Eurographics Symposium on Rendering (2004)
H. W. Jensen, A. Keller (Editors)

Light Space Perspective Shadow Maps

revised version June 10, 2005

Michael Wimmer, Daniel Scherzer and Werner Purgathofer

Vienna University of Technology, Austria

Abstract

In this paper, we present a new shadow mapping technique that improves upon the quality of perspective and uniform shadow maps. Our technique uses a perspective transform specified in light space which allows treating all lights as directional lights and does not change the direction of the light sources. This gives all the benefits of the perspective mapping but avoids the problems inherent in perspective shadow mapping like singularities in post-perspective space, missed shadow casters etc. Furthermore, we show that both uniform and perspective shadow maps distribute the perspective aliasing error that occurs in shadow mapping unequally over the available depth range. We therefore propose a transform that equalizes this error and gives equally pleasing results for near and far viewing distances. Our method is simple to implement, requires no scene analysis and is therefore as fast as uniform shadow mapping.

Categories and Subject Descriptors (according to ACM CCS): J.7.6 [Computer Applications]: Real time I.3.3 [Computer Graphics]: Picture/Image Generation - Display algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Virtual reality

1. Introduction

Shadows belong to the most important effects to convey realism in a computer-generated scene. One of the most popular shadow generation algorithms is *shadow mapping* [Wil78], due to its simplicity, generality and high speed. With shadow mapping, the scene is first rendered from the view of the light, storing the depth values in a separate buffer. When the scene is then rendered from the normal viewing position, each pixel is transformed again into the light view, and its depth value compared to the depth value stored in the shadow map. If the depth value of the shadow map is nearer to the light source, the pixel is in shadow.

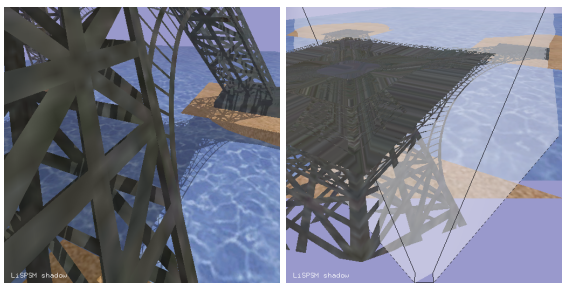

Figure 1: *Light space perspective shadow maps (LiSPSM) (left) and the corresponding warped light view (including the eye frustum) (right). Note the high shadow detail both for near and distant objects.*

Like all image-space algorithms, shadow mapping suffers from aliasing artifacts due to quantization and perspective projection. Recently, several approaches have tried to reduce those aliasing artifacts. One particularly promising idea, *perspective shadow maps* [SD02], is based on a perspective reparameterization of the shadow map. The original perspective shadow map technique suffers from a few drawbacks, which result mostly from the fact that the chosen perspective mapping is based on the observer projection:

- The lights have to be transformed into post-perspective space, and frequently change their type (from point to directional and vice versa, or from a normal to an "inverted" lightsource). Thinking in this particular post-perspective space is not intuitive.
- The mapping to post-perspective space has a singularity, which causes problems with shadow casters on or opposite the singularity. The practical solution to this problem

is to "move" the viewpoint backwards for the shadow map generation until all relevant objects are included in front of the singularity. However, this reduces the shadow-map quality.
- Due to a number of special cases, the implementation is quite involved.
- The increase in shadow map resolution near the viewer comes at the expense of a drastic reduction of resolution for distant objects.

In this paper, we introduce *light space perspective shadow maps (LiSPSM)*, a new shadow mapping technique based on a variable perspective mapping specified in light space (see Figure 1 for an example). The advantage of the technique is that the chosen perspective mapping has no relevant singularities, allows treating all lights as directional lights and does not change the direction of the light sources. Therefore, most of the problems of perspective shadow maps are avoided. One of the most important contributions of this paper is a thorough error analysis of all shadow mapping techniques based on perspective reparameterizations. This analysis is then exploited to derive optimal parameters for our LiSPSM technique.

An important insight is that perspective shadow maps trade near resolution against large shadow errors in distant objects. In contrast, our method can be tuned to distribute the error equally or in a user-specified way among different distance regions. Light space perspective shadow maps are robust, as fast as standard shadow maps and simple to implement. Their major limitation is inherent to all methods based on reparameterizing the shadow plane, i.e., they can only deal with *perspective aliasing*. Dealing with more kinds of shadow aliasing usually requires a potentially costly scene analysis per frame.

The remainder of the paper is organized as follows: after a discussion of previous work in Section 2, we describe the LiSPSM technique in Section 3. In Section 4, we provide a thorough error analysis of uniform, perspective and light space perspective shadow maps and show how to optimally calculate the free parameter of LiSPSMs (n), which controls the distribution of aliasing error. Results are given in Section 5, and conclusions and ideas for future work in Section 6.

2. Previous work

The two most important categories of shadow algorithms are *shadow volumes* [Cro77] and *shadow mapping* [Wil78]. Shadow volumes work in object space and are therefore potentially more accurate, but place a high burden on geometric and pixel processing and require well tesselated objects. Therefore, interest in shadow mapping techniques is large, since they work in image space and require only one additional rendering pass per light.

