of pixel error. This could be done by determining the difference between the

sampled texture coordinates and the texture coordinates when corrected by an accurate method (like raycasting on the height map as described by [Tatarchuk06]).

It would also be useful to correct some problems (like the mipmap construction) in order for the data collected to provide more reliable information. When done so, this could reveal other problems that were obscured by the ones detailed in this paper.

Another venue of study would be to study the error (pixel or otherwise) for the case where there is some geometric correction done. One example of such experiment would be the usage of more than one billboard, each representing different parts of the model (like a billboard for each node of a certain depth of an octree). Another example would be to place the geometry of the billboard such that it diminishes the sum of the absolute values in the height map (or placing the billboard in the centroid of the object instead of at the center of the bounding sphere). In the Batalha model, this would correspond to placing the billboard closer to the front seen in Figures 13 and 14 instead of being set on a plane that passes on the center of the sphere. This would provide some perspective correction, which could reduce the error observed (particularly in the Batalha model).

8. CONCLUSIONS

In this paper we presented an automatic approach for creating billboards and demonstrated that the billboards created with an orthographic projection generally displayed less pixel error when compared to billboards generated using perspective projections. We have also shown that this billboard method can be used at relatively close distances (when compared to typical use of billboards as impostors, which place them far away) with a good visual quality on models in which the height map created does not display a large gamut of values, therefore requiring less perspective correction (unlike the Batalha model, or the David model when viewed from above the head in the direction of the feet, which do display a large gamut of depths).

9. ACKNOWLEDGMENTS

We would like to thank INESC-ID and its VIMMI research group for the resources made available in the writing of this paper.We would also like to thank Instituto Português do Património Arquitectónico (IPPAR), now incorporated in Instituto de Gestão do Património Arquitectónico e Arqueológico (IGESPAR) and "Artescan, Tridimensional Digitization" for the model of the Batalha cathedral. For the Bunny and David scanned models, we also thank the Stanford University. The work presented in this paper was funded by the Portuguese Foundation for Science and Technology (FCT), VIZIR project grant (PTDC/EIA/66655/2006).

10. REFERENCES

[Aliaga99] Aliaga, D. Automatically reducing and bounding geometric complexity by using images. *PhD thesis, University of North Carolina at Chapel Hill*, 1999.

[Araújo05] Araújo, B., Guerreiro, T., et al. LEMe Wall: Desenvolvendo um Sistema de Multi-Projecção, *13º Encontro Português de Computação Gráfica*, 2005, 191-196.

[Bentley75] Bentley, J. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18 (1975), 509-517.

[Clark76] Clark, J. Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19 (1976), 547-554.

[Costa07] Costa, V., Pereira, J. et al. Tecnologias CAVE-HOLLOWSPACE para a Mina do Lousal. *15º Encontro Português de Computação Gráfica*, 2007, 69-78.

[FRAPS] A free application that can measure frame rates in OpenGL and DirectX applications `http://www.fraps.com`

[Fuchs80] Fuchs, H., Kedem, Z. et al. On visible surface generation by a priori data structures. *Proceedings of the 7th annual conference on Computer graphics and interactive techniques*, 1980, 124-133.

[Gobbetti08] Gobbetti, E., Kasik, D. et al. Technical strategies for massive model visualization. *Proceedings of the 2008 ACM symposium on Solid and physical modeling*, 2008, 405-415.

[Jackins80] Jackins, C., Tanimoto, S. Oct-trees and their use in representing three-dimensional objects. *Computer Graphics and Image Processing*, 14 (1980), 249-270.

[Jeschke02] Jeschke, S., Wimmer, M., Schuman, H. Layered environment-map impostors for arbitrary scenes. *Proceedings of Graphic Interface*, (2002), 1-8.

[Kaneko01] Kaneko, T., Takahei, T. et al. Detailed Shape Representation with Parallax Mapping. *Proceedings of the ICAT 2001*, 2001, 205-208.

