ShaderX 8 Supplementary Materials (2009), pp. 1–6
Fan Zhang, Chong Zhao, Adrian Egli (Authors)

# Extra Note of "Visualize Your Shadow Map Techniques"

Fan Zhang†, Chong Zhao, Adrian Egli

**Abstract**

*This extra note is a supplementary material for the ShaderX 8 chapter "Visualize Your Shadow Map Techniques".*

## 1. Derivation of Aliasing Functions

This section presents the detailed derivations of Eqs. (3.1) and (3.2) in our ShaderX8 chapter.

### 1.1. Preliminaries

Consider an arbitrary point $\mathbf{c}(x,y,z)$ in the eye's coordinates system $(O;X,Y,Z)$, whose projections on the screen and shadow plane are $\mathbf{p}(q,p)$ and $\mathbf{s}(t,s)$ respectively. A small increment $\mathrm{d}\mathbf{c} = (\mathrm{d}x,\mathrm{d}y,\mathrm{d}z)$ in the view space causes a shift $\mathrm{d}\mathbf{p} = (\mathrm{d}q,\mathrm{d}p)$ on the screen, and an offset $\mathrm{d}\mathbf{s} = (\mathrm{d}t,\mathrm{d}s)$ on the shadow plane. The aliasing errors in horizontal and vertical dimensions of the screen, $\mathrm{E}_q$ and $\mathrm{E}_p$ respectively, are quantified as

$$(\mathrm{E}_q,\ \mathrm{E}_p) = \left(\left|\frac{\mathrm{d}q}{\mathrm{d}t}\right|,\ \left|\frac{\mathrm{d}p}{\mathrm{d}s}\right|\right). \tag{1}$$

In this paper, the above $E_q$ and $E_p$ are termed *aliasing functions* in shadow mapping.

Since $q$, $p$, $t$ and $s$ are all functions with respect to the view coordinates $(x,y,z)$, the differentials involved in the aliasing functions can be explicitly calculated using the *total differential equation*. For instance,

$$\mathrm{d}q = \bigtriangledown q \cdot \mathrm{d}\mathbf{c} \tag{2}$$

where $\bigtriangledown = (\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z})$ and $|\cdot|$ denotes the dot product of vectors. For given scene configuration, the normalized screen coordinates $(q,p)$ are determined by the eye's projection transformation. Without loss of generality, in our analysis, we assume the field-of-views (FOVs) of the view frustum on both X and Y directions are $2\phi$. In Direct3D (similar in OpenGL), the normalized device coordinates $(q,p)$ are transformed from the view coordinates $(x,y,z)$ using the following perspective projection transform,

$$(q,\ p) = (\frac{1}{\tan\phi}\frac{x}{z},\ \frac{1}{\tan\phi}\frac{y}{z})$$

From Eq. (2), the differentials of screen coordinates are thus

$$\bigtriangledown q = \left(\frac{1}{\tan\phi}\frac{1}{z},\ 0,\ -\frac{1}{\tan\phi}\frac{x}{z^2}\right) \tag{3}$$

$$\bigtriangledown p = \left(0,\ \frac{1}{\tan\phi}\frac{1}{z},\ -\frac{1}{\tan\phi}\frac{y}{z^2}\right) \tag{4}$$

Substituting Eqs. (3) and (4) into Eq. (1),

$$(E_q,\ E_p) = \frac{1}{\tan\phi}\frac{1}{z}\left(\left|\frac{dx}{dt}\left(1-\frac{x}{z}\frac{dz}{dx}\right)\right|,\ \left|\frac{dy}{ds}\left(1-\frac{y}{z}\frac{dz}{dy}\right)\right|\right) \tag{5}$$

For the texture coordinates $t$ and $s$ in the above equation, the derivation of the explicit representations $t(x,y,z)$ and $s(x,y,z)$ is relatively complicated. The complexity mostly depends on how the texture transformation is designed when reparameterizing the shadow map.

### 1.2. Overview

To facilitate our analysis, the frequently used notations are listed in Table 1.

| | |
|---|---|
| $V$ | view frustum |
| $W$ | warping frustum |
| $\Theta$ | light vector |
| $\theta$ | angle between view and light directions |
| $n$ and $f$ | near- and far- plane values for $V$ |
| $\lambda$ | near plane value for $W$ |
| $\mu$ | depth range of $W$ |
| $\phi$ and $\phi_w$ | half FOV of $V$ and half FOV of $W$ |
| $(O;X,Y,Z)$ | eye's coordinates frame |
| $(O_w;X_w,Y_w,Z_w)$ | $W$'s coordinates frame |
| $(q,\ p)$ | normalized screen coordinates in 2D |
| $(t,\ s)$ | texture coordinates in 2D |
| $(E_q,\ E_p)$ | aliasing functions |

**Table 1:** *Notations.*

In this paper, we adopt the DirectX transformation matrices in our analysis: vectors are row-based and the coordinates system is left-handed. The coordinates for vectors (e.g.