Most of the publications dealing with shadow maps try to solve the associated aliasing artifacts. Percentage closer filtering [RSC87] alleviates reprojection problems by filtering. A number of papers have tried to solve *perspective aliasing* due to the perspective view-frustum projection. The most prominent of these is the perspective shadow map method by Stamminger and Drettakis [SD02], which tries to remove perspective aliasing by subjecting the shadow map to the same perspective transform as the viewer. Despite its drawbacks, this paper has inspired and opened the door to more general shadow map reparameterization approaches, of which we present one in this paper. Another way to reparameterize the shadow map is to tilt or warp the shadow plane directly [CG04, LI03]. Recent approaches propose to combine shadow maps with shadow volumes or other primitives [SCH03, GLY*03]. These techniques can potentially be combined with the light space perspective shadow maps presented in this paper.

Another approach to solve the aliasing problem are adaptive shadow maps [FFBG01], where shadow maps are stored in a hierarchical fashion in order to provide more resolution where it is required due to different aliasing artifacts. However, the approach requires multiple readbacks and does not map well to current graphics hardware. Second depth shadow mapping [WM94] can be used to reduce problems due to depth quantization and self occlusions. Brabec et al. [BAS02] improve uniform shadow map quality by focussing the shadow map to the visible scene.

As with all things related to real-time rendering, an excellent overview of shadow mapping and shadow algorithms in general can be found in Möller and Haines' Real-Time Rendering book [MH02]. Finally, concurrent to our own work, there have been alternative developments to reduce shadow map aliasing [Koz04, MT04, CG04, AL04].

3. Light space perspective shadow maps

3.1. Motivation

The goal of this work is to provide a fast, high-quality and robust shadow-map algorithm. Perspective shadow maps attempt to improve shadow quality by warping the shadow map according to a perspective transform given by the view transform. Our approach draws on this basic idea and improves upon it based on two main observations:

- While perspective transforms are valid tools to warp the shadow map, there is no reason that this perspective transform needs to be tied to the view frustum as in perspective shadow maps. In fact, any arbitrary perspective transformation could be used.
- Since the main goal of the perspective transform is to change the distribution of shadow map pixels, it is sufficient to use a warp that affects mainly the shadow map plane and not the axis perpendicular to the shadow map.

These two observations motivate a perspective transformation specified with respect to the coordinate axes of *light*

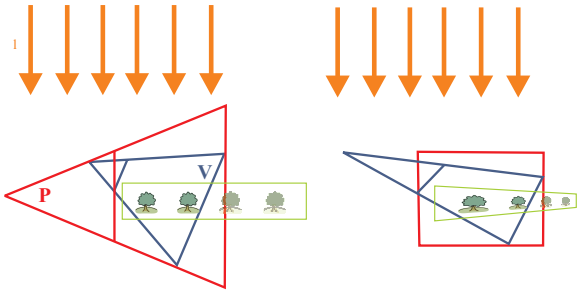

Figure 2: *An example configuration of light space perspective shadow maps with view frustum V and the frustum defining the perspective transform P. Left: Note how the light rays l are parallel to the near and far plane of P. Right: After perspective transformation, the light direction is unchanged.*

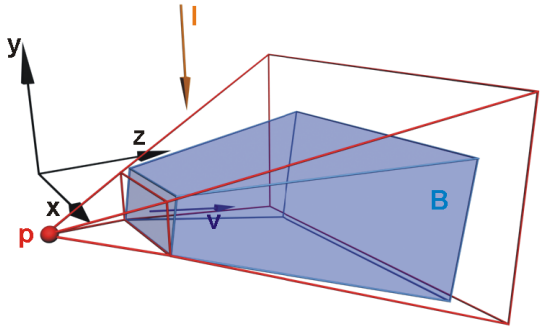

Figure 3: *Construction of the perspective frustum P in 3D.*

space. In contrast to perspective shadow maps, this transformation has the important property that it does not change the direction of the light sources, and has no relevant singularities, because the view plane is parallel to the light vector. Directional lights remain directional lights in post-perspective space, while point lights are converted to directional lights as well. This results in a much more intuitive transformation, at the same time avoiding many of the problems found in perspective shadow maps.

Note however that our method only deals with the same aliasing errors as perspective shadow maps. In particular, projection errors due to surfaces parallel to light rays are not handled. An analysis of shadow mapping errors can be found in Section 4.

3.2. Overview

Light space perspective shadow maps are applied in the following steps:

- Focus the shadow map on the convex body B that encloses all interesting light rays (i.e., the view frustum and all objects that can cast shadows into it).
- Enclose this body with an appropriate perspective frustum P that has a view vector parallel to the shadow map.
- By choosing the free parameter in P, the distance n of the projection reference point p to the near plane of the frustum, control the strength of the warping effect.
- Apply P both during shadow map generation rendering just as in standard shadow mapping.

Figure 2 shows an example configuration of light direction l, view frustum V and perspective frustum P, and the resulting warp.

