Enhanced Billboards for Model Simplification

Phongvarin Vichitvejpaisal
Faculty of Computer Engineering
Chulalongkorn University
Pathumwan
Bangkok 10330, Thailand
matt_varin@yahoo.com

Pizzanu Kanongchaiyos
Faculty of Computer Engineering
Chulalongkorn University
Pathumwan
Bangkok 10330, Thailand
pizzanu@cp.eng.chula.ac.th

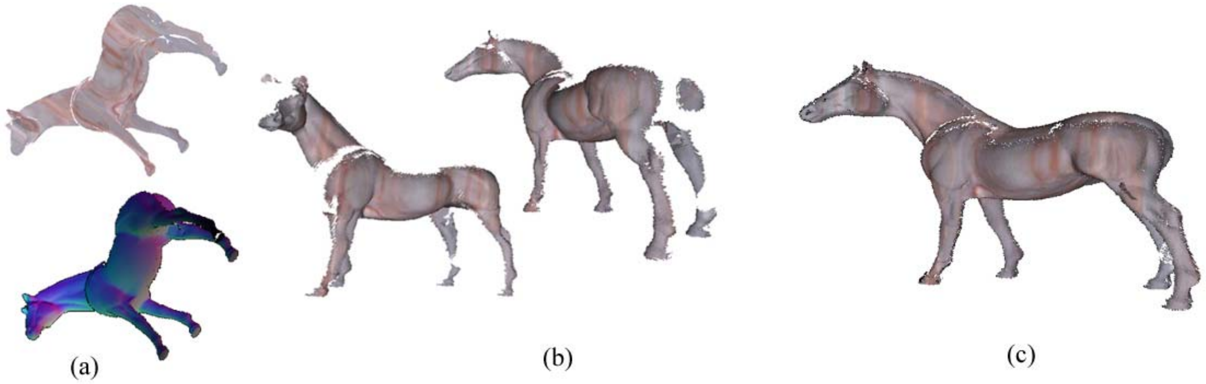

Figure 1: (a) An enhanced billboard consisting of a color map, a transparency map, a depth map and a normal map encoded into two images. (b) A horse model rendered with an enhanced billboard shown in (a), therefore representing only some portion of the model. (c) Complete horse model rendered with all enhanced billboards.

ABSTRACT

A set of billboards can represent 3D models for extreme simplification in real-time rendering. Unlike conventional polygon method, billboard-based technique has the rendering time of the model proportional to its contribution to the image. Thus, it has an automatically built-in Level of Detail. However, previous techniques still have limited viewing angle and do not accurately represent the model. We present an adaptive rendering method using enhanced billboards. First, each enhanced billboard, representing portion of the model, is defined to have four maps: depth map, normal map, color map and transparency map. The model is then projected onto a number of viewing planes in different viewing directions. Consequently, these enhanced billboards are rendered based on ray-height-field intersection algorithm implemented on GPU. This representation can maintain the geometry and the silhouette of the model with no limit in viewing direction. Moreover, real-time rendering is supported.

Keywords

image-based rendering, displacement, billboards, real-time rendering, simplification

Conference proceedings ISBN 80-86943-03-8
WSCG'2006, January 30-February 3, 2006
Plzen, Czech Republic.

1. INTRODUCTION

One of the important problems in computer graphics is how to handle the overwhelming complexity of the screen. A detailed scene can give a more attractive and realistic look but lower the performance. A typical scene nowadays may contain up to millions of polygons. In addition, programs such as games are performance-critical applications. Many techniques have been invented to manage the tradeoff between speed and quality of the picture. These techniques deal with new model representations as well as new rendering algorithms. The billboard is one of the various image-based model simplification methods with advantages in its simplicity and rendering speed. Typically, a billboard represents a 3D model with an image. The image is placed at the position of the model and rendered as a texture mapped polygon quad. The image-based property of the billboard enables its rendering time to scale linearly with its screen contribution. Unfortunately, the billboard has a lot of limitations. For example, it must be viewed with narrow viewing directions, otherwise the 2D structure of the image will be noticeable.

