OBJECTIVE QUALITY SELECTION FOR HYBRID LOD MODELS

Tom Jehaes, Wim Lamotte
Hasselt University, Expertise Centre for Digital Media
Interdisciplinary institute for BroadBand Technology (IBBT)
Wetenschapspark 2, 3590 Diepenbeek, Belgium

Nicolaas Tack
IMEC, Kapeldreef 75, 3001 Leuven, Belgium

Keywords: Progressive Geometry, Image-Based Rendering, Level-of-Detail, Objective Quality Selection.

Abstract: The problem of rendering large virtual 3D environments at interactive framerates has traditionally been solved by using polygonal Level-of-Detail (LoD) techniques, for which either a series of discrete models or one progressive model were determined during preprocessing. At runtime, several metrics such as distance, projection size and scene importance are used to scale the objects to such a resolution that a target framerate is maintained in order to provide a satisfactory user experience. In recent years however, image-based techniques have received a lot of interest from the research community because of their ability to represent complex models in a compact way, thereby also decreasing the time needed for rendering. One of the questions however that should be given some more consideration is when to switch from polygonal rendering to image-based rendering. In this paper we explore this topic further and provide a solution using an objective image quality metric which tries which we use to optimize render quality. We test the presented solution on both desktop and mobile systems.

1 INTRODUCTION AND RELATED WORK

In order to efficiently render large virtual 3D environments at interactive framerates, application designers have traditionally resorted to using polygonal Level-of-Detail (LoD) techniques. Usually, either a series of discrete models or one progressive model are determined during preprocessing. At runtime, a target framerate is maintained in order to provide a satisfactory user experience. This is achieved by employing several metrics such as distance, projection size and scene importance to scale the objects a suitable resolution. In recent years however, image-based techniques have received a lot of interest from the research community because of their ability to represent complex models in a compact way, thereby also decreasing the time needed for rendering.

An important question however, is when to switch from polygonal rendering to image-based rendering. Previously, a subjective quality metric was often used for determining the switching distance at runtime. In this paper we explore this topic further and provide a solution using an objective image quality metric which tries to optimize render quality.

Our following related work discussion is divided into three categories: firstly, polygonal LoD techniques which have been used traditionally, secondly, image-based rendering techniques which have been researched in previous years as an alternative rendering method for distant objects and finally, mixed systems, benefiting from the strengths of each system: polygonal rendering for objects close by in high quality, faster image-based rendering for objects further away providing sufficient detail.

1.1 Polygonal LoD

The first techniques for choosing the Level of Detail (LoD) of polygonal objects have used static heuristics based on the screen size of an average face of the object (Funkhouser et al., 1992) and/or the distance of the object to the user (Blake, 1987; Rossignac and Borrel, 1993). These simple heuristics improve the frame rate in many cases, but cannot guarantee a regulated or bounded execution time. To guarantee a

Jehaes T., Lamotte W. and Tack N. (2008).
OBJECTIVE QUALITY SELECTION FOR HYBRID LOD MODELS.
In *Proceedings of the Third International Conference on Computer Graphics Theory and Applications*, pages 241-248
DOI: 10.5220/0001099102410248

bounded frame rate, (Funkhouser and Séquin, 1993) have presented a predictive technique that uses an estimate of the execution time for the correct choice of the LoD to use. The core of this work is a multiple choice knapsack problem that maximizes the visual quality for a given maximum execution time. The greedy solution of Funkhouser and Séquin however only guarantees to be half as good as the optimal solution. Therefore, (Gobbetti and Bouvier, 2000) have proposed to use convex optimisation (interior point algorithm) with a guaranteed specified accuracy.

1.2 Image-based Rendering

As a solution for faster rendering of large geometrical models, Image-Based Rendering (IBR) has received many followers over the last years. The render times of image-based models are fairly constant and rely on the resolution of the reference images instead of polygon count. This means that for high polygon count models, image-based rendering can be a much faster render solution without introducing too much visual degradation. For closeup viewing of detailed models however, conventional polygon rendering is often still preferred. McMillan first proposed the fundamental 3D warping equation together with an occlusion compatible warping order to efficiently render new views based on a series of reference depth images (McMillan and Bishop, 1995). Extensions to this have been presented in (Shade et al., 1998), (Oliveira and Bishop, 1999) and (Chang et al., 1999) whereby separate models are represented by a cluster of reference images.