3.3. Focussing the shadow map

The first step proceeds exactly as described by Stamminger and Drettakis [SD02] for perspective shadow maps by *focussing* onto the convex body that is relevant for shadow calculation. This is done by calculating the convex hull of the view frustum and the light position (which is at infinity for a directional light) and intersecting this hull with the light frustum and the scene envelope (typically its bounding box). The result of this calculation is a convex body B described by a number of points.

3.4. The perspective frustum in light space

The parameters for the perspective transform P can best be found in *light space*. We construct light space in the following way (see Figure 3): The y-axis is defined by the light vector l (but pointing towards the light). In the case of point lights, the spot direction vector is used as light vector (non-spot lights are usually not used for shadow mapping). The z-axis is defined to be perpendicular to the light vector and to lie in the plane containing the observer view vector v and the light vector. The x-axis complements the other two axes in order to form an orthogonal coordinate system. Note that in this definition of light space, the xz-plane is parallel to the shadow-map plane, and the z-axis corresponds roughly to the depth coordinate of the eye coordinate system. For illustration purposes, we will use a left-handed light space coordinate system in this paper, i.e., the z-axis has the same orientation as the view vector. In practice, the z-axis will usually be reversed to give a right-handed coordinate system, as is common for example in OpenGL.

For point lights, light space is defined as above, but *after* the perspective transform associated with the point light. There are no singularities in the combined perspective mapping, because the body B is completely in "front" of the light position and the view frustum (provided no objects straddle the near plane of the point light frustum). Thus, point lights can be treated as directional lights from this point on.

The *near* and *far planes* for the perspective frustum P are

defined as planes parallel to the xy-coordinate plane, placed at the minimum and maximum light-space z-coordinate among the points of the body B.

The x and y coordinates of the *projection reference point* p for P are chosen so that the resulting frustum is roughly symmetrical, by taking the x-coordinate from the transformed viewpoint and the y-coordinate as the middle of the minimum and maximum y-coordinates of the body B.

3.5. Choosing the free parameter n

The *remaining free parameter* for P, the distance n of the projection reference point p to the near plane, influences how strong the shadow map will be warped. If it is chosen close to the near plane of P, perspective distortion will be strong, and the effect will resemble the original perspective shadow maps. If it is chosen far away from the far plane of P, the perspective effect will be very light, approaching uniform shadow maps. In Section 4, we will show that in the case of a view direction perpendicular to the light vector, the optimal choice for this parameter is $n_{opt} = z_n + \sqrt{z_f z_n}$, where z_n and z_f are the near and far plane distances of the eye view frustum. We will also show that in order to also give optimal results when the viewer is tilted towards the light or away from it, n has to be increased, so that it reaches infinity when the viewer looks exactly into the light or away from it. LiSPSMs then give exactly the same results as a uniform shadow map (which is optimal for this case).

Finally, the *frustum planes* are found by projecting all points of the body B onto the near plane of P and recording the maximum extents along the x and y axes of the near plane.

3.6. Applying the perspective frustum

Once the appropriate perspective frustum P has been found, its application is simple. The frustum combined with the usual projective mapping used for standard shadow maps. The mapping is used in two places, namely in the shadow map generation, and in the texture coordinate generation for shadow map rendering.

4. Analysis and optimal parameter estimation

In this section, we provide an analysis of shadow map aliasing errors, especially perspective aliasing, which is the error treated by our method. Our analysis is different from previous work in that we concentrate on worst-case errors in the center of view. We will show the ideal (yet impractical) logarithmic shadow map parameterization, and compare uniform, perspective and light space perspective shadow maps with regard to perspective aliasing. We will also show how the free parameter for light space perspective shadow maps can be chosen to provide optimal shadow resolution in the majority of cases.

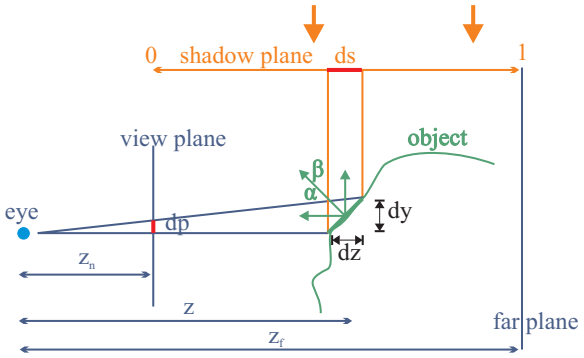

Figure 4: *Aliasing in shadow mapping.*

In this section we will focus on directional light sources for two reasons: first, aliasing artifacts are often much worse for uniform light sources due to the wide range they have to cover (e.g., outdoor lighting), and second, point lights are mapped to directional lights in our approach.

4.1. What is perspective aliasing

We will briefly reiterate the main causes for shadow map aliasing for directional lights. Figure 4 shows a configuration for a small edge.

A pixel on the shadow map represents a shaft of light rays passing through it and has the size $ds \times ds$ in the local parameterization of the shadow map. Note that we assume a local parameterization of the shadow map which goes from 0 to 1 between near and far planes of the viewer—this already assumes that the shadow map has been properly focussed to the view frustum, not wasting any resolution on invisible parts of the scene. Introducing a local parameterization at this point also has the advantage that we will be able to compare different parameterizations. In world space, the shaft of rays has the length $dz = (z_f - z_n)ds$ for uniform shadow maps as an example.