This paper shows how the billboard can be enhanced to produce high quality images without such limitations. We propose a new representation for the model, which is a series of images. Each image consists of a color map, a transparency map, a normal map and a depth map, or displacement map, called *enhanced billboards*. The enhanced billboard image is projected from different direction of the model. Therefore, we can represent all the information of the 3D model. The enhanced billboards are shown in Figure 1. Previous simplification techniques typically store only some information of the source model, and, therefore, much of the information is lost. We use a new ray-height-field intersection algorithm to render the enhanced billboards. Our method emphasizes on the proportion of processing time compared to the contribution of the outcome image. Thus, the enhanced billboard is effective in rendering rich detail models in distant scenes with unnoticeable artifacts. It maintains the silhouettes, parallax effect and smooth surface without any cracks.

2. RELATED WORK

Many image-based approaches are used to simplify the model. The Impostor Technique replaces distant geometry by projecting an object or a portion of the screen onto a plane and rendering it as an image. Bump Mapping [Bli78; Per97] uses the normal map to perturb the normal of the model, thus increasing the detail of the model. However, it cannot render the silhouette and shadow. Displacement Mapping [Hir04; Pha96; Sch99; Smi00], in contrast, simulates the actual geometry of the surface and can produce the silhouette and shadow effect. Nevertheless, displacement mapping increases the amount of micro-polygons to capture the detail of the model, which is not appropriate for real-time application.

The layered depth image [Sha98] is developed to handle the parallax effect. It is an image that contains depth information. The drawback of the layered depth image is that it can only be viewed at a narrow range.

Compared to our enhanced billboard, which uses a series of image from different views of the model, our method does not suffer from limited viewing directions. Meyer et al. [Mey98; Mey01; Phi04; Jak00] proposed a method to render a forest scene. They used a group of billboards, each being the sliced image of some group of trees. This method faced a problem with the viewing region, which makes it more suitable for a bird eye's view of the forest. The Billboard Cloud Technique [Xav03] also uses a number of billboards, each representing some portion of the model. The billboard cloud is rendered as a conventional texture map polygon. Because it uses a plane to approximate the nearby geometry, the billboard cloud results in cracks in the image.

Another simplification technique is Mesh Decimations [Bax02; Gar97; Hop04; Jes02], which reduces the number of polygons rendered on the screen. The number of polygons is decreased by using edge-collapse or vertex-removal algorithm. Thus, the quality of the model is lower than the original, and the results may be rough models with discontinuity. Some of the key visual details such as the nose of the face can be lost. These techniques depend on the property of the input model, such as topology or connectivity. Thus, a different algorithm has to be used for a different type of model. Another drawback is that this method requires that the model be stored in various resolutions. This will increase the storage cost and introduce a pop-up artifact when changing to different resolutions of the model. The Progressive Mesh [Hop96; Hop97] Technique is invented to solve the pop-up problem. It has an auxiliary data structure to smoothly add and collapse the vertex of the object. Thus, it can continuously increase or decrease the quality of the model without any noticeable effects.

Wang et al. [Wan03; Xi04] proposed a five dimensional representation GDM (general displacement map) to quickly render the displacement model. This technique stores the pre-compute distances from each displaced point to a reference surface. It requires a lot of storage and needs to be compressed.

Relief Texture [Fab05; Oli00] Mapping bears similarities to our work. The relief texture is mapped onto the surface to capture the detail of the model. It has a normal map and a depth map like ours. The relief texture map can capture the parallax effect correctly, but the silhouette effect cannot be produced. In contrast, our method is designed to render the complete model with silhouette information, not just the detailed surfaces.

