第6讲：着色

中国海洋大学 周佳然

上次课程内容

- 什么是着色（Shading）？
- 一个简单的着色模型：Blinn-Phong反射模型
- 着色频率
  - Flat shading
  - Gouraud shading
  - Phong shading
- 图形管线（Graphics Pipeline）
- 纹理映射（Texture Mapping）

本次课程内容

- 纹理映射
- 重心坐标（Barycentric coordinates）
- 纹理的应用
- 期中综述报告要求已发布

OCEAN UNIVERSITY OF CHINA

纹理映射（Texture Mapping）

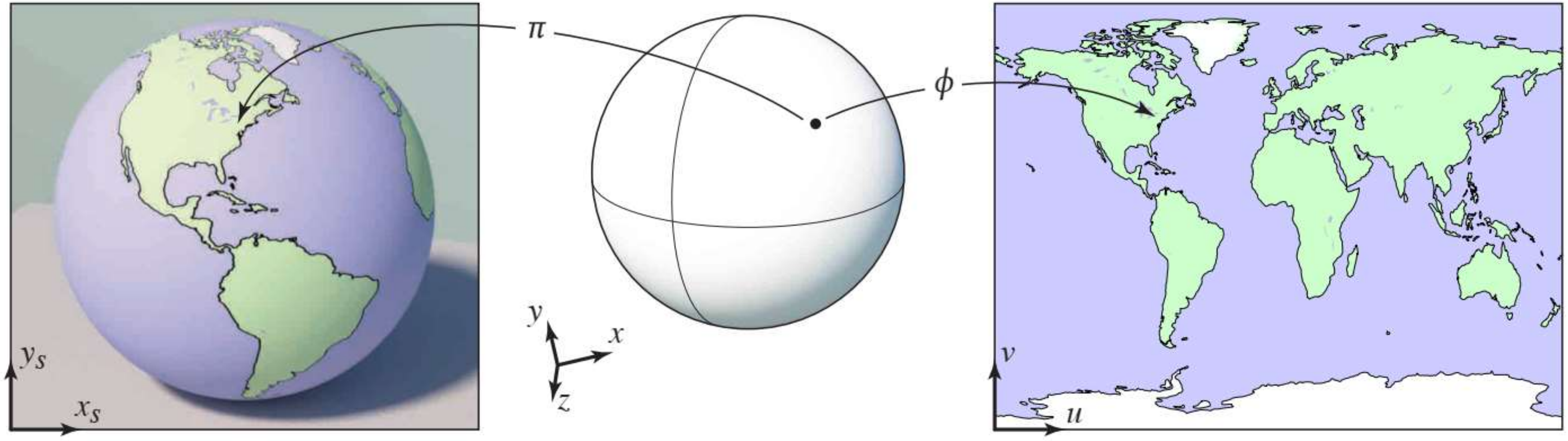
Pattern on ball

Wood grain on floor

中國海洋大學 OCEAN UNIVERSITY OF CHINA