The shaft hits a small edge along a length of $dz/\cos\beta$. This represents a length of $dy = dz\frac{\cos\alpha}{\cos\beta}$ in eye space, projecting to $dp = dy/z$ on screen. Note that we assume that the small edge can be translated along the z-axis. The shadow map aliasing error dp/ds is then

$$\frac{dp}{ds} = \frac{1}{z}\frac{dz}{ds}\frac{\cos\alpha}{\cos\beta}. \qquad (1)$$

Shadow map *undersampling* occurs when dp is greater than the size of a pixel, or, for a viewport on the near plane of height 1, when dp/ds is greater than $res_{shadowmap}/res_{screen}$. As already shown by Stamminger and Drettakis, this can happen for two reasons: *perspective aliasing* when $\frac{dz}{zds}$ is large, and *projection aliasing* when $\cos\alpha/\cos\beta$ is large.

Projection aliasing occurs usually for surfaces almost parallel to the light direction. Reducing this kind of error would require higher sampling densities in such areas, which can only be found through an expensive scene analysis for each frame. In this paper, however, we concentrate on an approach which works without feedback, just like uniform or in some cases perspective shadow maps.

Perspective aliasing, on the other hand, is caused by the perspective projection of the viewer. If the perspective foreshortening effect occurs along one of the axes of the shadow map, it can be influenced by the *parameterization of the shadow map*. For uniform shadow maps, dz/ds is constant, therefore dp/ds is large when $1/z$ is large, which happens close to the near plane. In order to reduce perspective aliasing, it is therefore useful to analyze different shadow map parameterizations.

Note that perspective aliasing is the only aliasing error that can be improved using a global transformation like a perspective transform. Another important point to note is that the reparameterization is most effective when the view direction is perpendicular to the light direction. Otherwise, the depth range that can be influenced by the reparameterization decreases, down to the limit case where the view direction is parallel to the light vector. In this case, there is no perspective aliasing, and no global reparameterization of the shadow map can improve shadow map quality. This means that any such reparameterization should converge to uniform shadow maps in this situation.

Other image quality errors in shadow maps include resampling aliasing, which can be solved by percentage closer filtering [RSC87]; "swimming" artifacts, i.e., shadows that seem to frequently change their shape (this happens for all shadow map methods when they are undersampled except for the original "unfocussed" uniform shadow maps, which stay fixed in world space); and self-occlusion artifacts due to depth quantization errors.

4.2. Logarithmic shadow mapping

As has been shown in the previous subsection, perspective aliasing is the only shadow mapping problem that can be improved with a global approach, namely by changing the shadow map parameterization. An *optimal parameterization* would make dp/ds constant $= 1$ over the whole available depth range. For the ideal case of view direction perpendicular to light direction, this is (constants notwithstanding) equivalent to

$$ds = \frac{dz}{z}, \text{ i.e., } s = \int_0^s ds = \int_{z_n}^{z} \frac{dz}{z} = \ln \frac{z}{z_n}.$$

This shows that the optimal parameterization for shadow mapping (at least for directional lights) is logarithmic. Unfortunately, such a parameterization is not practical for hardware implementation. The logarithm could be applied in a vertex program on modern programmable hardware, however, pixel positions and all input parameters for pixel programs are interpolated hyperbolically. This makes graphics hardware amenable to perspective mappings, but not logarithmic ones. Still, it would be interesting to implement logarithmic shadow maps either in software or in a vertex program for finely tesselated scenes in order to see their quality advantage.

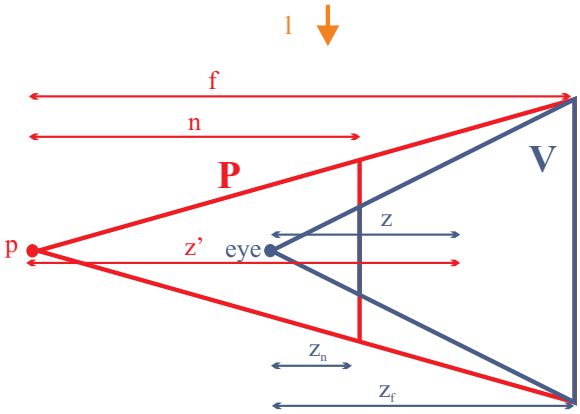

Figure 5: *The parameterization of light space perspective shadow maps (shows the yz-plane in light space). The parameter n is free and can vary between z_n (perspective shadow mapping) and infinity (uniform shadow mapping).*

4.3. Analysis of light space perspective shadow maps

We will expand the analysis from the previous subsections in order to compare light space perspective shadow maps to uniform and perspective shadow maps. We first discuss the case in which the view direction is perpendicular to the light direction. Note that this is also the ideal case for any method that tries to reduce perspective aliasing, since the whole depth range in the view frustum is available to influence the shadow map parameterization.

