Pixel Accurate Shadows with Shadow Mapping

Christian Luksch*

Institute of Computer Graphics and Algorithms
Vienna University of Technology
Austria

Abstract

Shadow mapping is a popular technique for shadow generation. However, it is still a difficult problem to generate high quality shadows in real time framerates for arbirary scenes. This work summarizes state-of-the-art techniques to achieve pixel accurate shadows and points out the various problems of generating artifact free shadows. Furthermore, we present a new technique that uses multiple jittered shadow maps with confidence-based accumulation to simulate shadow map resolutions beyond hardware capabilities. With a screen-spaced evaluation method we can guarantee pixel correct shadows.

Keywords: Pixel Accurate Shadows, Shadow Mapping, Deferred Shading

1 Introduction

The shadow mapping algorithm introduced by Williams [12] is an efficient way to determine the shadow projected by a light in a scene. Thereby the light-view depth values are rendered to a texture which is used to classify the visibility of the scene-fragments relative to the light. In theory this algorithm has very little limitations and performs well on modern graphics hardware. On the other hand shadow mapping hugely suffers from aliasing artifacts, which all together make pixel accurate shadows for all kinds of scenes and camera positions very difficult.

Shadow map aliasing will occur when there is not enough information in the shadow map to do an accurate shadow test for a fragment. Because of the finite resolution of the shadow map, for a fragment in eye-space the corresponding depth value in light-space can only be approximated by sampling the nearest value or doing some sort of interpolation. The lack of accuracy causes a blocky appearance and unaesthetic incorrect shadowing results.

In this paper, we introduce a new technique that renders pixel accurate shadows using shadow mapping. The idea is to approximate the correct depth image with multiple slightly jittered shadow maps. Only the most accurate samples are selected on a confidence-based method to generate the shadow. We show that is also possible to calculate the required confidence for pixel accurate shadows, so that our new technique can guarantee a correct result.

In the following sections we give a brief overview of typical shadow mapping problems and previous work. In Section 4 we present our research on confidence-based shadows and elaborate our new technique to guarantee pixel accurate shadows. In Section 5 we give insight into our demo application and details of our implementation. Finally, we summarize the discussed techniques in a comparison, followed by the conclusion.

2 Shadow Mapping Problems

The whole shadow mapping process suffers from two types of aliasing artifacts, perspective and projection aliasing. Additionally there is a self shadowing problem and the fact that shadow maps can not be filtered like common textures.

Perspective aliasing: In a perspective view objects near the camera are larger than distant objects. When the shadow map is rendered the scene is regularly sampled, which results in undersampling near the camera and oversampling in the distance.

Projection aliasing: This type of aliasing is independent of the camera, it only depends on the angle between the light direction and the surface normal. If the angles is almost perpendicular, the surface area has barley any shadow map resolution. It is difficult to counteract this and cannot be solved by a simple global method.

Incorrect Self-Shadowing: The shadow map can be seen as a regular grid of depth samples taken from the scene, which are resampled during the shadow test. This leads to incorrect self-shadowing artifacts. Therefore some sort of distinction or biasing must be used.

Shadow Map Filtering: Filtering is very important to hide undersampling artifacts or to get anti-aliased shadow outlines in oversampled areas and increases the overall shadow quality. Common texture filtering can not be used, because interpolated depth values make no sense along object edges and will still generate sharp outlines. Special filtering techniques developed for shadow mapping have to be used.

*christian.luksch@aon.at

Several techniques have been developed to improve the quality of the shadow mapping algorithm. The next section gives a brief overview and classifies these techniques.

3 Previous Work

Most of the shadow mapping techniques try to overcome the aliasing artifacts, which are the result of undersampling due to limited resolution. The ideal solution would be to generate a depth sample in light space for each fragment in screen space. This approach has been followed by Aila et al with so called "Alias-Free Shadow maps" [1]. Unfortunately, this requires irregular shadow map samples, which makes it hard to implement the algorithm efficiently.

Many pixel exact shadow mapping techniques use some sort of hierarchical tiling to achieve the required sampling resolution where it is needed. To this class of algorithms belongs Adaptive Shadow Maps [4], Tiled Shadow Maps [2], Queried Virtual Shadow Maps [6] and Fitted Virtual Shadow Maps (FVSM) [5].

A widely used class of techniques are those that create a view-dependent reparameterization of the shadow map, so that there are more samples close to the view point. In this category belongs Perspective Shadow Mapping (PSM) [11], Trapezoid Shadow Mapping (TSM) [8] and Light Space Perspective Shadow Mapping (LiSPSM) [13].
In comparison to standard shadow mapping the complexity of these techniques is almost the same, which makes them practical for real-time rendering. However, the quality depends on the view point and will change when the camera moves, which even can get as bad as standard shadow mapping in the so called *Duelling Frusta Case*, where the view direction is almost parallel to the light direction.

