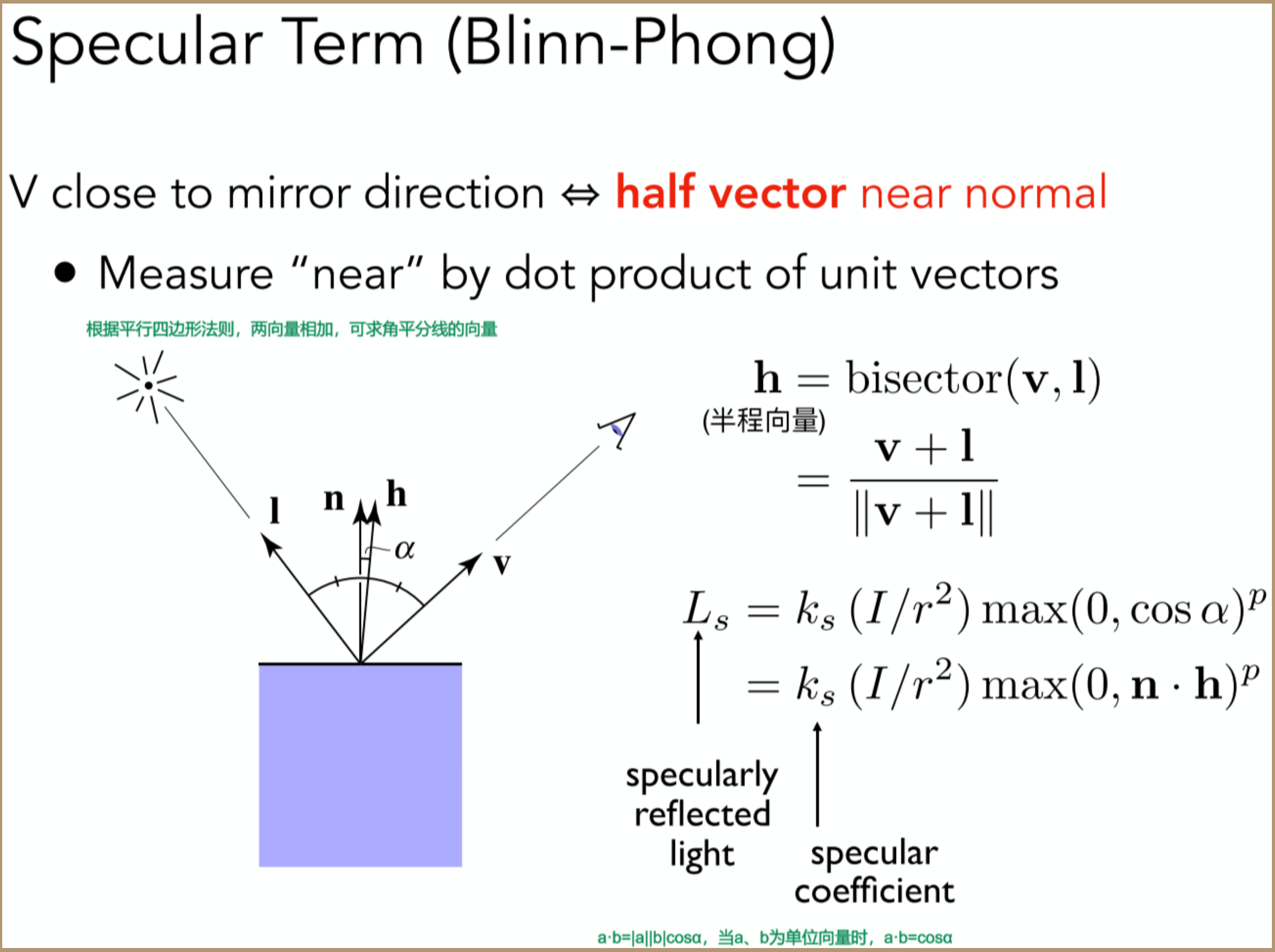


Specular Term (Blinn-Phong)