More recently, (Oliveira et al., 2000) shows that a factorisation of the 3D warping equation, called relief-texture mapping (RTM), enables the use of fast graphics hardware for part of the calculation to speed up the warping. Warping is done in two steps: a pre-warp followed by a simple texture mapping. This delivers a significant speedup because the pre-warp is implemented using a fast two-pass reconstruction algorithm and the texture mapping can be done on fast graphics hardware. Layered Relief Textures presented in (Parilov and Stuerzlinger, 2002) combine the idea of storing multiple samples per pixel with fast hardware assisted warping of RTM. Finally, (Fujita and Kanai, 2002) incorporate dynamic shading into the RTM approach by using per-pixel shading hardware.

1.3 Qos for Mixed Systems

Conventional texture mapping can be seen as a very simple form of image-based rendering that does not take depth information into account. Based on this, (Maciel and Shirley, 1995) focus on maintaining a high framerate by replacing clusters of objects with simple texture mapped primitives. Similarly, (Shade et al., 1996) use a BSP tree scene representation for which they cache images of nodes that were rendered in previous frames. Taking frame-to-frame coherence into account they reused these cached images for rendering subsequent frames, thereby gaining a significant rendering speedup.

Later on, making use of depth image representations, (Rafferty et al., 1998) extend a portal culling renderer in which they determine the view through a portal by warping a precalculated Layered Depth Image (LDI) that captures the view through that portal. This approach is generalized for massive model rendering in (Aliaga and Lastra, 1999). A grid of viewpoints is constructed for which the far geometry is determined after which LDIs are created that represent this far geometry. The renderer can then first render far geometry from LDIs followed by polygon based rendering of the near geometry. The MMR system (Aliaga et al., 1999) replaces the LDIs in the previous approach by Textured Depth Meshes (TDMs) to make optimal use of current graphics hardware.

Two fundamental drawbacks of the former hybrid techniques are the often very long preprocessing times and huge storage requirements. Therefore, a system that creates image-based representations on demand without the need for additional storage was presented in (Hidalgo and Hubbold, 2002). They employ a dual renderer setup in which the hybrid renderer (HR) can request reference depth images representing the current far geometry from a reference image generator (RIG) that runs in parallel. While the RIG is working on the requested data, the HR can use warping on the previous reference depth image to render its frames. Prediction is used to request optimal reference images.

A specific optimization algorithm for a terrain flyover application was proposed in (Zach et al., 2002). Interesting in this approach is that they not only use discrete and continuous Level Of Detail for polygonal rendering but also point-based rendering for the trees on the terrain.

A major drawback of these approaches is that they are not well suited for handling dynamic scenes for which the contents is not known beforehand. This is the case for instance for the increasingly popular MMORPGs. We have therefore selected to use the hybrid rendering technique presented in (Jehaes et al., 2004), which was extended to mobile devices in (Jehaes et al., 2005). This technique uses a combination of progressive geometry and relief texture mapped objects for representing separate objects in the scene. The authors have however only used a subjective quality metric for selecting at run-time which render method should be used for each object instance. In this paper we therefore present a objective metric based on image quality.

In the next section we give some more details about hybrid object representation. Following that, we present our solution for an objective LoD selection metric which is meant to maintain a high level of image quality. Finally, we present our test setup and results on both desktop and mobile systems and end with conclusions and some pointers for future work.

2 HYBRID MODELS

The hybrid geometric/image-based rendering scheme, which was presented in (Jehaes et al., 2004), makes use of geometry simplification based on (Hoppe, 1996), and the relief-texture mapping technique which was introduced by Oliveira et al. (Oliveira et al., 2000). This last technique is used because it makes efficient use of the texturing capabilities of current graphics cards to speed up rendering of the image-based models. It also integrates easily into the standard geometry rendering pipeline. An example of relief texture mapped model representations can be seen in figure 7(b). The relief textures capture the appearance of the model as seen from each side of the bounding box. During rendering, each visible relief texture is pre-warped into a texture which is subsequently mapped onto the corresponding bounding box quad, resulting in a correct view of the model. Because the pre-warp equation can be efficiently calculated, the total render time will be small and mostly depends on the resolution of the relief textures.