Another category of techniques are those that split the view frustum in smaller parts and create a shadow map for each of them. A possible partitioning scheme is to split by the face edges of the view frustum seen from the light, which allows to build a reparameterization for each face to optimize the sample positions. Another possibility is to slice the view frustum along the view axis, which has been presented with Parallel Split Shadow Maps (PSSM) [14] or also called z-partitioning. It can be combined with shadow map reparameterization as well.

Lloyd et al. [7] extensively analyzes the aliasing error of the latter two classes of techniques. It is shown that z-partitioning used with a warping technique like LiSPSM should be the best scheme to render shadows in scenes with a high depth range to reduce perspective aliasing.

Scherzer et al. [9] presented a technique that reuses already rendered shadow information through temporal reprojection and use a confidence-based method to merge with the shadow rendered in the next frame. A single reparameterized shadow map is rendered each frame to achieve a high frame rate. With additional jittering exact shadows will be produced after a certain number of frames.

Beside these techniques shadow map filtering will also be required to render artifact free shadows. Percentage Closer Filtering (PCF) is a widley used technique. Another approach is Variance Shadow Mapping (VSM), which enables to use common hardware texture filters and makes large filter kernels more efficient. [3]

4 Confidence-Based Shadows

In this section we present our confidence-based pixel accurate shadow mapping technique. To generate high accuracy shadows a very high resolution shadow map in a dimension that is far beyond graphics hardware capabilities would be required. A series of slightly jittered shadow maps can be used to simulate higher shadow map resolutions. The shadow maps are combined in a screen-space buffer, which uses a confidence value [9] to preserve the best samples. Deferred shading is used to reduce the rendering overhead [10].

When a fragment of the scene is shaded, it is transformed into light-space and the nearest depth value is read from the shadow map. Unfortunately, the exact depth value is only known at the center of a texel, which usually will not be hit. Therefore, we also store a confidence value of this shadow test in the accumulation shadow buffer, based on the distance to the nearest texel center, where the depth information has been taken from. The confidence can be described through the texture coordinate *tc* and the resolution *sm*:

$$conf = 1 - \max\left(|\{tc_x \cdot sm_{xl}\} - 0.5|, |\{tc_y \cdot sm_{yl}\} - 0.5|\right) \cdot 2 \quad (1)$$

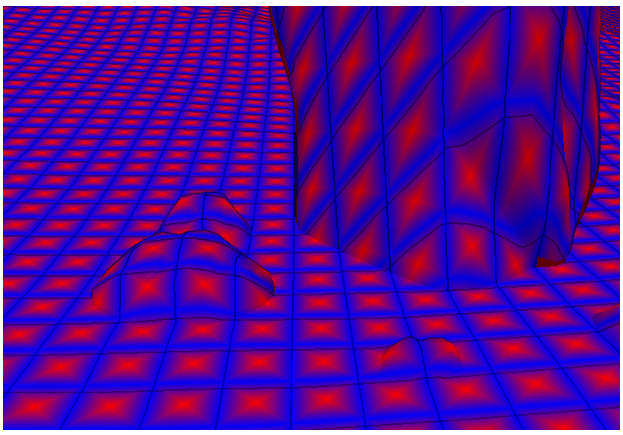

Figure 1: Illustrates the confidence of the sampling position of a projected shadow map. The confidence is high (red) at the center of the shadow map texels, because that is where the depth values has been rendered. The farther apart the lower the confidence.

Figure 1 visualizes this confidence value. Such a confidence value is bound to the light space and the shadow map resolution. This will be relevant in the discussion in Section 4.5.

4.1 Jittering

The next step is to generate a series of shadow maps with different rasterization. This is achived by a jittering method based on a random number sequence with the following possibilities: *Translational jittering*, whereby each shadow map is rendered with an offset along the light view plane in sub-pixel scale. *Rotational jittering*, where the light view is rotated around the light direction. *LiSPSM-n jittering*, a new technique which is discussed in Section 4.5. A combination of these methods can be used as well.
The offsets and rotations are controlled by a series of random numbers. To be able to reproduce the result the series has to be the same. A simple random number sequence could be used, but the Halton sequence numbers are more convenient, because they guarantee a nearly uniform distribution and appear to be random at the same time. Alternatively, Poisson Disc Sampling can be used which has a similar well behavior. An illustration of the sample positions is shown in Figure 2.