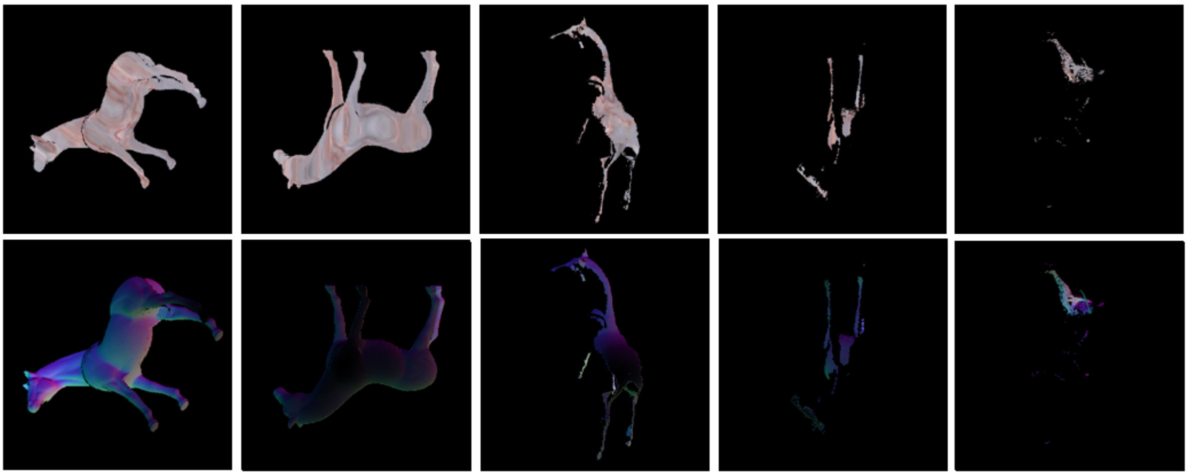

Figure 2: The first five enhanced billboards of the horse model encoded as color-alpha images and normal-depth images.

3. ENHANCED BILLBOARDS

We introduce *enhanced billboards* as an alternative for representing an arbitrary polygon model. An enhanced billboard consists of a color map, a transparency map, a normal map and a depth map. These maps are stored in two images. One image stores color map in the RGB channel and the transparency map in the alpha channel. The other image stores the normal map in the RGB channel and the depth map in the alpha channel since an image is a 2D data structure. The depth map is used to store the third dimension of the model encoded as depth information. The normal map is used in the shading algorithm, similar to a bump map.

The model is projected onto a set of planes. Each plane captures some polygons in the model. The projected polygon stored as an enhanced billboard includes all the color, normal and depth information of that polygon. The direction of the projected plane is chosen to minimize the number of planes needed to capture all the polygons in the model. With a smaller number of enhanced billboards, less storage and time will be required for rendering. Thus, we use the greedy algorithm that relies on the normal vector of the polygon to choose the projection plane. The enhanced billboards are rendered with our new ray-height-field algorithm by the help of GPU.

The enhanced billboards can store all the information of the model, so no information is lost nor approximated, thus enabling enhanced billboards to be viewed in any direction with the correct silhouette. The amount of time used to render the enhanced billboards scales linearly to the number of pixels they cover. Thus, enhanced billboards are suitable for representing distant objects while preserving the geometric feature of the model.

3.1 Building Enhanced Billboards

This section shows how the polygon model is represented as enhanced billboards (see Fig 2.). An enhanced billboard can only represent polygons that are facing it. In addition, an enhanced billboard cannot store the polygons that occlude one another in the same image. Thus, the polygons that are occluded by some other polygons have to be stored in another enhanced billboard. We have to choose a robust way to pick projected directions that will result in small number of enhanced billboards used, and the chosen technique is based on using the normal vectors of the polygons.

First, we build an array of direction buffers. Each element of the array represents a direction in space. We use an array size of 36x18 to represent all the directions around the object. Each array element is used to accumulate the number of polygons that is facing the direction of the array element. Then, we choose the direction that has the most number of polygons. Although our method may not guarantee the smallest number of billboards, our greedy algorithm is easy to implement and still results in a small number of billboards.

We then have the direction for projecting the model. However, we cannot project the model yet because the polygons may be overlapping or the polygon may be far apart from each other. Our enhanced billboards will not store overlapping polygons in the same image. Moreover, enhanced billboards must have the same limited depth value. This prevents the varying depth of each billboard and results in inconsistency in rendering time. Therefore, we cull the rear-facing polygons and the polygons that are farther from the projecting plane than the limited depth value. Then, we cull the occluded polygons by using the occlusion query instruction in the graphics library. Only the polygons that are not at all occluded are chosen.