The render scheme also supports animated objects such as avatars. Often, the animations for objects further away do not add much to the visual quality, so they can be suppressed. Furthermore, these distant animated objects can then be replaced by RTM objects, using the same LoD rendering scheme.

3 OBJECTIVE HYBRID LOD SELECTION

Triangular rendering is nowadays one of the most efficient methods for real-time rendering of 3D content. However, in some cases, triangles can become so small that the processing of their three vertices and the rasterization are done for shading only one pixel. In this case, other rendering algorithms, such as IBR, become more efficient. Therefore, Hybrid LoD selection also selects the most efficient rendering method next to the correct LoD for the rendered object. Subsections 3.1 and 3.2 respectively discuss LoD selection for the polygonal and IBR objects. Subsection 3.3 then discusses the selection of the most efficient rendering algorithm.

3.1 Polygonal LoD Selection

The LoD selection mechanism for polygonal objects uses a Pareto optimisation (Tack et al., 2006) for the selection of the optimal LoDs in terms of quality and render time. Central in this approach is the use of the Pareto plot, which is a collection of Pareto optimal points for which it is impossible to improve the visual quality without increasing the cost. The Pareto plots are measured - for each separate polygonal object in the scene - in off-line preprocessing steps and encoded with the objects. The online steps of the optimization use a gradient descent algorithm to combine the Pareto plots of the visible objects in the scene with a minimum of online parameters (e.g. object distance) to find the optimal trade-off between quality and cost.

3.2 LoD Selection for Relief Texture Mapping

Advanced LoD selection mechanisms use a model for estimating the performance and the quality (Gobbetti and Bouvier, 2000). However, these systems control the render time of polygonal rendering, which is a very regular algorithm and hence easy to model. Relief Texture Mapping has an irregular flow and its performance model is much more complicated, i.e. it is necessary to track a lot of online parameters to obtain a good estimate of the render time. Consequently, using this performance model at run-time has a negative effect on frame render times. We have therefore chosen a different approach for the RTM control algorithm which requires much less run-time processing: the selection of the switching distance and resolution of the relief texture mapped objects by using a target quality of the rendered image as the control parameter.

In order to specify a target quality, we have used two objective quality metrics for measuring the visual quality of an RTM model: the Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) (Wang et al., 2004) metrics. These metrics compare a reference image with the image that contains the rendered RTM object. The reference image is obtained by rendering the polygonal object at

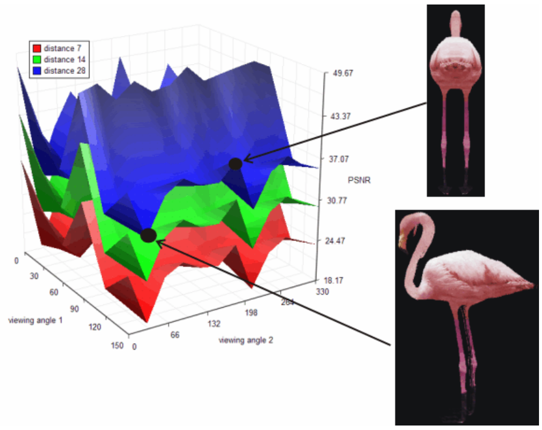


Figure 1: The PSNR as function of the viewing angles φ and θ (30° steps) for three different distances to the object for a single resolution RTM object.

full quality. Next, the quality is sampled for different viewing angles around the object. The viewpoint is first located at a fixed distance and both the viewing angles φ and θ are set to 0, the quality is derived and the viewing angle φ is increased by a preset amount. This is repeated until the viewing angle φ is 360° , then φ is reset to 0 and θ is increased. The procedure is stopped at θ equal to 180° and φ equal to 360°. This procedure is then repeated for different distances.