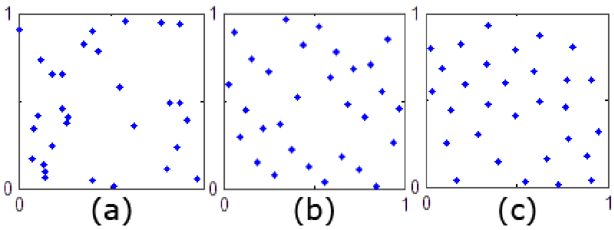

Figure 2: 20 sample positions generated with (a) a random function, (c) the Halton sequence and (c) Poisson Disc Sampling.

4.2 Accumulation

With every new pass the different rasterizations contribute new shadow information. There are several ways to achieve this, thereby it has to be discovered when to stop rendering new passes and if the result is actually the exact one.
The first implemented method draws only shadow fragments with a certain confidence, we will refer to this method as *Simple Confidence*. This will draw dots at the center of the shadow map texel. After enough passes to cover the whole texel have been rendered, a continuous shadow in a higher quality is generated. Thereby a certain confidence value always requires a certain number of passes to cover the sample area, whereby the quality is also increased by a certain factor.
Figure 3 shows the first five samples of translational jittering with a very high confidence value and a low shadow map resolution to visualize the process. For this technique the confidence value and number of passes has to be configured manually and it has to be ensured that the whole texel is covered.
A problem is that the shadow outline also increases in relation to the size of the projected shadow map texels and the configured confidence value.

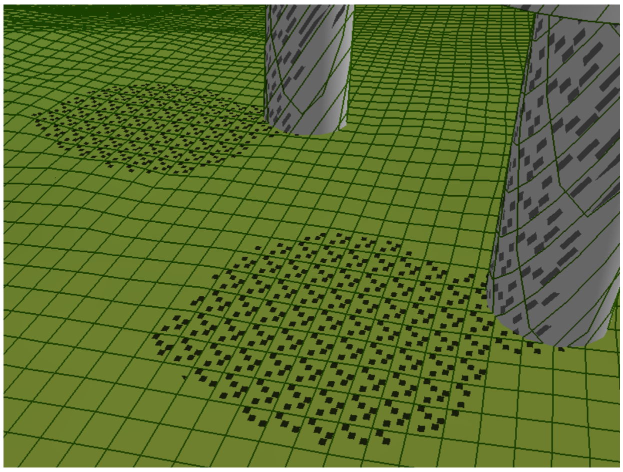

Figure 3: Translational jittering using the first five offsets given by the Halton sequence. The points represent the jittered texel centers and the grid the unjittered shadow map.

Based on the insight gained from the first method, an advanced method to accumulate confidence-based shadows has been developed, which will be referred to as *Adapted Confidence*.
To no longer depend on configuring the confidence value and a fixed number of passes, new fragments are only rendered if their confidence value is higher than the current one regardless of its shadow. This principle can easily be implemented using an additional hardware depth buffer which holds the current confidence. This method also obtains that the shadow is always continuous and not composed by dots, which allows to stop at any pass without leaving falsely unshadowed holes.
However, an automatic stopping criteria can not be used, because each pass only simulates a higher shadow map resolution, which only increases the confidence value of the shadow tests, but there is no correlation to the scene properties. Unless we know how much confidence is required, such a method is not possible.

4.3 Optimal Confidence

An efficient pixel exact shadowing method should adapt on the unique scene properties of each setting. The required shadow map resolution has to be evaluated, which has to be done on fragment basis.
So far our method increases the accuracy of the shadow

test globally by simulating higher shadow map resolutions with each additional pass through jittering. Actually each fragment requires a certain minimum shadow map resolution to be exactly shadowed, which is equivalent to a certain confidence value. To adapt the quality locally it is required to know which confidence is needed per fragment. This value depends on how the shadow map is projected onto a fragment. To approximate this projection we use the neighbouring fragments and project all to light-space and calculate their shadow map texture coordinates. The spanning area gives information of the required confidence. This process is illustrated in Figure 4.

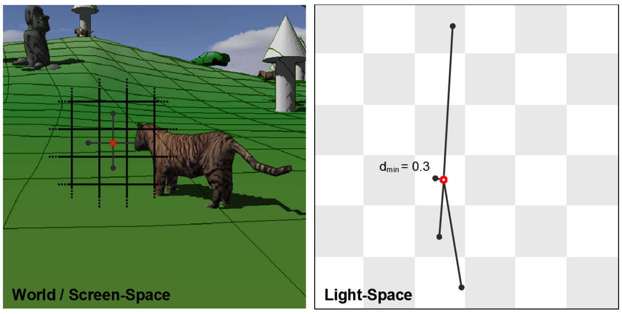

Figure 4: Illustrates how the optimal confidence is calculated. For each fragment the neighboring fragments are projected to light-space and the minimum texture coordinate distance is used to determine the required confidence. It also shows a case of discontinuity.