V close to mirror direction ⇔ **half vector** near normal

- Measure "near" by dot product of unit vectors

根据平行四边形法则，两向量相加，可求角平分线的向量

$$\mathbf{h} = \mathrm{bisector}(\mathbf{v}, \mathbf{l})$$

(半程向量)

$$= \frac{\mathbf{v} + \mathbf{l}}{\|\mathbf{v} + \mathbf{l}\|}$$

$$L_s = k_s\,(I/r^2)\max(0, \cos\alpha)^p$$

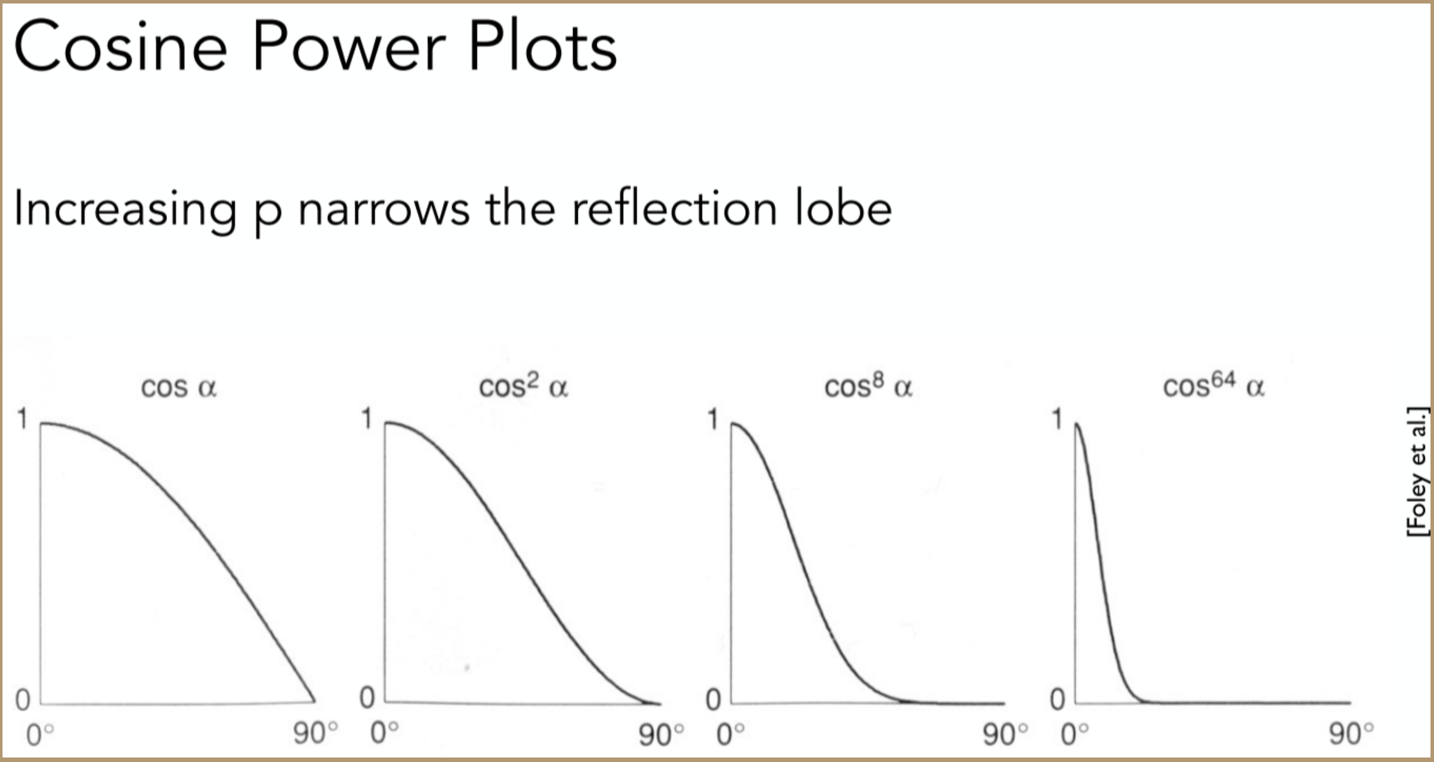
$$= k_s\,(I/r^2)\max(0, \mathbf{n}\cdot\mathbf{h})^p$$

specularly reflected light

specular coefficient

a·b=|a||b|cosα，当a、b为单位向量时，a·b=cosα

Specular intensity depends on view direction, bright near mirror reflection direction