Figure 1 illustrates the result for an RTM object (flamingo) with a resolution of $512x512$. The PSNR is plotted as a function of the viewing angles φ and θ and distance between the observer and the object. From figure 1, one can derive the parameters onto which the quality depends:

- The quality increases with a larger distance between the viewpoint (observer) and the object.
- The quality is the highest for viewpoints near to the viewpoints for which the Relief Texture Maps were derived: φ equal to 0° , 90° , 180° and 270° and θ equal to 0° , 90° and 180°. For in-between viewpoints, interpolations between RTMs are needed and the quality decreases because of occlusions in the used relief texture map.

An additional parameter, which is not shown in figure 1 but that also influences the quality is the resolution of the RTMs. This is illustrated in Figure 2, which shows the PSNR as a function of the distance to the viewpoint for an RTM object at a single viewing angle and different resolutions.

Figure 1 and Figure 2 can be summarised in a table, which stores the resolution of the RTM object as a function of quality, distance and viewing angle (see Table 1 for a single viewing angle). This table is derived in preprocessing and is used at run-time for the selection of the correct RTM resolution.

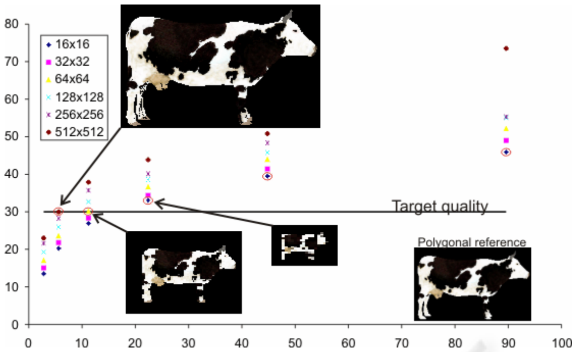


Figure 2: The quality (PSNR) as a function of distance to the viewpoint and resolution. For small distances, a high resolution is needed for a acceptable target quality of 30 dB; for large distances, this needed resolution decreases.

Table 1: The selection of the RTM resolution as a function of requested quality and distance.

dist/Q	20 dB	25 dB	30 dB	35 dB
5 m	16x16	128x128	512x512	-
10 m	16x16	16x16	64x64	256x256
20 m	16x16	16x16	16x16	64x64
30 m	...	...	...	...

3.3 Hybrid LoD Selection

To be able to select the most efficient rendering algorithm, a comparison between polygonal rendering and RTM is needed. Figure 3 therefore shows the render time as a function of the quality (PSNR), object distance and the used rendering technique. The quality is varied by changing the resolution of the relief textures for RTM and the number of triangles for polygonal rendering. The render time was measured on a Pentium M processor with a frequency of $1.86GHz$ and an Intel 915GM graphics accelerator. The time reported in Figure 3 for RTM is the total time needed for a complete warping operation, of which the pre-warping takes 90% (Oliveira et al., 2000). However, the pre-warping is only done when the relative position of the camera to the object changes and the real impact of RTM on the total render time will therefore depend on the user navigation and object animations.

Figure 3 shows an exponential relationship between the execution time and quality for RTM, while the relationship is linear for polygonal rendering. From Figure 3, it is clear that high resolution relief textures (points with high execution time) must be used for content which is close to the observer.E.g., for the distance of 2.8m (RTM:2.8), the technique gives a quality of 23 dB for an execution time of 23 ms. Even if the pre-warping is not executed for ev-

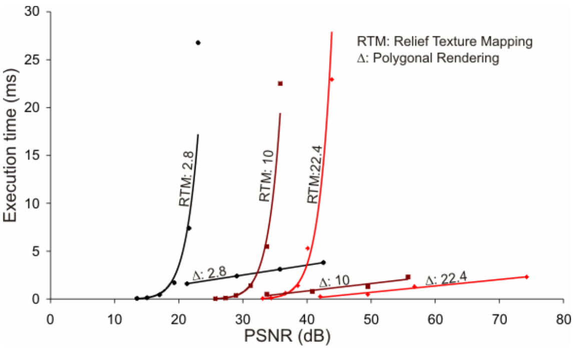


Figure 3: Polygonal rendering versus RTM: execution time as a function of quality and distance (2.8m, 10m and 22.4m).

ery frame, its impact is still too high to use RTM for close content. However, for distant content (e.g. RTM:22.4), low resolution relief textures give an acceptable quality ($> 30dB$) for a low execution time. Even if the pre-warping is executed every frame, the render time is still comparable with the polygonal rendering time. If the pre-warping is not needed, then the rendering is limited to rendering of the texture mapped bounding box and the rendering time is negligible.