We calculate the distances to the neighboring fragments and use the minimum distance $tcDist_{min}$ to determine the required confidence $optConf$ in the following way:

$$optConf = 1 - tcDist_{min} \cdot smSize \qquad (2)$$

In the case of discontinuities the distance of such samples will be significantly greater than of connected ones and therefore does not influence the result.
The optimal confidence is calculated in a separate pass after the scene is rendered and stored in full-screen texture. This factor is now used to bias the confidence written to the depth buffer $finalConf$, in such way that the value will be 1 once the required confidence is reached and will not get overwritten anymore.

$$finalConf = \text{saturate}(conf + (1 - optConf)) \qquad (3)$$

With this method it is possible to use occlusion queries to determine when to stop rendering new passes. The number of rendered fragments will drop each pass and gives an approximation of how far the shadow is converged. We use a simple heuristic with two thresholds to configure the stopping criteria. The process is aborted when less than ε fragments are rendered for N passes. Such a rule is reasonable, because even when zero fragments have been rendered in the last pass, it is not guaranteed that all fragments reached their required confidence and a new rasterization still might contribute new fragments.
Figure 5 visualizes the required confidence calculated with this method. The high perspective aliasing in this setting shows that uniform shadow mapping requires a shadow map resolution about 50 times of the current one near the camera, on the other hand the distribution with LiSPSM is well balanced.

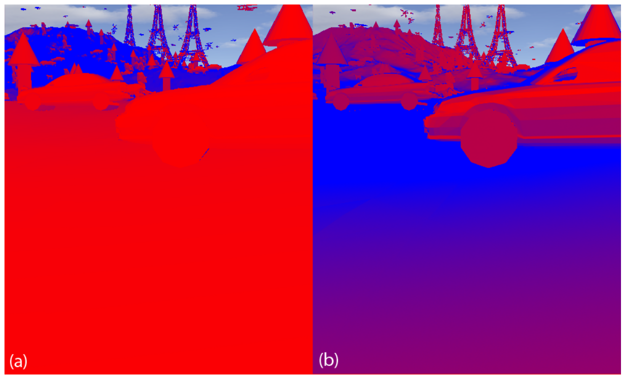

Figure 5: Minimum required confidence (red: high, blue: low) needed to generate pixel accurate shadows. (a) uniform shadow mapping, (b) LiSPSM.

4.4 Early Results

The number of passes or achievable shadow quality with the adapted confidence method depends on the initial shadow map sample distribution. On settings where LiSPSM can already eliminate most of the perspective aliasing, in a few passes projection aliasing artifacts can be eliminated very well. However, in the duelling frustum case where only uniform shadow mapping can be used, the required confidence can get too high in some areas to get a pixel accurate result in a practical number of passes.
Figure 6 shows such a duelling frustum case, but there are no objects very close to the camera so that the undersampling is moderate. It compares uniform shadow mapping with confidence-based shadow rendered in 20 passes and using a 2048^2 shadow map. (b) has been rendered with the simple confidence method using a confidence threshold of 0.5, thereby the increased shadow outline, which is about half a shadow map texel, can be noticed. (c) uses the adapted method and blending with the optimal confidence. The outlines look very fringed, because the required confidence is still not reached. This effect is only in dimension of one pixel, nevertheless, blending or blurring should hide it very well.
A problem is that shadow map filtering to generate smooth shadow outlines does not work with our confidence-based shadow accumulation. Pixel exact anti-aliased shadow outlines would require a much more expensive accumulation.