The projected polygons may cover varying areas of the plane, which may result in a variety of billboard sizes. Therefore, we need to limit the size of enhanced billboards as well. In the implementation, we use only 2 image sizes, 512x512 and 256x256 pixels. We have a scale factor to scale the coordinates of the projected polygons to fit in the billboards. The scale factor scales the diagonals of the model's bounding box to a value of 512. Thus, it guarantees that all the polygons in the model can fit in the 512x512 pixel billboard. We also store the bounding box of each remaining polygon. The bounding box data is bound within an enhanced billboard and will be used in rendering time.

```
Build Enhanced billboards
Input:  polygon model
Output:  images representing enhanced billboards
    Compute bounding box of the model
    Set model-to-texture scale factor
    Set of faces left $F$ = input model
    while  $F > 0$
        Choose direction with highest faces count
        Cull back faces
        Cull exceed depth value faces
        Choose Texture resolution of output images
        Cull occluded faces
        Orthogonally project remaining faces
        Save images as enhanced billboards
```

Figure 3: Pseudo-code of algorithm for construction of enhanced billboards

The remaining polygons can now be projected onto the plane of enhanced billboards using orthogonal projection. The free space on the billboard not covered by polygons is rendered as transparent pixels. The projected polygons will be discarded from the source model and the whole process is repeated until there is no polygon left. This results in a series of enhanced billboards representing some portions of the model from different viewing directions. By using all the enhanced billboards, we can reconstruct the complete model. Figure 3 shows the pseudo-code of the algorithm for the construction of enhanced billboards.

3.2 Rendering Enhanced Billboards

Because we treat an enhanced billboard as a height map, we can use a ray-height field intersection algorithm to render our enhanced billboards. The previous algorithm [Fab05] utilizes the programmability of the graphics hardware to do per-pixel ray-height field intersection. The height map image is rendered as a polygon quad. This method casts a ray from the eye to the texture coordinate in the height map polygon quad. The ray travels down the image and intersects at the first-hit position in the height map. The coordinate of the hit position is used to retrieve the color and normal information used in shading.

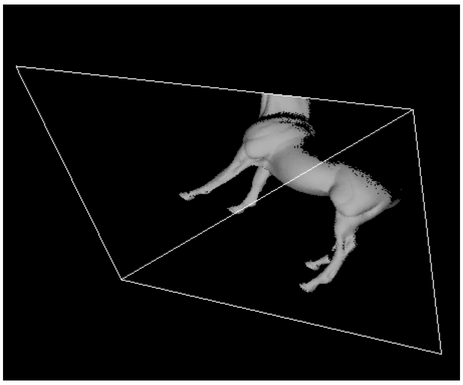

Figure 4: Rendering an enhanced billboard using only one plane. Some parts of the geometry are missing.

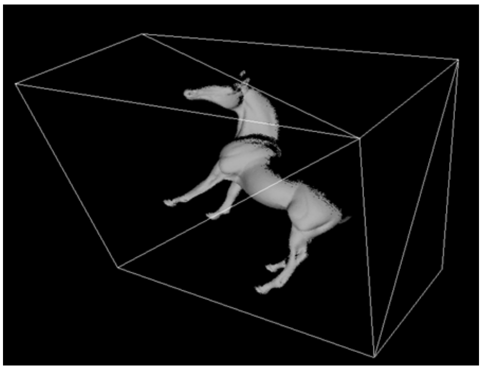

Figure 5: Rendering an enhanced billboard with all the bounding box planes. Thus, all the geometries are captured.

Unfortunately, previous algorithms could not produce an image with the silhouette feature; the boundary of the image will be flat. Furthermore, if we look at the image from the side view we will see a thin plane with nothing rendered. The earlier methods were built to render a patch of surface.

However, to render a complete model, many aspects of the algorithm had to be improved.

In order to correctly produce the silhouette, we use the bounding box information that comes with the enhanced billboard. For each enhanced billboard, we construct a bounding box of that enhanced billboard in screen space. Instead of rendering one plane with the previous ray casting algorithm, we render the enhanced billboard with all sides of its bounding box (see Fig.4 and Fig.5), which guarantees that all the geometry represented in the enhanced billboards will be rendered. In the view direction, three sides of the box are visible at most: the top and the two sides. The bottom of the box does not need to be rendered because the bottom plane shows the rear of the polygons.

The algorithm reports a hit if the depth of the traced ray is close to the depth value in the depth map. If the ray does not hit any position in the height map, the algorithm must report no hits (see Fig.6), and the pixel of the missed ray will not be rendered. Fig 7 shows the pseudo-code of the algorithm.