Cosine Power Plots

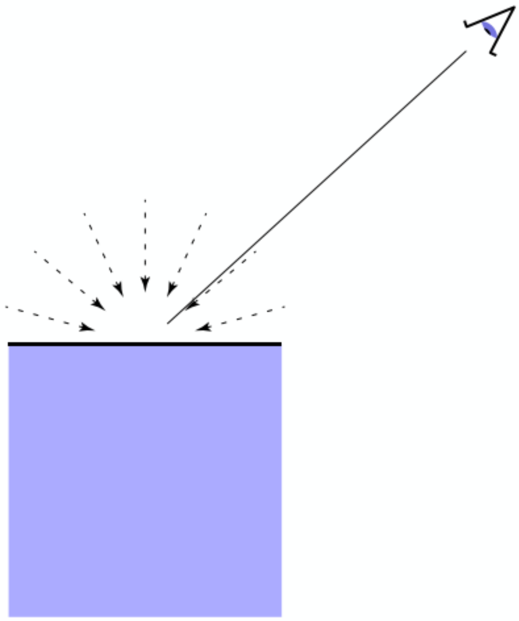
Increasing p narrows the reflection lobe

[Foley et al.]

Ambient Term

Shading that does not depend on anything

- Add constant color to account for disregarded illumination and fill in black shadows
- This is approximate / fake!