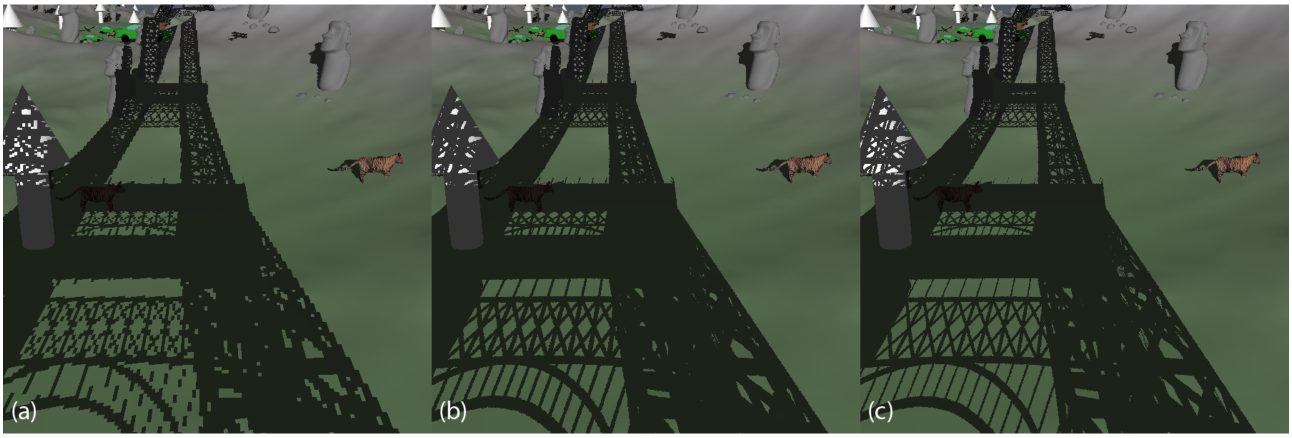

Figure 6: Comparison of uniform shadow mapping with a 20-pass confidence based shadow. (a) uniform shadow mapping (b) simple confidence based shadow: drawing dots with conf > 0.5 (c) adapted confidence based shadow accumulated with optimal confidence.

4.5 LiSPSM n-Parameter Jittering

LiSPSM uses a parameter that controls the balance of the quality between front and back. The required confidence in Figure 5 (b) indicates a very well balanced sample distribution. It is the result of the automatic calculation of the LiSPSM n-parameter according to the paper. A detailed explanation how the n-parameter works can be found in [13].

The problem is that this technique suffers from biasing artifacts in the back, but for accurate shadows the suitable bias would be too high. Our solution is to vary the n-parameter to yield different rasterizations. Thereby we manually selected a factor for n that produces accurate shadows in the back. Then every second pass n is multiplied with that, whereby samples are alternately focused in the front and the back. An additional random factor between $[0.5, 1.5]$ does the final jittering, but its hard to find the optimal amount of jittering and it does not yield the same result in all cases. Furthermore, if confidence-based accumulation should be used, a different reparameterization would mean another required confidence and therefore two confidence values from different passes have no relation and actually can not be compared. When the confidence values are biased by the optimal confidence the confidence-based accumulation is correct again, but it means that the optimal confidence has to be recalculated whenever n is changed.

Figure 7 compares translational jittering (a) with our implementation of n jittering (b) after 10 passes confidence-based shadow accumulation. A 1024^2 shadow map has been used, whereby markedly biasing artifacts occur near the far plane, but increasing the bias to an artifact free amount would cause hugely misplaced shadows.

Translational jittering produces perfect shadows near the camera, but it is not capable of completely removing the biasing artifacts in the distance after 10 passes. With our n jittering method the markedly biasing artifacts nearly vanish, while the shadow quality near the camera marginally loses some accuracy. Overall, this is an additional jittering method that can be used in scenes with high depth range where such biasing artifacts are typical problems.

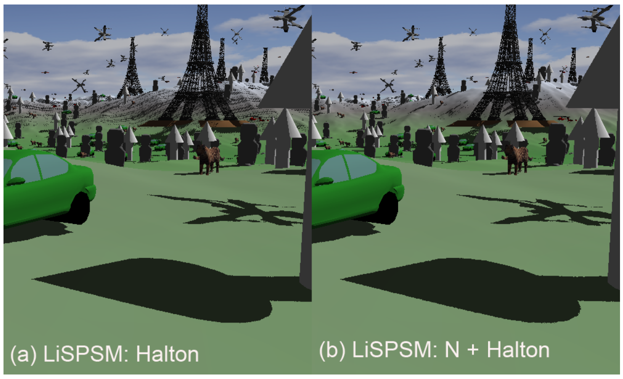

Figure 7: Comparison of translational (a) with n (b) jittering after 10 passes with a 1024^2 shadow map.

4.6 Confidence with frustum reduction

A simple but very effective splitting scheme is used by Parallel Split Shadow Maps (PSSM) [14]. It is based on the observation that objects in different depth layers from the camera require a different shadow map resolution, which is achieved by splitting the view frustum in depth slices and a shadow map is rendered for each. A practical slicing scheme is a combination of a linear and a logarithmic function.

In this section we present two new techniques, a simple approach and an extension to guarantee pixel accurate shadows that combine frustum splitting and confidence based shadow accumulation.

When a uniform shadow map is rendered, the shadow near the far plane is usually oversampled and therefore requires no further refinement. For our adapted confidence method this means that new samples there will be rejected. Our first method is to reduce the shadowed view frustum each pass. The reduction step is a simple factor that can be defined with a similar method like used with PSSM. However, it has to be considered that the whole frustum is shadowed instead of a depth slice and therefore a slightly higher factor may be needed, which results in a few more required passes.
Since confidence-based shadow accumulation should be used, the calculated required optimal confidence has to be adapted to the new projection of the reduced frustum. Due to a frustum reduction results in a uniform increment of the sampling rate in the focused area, the original calculated optimal confidence can simply be scaled by:

$$finalConf = \text{saturate}(conf + (1 - optConf) \cdot f_{reduced}/f), \quad (4)$$

where f is the view frustum far plane distance and $f_{reduced}$ the current reduced far plane distance.