Figure 5 shows the setup for light space perspective shadow maps. The view frustum V (which is assumed to be identical to B here) is enclosed by the perspective frustum P as described in Section 3.4. The values z_n, z_f and z describe the distance of the near plane, the far plane and an arbitrary point respectively to the viewpoint, whereas n, f and z' represent the distances of the same entities to the projection reference point p. In order to analyze the aliasing error dp/ds, we need to find the shadow map parameterization $s = s(z)$ that is induced by the perspective transform P. The effect of P on s is more easily described using the z'-coordinate (when using, for example, the OpenGL `glFrustum` command to generate P):

$$s = \frac{1}{2} + \frac{f+n}{2(f-n)} + \frac{fn}{z'(f-n)}. \quad (2)$$

After substituting $z' = z - z_n + n$, differentiation gives:

$$\frac{ds}{dz} = \frac{fn}{(z-z_n+n)^2(f-n)}.$$

Substituting $f = n + z_f - z_n$ and plugging this into equation (1) (assuming projection aliasing $= 1$) gives:

$$\frac{dp}{ds} = \frac{(z-z_n+n)^2}{z} \frac{(z_f-z_n)}{n(n+z_f-z_n)}. \quad (3)$$

Equation (3) can now be used to analyze all three shadow mapping methods. Sending n to infinity corresponds to uniform shadow maps and gives $(z_f - z_n)/z$, i.e., the error decreases hyperbolically with increasing distance, which is of course to be expected. More interesting is the case for perspective shadow maps, where $n = z_n$. Surprisingly, the result is a linear dependence on z:

$$\frac{dp}{ds} = z\frac{z_f - z_n}{z_n z_f}.$$

This means that for perspective shadow maps, the perspective aliasing error is very small at the near plane. However, with increasing distance, the error increases rapidly. This can be seen in Figure 6, which plots the perspective errors of uniform and perspective shadow maps for $z_n = 1$ and $z_f = 100$. See Section 4.5 for an explanation why this is consistent with the claim that PSMs are optimal for this situation, given a different error analysis. Note again that uniform shadow maps are assumed to be focussed.

Light space shadow maps lie between these two extremes of uniform and perspective shadow maps, depending on the parameter n. There are many ways to choose this parameter, including user choice or sampling several positions on screen. However, we opted for a choice which gives the optimal error distribution along the whole depth range with respect to the $\mathbf{L}_{max}$ norm. First, analyzing equation (3) shows that the function has only one positive local extremum, a minimum at location $n - z_n$. Therefore, the maxima within the relevant z-range $[z_n, z_f]$ are located at the boundaries of the range. Consequently, the minimal $\mathbf{L}_{max}$ norm is achieved when both maxima are equal. Substituting first z_n and then z_f into equation (3) and solving for n yields the desired parameter value for the ideal case of view direction perpendicular to light direction:

$$n_{opt} = z_n + \sqrt{z_f z_n}.$$

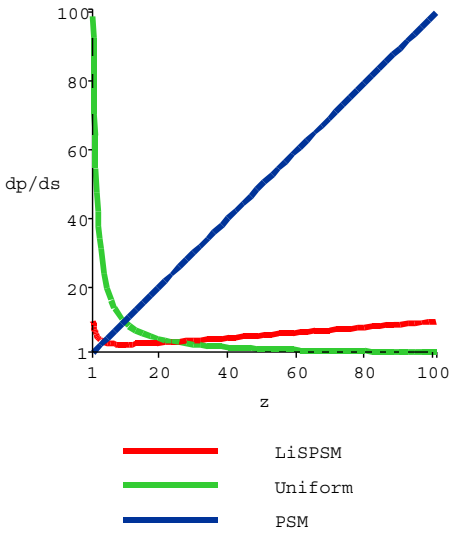

Figure 6: *Perspective aliasing errors plotted against z-coordinate for different shadow mapping techniques.*

The associated maximum error rises roughly like $\sqrt{z_f}$, which is much better than both uniform and perspective shadow maps, which have a maximum error of z_f. In addition to the perspective errors for uniform and perspective shadow maps, Figure 6 also shows the error for light space perspective shadow maps (LiSPSM) with $n = n_{opt}$. It can be seen that the error for LiSPSMs decreases quickly to the local minimum at $\sqrt{z_n z_f}$, then rises practically linearly until reaching again the maximum error at the far plane. One could say that LiSPSMs inherit the hyperbolic falloff of uniform shadow maps at the near plane, and the linear increase from perspective shadow maps from the minimum on, but both with far smaller error maxima.

Finally, this treatment has only dealt with perspective aliasing in the z-direction. In the x-direction, the situation is slightly different. The problematic behavior of perspective shadow maps for the z-direction arises from the fact that the z coordinate is projected into the shadow map s coordinate, which is subject to foreshortening. The x-coordinate, however, is projected into the shadow map t coordinate which is orthogonal to s, and this coordinate is *not* subject to perspective foreshortening. On the contrary, t undergoes the same perspective transformation as x, and is therefore ideal in the perspective shadow map approach, i.e., dp/dt is constant.