† Corresponding Author: andy.fzhang@gmail.com

$\Theta$) and 3D points (e.g. $O_w$) are defined in the eye's coordinates system $(O;X,Y,Z)$, which means the texture matrix in shadow mapping transforms points from the view space to the texture space.

### 1.3. Local Representation of Perspective Aliasing

To approximate the aliasing functions represented by Eq. (5), the following assumptions are used in previous reparameterizations [SD02] [WSP04] [LTYM06] [ZXTS06] [ZSZW08].

$$1-\frac{x}{z}\frac{\mathrm{d}z}{\mathrm{d}x}\equiv\text{const.}\quad 1-\frac{y}{z}\frac{\mathrm{d}z}{\mathrm{d}y}\equiv\text{const.}\quad \frac{\mathrm{d}y}{\mathrm{d}z}\equiv\text{const.}\qquad(6)$$

To satisfy the above assumptions, previous work only considers the points $\mathbf{c}(0,0,z)$ and restricts the small increment $\mathrm{d}\mathbf{c}$ moving on small lines $\{\mathbf{l}: y=kz+b\}$, where $k$ is a constant slope. With the assumptions, the aliasing functions are simplified as

$$E_q\sim\frac{1}{z}\left|\frac{\mathrm{d}x}{\mathrm{d}t}\right|\quad\text{and}\quad E_p\sim\frac{1}{\tan\phi}\frac{1}{z}\left|\frac{\mathrm{d}z}{\mathrm{d}s}\frac{\mathrm{d}y}{\mathrm{d}z}\right|\sim\frac{1}{z}\left|\frac{\mathrm{d}z}{\mathrm{d}s}\right|\qquad(7)$$

where $\sim$ stands for the linear dependence.

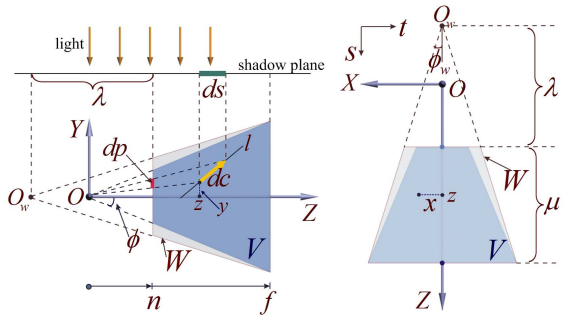


**Figure 1:** *Perspective reparameterizations in the ideal case. Left and right are the side view and the light's view respectively.*

For the overhead light on the YZ plane in Fig. 1, perspective shadow maps are generated in the post-perspective space of the warping transform $W$. The texture coordinates in standard shadow maps (SSMs) are computed by normalizing the projective coordinates into $[0,1]\times[0,1]$,

$$t(x,z)=\frac{1}{2}\frac{1}{\tan\phi_w}\frac{x}{z}+\frac{1}{2}\quad\text{and}\quad s(z)=\frac{\lambda+\mu}{\mu}\left(1-\frac{\lambda}{z}\right)$$

A simple analysis of this scenario gives

$$\mu=f-n\quad\text{and}\quad\frac{\tan\phi_w}{\tan\phi}=\frac{f}{\mu+\lambda}$$

Combining them together and replacing $\mathrm{d}x/\mathrm{d}t$ in Eq. (7) with $(\partial t/\partial x)^{-1}$, we get the following explicit representations of aliasing functions in the ideal case (note that $\approx$ occurs when approximating $\mathrm{d}x/\mathrm{d}t$ with $(\partial t/\partial x)^{-1}$).

$$E_q\approx\frac{2f}{f-n+\lambda}\frac{z-n+\lambda}{z}\qquad(8)$$

$$E_p=\frac{f-n}{(f-n+\lambda)\lambda\tan\phi}\frac{(z-n+\lambda)^2}{z}\qquad(9)$$

Eqs. (8) and (9) are termed the *local representation* of aliasing functions. In variant reparameterizations, the free parameter $\lambda$ is adjusted to control the aliasing distribution over the whole depth range. SSM is a special perspective reparameterization using a frustum with $\lambda_{\mathbf{SSM}}=\infty$. Hence,

$$E_q^{\mathbf{SSM}}\sim 1/z\quad\text{and}\quad E_p^{\mathbf{SSM}}\sim 1/z$$

It means the aliasing errors increase hyperbolically as the object moves closer to the view plane. On the contrary, PSM warps the scene and light using the view frustum, i.e. $\lambda_{\mathbf{PSM}}=n$, such that

$$E_q^{\mathbf{PSM}}\sim 1\quad\text{and}\quad E_p^{\mathbf{PSM}}\sim z$$

The linear aliasing distribution of $E_p$ dramatically improves the shadow qualities in the area near the viewer.