Our hybrid LoD selection first uses table 1 to select the resolution of the RTM object to meet the requested quality. If this resolution is higher than a hard limit (e.g. 128x128), polygonal rendering is used for that object. The RTM objects are then rendered and the render time is measured. This render time is then subtracted from the render time budget and the Pareto optimisation for polygonal rendering is executed for maximising the quality for the polygonal objects for the computed time budget.

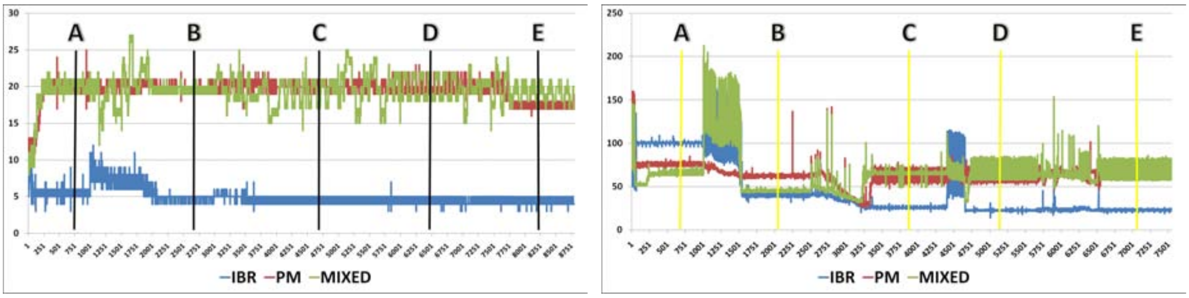
In order to optimize user experience and interactivity we set a target framerate and make adjustments to the target quality based on the current framerate. When the current framerate is either higher or lower than the target framerate, we increase or decrease the target quality. Since our LoD selection is directly linked to the target image quality, switching distances and resolutions are automatically adjusted resulting in a higher or lower framerate.

4 RESULTS

In order to determine the usefulness of our image-based quality metric in an actual application, we set up a test for both desktop and mobile systems. Concentrating first on the desktop scenario, we set out to render a scene consisting of 400 object instances which were randomly chosen from 8 different base models. Each model representation consisted of a progressive mesh representation, with a corresponding Pareto plot, and a RTM representation for which the resolutions and switching distances were determined by using the image-based metric presented above. For reasons of comparison, we defined a camera path through the scene which would be used for each of the 4 render scenarios: full resolution geometry, image-based only (IBR), progressive mesh only (PM) and mixed representations. As can be seen in figure 5(a), using full resolution geometry rendering would result in a maximal triangle count of more than 1600000 for which we got a framerate of about 4 fps. Note that the overly large frametimes for the full geometry walkthrough were left out of the graphs for reasons of readability. Fortunately, we can increase the framerate considerably by using one of the other representation types as shown in figure 4(a). The target frametime for both the PM and mixed scenarios was set to 20ms, while the IBR scenario would render at maximum speed. We could have also regulated the IBR framerate by increasing the RTM resolution, but we wanted to show the speedup benefits of using the IBR approach.