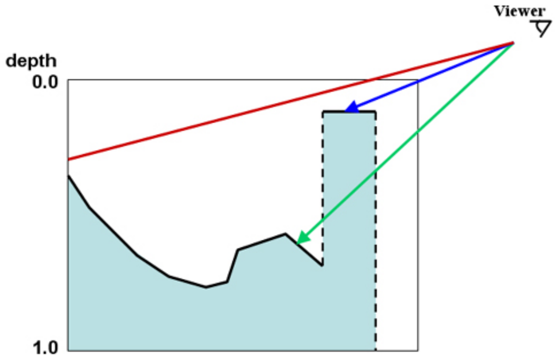

Figure 6: The red ray does not hit the depth map, thus reporting no hits. The green and blue rays hit the depth map at the arrow tips and report hits. Notice that the green ray passes through the dash line because only the black line represents the geometry while the dash line does not.

In the rendering algorithm, a ray is cast onto each pixel of the image. The algorithm performs a linear search to find the intersection of the ray and the depth map by stepping in the direction of the ray. We then compare the ray depth with the depth map. If the ray depth is in range (e = 0.01), the algorithm reports a hit. The step size is computed from $\sqrt{2}$ (diagonal of texture coordinate, which is the maximum length of the ray) divided by the total number of steps (user-defined).

```
Render Enhanced Billboards
Input:  enhanced billboards images
Output: picture rendered on screen
e = hit range, maxStep = number of step
For each enhanced billboards
    Build bounding box of enhanced billboards
    For each texel on the plane of box
        start = texel position
        dir = direction from eye position to start
        d = 0; hitd = 0; hit = false;
        step = $\sqrt{2}$ / maxStep
        For 1 to maxStep
            ray = start + dir*d
            depth = GetDepthMapValue(ray.xy)
            if ( ray.z > depth) and (ray.z – e < depth)
                hit = true; hitd = d; exit loop
            d = d + step
        if (hit)
            hitpos = start + dir*hitd
            Render_Light(hitpos)
        else
            render as transparent pixel
```

Figure 7: Pseudo-code of algorithm for rendering ray-height field of enhanced billboards.

4. IMPLEMENTATION AND RESULT

We have implemented our rendering techniques described in the paper using HLSL. The programmability of the GPU is used for casting rays through enhanced billboards. Our testing system is a 2.8 GHz PC with 512 MB of memory, using a GeForce FX6800 with 128 MB of memory. We tested our method with a variety of polygon models, containing around 5,000 to 50,000 polygons.

The model was represented as enhanced billboards (see Fig.8 and Table 1.). The construction algorithm generated an output of around 8 to 10 enhanced billboards, each stored as 2 images. There might be a problem if the input model had two intersecting polygons, so the occlusion query was used to report that these two polygons were occluded and therefore not chosen. We dealt with this problem by forcing these intersecting polygons to bypass the occlusion test. There might be some polygons in the model which were so tiny that their projected areas were less than a pixel. We discarded these polygons. Some enhanced billboards that were generated from