**Limitations:** The local representation clearly explained the motivation of perspective parameterizations. However, a few limitations still remaine due to the assumptions used in Eq. (6). Since the assumptions are satisfied only for the points on the view direction, there's no explicit way to quantitatively analyze the aliasing elsewhere. Furthermore, the local representation is only valid for the ideal case where $\theta=\pi/2$. When the light or viewer moves in the general case, the local representations can not guide us to adaptively select the appropriate warping transform according to the application-specific requirement.

### 1.4. Factorization of Aliasing Functions

To explicitly represent the aliasing functions in the general case, we consider the differential $\mathrm{d}\mathbf{c}=(\mathrm{d}x,\mathrm{d}y,\mathrm{d}z)$ on a small shaft $\mathbf{l}$. The small shaft $\mathbf{l}$ is parameterized with $x=k_xz-b_x$ and $y=k_yz-b_y$, where $k_x$ and $k_y$ denote the slopes of the line on the XZ and YZ planes respectively.

From Eqs. (2) and (5), the aliasing functions are parameterized into

$$(E_q,\,E_p)=\left(\frac{\frac{1}{\tan\phi}\frac{1}{z}\left(k_x-\frac{x}{z}\right)}{\frac{\partial t}{\partial x}k_x+\frac{\partial t}{\partial y}k_y+\frac{\partial t}{\partial z}},\ \frac{\frac{1}{\tan\phi}\frac{1}{z}\left(k_y-\frac{y}{z}\right)}{\frac{\partial s}{\partial x}k_x+\frac{\partial s}{\partial y}k_y+\frac{\partial s}{\partial z}}\right)\qquad(10)$$

Perspective reparameterizations approximate the aliasing functions based on an important observation: the unknown coefficients $(k_x,k_y)$ depend on local geometry details. Any reparameterization at a global scale cannot reduce this type of aliasing everywhere. For a light direction $\Theta$, a practical solution is to factorize the aliasing functions in Eq. (10) into

$$E(\mathbf{c},\Theta,\mathbf{l})=E^{\text{pers}}(\mathbf{c},\Theta)\times E^{\text{proj}}(\mathbf{c},\Theta,\mathbf{l})\qquad(11)$$