[McMillan95a] McMillan, L., Bishop, G. Plenoptic Modeling: An Image-Based Rendering System. *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, 39-46.

[McMillan95b] McMillan, L., Bishop, G. Head-tracked stereoscopic display using image warping. *Proceedings of SPIE*, 2409 (1995), 21-30.

[McMillan97] McMillan, L. An image-based approach to three-dimensional computer graphics. *PhD thesis, University of North Carolina at Chapel Hill*, 1997.

[Oliveira06] Oliveira, J., Buxton, B. PNORMS: platonic derived normals for error bound compression. *VRST '06: Proceedings of the ACM symposium on Virtual reality software and technology*, 2006, 324-333.

[Oliveira08] Oliveira, J. Vertex Classification for non-uniform geometry reduction. *PhD thesis, Department of Computer Science, University College London*, 2008, 194.

[Purnomo04] Purnomo, B., Cohen, J. et al. Seamless texture atlases. *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry Processing*, 2004, 65-74.

[Sajadi09] Sajadi, B., Huang, Y. et al. A Novel Page-Based Data Structure for Interactive Walkthroughs. *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, 2009, 23-29.

[Schaufler98] Schaufler, G. Image-based representation by layered impostors. *Proceedings of the ACM symposium on Virtual reality software and technology*, 1998, 99-104.

[Tatarchuk06] Tatarchuk, N., Dynamic Parallax Occlusion Mapping with Approximate Soft Shadows. *Proceedings of ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, 2006, 63-69.

11. APPENDIX

Figure 19. Bunny model using a billboard generated with a FOV of 40 degrees.

Figure 20. The difference of Figure 19 to a correct render of the model.

Figure 21. Bunny model using a billboard generated with a FOV of 90 degrees.

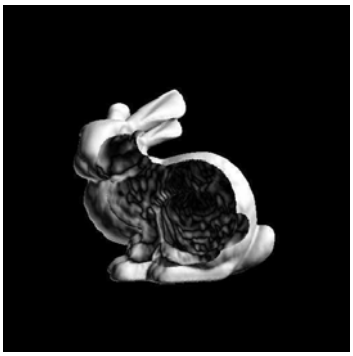

Figure 22. The difference of Figure 21 to a correct render of the model.

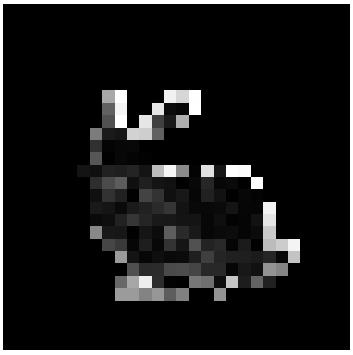

Figure 23. The Bunny model at a relative distance of 16, zoomed in. Note how the borders still have a large difference.

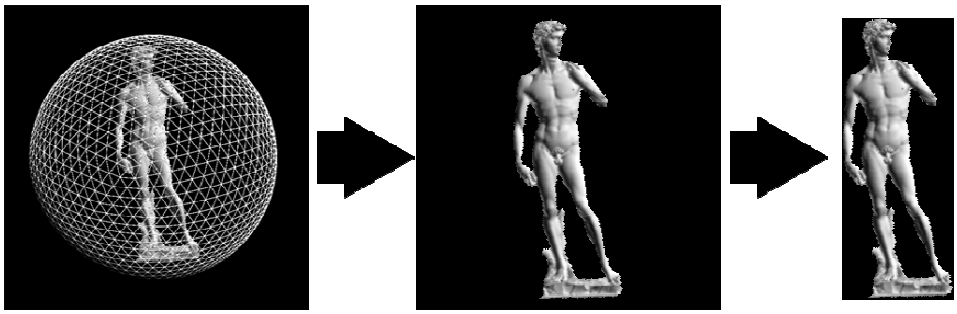

Figure 24. The generation process: the scene (with subdivided bounding sphere), a generated texture and a cropped texture.