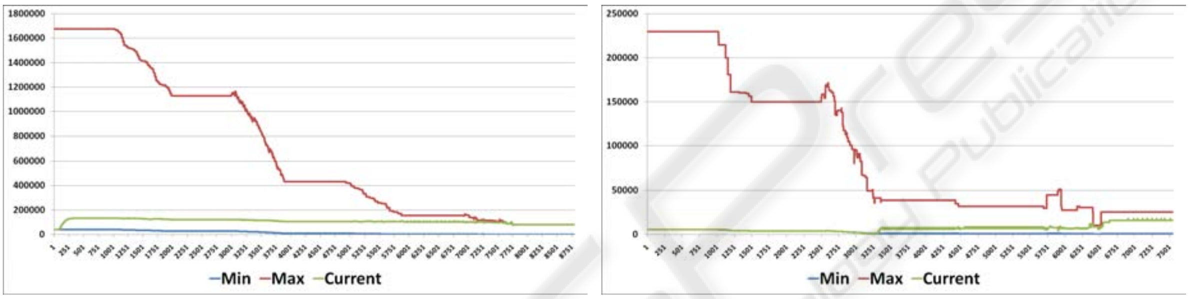
During our experiments we tested with both the PSNR and SSIM image quality metrics and found that the SSIM metric gave results which were more consistent compared to our own subjective evaluation of the image quality. After performing some tests comparing the results of both metric, we clearly noticed that the some PSNR results were inconsistent with what was to be expected. The SSIM metric exhibited a much more consistent result, so we therefore switched over to this metric. By combining the frametime graphs with the results of image quality measurements shown in table 2, which were taken at the corresponding points indicated on figure 4(a), we can conclude that our mixed representation scenario, using the image-based metric results in higher image quality at comparable frametimes which was what we set out to achieve. The difference in image quality between the four scenarios is indicated in figure 6. The use of the mixed scenario clearly allows for much higher resolution meshes to be used for objects close to the viewer. As our camera moves through the scene, less objects are in the view and the PM and mixed scenarios converge to the same image quality, which can also be concluded from figures 5(a) and 5(c).

For our second test we set up a scene consisting of 60 object instances, using the same base models as in the desktop test. The application was deployed on Dell X51v PDAs, which incorporate the Intel 2700G GPU. From figure 5(b) it can be seen that the maximal triangle count at application start is about 230000 for

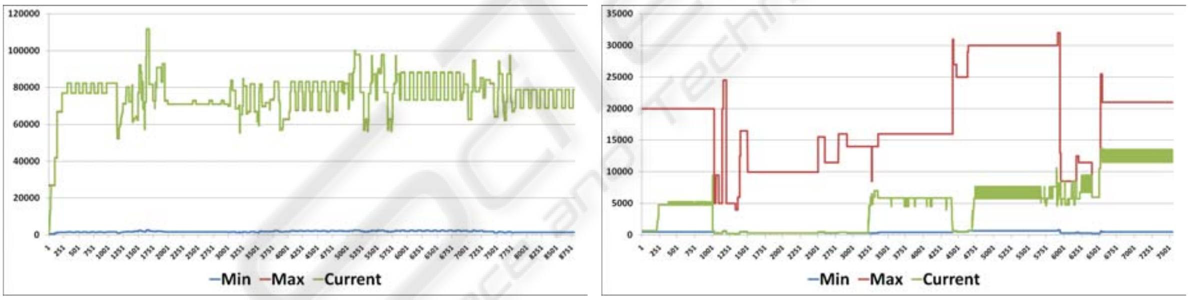


(a) Desktop (b) PDA

Figure 4: Frametimes (ms) measured during a walkthrough (Target frametime was set to 20ms for desktop and 66ms for PDA). Image quality was measured at each of the specified points (A-E).



(a) PM only (Desktop) (b) PM only (PDA)



(c) Mixed (Desktop) (d) Mixed (PDA)

Figure 5: Number of PM triangles rendered for desktop and PDA during the walkthrough. Maximum and minimum represent lowest and highest resolutions for all objects combined. Current denotes the selected object quality at runtime.

(a) (b) (c) (d)

Figure 6: Desktop screenshots: (a) Full geometry (b) IBR (c) PM only (d) Mixed PM and IBR.

Table 2: SSIM quality measurements for desktop.

SSIM	IBR	PM	MIXED
A	0.8756	0.9747	0.9804
B	0.9318	0.9865	0.9912
C	0.9278	0.9882	0.9921
D	0.9408	0.9935	0.9992
E	0.9266	1.0000	1.0000

Table 3: SSIM quality measurements for PDA.