where $E\in\{E_q,E_p\}$. Shadow map under-sampling can

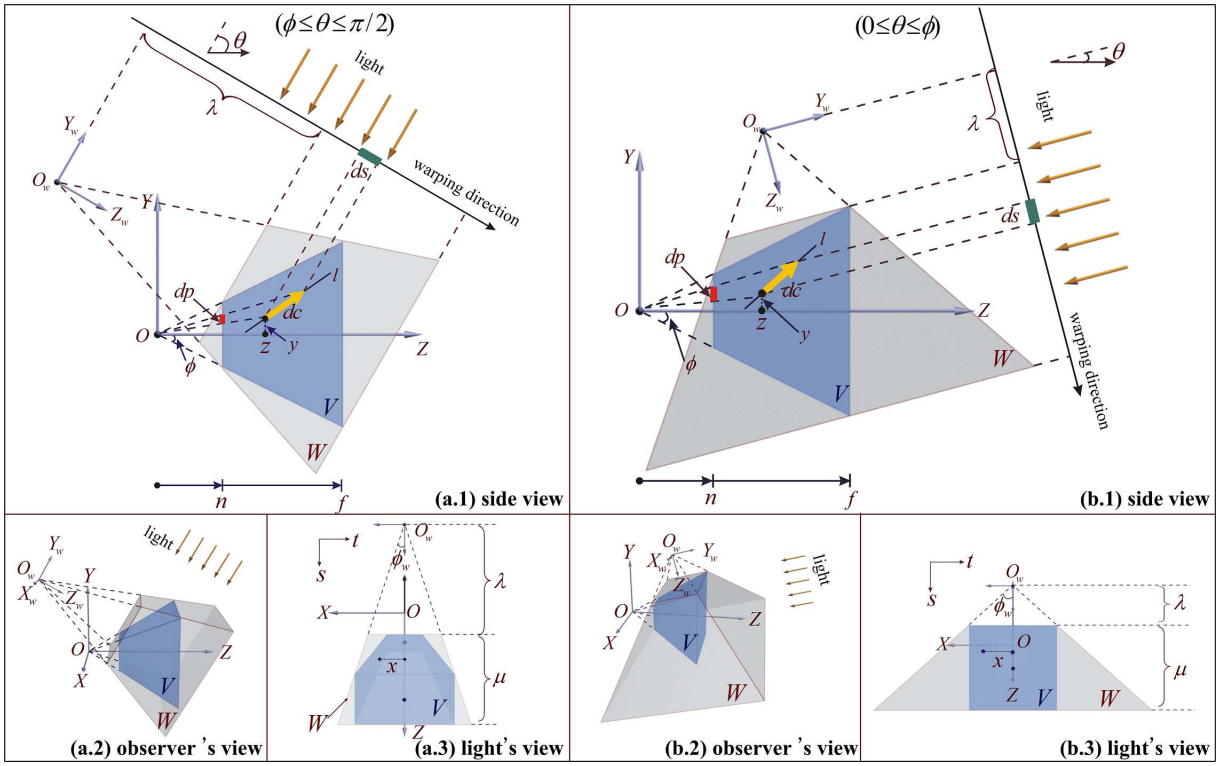


**Figure 2:** *Construction of the warping frustum $W$ when $\phi \leq \theta \leq \pi/2$ (from **a.1** to **a.3**) and $0 \leq \theta \leq \phi$ (from **b.1** to **b.3**).*

happen when *perspective aliasing errors* $E^{\text{pers}}$ or *projection aliasing errors* $E^{\text{proj}}$ becomes large. Projection aliasing usually happens for surfaces almost parallel to the light direction. Since projection aliasing depends on the geometrical details, the local increase of sampling densities on these surfaces is needed to reduce this type of aliasing. An inevitably expensive scene analysis at each frame is required such that using hardware-acceleration is impractical. On the other hand, perspective aliasing comes from the perspective foreshortening effect and can be reduced by warping the shadow-map. In this paper, we address the reduction of perspective aliasing errors.

### 1.5. Simplifications

As we mentioned before, the computation for the explicit representations of texture coordinates $s(x,y,z)$ and $t(x,y,z)$ is relatively complicated. The complexity is from two aspects:

**Type of the light source:** the texture matrix for point lights contains the light's perspective projection transform. It makes the mapping from view coordinates to texture coordinates not intuitive. Even a preliminary result was presented for point lights using the local representation [LGT*06], the comprehensive analysis in the general case remains challenging. In this paper, we only consider directional light sources.

**Warping direction:** the selection of the warping direction strongly influences the implementation complexity, because the type of light might be frequently switched between directional and point in the post-transformed space. When an inappropriate warping direction is selected, mapping singularities might be produced such that the aliasing analysis is difficult. Since any arbitrary perspective transformation can be used to warp the distribution of shadow map texels, it is sufficient to use a warping direction that's not perpendicular to the view direction like PSMs. This observation inspires Wimmer et al. [WSP04] to use a warping direction in parallel with the shadow plane, and construct the warping frustum in the light space. The main advantage of using this warping direction is that the direction of the light source doesn't change in the post-transformed space, thus no mapping singularities are generated. It greatly simplifies the implementation and aliasing analysis. In this paper, our computational model adopted the same warping direction.

### 1.6. Global Representation of Perspective Aliasing

With the simplifications in subsection 1.5, the warping frustum $W$ is constructed as shown in Fig. 2. The $Z_w$ and $Y_w$ axes are parallel to the shadow plane and light vector respectively. The near and far planes of $W$ bound the view frustum $V$. The projection reference point $O_w$ of the warping frustum is determined by the $\lambda$ selection in variant reparameterizations. A important assumption we use here is that the movement of the light is confined on the YZ plane. Without using this

assumption, representing texture coordinates $(s,t)$ as a function of $(x,y,z)$ seems impractical.

We only need to consider the light directions from the upper hemisphere over $O$ due to symmetry. Since $\theta$ can be replaced by $\pi-\theta$ in our computation when $\theta > \pi/2$, we suppose $0 \le \theta \le \pi/2$ in our analysis from now on. The shadow-map reparameterization $(t,\ s)$ is induced by applying the perspective projection transform $W$ to both the scene and light. Each point $(x,y,z)$ in the view space is transformed to $(x_w,y_w,z_w)$ in the coordinates system $(O_w;X_w,Y_w,Z_w)$, and then projected to $(x_w^c,y_w^c,z_w^c)$ in $W$'s post-perspective space. Finally, $(x_w^c,z_w^c)$ is normalized into $[0,1]\times[0,1]$ to output the texture coordinates below.