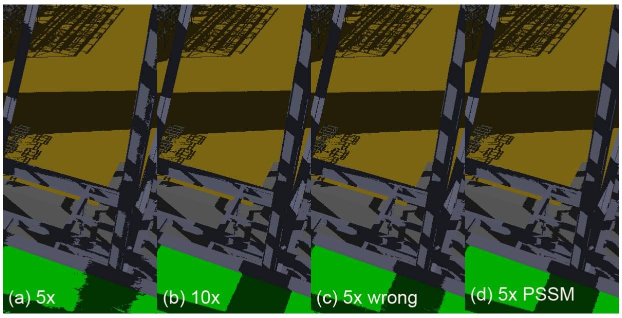

Figure 8: Frustum reduction by factor 0.55 after 5 (a) and 10 (b) passes. (c) reduction factor of 0.45 and 5 passes with wrong shadow results. (d) comparison to PSSM in 5 slices.

Figure 8 shows the shadow result after 5 (a) and 10 (b) passes using a 2048 shadow map and a reduction factor of 0.55. Less passes or a lower factor will leave pixels that have wrong shadow results (c), however, by not using the confidence-based accumulation and by simply overdrawing the shadow result with the new one, these artifacts would be resolved. Nevertheless, most of the shadow maps would be wasted and PSSM with focusing on the depth slice would make more sense. In comparison to PSSM the quality is similar, but thereby already 5 slices were sufficient to generate an acceptable shadow result (d).
The first method comes to a final result in a fixed number of passes, but does not guarantee pixel accurate shadows. As second approach we implemented an extension that tests the backmost area whether the shadow is converged. This can be determined by using hardware occlusion queries in a pass that draws pixel only of fragments in this depth area that have not reached the optimal confidence. Now jittering is continued until the occlusion query returns a value below a certain threshold. It turned out that the shadow sometimes does not converge fast enough in the back, because of projection aliasing. A suitable occlusion query threshold is hard to find, because it depend on the scene. Therefore, it takes many passes until the front shadow quality is usable. Hardware occlusion query also cost quite much performance.
Overall, the result is quite similar to PSSM and sometimes even superior, because of additional jittering. The total rendering costs of our frustum reduction approach are mostly higher due to parts of the shadow frustum are rendered several times. However, in a case where the light direction is almost parallel to the view direction, PSSM also has to render lots of shadow casters several times and the shadow frusta of the slices will overlap. In other words, our frustum reduction method always proceeds like in such a case and additionally uses confidence-based accumulation.

5 Implementation Overview

To compare and evaluate differences between various shadow mapping techniques, it is important to have a common test basis. We implemented our own application, which is a powerful tool to experiment with alternative approaches and it gives more control over the implemented techniques. Only a few other published shadow mapping demo applications exist and most of them only support a single technique.
Our application has been implemented in C++ and uses the DirectX 9 graphics API. The test system is a Core2Duo @ 3Ghz with 4GB Ram and a Nvidia GeForce 8800 GTS with 640MB video ram.
We used two test scenes, one random generated terrain with numerous static and dynamic objects, in which the large size and perspective aliasing plays a major role. Second a scene with the power plant mesh which has lot of fine structures and much more projection aliasing. Screenshots can be found in Figure 9.
Further a set of camera positions has been composed that show various cases of shadowing scenarios. Thereby comparable analyses can be done at different times and also allows accurate benchmarking.
As comparison techniques FVSM [5] and PSSM [14] have been implemented.
FVSM: A scene analysis calculates the required shadow map resolution for each pixel on the screen after the initial deferred rendering pass. This information is used to build a hierarchical update structure of a huge virtual shadow map, which then is rendered in several passes using quad-tree like tiling and a screen space accumulation buffer.
PSSM: Uniform shadow mapping is used for the depth slices. The number of passes and the slicing scheme has to be tuned manually depending on the scene. Because of flexibility we chose to apply each slice separately