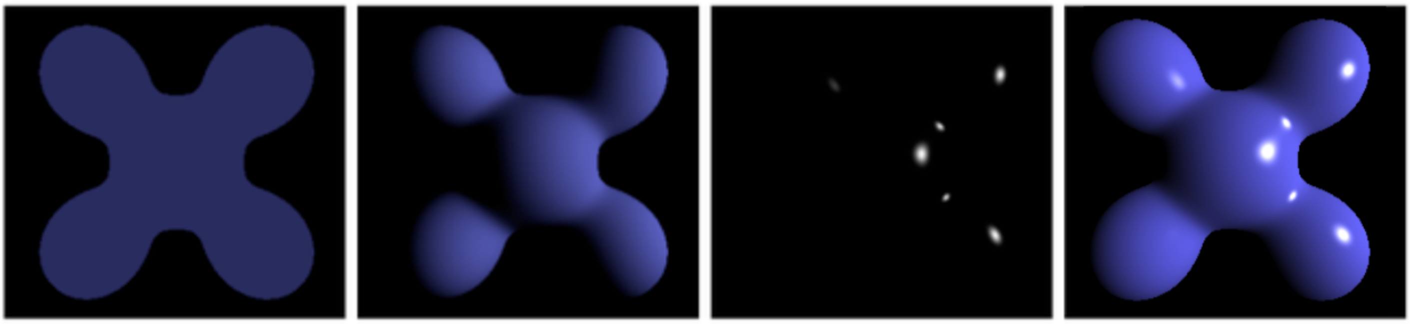


$$L_a = k_a \, I_a$$

reflected ambient light

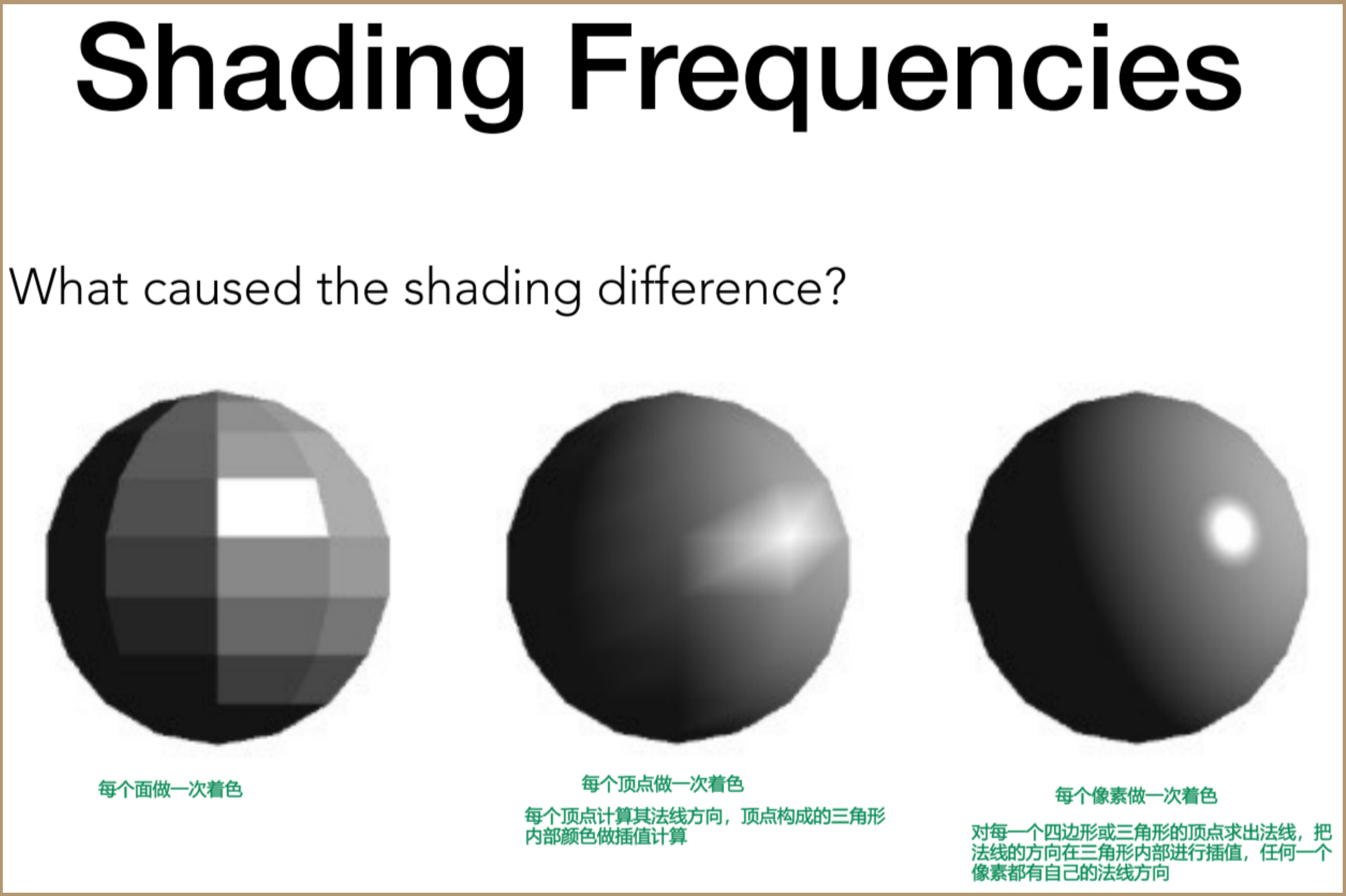
ambient coefficient

Blinn-Phong Reflection Model



Ambient + **Diffuse** + **Specular** = **Blinn-Phong Reflection**

$$
\begin{aligned}
L &= L_a + L_d + L_s \\
&= k_a \, I_a + k_d \, (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{l}) + k_s \, (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{h})^p
\end{aligned}
$$



Shading Frequencies

What caused the shading difference?

