Improved Geometric Specular Antialiasing
(Supplemental Document)

Yusuke Tokuyoshi
SQUARE ENIX CO., LTD.
tokuyosh@square-enix.com

Anton S. Kaplanyan
Facebook Reality Labs
kaplanyan@fb.com

1 Non-Axis-Aligned Anisotropic BRDF

Shadowing-masking Function. The Smith masking function [Smi67] is defined as
\[ G_1(i, h) = \frac{\chi^+(i \cdot h)}{1 + \Lambda(i)} \]
where \( \chi^+(i \cdot h) \) is the Heaviside function: 1 if \( i \cdot h > 0 \) otherwise 0. \( \Lambda(i) \) is a function which depends on the NDF model. The height-correlated masking-shadowing function [Hei14] is given as
\[ G_2(i, o) = \frac{\chi^+(i \cdot h) \chi^+(o \cdot h)}{1 + \Lambda(i) + \Lambda(o)}. \]

In this paper, \( \Lambda(o) \) for the anisotropic GGX NDF model is described in the later paragraphs.

Axis-aligned Anisotropic GGX BRDF The axis-aligned anisotropic GGX NDF is defined as follows:
\[ D(h) = \frac{\chi^+(h_z)}{\pi^{\alpha_x \alpha_y} (\frac{h_x^2}{\alpha_x^2} + \frac{h_y^2}{\alpha_y^2} + h_z^2)^{\frac{3}{2}}}. \]

For this NDF, the masking-shadowing function is obtained using the following function:
\[ \Lambda(o) = -0.5 + \sqrt{\frac{\alpha_x^2 o_x^2 + \alpha_y^2 o_y^2 + o_z^2}{2|o_z|}}, \]
where \([o_x, o_y, o_z] \) is the outgoing direction \( o \) in tangent space.

Non-axis-aligned Anisotropic GGX BRDF For shading antialiasing, we use the \( 2 \times 2 \) roughness matrix \( A \) instead of \( \alpha_x \) and \( \alpha_y \). The anisotropic NDF can be generalized using this matrix [Hei14] as follows:
\[ D(h) = \frac{\chi^+(h_z)}{\pi \sqrt{\det(A)} (\langle [h_z, h_y] A^{-1} [h_z, h_y]^T, h_z^2 \rangle)^{\frac{3}{2}}}. \]
For this NDF, the masking-shadowing function is obtained using the following function:

$$\Lambda(o) = -0.5 + \sqrt{\langle o_x, o_y \rangle A [o_x, o_y]^T + o_z^2} / |o_z|.$$ 

For this microsurface model, the slope of a microsurface is stretched in the directions of the eigenvectors of the roughness matrix $A$. The stretching scale for each eigenvector is the reciprocal square root of the eigenvalue of $A$.

**Practical Implementation.** The determinant $\det(A)$ can produce a large precision error due to floating point arithmetic, especially when using an elongated kernel for NDF filtering. To improve the numerical stability, this paper clamps $\det(A)$ by a small value $\tau$ for NDF:

$$D(h) = \frac{\chi^+(h_z)}{\pi \sqrt{\max(\det(A), \tau)}} (\langle h_x, h_y \rangle A^{-1} [h_x, h_y]^T + h_z^2)^2.$$ 

To compute $A^{-1}$, we also use this clamped determinant as follows:

$$A^{-1} = \frac{\text{adj}(A)}{\max(\det(A), \tau)}.$$ 

For NDF filtering, since $\sqrt{\det(A)}$ must be equal or greater than the original squared roughness parameter, we use $\tau = \alpha_x^2 \alpha_y^2$.