SSIM	IBR	PM	MIXED
A	0.9686	0.9775	0.9812
B	0.9696	0.9852	0.9855
C	0.9694	0.9884	0.9913
D	0.9791	0.9975	0.9977
E	0.9683	0.9996	0.9996

which we got a framerate of 0.41 fps (not shown) using full geometry rendering. Again, by applying one of the other representation types, we can substantially increase the framerate. Note that the target frametime for this mobile application was set to 66ms. When we compare the resulting frametimes graph of figure 4(b) to figure 4(a), we can see that on the PDA, it is much more difficult to stay close to the target frametime. This is a result of the inferior processing capabilities of the mobile device. Furthermore, for both the IBR and mixed scenarios, we see a sudden increase in frametime at two periods during the walkthrough. These two periods consist of camera movement, thereby resulting in the need for updates to the RTM objects. When looking at figures 4(b) and 5(d) we can see however that the mixed scenario is much less affected during the second movement period because at this moment much less objects are visible and rendered using the RTM technique than during the first period.

Another difference can be seen when comparing the graphs in figures 5(c) and 5(d). On the PDA we are never able to render the maximal object quality, while on the desktop, objects near to the viewer are always rendered at maximum resolution. Even so, from table 3 and the screenshots presented in figure 7 it is obvious that by using the mixed representation we can greatly increase image quality while rendering at comparable framerates. Furthermore, notice that when using the PM only approach, the application is unable to bring the frametime down to the target frametime, even by using the lowest resolution versions for all object instances while the mixed approach can render at the target frametime using higher resolution versions for the near objects.

5 CONCLUSIONS

We presented our solution for determining when to switch from standard geometry representations to image-based representations during rendering of a complicated scene. During preprocessing, an image quality metric is used for determining at which distance the image-based representation provides sufficient image quality compared to the geometrical version. These measurements are performed for a subset of viewpoints on the viewing sphere and stored for later use. We successfully tested the usefulness of this metric during a test that was performed on both a desktop and a mobile system.

With regard to future work, we will be looking into other image metrics for determining the quality difference between the images rendering using the different representation types because we have noticed that, even by using the SSIM metric, the image metric can sometimes differ from the quality perceived by users themselves.

ACKNOWLEDGEMENTS

This work has been done under IBBT project A4MC3.

REFERENCES

Aliaga, D. G., Cohen, J., Wilson, A., Baker, E., Zhang, H., Erikson, C., III, K. E. H., Hudson, T., Sturzlinger, W., Bastos, R., Whitton, M. C., Jr., F. P. B., and Manocha, D. (1999). MMR: an interactive massive model rendering system using geometric and image-based acceleration. In *Symposium on Interactive 3D Graphics*, pages 199–206. ACM.

Aliaga, D. G. and Lastra, A. (1999). Automatic image placement to provide a guaranteed frame rate. In Rockwood, A., editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 307–316, Los Angeles. Addison Wesley Longman.

Blake, E. (1987). A metric for computing adaptive detail in animated scenes using object-oriented programming. In *Eurographics Conference Proceedings*, pages 295–307.

Chang, C.-F., Bishop, G., and Lastra, A. (1999). Ldi tree: A hierarchical representation for image-based rendering. In Rockwood, A., editor, *Siggraph 1999, Computer Graphics Proceedings*, pages 291–298, Los Angeles. Addison Wesley Longman.

Fujita, M. and Kanai, T. (2002). Hardware-assisted relief texture mapping. *Eurographics 2002 short paper presentation*, pages 257–262.

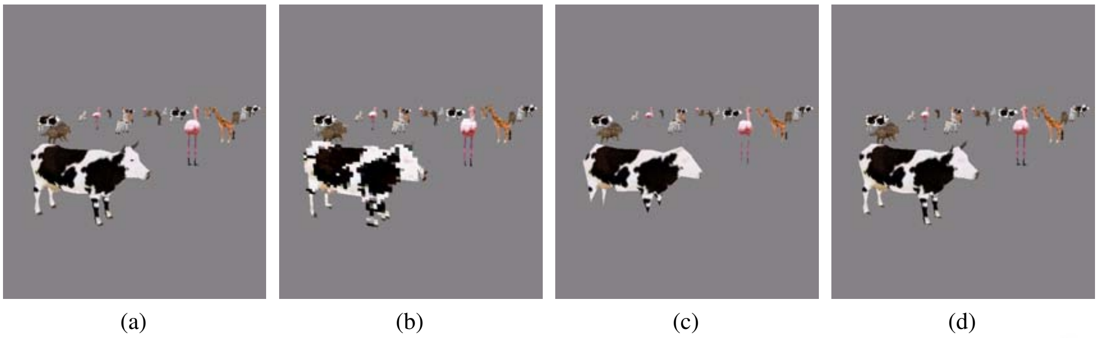

(a) (b) (c) (d)