每个面做一次着色

每个顶点做一次着色

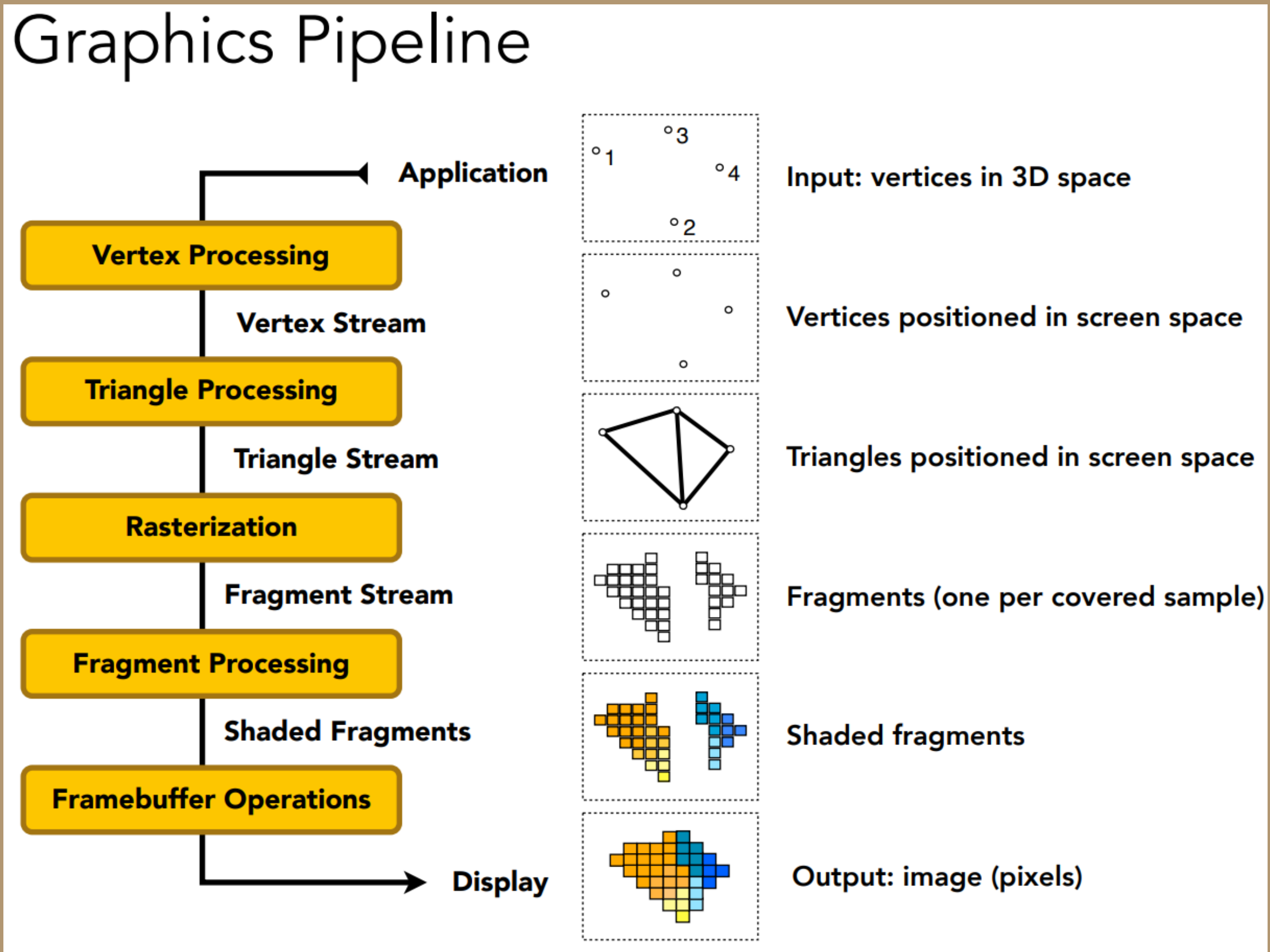
每个顶点计算其法线方向，顶点构成的三角形
内部颜色做插值计算

每个像素做一次着色

对每一个四边形或三角形的顶点求出法线，把
法线的方向在三角形内部进行插值，任何一个
像素都有自己的法线方向

Flat Shading
可使用三角形的两条边做叉积，求出一个面的法线方向

如何求出一个顶点的法线？
对顶点周围的三角面的法线做一个加权平均，面积更大的三角形贡献的更多



Graphics Pipeline

Application

Vertex Processing

Vertex Stream

Triangle Processing

Triangle Stream

Rasterization

Fragment Stream

Fragment Processing

Shaded Fragments