$$\mathbf{s} = (t,\ s) = \left(\frac{1}{2}\frac{x_w}{z_w \tan\phi_w} + \frac{1}{2},\ \frac{\lambda+\mu}{\mu}\left(1-\frac{\lambda}{z_w}\right)\right) \tag{12}$$

In Fig. 2, the light views (right column) tells us

$$x_w = x \quad \text{and} \quad \tan\phi_w = \begin{cases} \frac{f\tan\phi}{\mu+\lambda} & \theta \in [\phi, \frac{\pi}{2}] \\ \frac{f\tan\phi}{\lambda} & \theta \in [0,\phi] \end{cases}$$

The side views (middle column) gives

$$\mu = \begin{cases} (n\frac{\tan\phi}{\tan\theta} + f - n + f\frac{\tan\phi}{\tan\theta})\sin\theta & \theta \in [\phi, \frac{\pi}{2}] \\ 2f\tan\phi\cos\theta & \theta \in [0,\phi] \end{cases} \tag{13}$$

and

$$z_w = \begin{cases} (z - n + n\frac{\tan\phi}{\tan\theta} - \frac{y}{\tan\theta})\sin\theta + \lambda & \theta \in [\phi, \frac{\pi}{2}] \\ \left(f\tan\phi - (f-z)\tan\theta - y\right)\cos\theta + \lambda & \theta \in [0,\phi] \end{cases}$$

The explicit expressions of $(t,s)$ are then obtained by substituting the above $x_w$, $\mu$ and $z_w$ into Eq. (12). Finally we derive the following factorizations of aliasing functions using Eq. (10).

$$E_q = E_q^{\text{pers}} \times E_q^{\text{proj}} = \left(\frac{2\tan\phi_w}{\tan\phi}\frac{z_w}{z}\right) \times \left(\frac{k_x - \frac{x}{z}}{k_x - \frac{x}{z_w}\frac{\partial z_w}{\partial y}k_y - \frac{x}{z_w}\frac{\partial z_w}{\partial z}}\right)$$

$$E_p = E_p^{\text{pers}} \times E_p^{\text{proj}} = \left(\frac{1}{\tan\phi}\frac{\mu}{(\lambda+\mu)\lambda}\frac{z_w^2}{z}\right) \times \left(\frac{k_y - \frac{y}{z}}{\frac{\partial z_w}{\partial y}k_y + \frac{\partial z_w}{\partial z}}\right)$$

or

$$E_q^{\text{pers}} = \frac{2\tan\phi_w}{\tan\phi}\frac{z_w}{z} \quad \text{and} \quad E_q^{\text{proj}} = \left|\frac{k_x - \frac{x}{z}}{k_x - \frac{x}{z_w}\left(\frac{\partial z_w}{\partial y}k_y + \frac{\partial z_w}{\partial z}\right)}\right|$$

$$E_p^{\text{pers}} = \frac{1}{\tan\phi}\frac{\mu}{(\lambda+\mu)\lambda}\frac{z_w^2}{z} \quad \text{and} \quad E_p^{\text{proj}} = \left|\frac{k_y - \frac{y}{z}}{\frac{\partial z_w}{\partial y}k_y + \frac{\partial z_w}{\partial z}}\right|$$

To facilitate our analysis of perspective aliasing, we introduce two auxiliary functions $F$ and $G$ to further factorize $E_q^{\text{pers}}$ and $E_p^{\text{pers}}$ into

$$E_q^{\text{pers}} = F_q(\lambda,\theta)\frac{z+G(y,\lambda,\theta)}{z}$$

and

$$E_p^{\text{pers}} = F_p(\lambda,\theta)\frac{\left(z+G(y,\lambda,\theta)\right)^2}{z}$$

where

$$G(y,\lambda,\theta) = \begin{cases} \frac{\lambda}{\sin\theta} - n + \frac{n\tan\phi - y}{\tan\theta} & \theta \in [\phi, \frac{\pi}{2}] \\ \frac{\lambda}{\sin\theta} - f + \frac{f\tan\phi - y}{\tan\theta} & \theta \in [0,\phi] \end{cases}$$

$$F_q(\lambda,\theta) = \begin{cases} \frac{2f\sin\theta}{\lambda + (f-n)(1-\frac{\tan\phi}{\tan\theta})\sin\theta} & \theta \in [\phi, \frac{\pi}{2}] \\ \frac{2f\sin\theta}{\lambda} & \theta \in [0,\phi] \end{cases}$$

$$F_p(\lambda,\theta) = \begin{cases} \frac{\sin^2\theta\left(f-n+(f+n)\frac{\tan\phi}{\tan\theta}\right)}{\tan\phi\lambda\left(\lambda+\left(f-n+(f+n)\frac{\tan\phi}{\tan\theta}\right)\sin\theta\right)} & \theta \in [\phi, \frac{\pi}{2}] \\ \frac{2f\sin^2\theta}{\lambda(\lambda+2f\tan\phi\cos\theta)\tan\phi} & \theta \in [0,\phi] \end{cases}$$