Figure 7: PDA screenshots: (a) Full geometry (b) IBR (c) PM only (d) Mixed PM and IBR.

Funkhouser, T. A. and Séquin, C. H. (1993). Adaptive display algorithm for interactive frame rates during visualization of complex virtual environments. *Computer Graphics*, 27(Annual Conference Series):247–254.

Funkhouser, T. A., Squin, C. H., and Teller, S. J. (1992). Management of large amounts of data in interactive building walkthroughs. In *1992 Symposium on Interactive 3D Graphics*, pages 11–20.

Gobbetti, E. and Bouvier, E. (2000). Time-critical multiresolution rendering of large complex models. *Journal of Computer-Aided Design*, 32(13):785–803.

Hidalgo, E. and Hubbold, R. (2002). Hybrid geometric - image based rendering. *Computer Graphics Forum*, 21(3).

Hoppe, H. (1996). Progressive meshes. *Computer Graphics (ACM SIGGRAPH '96 Proceedings)*, 30(Annual Conference Series):99–108.

Jehaes, T., Quax, P., and Lamotte, W. (2005). Adapting a large scale networked virtual environment for display on a pda. In *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 217–220, New York, NY, USA. ACM Press.

Jehaes, T., Quax, P., Monsieurs, P., and Lamotte, W. (2004). Hybrid representations to improve both streaming and rendering of dynamic networked virtual environments. In *Proceedings of the 2004 International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI2004)*.

Maciel, P. W. C. and Shirley, P. (1995). Visual navigation of large environments using textured clusters. In *Symposium on Interactive 3D Graphics*, pages 95–102, 211.

McMillan, L. and Bishop, G. (1995). Plenoptic modeling: An image-based rendering system. *SIGGRAPH 95 Conference Proceedings*, pages 39–46.

Oliveira, M. M. and Bishop, G. (1999). Image-based objects. In *Symposium on Interactive 3D Graphics*, pages 191–198.

Oliveira, M. M., Bishop, G., and McAllister, D. (2000). Relief texture mapping. In Akeley, K., editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 359–368. ACM Press / ACM SIGGRAPH / Addison Wesley Longman.

Parilov, S. and Stuerzlinger, W. (2002). Layered relief textures. *Journal of WSCG*, 10(2):357–364. ISSN 1213-6972.

Rafferty, M. M., Aliaga, D. G., Popescu, V., and Lastra, A. A. (1998). Images for accelerating architectural walkthroughs. *IEEE Computer Graphics and Applications*, 18(6):38–45.

Rossignac, J. and Borrel, P. (1993). Multi-resolution 3d approximations for rendering complex scenes. In *Conference on Geometric Modeling in Computer Graphics*, pages 455–465.

Shade, J., Lischinski, D., Salesin, D. H., DeRose, T., and Snyder, J. (1996). Hierarchical image caching for accelerated walkthroughs of complex environments. *Computer Graphics*, 30(Annual Conference Series):75–82.

Shade, J. W., Gortler, S. J., He, L.-W., and Szeliski, R. (1998). Layered depth images. *Computer Graphics*, 32(Annual Conference Series):231–242.

Tack, N., Lafruit, G., Catthoor, F., and Lauwereins, R. (2006). Platform independent optimization of multiresolution 3d content for enabling universal media access. *The Visual Computer*, 22(8):577–590.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Zach, C., Mantler, S., and Karner, K. (2002). Time-critical rendering of discrete and continuous levels of detail. In *VRST '02: Proceedings of the ACM symposium on Virtual reality software and technology*, pages 1–8, New York, NY, USA. ACM Press.