

Light Space Perspective Shadow Maps

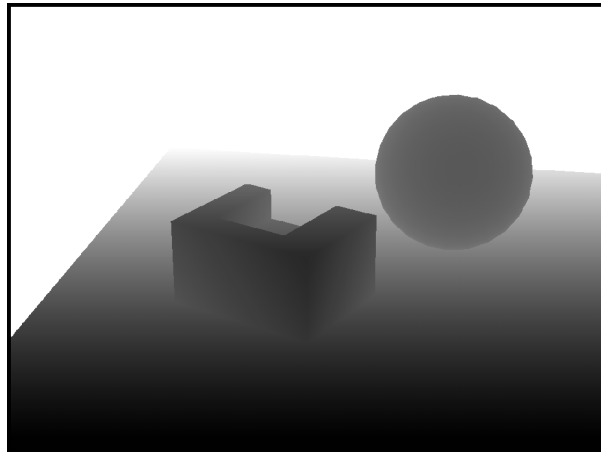
A survey under consideration of the material presented in the course Projective Geometry held by Wolfgang Förstner at the TU-Wien in 2004 written by Daniel Scherzer 9727072 E881

Introduction

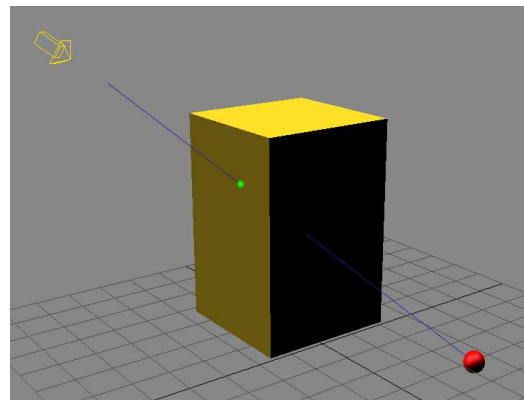
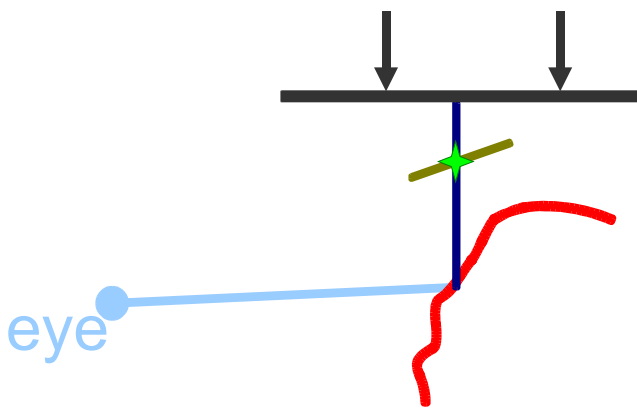
Light Space Perspective Shadow Maps (LiSPSM) are based on the well understood Shadow Mapping algorithm by Williams. One drawback of the Standard Shadow Mapping are the artefacts which are caused by perspective aliasing. The Light Space Perspective Shadow Map approach tries to decrease this artifacts with a carefully chosen perspective transformation and therefore increases the resulting shadow mapping quality.

Standard Shadow Mapping

Standard Shadow Mapping (SSM) is a two pass algorithm. In the first pass the scene is viewed from the point of view of the light source and a 2D depth image from this view is saved (== shadow map). A sample depth image is shown below.