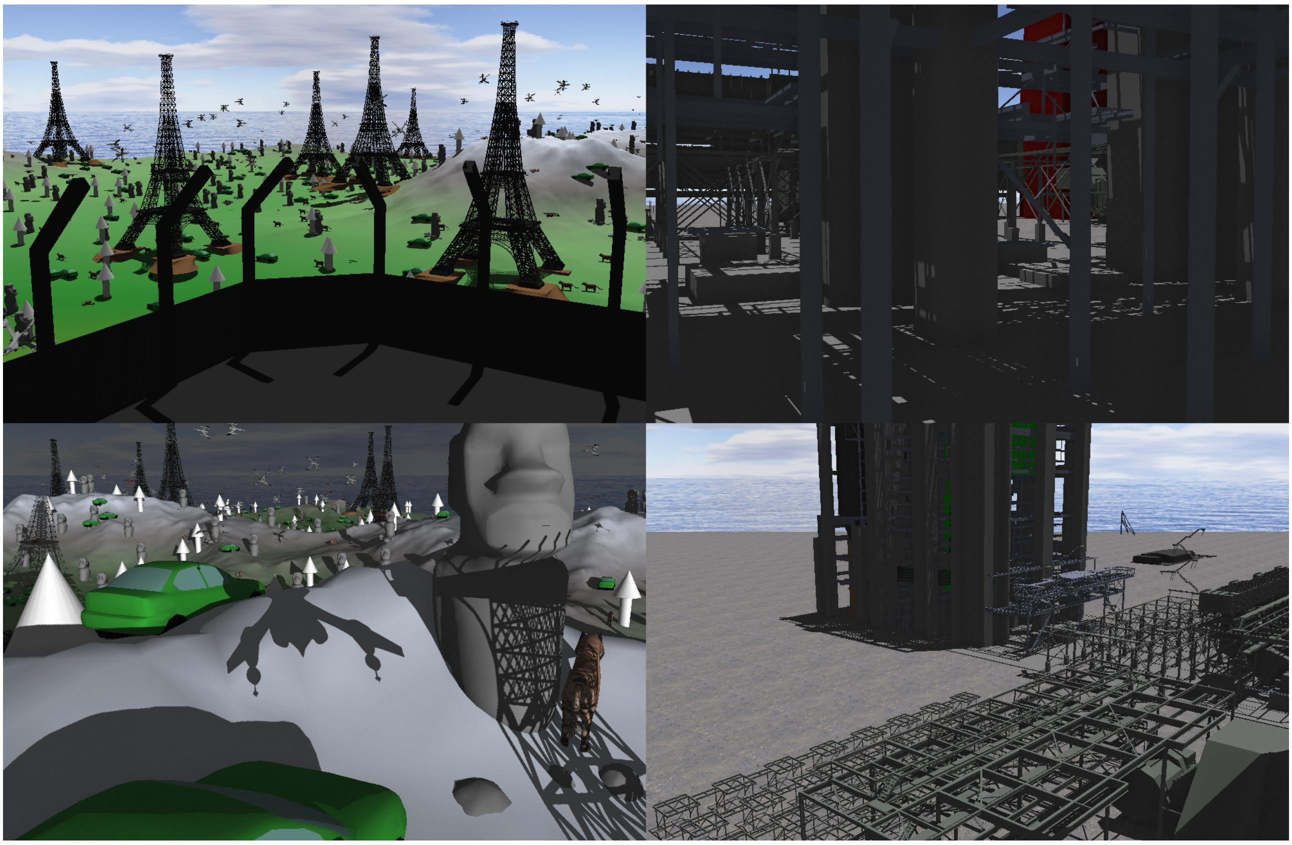

Figure 9: Screenshots of the demo application. Terrain scene (left) and power plant (right).

using our accumulation buffer, which costs a bit more performance than the method suggested in the original paper.

6 Comparison

This section compares the presented techniques in our two test scenes, a terrain and the power plant. A performance benchmark had been made, where each technique had to render ten different camera settings for three seconds. All techniques had been configured with practical settings to achieve the best possible shadow quality with the least required rendering costs. Figure 10 shows the achieved frames per second (FPS) in both scenes on the test system.

Because a previous performance analysis showed that a 2048^2 shadow map is the most efficient on the test system, this setting had been used with all techniques.
Adapted Confidence: To get a well initial shadow map alignment, LiSPSM had been used. The stopping criteria was when less than 3000 pixels had been refined in the last pass. In some unfavorable conditions the shadow converged very slowly and the process had to be aborted after 30 passes, whereby the shadow outlines were still fringed. The graph shows that the performance strongly depended on the setting and overall was clearly below all other.

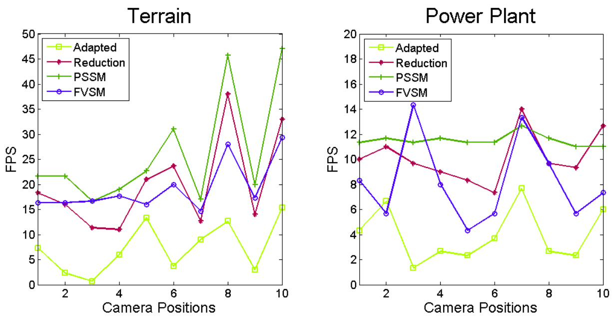

Figure 10: Performance comparison between Adapted Confidence, Frustum Reduction, PSSM and FVSM.