By ignoring projection aliasing errors, the aliasing functions can be approximated by

$$E_q \approx E_q^{\text{pers}} = F_q(\lambda,\theta)\frac{z+G(y,\lambda,\theta)}{z} \tag{14}$$

$$E_p \approx E_p^{\text{pers}} = F_p(\lambda,\theta)\frac{\left(z+G(y,\lambda,\theta)\right)^2}{z} \tag{15}$$

Eqs. (14) and (15) are termed the *global representation* of aliasing errors in perspective reparameterizations. The global representation extends the local representation to all points within the view frustum without requiring the assumptions shown in Eq. (6). Given an arbitrary point $(x,y,z)$, the global representation can quantitatively evaluate the aliasing error at this point. On the contrary, as we mentioned in subsection 1.3, the local representation is only able to quantify the aliasing errors at points with the form $(0,0,z)$. More importantly, the global representation is derived based on an analysis of the general case, such that the aliasing distribution remains analyzable for dynamic lights/viewer.

**Aliasing metrics:** We considered three kinds of aliasing metrics: $E_p(z,y,\lambda,\theta)$, $E_q(z,y,\lambda,\theta)$ and $E_{p\times q}(z,y,\lambda,\theta) = E_p \times E_q$. $E_p$ and $E_q$ quantify the aliasing in the vertical and horizontal dimensions of the screen independently, while $E_{p\times q}$ is an aggregate measure of the aliasing errors in both dimensions. In the factorization of each metric, the part of projection aliasing errors is ignored.

## 2. Derivation of $\lambda$ Values

**Impoartant Notes: For better illustration, the $\lambda$ value shown in Figure 6 in the paper is actually $\lambda/\mu$ instead. This *normalization* makes the value only depends on the ratio $f/n$ rather than the absolute values of $n$ and $f$.**

### 2.1. LiSPSM

In LiSPSM [WSP04], $\lambda^*_{\text{LiSPSM}} = n + \sqrt{fn}$ in the ideal case satisifies the condition $E_p|_{z=n,\theta=\pi/2} = E_p|_{z=f,\theta=\pi/2}$. For general $\theta$ values, the authors suggest to use the approximation $\lambda(\theta) = \lambda^*_{\text{LiSPSM}}/\sin\theta$. Therefore,

$$\lambda_{\text{LiSPSM}} = \frac{(n+\sqrt{fn})}{\sin\theta}$$

### 2.2. TSM

The "$\kappa-\xi$" rule in TSM maps the focus point $\mathbf{c}_\kappa(0,0,z_\kappa)$ in the view space to the point $(0,\xi)$ in the *trapezoidal space* [MT04], where $z_\kappa = n+(f-n)\kappa$. Plugging Eq. (12) into $s(z_\kappa)=\xi$ gives

$$\lambda_{\text{TSM}} = \begin{cases} \frac{(f-n)\kappa(1-\xi)+n(1-\xi)\frac{\tan\phi}{\tan\theta}}{(f-n)(\xi-\kappa)+(f\xi-n(1-\xi))\frac{\tan\phi}{\tan\theta}}\mu & \theta\in[\phi,\frac{\pi}{2}] \\ \frac{(f-n)(\kappa-1)(1-\xi)\frac{\tan\theta}{\tan\phi}+(1-\xi)f}{(f-n)(1-\kappa)\frac{\tan\theta}{\tan\phi}+(2\xi-1)f}\mu & \theta\in[0,\phi] \end{cases} \quad (16)$$

Refer to Eq. (13) for the representation of $\mu$. For the typical setting $\xi=0.8$ and $\kappa=0.5$, the above equation can be simplified into

$$\lambda_{\text{TSM}} = \begin{cases} \frac{f-n+2n\frac{\tan\phi}{\tan\theta}}{3(f-n)+(8f-2n)\frac{\tan\phi}{\tan\theta}}\mu & \theta\in[\phi,\frac{\pi}{2}] \\ \frac{2f-(f-n)\frac{\tan\theta}{\tan\phi}}{5(f-n)\frac{\tan\theta}{\tan\phi}+6f}\mu & \theta\in[0,\phi] \end{cases}$$

To better handle the shadow quality in degenerate cases, the TSM authors propose an anternative implementation which will be explained in the next section. For clarity, our ShaderX 8 paper only consider the above $\lambda_{\text{TSM}}$.