For light space perspective shadow maps, the situation is a bit different. The perspective aliasing error dp/dt evaluates to

$$\frac{ds}{dt} = \frac{z - z_n + n}{z} = 1 + \frac{\sqrt{z_n z_f}}{z},$$

where n has been replaced by n_{opt} in the right term. So, the error starts with a maximum about the same order of magnitude than dp/ds and then falls off hyperbolically. This means that error is distributed fairly between the x and z direction, whereas in perspective shadow maps, the x direction is ideal and the z direction can show dramatic errors. Note that the unequal treatment of errors in the z and x direction is inherent to all approaches that reparameterize the shadow plane.

4.4. General case

So far, the discussion has only dealt with the ideal case of a view direction perpendicular to the light direction. In practice, the light will rarely come directly from above, and the observer will also want to look up and down. As discussed in Section 4.1, this decreases the available depth range, and in the limit case of the light direction parallel to the view direction, no reparameterization of the shadow map can improve its quality, so the parameterization should converge to uniform shadow mapping.

In the general case, the eye space z-coordinate does not correspond directly to the light space z-coordinate. This has to be taken into account in the derivation of n_{opt} via the tilt angle γ, i.e., the angle between the light direction and the view vector. The main difference in the derivation of the perspective aliasing error dp/ds is that in equation (2), z' has to be assumed as $z' = (z - z_n)/\sin\gamma + n$ instead of the formula given there. Leading this through, the optimal parameter value only has to be adjusted slightly as $n'_{opt} = n_{opt}/\sin\gamma$.

This choice causes the perspective warping effect to decrease with increasing tilt angle. For $\gamma = 0$, n'_{opt} goes to infinity, totally removing the perspective transform. This means that LiSPSMs converge to uniform shadow maps as desired. For $\gamma = \pi/2$, on the other hand, the original formula results.

There are certain additional intricacies involved in the general case. Due to the tilt of the view frustum with respect to the light direction, the warping effect of the shadow map reparameterization is also tilted with respect to the view frustum, which can reduce the range of z values that are actually affected by the warp. Figure 7 shows such a constellation. In such cases, it is advisable to replace in the calculation of n'_{opt} z_f by $z_n + \Delta z$, where $\Delta z = (f - n)\sin\gamma$. This will cause the mapping to converge faster to the uniform shadow map.

4.5. Discussion

Uniform and perspective shadow maps (PSMs) are at the opposite ends of a spectrum of error distributions: while uniform shadow maps show most error close to the viewer and then gain rapidly in quality, PSMs have the best quality near the viewer and then lose quality. The good results reported for PSMs seem surprising seeing that in Figure 6, uniform shadow maps surpass PSMs quite quickly in quality, even though this is supposed to be the ideal case for PSMs. The main reason for this is that our error analysis is more robust than the one used by Stamminger and Drettakis.

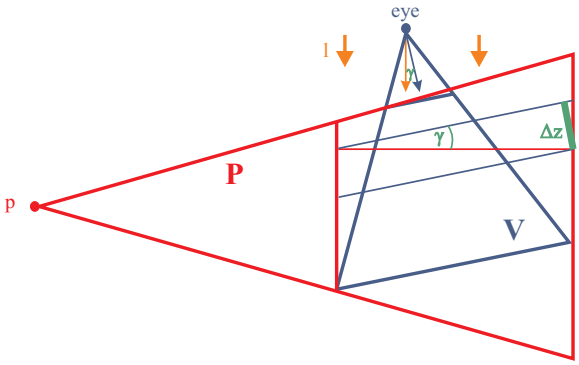

Figure 7: *The z-range affected by the reparameterization is small when the view direction gets near the light direction. For perspective shadow maps, V and P would be identical, but the effect of the warp on the z-distribution would be the same.*

Our analysis of dp/ds (here discussed again for the case of view direction perpendicular to light direction) is based on a small edge at the center of the viewport that is translated in z-direction. This occurs for example when the viewpoint or an object is translated in z-direction, or for objects in a scene that have varying depths (see Figure 8). To extend the analysis to a general edge, it can be shown that the term $1/z$ (together with the trigonometric term) in equation (1) needs to be replaced by $k/z - y/z^2$, where k is the slope of the edge. For $y = 0$ and $k = 1$, our analysis results. In contrast, the error analysis done by Stamminger and Drettakis assumes small edges that are distributed within a single line (in 2D). This corresponds to restricting y to lie on a line with the same slope as the edge: $y = kz - y_0$ with y_0 arbitrary. The term $1/z$ from our analysis then becomes y_0/z^2, which, if used for the derivation of equation (3), indeed gives the claimed constant error dp/ds along the chosen line for PSMs (Figure 8). However, the problem for PSMs is that the value of this constant can be quite large—it essentially behaves as shown in Figure 6 when translated in z-direction. This can also be seen in typical light views for PSMs, where almost no area is reserved for far objects. In order to somewhat mitigate this quick increase in error, the authors push the near plane as far as possible into the scene. Unfortunately, this requires a readback from graphics hardware, which we avoid for LiSPSMs.