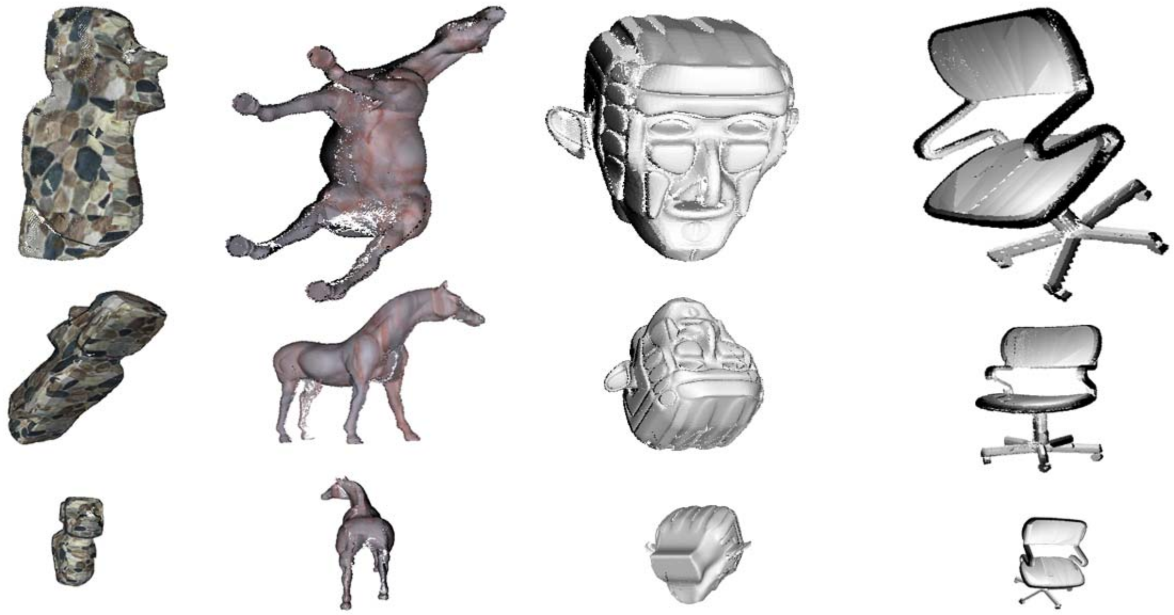

Figure 8: Variety of models rendered using enhanced billboards in different views and resolutions. Each of these images use around 4 – 5 enhanced billboards. Notice the correct silhouette. The render times are shown in Table 1.

Model	Polygons in Original Model	Build Time	Render Time High Resolution (800x600)	Render Time Medium Resolution (400x300)	Render Time Low Resolution (200x150)
Moai	5,000	26	7.11	15.31	35.36
Horse	40,000	195	10.02	18.34	40.12
Head	35,000	180	14.22	26.89	50.01
Chair	7,200	50	7.46	15.10	30.28

Table 1: The build times (seconds) and render times (fps) of different models. With different resolutions, enhanced billboards still represent the same amount of polygons. However, the frame rates are increasing because the models cover fewer pixels. Notice that the render time is not significantly dependent on the number of polygons in the original model.

the algorithm might contain little polygons seen as fragments of points in the image. To increase rendering performance, we also discarded these enhanced billboards because they had little contribution to the final rendered image. We decided not to project the polygons if they were almost perpendicular to the chosen direction even though they were facing forward. This was due to the fact that the enhanced billboard could not capture the information of almost perpendicular polygons. These polygons covered a small area of the enhanced billboard and thus were not sufficient for reconstructing the model.

Enhanced billboards are usually limited by the fill rates of the pixel shader, which uses a lot of pixel shader instructions. However, results have shown that enhanced billboards scale well with the pixels rendered. Thus, it was efficient in rendering objects at medium to high distance in any direction (see Fig 8.) with complete effects, such as parallax or silhouette. In contrast, the polygon representation methods did not scale according to the pixels they covered. Many distant polygons were rasterized to the same pixel, resulting in an alias in the rendered image.

5. CONCLUSION AND FUTURE WORK

We have presented a method for simplifying the model using a new representation and rendering technique. The enhanced billboards, each representing some portion of the model, store all the information of the original model. The information of the model is stored as a depth map, a normal map, a color map and a transparency map projected onto a plane by the greedy algorithm. Then, the enhanced billboards are rendered using the new ray-height field intersection algorithm. Our enhanced billboards can produce effects such as parallax and silhouette in any viewing directions. The quality of the enhanced billboard is comparable to the source model, and the frame rate does not depend on the number of polygons in the original model. Thus, this technique creates consistency in rendering time. Our result shows that the enhanced billboard is most suitable for rendering objects at medium to high distance where the pixels covered are less than those of the closer object.

We plan to improve the quality of enhanced billboards to solve the problem of missing fragments in the model, which occurs because some projected polygons have an area that is less than a pixel and are therefore discarded. Discarding consecutive polygons thus results in the missing fragments in the model.

The enhanced billboard technique can be easily extended to render shadow using the shadow map. The shadow map algorithm for enhanced billboards is not different from the polygonal model. First, the enhanced billboards are rendered from the light viewpoint and stored as a shadow map using the same rendering algorithm when viewing enhanced billboards. Then, the shadow map is used in rendering time the same way as in the polygonal model. Enhanced billboards can be used to speed up the ray intersection test for a ray-tracing algorithm. Usually, this algorithm builds an accelerated structure, such as a grid to query all the geometries occupied in the grid cell where the ray is located. However, enhanced billboards already represent the positions of all the geometries located inside the bounding box of the enhanced billboards. Thus, there is no need to build an auxiliary data structure.

Since enhanced billboards are stored as images, the image compression techniques can be used to lower the size of image files. In recent years, wavelet compression has gained popularity [Ren03] due to its compression factor combined with minimal loss of overall image feature. Enhanced billboards can benefit from using wavelet compression to lower the amount of storage needed.