### 2 Derivation of the Jacobian Matrix

Let $\psi_x$ be an angle on the great circle passing through the halfvector $h$ and normal $n$, and $\psi_y$ be an angle on the great circle passing through the halfvector $h$ and $\frac{h \times n}{\|n \times h\|}$: then its Cartesian coordinate is given as

$$m_x = \cos \psi_y \sin \psi_x,$$
$$m_y = \sin \psi_y,$$
$$m_z = \cos \psi_y \cos \psi_x.$$ 

(1)

Thus, the Jacobian matrix of the transformation from $[\psi_x, \psi_y]$ to $[m_x, m_y]$ at $\psi_x = 0$ and $\psi_y = 0$ is yielded as

$$J_{\psi \rightarrow m} = \begin{bmatrix} \frac{\partial m_x}{\partial \psi_x} & \frac{\partial m_x}{\partial \psi_y} \\ \frac{\partial m_y}{\partial \psi_x} & \frac{\partial m_y}{\partial \psi_y} \end{bmatrix} = \begin{bmatrix} \cos \psi_y \cos \psi_x & -\sin \psi_y \sin \psi_x \\ 0 & \cos \psi_y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. $$

(2)
The tangent-space halfvector can be represented using a polar coordinate system \([\theta, \phi]\). Using this \(\theta\) and this \(\phi\), the rotation from the local-space halfvector to tangent-space halfvector is given by

\[
\begin{bmatrix}
h_x \\
h_y \\
h_z
\end{bmatrix} =
\begin{bmatrix}
\cos \theta \cos \phi & -\sin \phi & \sin \theta \cos \phi \\
\cos \theta \sin \phi & \cos \phi & \sin \theta \sin \phi \\
-\sin \theta & 0 & \cos \theta
\end{bmatrix}
\begin{bmatrix}
m_x \\
m_y \\
m_z
\end{bmatrix},
\]

where \([m_x, m_y, m_z] = [0, 0, 1]\) (i.e., \(\psi_x = 0\) and \(\psi_y = 0\)). Therefore, the Jacobian matrix of the orthographic projection is derived as

\[
J_{o \rightarrow \perp} = J_{\perp \rightarrow \perp} J_{o \rightarrow \perp} = \begin{bmatrix}
\frac{\partial h_x}{\partial m_x} & \frac{\partial h_x}{\partial m_y} \\
\frac{\partial h_y}{\partial m_x} & \frac{\partial h_y}{\partial m_y}
\end{bmatrix}
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
= \begin{bmatrix}
\cos \theta \cos \phi & -\sin \phi \\
\cos \theta \sin \phi & \cos \phi
\end{bmatrix}
= \frac{1}{\sqrt{1 - h_z^2}} \begin{bmatrix}
h_x h_z \\
h_y h_z
\end{bmatrix}.
\]

The slope of the halfvector is given as

\[
\begin{bmatrix}
h \parallel_x \\
h \parallel_y
\end{bmatrix} = \begin{bmatrix}
\frac{h_x}{\sqrt{1 - h_x^2 - h_y^2}} - \frac{h_y}{\sqrt{1 - h_x^2 - h_y^2}}
\end{bmatrix}.
\]

Therefore, the Jacobian matrix of the transformation from the projected unit disk to slope space is yielded as follows:

\[
J_{\perp \rightarrow \parallel} = \begin{bmatrix}
\frac{\partial h_{\parallel_x}}{\partial h_x} & \frac{\partial h_{\parallel_x}}{\partial h_y} \\
\frac{\partial h_{\parallel_y}}{\partial h_x} & \frac{\partial h_{\parallel_y}}{\partial h_y}
\end{bmatrix}
= \frac{1}{h_z^3} \begin{bmatrix}
1 - h_y^2 & h_x h_y \\
1 - h_x^2 & -h_x h_y
\end{bmatrix}.
\]

Hence, the Jacobian matrix of the transformation from spherical space to slope space is obtained as

\[
J_{o \rightarrow \parallel} = J_{\perp \rightarrow \parallel} J_{o \rightarrow \perp} = \frac{1}{h_z^2 \sqrt{1 - h_z^2}} \begin{bmatrix}
h_x \\
h_y \\
h_x h_z
\end{bmatrix}.
\]

References