In essence, LiSPSMs are more robust because they minimize the worst-case perspective aliasing error, which oc-

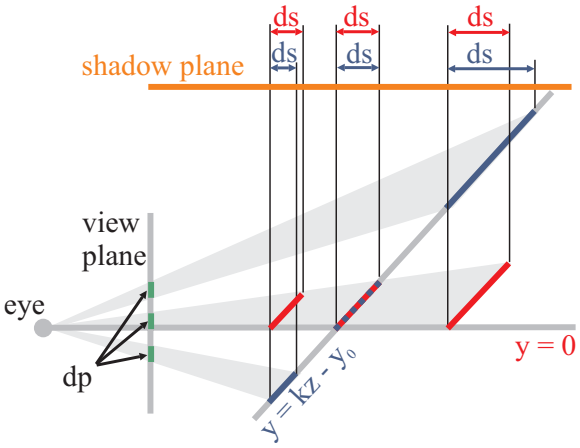

Figure 8: *Shows the lines along which the perspective error is analyzed. PSMs have constant error on small edges along the sloped line, whereas LiSPSMs distribute error evenly for small edges along the line with $y = 0$. The warping effect for PSMs is much stronger, which is in general not desirable.*

curs near the center of the viewport, by blending smoothly between uniform and perspective shadow maps. This is in contrast to PSMs, which equalize the error along any given plane, but this error can be quite large depending on the distance of this plane. Our optimal parameter choice limits the error of LiSPSMs to about the square root of the maximum error of the other two approaches (which both have the same maximum error) in our error analysis. Note that $\mathbf{L}_{max}$ with respect to the z-coordinate might not be the best error criterion, as nearby objects are far more important visually. On the other hand, the ratio between errors in near and far regions can be easily modified, giving, for example, even more weight to nearby objects at the expense of distant objects.

We also want to stress that the analysis in this section was based on directional light sources. While point lights are mapped to directional lights automatically by our approach, the perspective transform native to the point light may invalidate some of these findings for the case of point lights. However, we do not consider this a big limitation of our analysis. On the one hand, point lights in indoor environments or other "near", i.e., very "perspective" point lights are not as prone to perspective aliasing anyway, and uniform shadow maps might be sufficient in these cases. On the other hand, "far" point lights behave more like directional lights and should therefore be covered well by our analysis. Furthermore, there is a growing interest in using directional lights for large outdoor environments, for example in games.

5. Results

We have implemented the light space perspective shadow map algorithm, and for comparison also uniform shadow mapping and perspective shadow mapping. The implementation runs on GeForce3-class or higher hardware and uses only OpenGL ARB extensions. Performance was not measured separately, as LiSPSMs run as fast as standard shadow mapping (i.e., with at least 75 fps for all tested scenes on an Intel Pentium4 2.4GHz and an NVIDIA GeForceFX 5900 graphics card). All images presented in the following (Figure 9) were captured using a 512x512 pixel viewport and a shadow map resolution of 1024x1024 pixels, using varying directional light directions. Bilinear filtering was enabled, but we didn't implement percentage-closer filtering. The view frustum field of view was set to 60°, the near plane to 1.0, and the far plane to 400, which corresponds approximately to the the scene extent.

Uniform shadow maps were always focussed on the relevant scene parts (unfocussed uniform shadow maps are practically unusable for larger scenes). For perspective shadow maps, we did not implement a read-back of the frame buffer as suggested by the authors, but pushed the near plane to the nearest intersection with an object bounding box in the view.

The images demonstrate our findings that uniform shadow maps work well for distant objects, perspective shadow maps for close objects, whereas LiSPMs distribute the perspective error equally among near and distant objects, providing good results for both. Note that in some cases, projection aliasing can be seen, which cannot be solved with any technique relying on reparameterization.

6. Conclusions and future work

We have presented light space perspective shadow maps, a practical shadow mapping technique that combines the advantages of perspective and uniform shadow maps, provides overall high shadow quality and avoids the pitfalls of perspective shadow maps. The algorithm is robust, simple to implement and as fast as standard shadow maps.

We have also conducted a thorough error analysis of perspective aliasing errors in different shadow mapping techniques and shown our algorithm to be in a certain sense optimal among perspective shadow map reparameterizations, and not far from the ideal, logarithmic, parameterization.

In terms of future work, it would be interesting to investigate logarithmic and hierarchical parameterizations. Another idea is using multiple shadow maps for different depth regions. Figure 6 practically invites such a partition into two depth ranges, for example at the crossover point between uniform and perspective shadow map errors.

References

[AL04] AILA T., LAINE S.: Alias-free shadow maps. In *Proceedings of the Eurographics Symposium on Rendering 2004* (June 2004), Eurographics, Eurographics Association. 2