6. REFERENCES

[Bax02] Baxter, B., Sud, A., Govindaraju, N., and Manocha, D. Gigawalk:Interactive walkthrough of complex 3d environments, *Proc. of Eurographics Workshop on Rendering*, 2002

[Bli78] Blinn, J. Simulation of Wrinkled Surfaces, *SIGGRAPH 78*, pp. 286-292, 1978

[Fab05] F´abio Policarpo, M.M.O., Jo˜ao L. D. Comba Real-Time Relief Mapping on Arbitrary Polygonal Surfaces, *ACM SIGGRAPH 2005 Symposium on Interactive 3D Graphics and Games 2005*, 2005

[Gar97] Garland, M., and Heckbert, P. Surface simplification using quadric error bounds, *Pro. of ACM SIGGRAPH*, pp. 209-216, 1997

[Hir04] Hirche, J., Ehlert, A., Guthe, S.,and Doggett, M. Hardware accelerated per-pixel displacement mapping, *Graphics Interface*, pp. 153 – 158, 2004

[Hop04] Hoppe, H. Geometry clipmaps: Terrain rendering using nested regular grids, *ACM SIGGRAPH 2004*, pp. 769-776, 2004

[Hop96] Hoppe, H. Progressive meshes, *ACM SIGGRAPH 1996*, pp. 99-108, 1996

[Hop97] Hoppe, H. View dependent refinement of progressive meshes, *ACM SIGGRAPH Conference Proceedings*, pp. 189-198, 1997

[Jak00] Jakulin, A. Interactive vegetation rendering with slicing and blending, *Proc. Eurographics 2000 (short papers) (Aug. 2000)*, 2000

[Jes02] Jeschke, and Wimmer, M. Textured depth mesh for real-time rendering of arbitrary scenes, *Proc. Eurographics Workshop on Rendering*, 2002

[Mey98] Meyer A, N.F. Interactive volumetric textures, *Eurographics Rendering Workshop*, 1998

[Mey01] Meyer A., N.F., POULIN P. Interactive rendering of trees with shading and shadows, *Eurographics Workshop on Rendering (Jul 2001)*, pp. 183-196, 2001

[Oli00] Oliveira, M.M., Bishop, G., and Mcallister, D. Relief texture mapping, *Siggraph 2000, Computer Graphics Proceedings*, pp. 359-368, 2000

[Per97] Peercy, M., Airey, J., and Cabral, B. Efficient bump mapping hardware, *SIGGRAPH '97*, pp. 303-306, 1997

[Pha96] Pharr, M., and Hanrahan, P. Geometry caching for ray-tracing displacement maps, *Eurographics Rendering Workshop 1996*, pp. 31-40, 1996

[Phi04] Philippe Decaudin, F.N. Rendering Forest Scenes in Real-Time, *Eurographics Rendering Workshop 2004*, 2004

[Ren03] Ren Ng, R.R., Pat Hanrahan. All-Frequency Shadows Using Non-linear Wavelet Lighting Approximation, *SIGGRAPH 03*, 2003

[Sch99] Schaufler, G., and Priglinger, M. Efficient displacement mapping by image warping, *Eurographics Rendering Workshop 1998*, pp. 175-186, 1999

[Sha98] Shade, J.W., Gortler, S. J., HE, L.-W., andSzeliski, R. Layered depth images, *Siggraph 1998, Computer Graphics Proceedings*, pp. 231-242, 1998

[Smi00] Smits, B.E., Shieley, P., and Stark, M. M. Direct ray tracing of displacement mapped triangles, *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, pp. 307-318, 2000

[Wan03] Wang, L., Wang, X., Tong, X., Lin, S., Hu, S., Guo, B., and Shum, H.-Y. View-dependent displacement mapping, *ACM Trans. Graph. 22*, pp. 334-339, 2003

[Xav03] Xavier Decoret, F.D., Franc¸ois X. Sillion, Julie Dorsey. Billboard Clouds for Extreme Model Simplification, *SIGGRAPH 03*, 2003

[Xi04] Xi Wang, X.T., Stephen Lin, Shimin Hu, Baining Guo, Heung-Yeung Shum. Generalized Displacement Maps, *Eurographics Rendering Workshop 2004*, 2004