Framebuffer Operations

Display

Input: vertices in 3D space

Vertices positioned in screen space

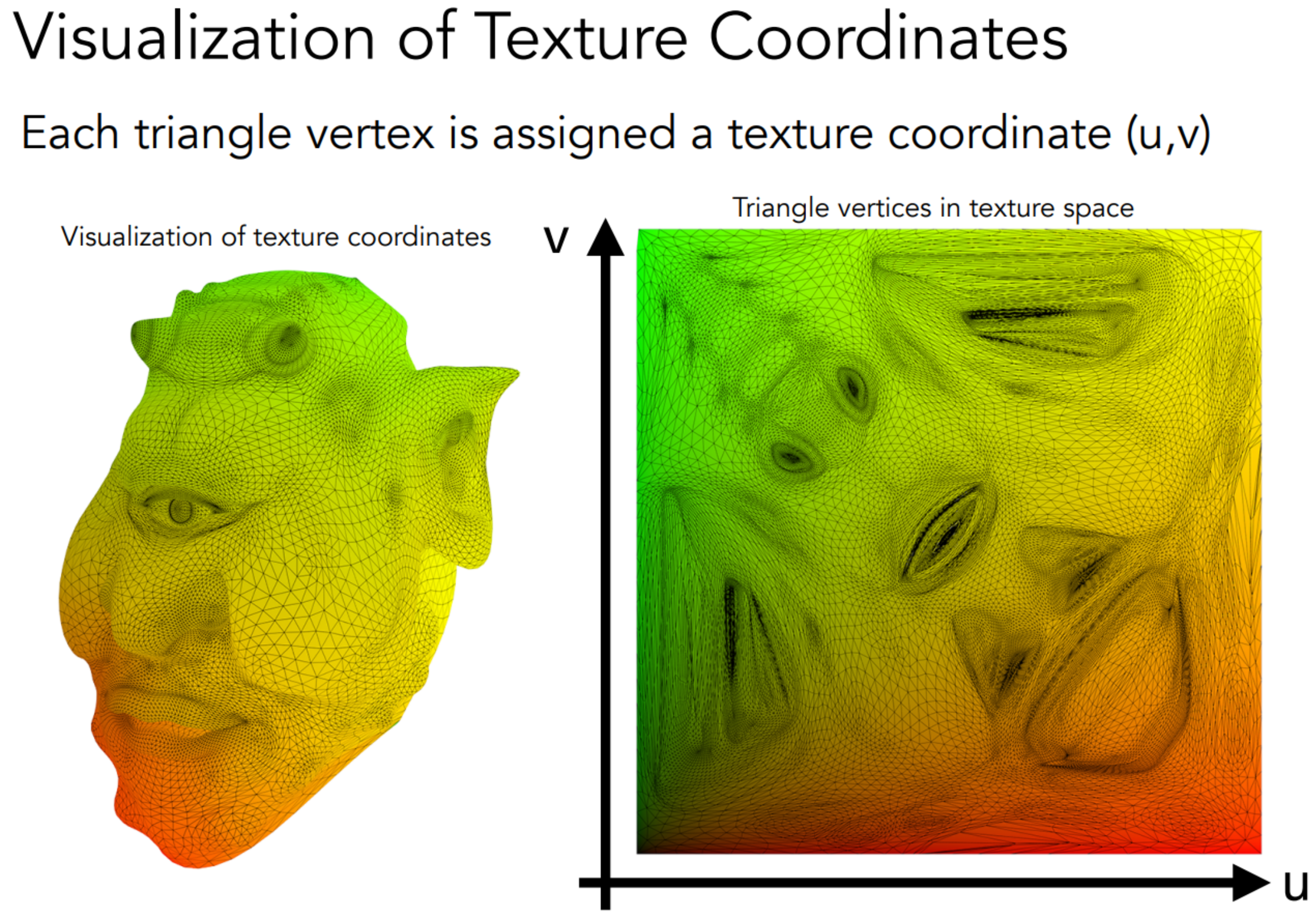
Triangles positioned in screen space

Fragments (one per covered sample)

Shaded fragments

Output: image (pixels)

Texture Mapping



Visualization of Texture Coordinates

Each triangle vertex is assigned a texture coordinate (u,v)

Visualization of texture coordinates

Triangle vertices in texture space