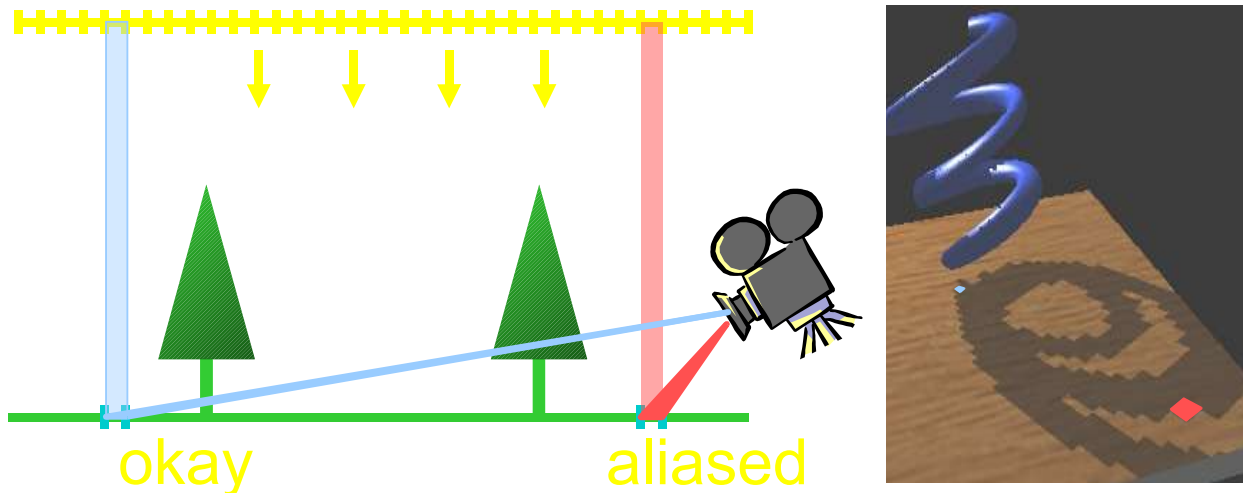


In the second pass the scene is rendered from the point of view of the observer. As each "pixel" is drawn it is transformed into the light source space and tested for visibility to the light source. If it's not visible to the light source, it is in shadow and the shading of the "pixel" should be altered accordingly. All shadow mapping techniques are somehow based on this algorithm. [Will1978]



Motivation

The main motivation for the LiSPSM lies in the perspective aliasing artifacts present in SSM. Especially objects near the viewer get insufficient shadow map resolution.



Related Work

Due to the importance of shadows for scene realism many different approaches exist to increase the quality of SSM. The LiSPSM approach is influenced by the perspective shadow mapping (PSM) approach. This approach introduces a perspective transformation to decrease the before mentioned artifacts. The difference between standard shadow mapping and perspective shadow mapping lies in the transformation to the point of view of the light source. In PSM this transformation is calculated in the post perspective space of the observer. [Stam2002]

In simpler words: First you transform to the post perspective space of the observer ($pm := \text{perspective matrix} * \text{modelview matrix}$) and then apply the light source transformations on the resulting matrix ($\text{light source perspective matrix} * \text{light source modelview matrix} * pm$). This means this transformation is dependent on the point of view of the observer. Nearer objects are enlarged and far away objects are shrunk, because of the perspective matrix of the observer, which reduces perspective aliasing.

This approach has various problems:

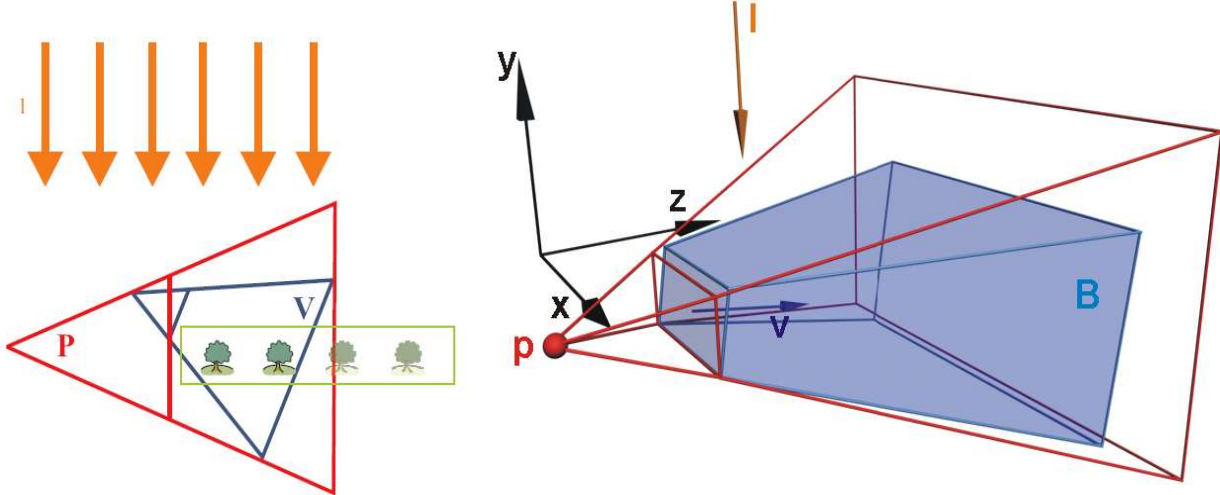
- Shadows from behind ->
This requires a virtual move back of the camera due to the singularity of the perspective transformation.
- Lights change their type (point/directional/inverted)
- Many special cases
- Non-intuitive post-perspective space
- View-dependent shadow quality
- Uneven z-distribution as can be seen to the right



Light Space Perspective Shadow Mapping[Wimm2004]

We want to redistribute samples in the shadow map. A perspective transformation is a good option, because it can do the required redistribution and is supported by hardware, but why should we choose the perspective transformation based on the observer transformation, like PSM do?