## 3. An Alternative TSM Implementation in Degenerate Cases

To simplify our discussion in this section, we consider the problem in the normalized warping space where $\mu=1$. Eq. (16) implies

$$\lim_{\theta\to 0}\lambda_{\text{TSM}} = \lim_{\theta\to 0}\lambda^{(2)}_{\text{TSM}}(\theta) = \frac{1-\xi}{2\xi-1} \neq \infty.$$

Note that any perspective reparameterization should converge to SSM as $\theta$ goes to zero, i.e. $\lim_{\theta\to 0}\lambda_{\text{TSM}}=\infty$. But the above equation does not follow this rule which should be obeyed by all perspective reparameterizations. In the dueling frusta case $\theta=0$, such $\lambda_{\text{TSM}}$ warps the focus region into very few texels in the shadow map. The result is that a large portion of shadow map resolution is not utilized. From Eq. (16), we know $\lambda_{\text{TSM}}=1/3$ at $\theta=0$ for the typical setting $\kappa=0.5$ and $\xi=80\%$. Substitute this $\lambda$ value into Eq. (12) gives $s(1/4)\approx 4/7$. Fig. 3 illustrates this result in which the first 4/7 fraction of the shadow map resolution is wasted for the distant region.

Due to the problem shown in Fig. 3, the TSM authors suggest using an alternative warping transformation to improve the shadow quality in degenerate cases. Instead of satisfying the "$\kappa$-$\xi$ *rule*", this alternative transformation maximizes the size of the transformed focus region in the trapezoidal space.

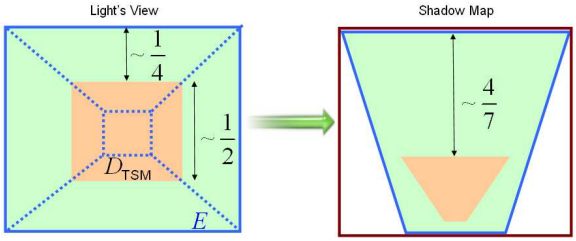


**Figure 3:** *Deformation of the focus region in TSM at $\theta=0$. Left: the focus region $D_{\text{TSM}}$ occupies $\sim 1/2$ of the shadow map in the light space. Right: the first $\sim 4/7$ fraction of the shadow map resolution is wasted on the distant region. This plot uses the setting $\kappa=0.5$ and $\xi=80\%$.*

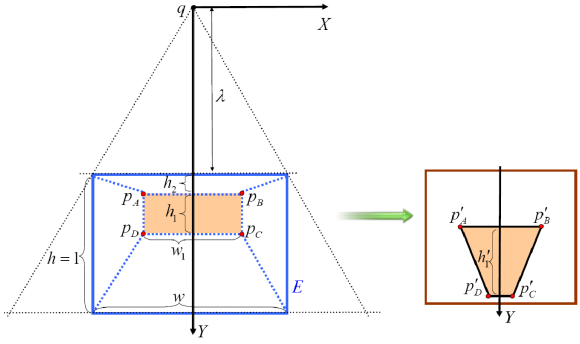


**Figure 4:** *Alternative TSM implementation in degenerate cases. Left: light space. Right: trapezoidal space.*

We now explain the alternative solution of TSM actually follows the rule $\lim_{\theta\to 0}\lambda_{\text{TSM}}=\infty$. In Fig. 4, the rectangle $p_Ap_Bp_Cp_D$ represents the projection of the focus region $D_{\text{TSM}}$ on the shadow map. The coordinates of the four vertices are listed below,

$$\begin{aligned} p_A &= (-w_1/2, \quad \lambda+h_2) \\ p_B &= (w_1/2, \quad \lambda+h_2) \\ p_C &= (w_1/2, \quad \lambda+h_1+h_2) \\ p_D &= (-w_1/2, \quad \lambda+h_1+h_2) \end{aligned}$$