Frustum Reduction: The number of required passes depends on the visible depth range. In most cases a few passes were sufficient and could had been rendered in a useful frame rate. However, a pixel correct shadow result is not guaranteed and in rare cases some artifacts could had been seen.
PSSM: To get nearly perfect shadows in all settings, five slices had been configured. A constant high frame rate had been achieved in nearly all settings. Transitions between slices were not visible.

FVSM: The scene analysis keeps the number of required passes quite low and does not cost much performance itself. Overall the performance was almost on the top, whereat artifact free pixel correct shadow had been rendered.

This final table puts all techniques together and compares some important aspects against each other:

	Pixel Accurate	Passes	Implementation	Artifacts
Adapted	yes	many	complex	fringed outlines
Reduction	optional	few	complex	marginal
PSSM	no	few	simple	(transitions)
FVSM	yes	average	complex	none

7 Conclusion

Deferred shading is definitely the way to go when multiple shadow maps should be accumulated successively. The additional rendering cost in the first pass already compensate when the scene has a high depth complexity and when expensive shading computations should be done in the fragment shader, which is the case especially in the power plant scene.
We have shown that confidence-based shadows can produce very accurate shadows. With the optimal confidence and occlusion queries it is possible to automatically adapt on the scene properties, but the number of required passes can exceed the practical limit. Therefore, a combination with temporal reprojection [9], is possibly a better way to accumulate high quality shadows when the frame rate is high enough.
Furthermore, PSSM turned out to be a very powerful technique to overcome perspective aliasing, but the only way to completely avoid undersampling would be to make an extensive scene analysis to set optimal splits. FVSM does this and uses a simple shadow map tiling approach to refine the shadow quality, but it seem not to be the best way to tile the shadow map, because a nearly exact shadow result can be generated with only a few depth slices in most cases as well.

References

[1] Timo Aila and Samuli Laine. Alias-free shadow maps. In *Proceedings of Eurographics Symposium on Rendering 2004*, pages 161–166. Eurographics Association, 2004.

[2] Jukka Arvo. Tiled shadow maps. In *CGI '04: Proceedings of the Computer Graphics International*, pages 240–247. IEEE Computer Society, 2004.

[3] William Donnelly and Andrew Lauritzen. Variance shadow maps. In *SI3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 161–165. ACM Press, 2006.

[4] Randima Fernando, Sebastian Fernandez, Kavita Bala, and Donald P. Greenberg. Adaptive shadow maps. In *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 387–390. ACM Press / ACM SIGGRAPH, 2001.

[5] Markus Giegl and Michael Wimmer. Fitted virtual shadow maps. In *Proceedings of Graphics Interface 2007*, pages 159–168. Canadian Human-Computer Communications Society, 2007.

[6] Markus Giegl and Michael Wimmer. Queried virtual shadow maps. In *Proceedings of ACM SIGGRAPH 2007 Symposium on Interactive 3D Graphics and Games*, pages 65–72. ACM Press, 2007.

[7] Brandon Lloyd, David Tuft, Sung-eui Yoon, and Dinesh Manocha. Warping and partitioning for low error shadow maps. In *Proceedings of the Eurographics Symposium on Rendering 2006*, pages 215–226. Eurographics Association, 2006.

[8] Tobias Martin and Tiow-Seng Tan. Anti-aliasing and continuity with trapezoidal shadow maps. In *Proceedings of the 2nd EG Symposium on Rendering*, pages 153–160. Eurographics Association, 2004.

[9] Daniel Scherzer, Stefan Jeschke, and Michael Wimmer. Pixel-correct shadow maps with temporal reprojection and shadow test confidence. In *Rendering Techniques 2007 (Proceedings Eurographics Symposium on Rendering)*, pages 45–50. Eurographics Association, 2007.

[10] Oles Shishkovtsov. *GPU Gems 2*, chapter Deferred Shading in S.T.A.L.K.E.R., pages 143–166. Addison-Wesley, 2005.

[11] Marc Stamminger and George Drettakis. Perspective shadow maps. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 557–562. ACM, 2002.

[12] Lance Williams. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph.*, 12(3):270–274, 1978.

[13] Michael Wimmer, Daniel Scherzer, and Werner Purgathofer. Light space perspective shadow maps. In *Rendering Techniques 2004 (Proceedings Eurographics Symposium on Rendering)*, pages 143–151. Eurographics Association, 2004.

[14] Fan Zhang, Hanqiu Sun, Leilei Xu, and Lee Kit Lun. Parallel-split shadow maps for large-scale virtual environments. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, pages 311–318. ACM Press, 2006.