The basic idea of LiSPSM is to specify the perspective transformation in light space. We therefore introduce an additional frustum p , marked in red. Blue is the view frustum and the light direction l is marked in orange. How do we specify p ?



P has to enclose the convex body B that encloses all interesting light rays (i.e., the view frustum and all objects that can cast shadows into it). Then we enclose this body with an appropriate perspective frustum P that has a view vector parallel to the shadow map. We can choose the up vector for the coordinate frame transformation freely, but if we choose one, which lies in the plane spanned by the vectors v (view direction) and l (light direction) and is perpendicular to l (light direction), we get a simple perspective transformation in a central position. Note: The following calculations are for a right handed coordinate frame.

$$\text{up} = (l \times v) \times l$$

To complete the frame change to light space we have to translate into the eye position:

$$\begin{pmatrix} l_y \cdot \text{up}_z - l_z \cdot \text{up}_y & -l_x \cdot \text{up}_z + l_z \cdot \text{up}_x & l_x \cdot \text{up}_y - l_y \cdot \text{up}_x & 0 \\ \text{up}_x & \text{up}_y & \text{up}_z & 0 \\ -l_x & -l_y & -l_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -\text{eye}_x \\ 0 & 1 & 0 & -\text{eye}_y \\ 0 & 0 & 1 & -\text{eye}_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

The more interesting part of the transformation is the perspective part. A general mapping of the type:

$$\begin{pmatrix} \frac{x}{d+b \cdot y} \\ \frac{c+a \cdot y}{d+b \cdot y} \\ \frac{z}{d+b \cdot y} \end{pmatrix}$$

is needed. We need a perspective transformation for y that maps our input range to the output range:

$$y \in \{f, n\} \Rightarrow y' \in \{-1, 1\} \quad \frac{x}{y} = \frac{x}{d+b \cdot y} \Rightarrow b = 1 \wedge d = 0$$

$$-1 = \frac{c+a \cdot n}{n} \Rightarrow c = -n \cdot (a+1) \quad 1 = \frac{a \cdot f - n \cdot (a+1)}{f} \Rightarrow a = \frac{f+n}{f-n}$$

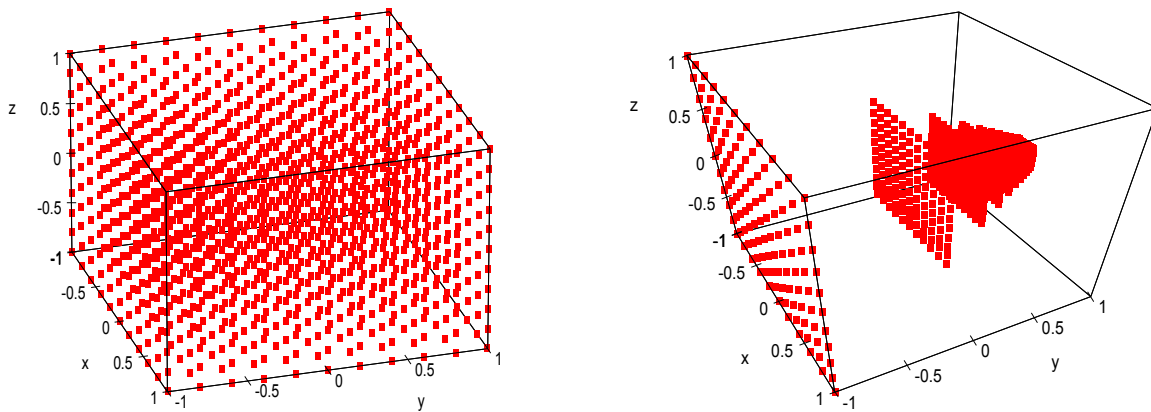
$$c = \frac{2 \cdot f \cdot n}{n - f}$$

And results in:

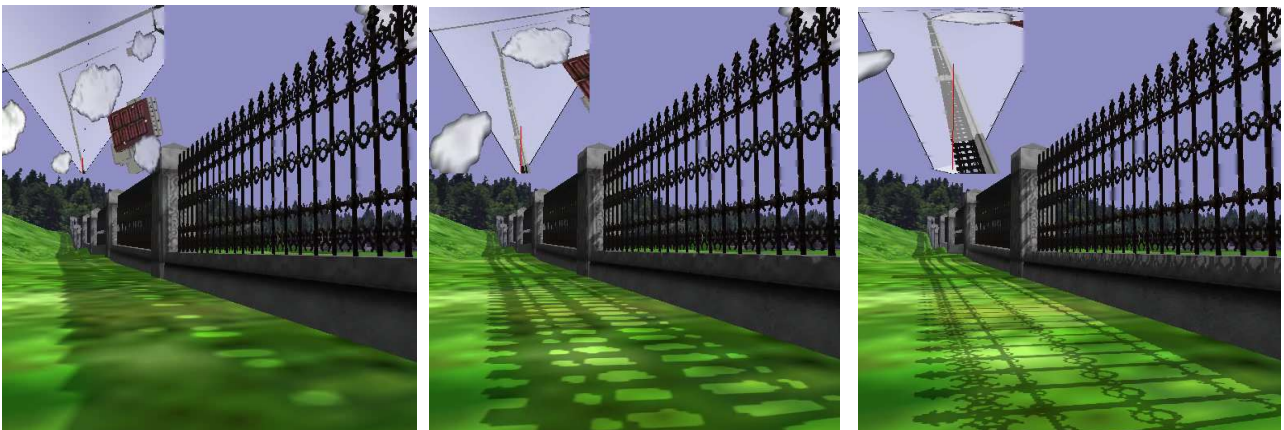
$$\begin{pmatrix} x \\ \frac{2 \cdot f \cdot n}{-f+n} + \frac{-f \cdot y - n \cdot y}{-f+n} \\ z \\ y \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{f+n}{f-n} & 0 & -\frac{2 \cdot f \cdot n}{f-n} \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

What are the properties of this perspective transformation matrix?

An equally space point cloud (left) is transformed into the point set shown to the right (n=1, f=10)



This is exactly what we wanted to achieve. If you let the light direction be the negative z-axis you can see that the near point samples are untouched and the farther away point samples are shrunk together. This is the required redistribution of the point samples. By choosing an appropriate n and f we can control this redistribution assigning more or less shadow map resolution to the near pixels. You can fix one of the two parameters and vary the other. For example the following pictures are taken with decreasing n, which gives more and more resolution to the near pixels in the shadow map:



In the upper left corner the scene is shown from the light point of view.

Is this still a directional light?

A directional light has the property that all light rays are parallel.

To show that the orthogonal light properties are preserved, we have to show that the light rays from the light source stay all parallel and pointing into the same direction after the application of the transformation. We simply have to show that for a general light ray the direction stays the same after transformation with the concatenation of the matrices, called warp matrix.

This can be shown in two (equal) ways:

1. point wise: a light ray direction vector AB is made out of two points:

$$A = [x, y, z, 1] \quad B = [x, y, z \cdot w, 1] \quad \text{ray} = B - A \quad \text{with } w \neq 0$$

we transform each point by the warp matrix and calculate the ray:

$$W \cdot B - W \cdot A =$$

$$\begin{pmatrix} 0 \\ 0 \\ z \cdot (-w + 1) \\ 0 \end{pmatrix}$$

this vector is pointing along the -z axis

2. directly with the direction:

$$\text{ray} = B - A = [0, 0, z \cdot (1 - w)] \quad \text{with } w \neq 0$$

$$W \cdot \text{ray} =$$

$$\begin{pmatrix} 0 \\ 0 \\ z \cdot (-w + 1) \\ 0 \end{pmatrix}$$

which leads to exactly the same result. the warp matrix works for points and directions and preserves the properties of a directional light source. q.e.d.

Bibliography

Will1978: Williams L., Casting Curved Shadows On Curved Surfaces, 1978,

Stam2002: Stamminger M. & Drettakis G., Perspective Shadow Maps, 2002, <http://www-sop.inria.fr/revues/publications/data/2002/SD02/PerspectiveShadowMaps.pdf>

Wimm2004: Wimmer M. & Scherzer D. & Purgathofer W., Light Space Perspective Shadow Maps, 2004, <http://www.cg.tuwien.ac.at/research/vr/lispsm>