All notations are self-explained in Fig. 4. We consider the problem in the normalized space (i.e. $h=1$). The rectangle $p_Ap_Bp_Cp_D$ is transformed into $p'_Ap'_Bp'_Cp'_D$ in the trapezoidal space. Using the standard projection matrix in Direct3D, the

coordinates of transformed vertices $p'_A$, $p'_B$, $p'_C$ and $p'_D$ are

$$\begin{aligned}
p'_A &= \left(-\frac{\lambda}{\lambda+h_2}\frac{w_1}{w}, \quad (\lambda+1)\frac{h_2}{\lambda+h_2}\right)\\
p'_B &= \left(\frac{\lambda}{\lambda+h_2}\frac{w_1}{w}, \quad (\lambda+1)\frac{h_2}{\lambda+h_2}\right)\\
p'_C &= \left(\frac{\lambda}{\lambda+h_1+h_2}\frac{w_1}{w}, \quad (\lambda+1)\frac{h_1+h_2}{\lambda+h_1+h_2}\right)\\
p'_D &= \left(-\frac{\lambda}{\lambda+h_1+h_2}\frac{w_1}{w}, \quad (\lambda+1)\frac{h_1+h_2}{\lambda+h_1+h_2}\right)
\end{aligned}$$

$h_1$, the depth range covered by the focus region, is transformed to $h'_1$ as the following,

$$h'_1 = (\lambda+1)\left(\frac{\lambda}{\lambda+h_2}-\frac{\lambda}{\lambda+h_1+h_2}\right)$$

The size of $D_{\mathrm{TSM}}$ in the trapezoidal space $\mathrm{S}_{p'_Ap'_Bp'_Cp'_D}$ is then

$$\begin{aligned}
\mathrm{S}_{p'_Ap'_Bp'_Cp'_D} &= (p'_Ap'_B+p'_Cp'_D)h'_1/2\\
&= 4\frac{w_1}{w}h_1\frac{\lambda^2(\lambda+1)(\lambda+\frac{h_1}{2}+h_2)}{(\lambda+h_2)^2(\lambda+h_1+h_2)^2}\\
&\le 4\frac{w_1}{w}h_1\frac{\lambda^2(\lambda+1)(\lambda+\frac{h_1}{2}+h_2)}{\lambda^2(\lambda+h_1+2h_2)^2}
\end{aligned}$$

where $w_1/w$ is a constant, $h_1$ is also a constant for a given $\theta$ value, and $h_1+2h_2\le 1$ when $0\le\theta\le\phi$. Let $2a=h_1+2h_2$, we have

$$\mathrm{S}_{p'_Ap'_Bp'_Cp'_D} \le 4\frac{w_1}{w}h_1\frac{(\lambda+1)(\lambda+a)}{(\lambda+2a)^2} \le 4\frac{w_1}{w}h_1\frac{(\frac{\lambda+1+\lambda+a}{2})^2}{(\lambda+2a)^2}$$

$$= 4\frac{w_1}{w}h_1\left(\frac{\lambda+\frac{a+1}{2}}{\lambda+2a}\right)^2.$$

Obviously, we have $\lim_{\theta\to 0}a=1/2$ and $(a+1)/2\le 2a$ when $a\ge 1/3$. Therefore, small $\theta$ values give

$$\mathrm{S}_{p'_Ap'_Bp'_Cp'_D} \le 4\frac{w_1}{w}h_1$$

To maximize $\mathrm{S}_{p'_Ap'_Bp'_Cp'_D}$, $\lambda=\infty$ should be satisfied such that $\mathrm{S}_{p'_Ap'_Bp'_Cp'_D}$ achieves the upper bound $4w_1h_1/w$. It proves that the alternative implementation of TSM still follows the rule $\lim_{\theta\to 0}\lambda=\infty$.

## References


[LGT*06] LLOYD B., GOVINDARAJU N. K., TUFT D., MOLNAR S., MANOCHA D.: Practical logarithmic shadow maps. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches* (New York, NY, USA, 2006), ACM Press, p. 103.

[LTYM06] LLOYD B., TUFT D., YOON S., MANOCHA D.: Warping and partitioning for low error shadow maps. In *Proceedings of the Eurographics Symposium on Rendering 2006* (2006), Eurographics Association, pp. 215–226.

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

[SD02] STAMMINGER M., DRETTAKIS G.: Perspective shadow maps. In *Proceedings of SIGGRAPH '02* (New York, NY, USA, 2002), ACM Press, pp. 557–562.

[WSP04] WIMMER M., SCHERZER D., PURGATHOFER W.: Light space perspective shadow maps. In *the Eurographics Symposium on Rendering 2004* (2004), Eurographics, Eurographics Association.

[ZSZW08] ZHANG F., SUN H., ZHAO C., WANG L.: Generalized minimum-norm perspective shadow maps. *Comput. Animat. Virtual Worlds 19*, 5 (2008), 553–567.

[ZXTS06] ZHANG F., XU L., TAO C., SUN H.: Generalized linear perspective shadow map reparameterization. In *VRCIA '06: Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications* (New York, NY, USA, 2006), ACM Press, pp. 339–342.