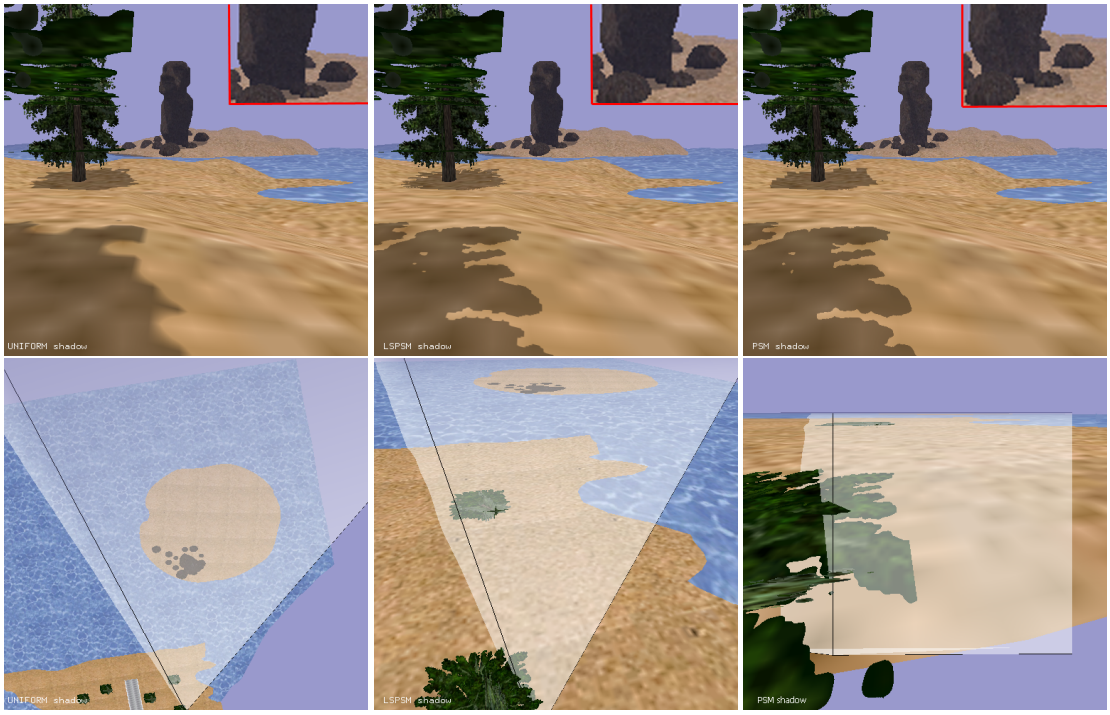

Figure 9: *Uniform (left), light space perspective (middle) and perspective (right) shadow maps. The second row shows images rendered from the the warped light views, which also include the eye frusta (shaded transparently). The view positions show shadows for near, medium and distant (zoomed in the inset) objects. Uniform shadow maps capture the distant shadow best, perspective shadow maps the near shadow, and LiSPSM work well for both. Models courtesy of www.3dcafe.com. For more results see also the attached high-resolution images and videos. A sample implementation with source code is available at the author's webpage.*

[BAS02] BRABEC S., ANNEN T., SEIDEL H.-P.: Practical shadow mapping. *Journal of Graphics Tools: JGT 7*, 4 (2002), 9–18. 2

[CG04] CHONG H., GORTLER S.: A lexel for every pixel. In *Proceedings of the Eurographics Symposium on Rendering 2004* (June 2004), Eurographics, Eurographics Association. 2

[Cro77] CROW F. C.: Shadow algorithms for computer graphics. In *Computer Graphics (SIGGRAPH '77 Proceedings)* (July 1977), George J., (Ed.), vol. 11, pp. 242–248. 2

[FFBG01] FERNANDO R., FERNANDEZ S., BALA K., GREENBERG D. P.: Adaptive shadow maps. In *SIGGRAPH 2001 Conference Proceedings* (2001), pp. 387–390. 2

[GLY*03] GOVINDARAJU N. K., LLOYD B., YOON S.-E., SUD A., MANOCHA D.: Interactive shadow generation in complex environments. *ACM Transactions on Graphics 22*, 3 (July 2003), 501–510. 2

[Koz04] KOZLOV S.: Perspective shadow maps - care and feeding. In *GPU Gems* (2004), Addison-Wesley. 2

[LI03] LOW K.-L., ILIE A.: Computing a view frustum to maximize an object's image area. *Journal of Graphics Tools: JGT 8*, 1 (2003), 3–15. 2

[MH02] MÖLLER T., HAINES E.: *Real-Time Rendering, Second Edition*. A. K. Peters Limited, 2002. 2

[MT04] MARTIN T., TAN T.-S.: Anti-aliasing and continuity with trapezoidal shadow maps. In *Proceedings of the Eurographics Symposium on Rendering 2004* (June 2004), Eurographics, Eurographics Association. 2

[RSC87] REEVES W. T., SALESIN D. H., COOK R. L.: Rendering antialiased shadows with depth maps. In *Computer Graphics (SIGGRAPH '87 Proceedings)* (July 1987), Stone M. C., (Ed.), vol. 21, pp. 283–291. 2, 5

[SCH03] SEN P., CAMMARANO M., HANRAHAN P.: Shadow silhouette maps. *ACM Trans. Graph. 22*, 3 (2003), 521–526. 2

[SD02] STAMMINGER M., DRETTAKIS G.: Perspective shadow maps. In *Siggraph 2002 Conference Proceedings* (July 2002), vol. 21, 3, pp. 557–562. 1, 2, 3

[Wil78] WILLIAMS L.: Casting curved shadows on curved surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)* (Aug. 1978), vol. 12, pp. 270–274. 1, 2

[WM94] WANG Y., MOLNAR S.: *Second-Depth Shadow Mapping*. Tech. rep., University of North Carolina at Chapel Hill, 1994. 2