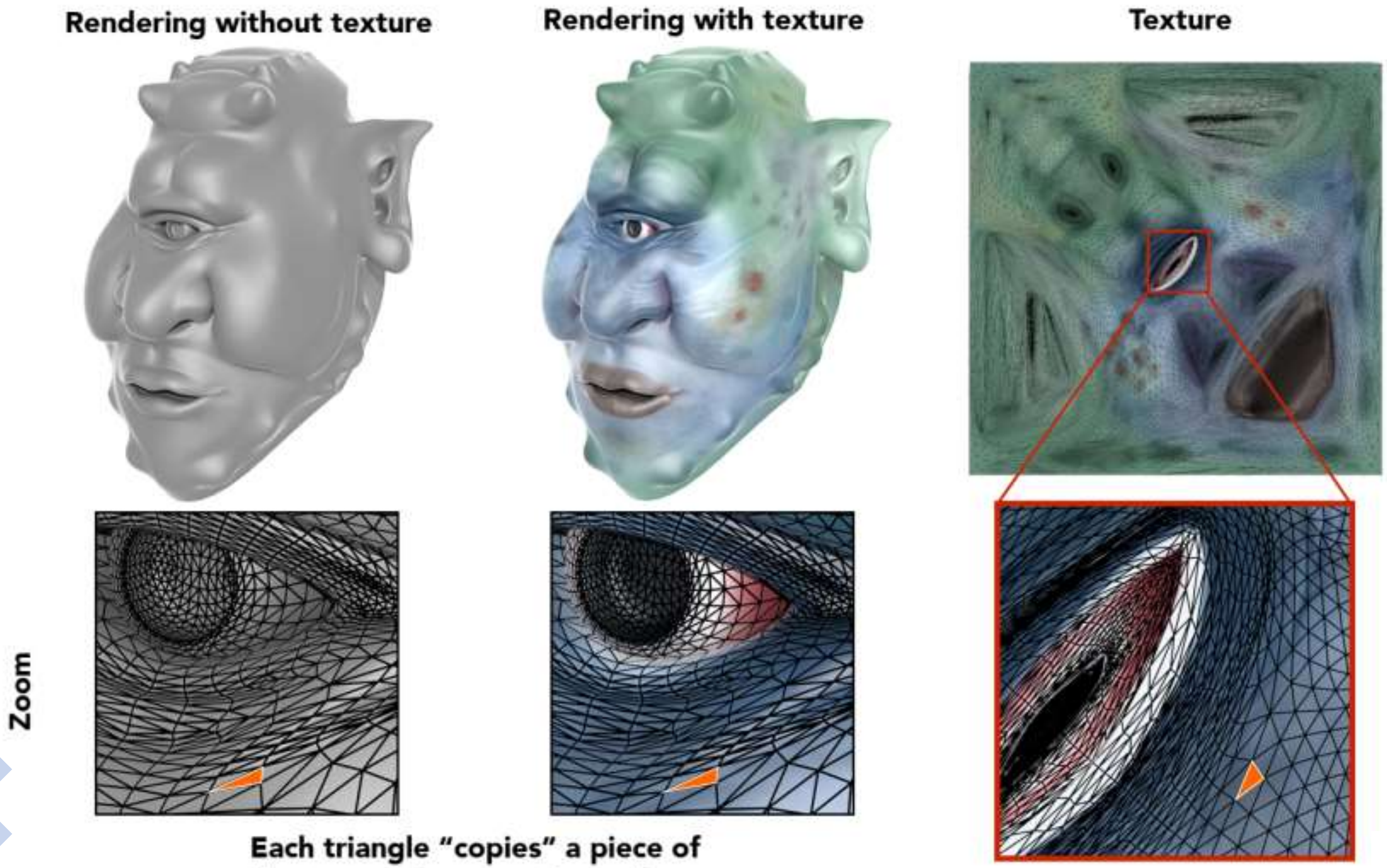
表面（surface）是二维的

- 三维空间中的表面上的一点总可以对应于二维图像（纹理）上的一点



Screen space　　World space　　Texture space

将纹理应用于表面

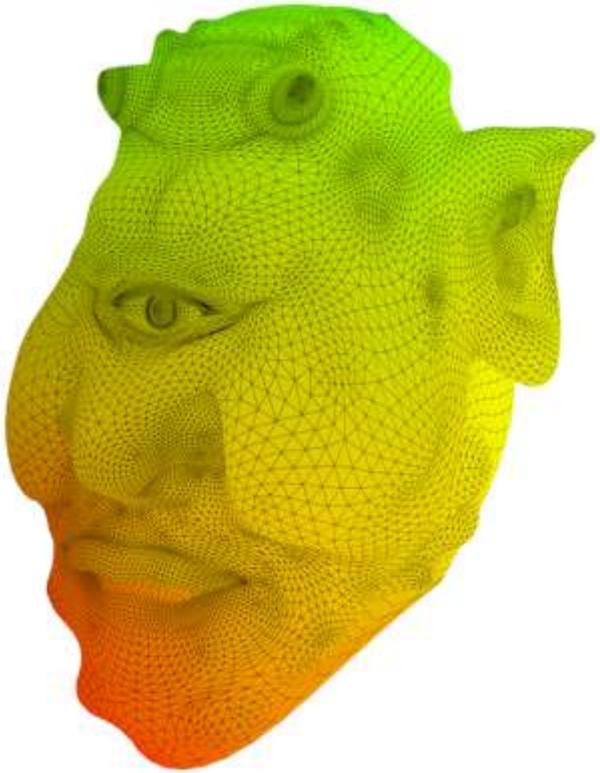


Each triangle "copies" a piece of
the texture image to the surface.

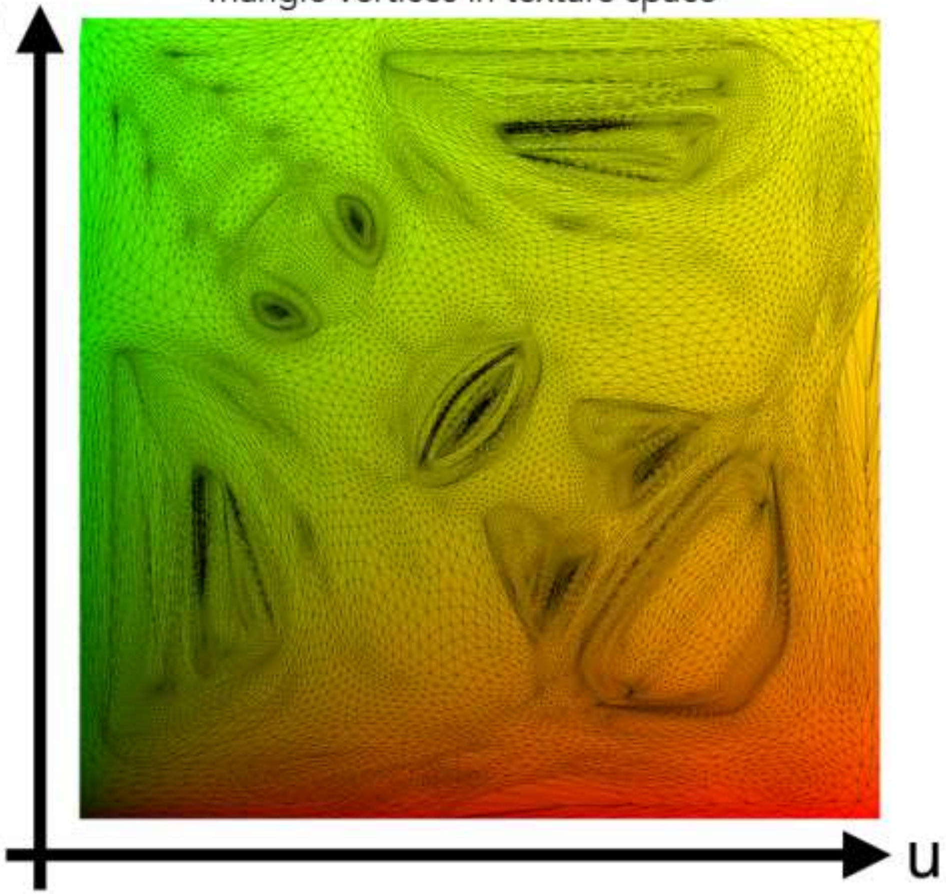
纹理坐标

- 每一个三角形顶点都分配有一个纹理坐标（u，v）

Visualization of texture coordinates



Triangle vertices in texture space

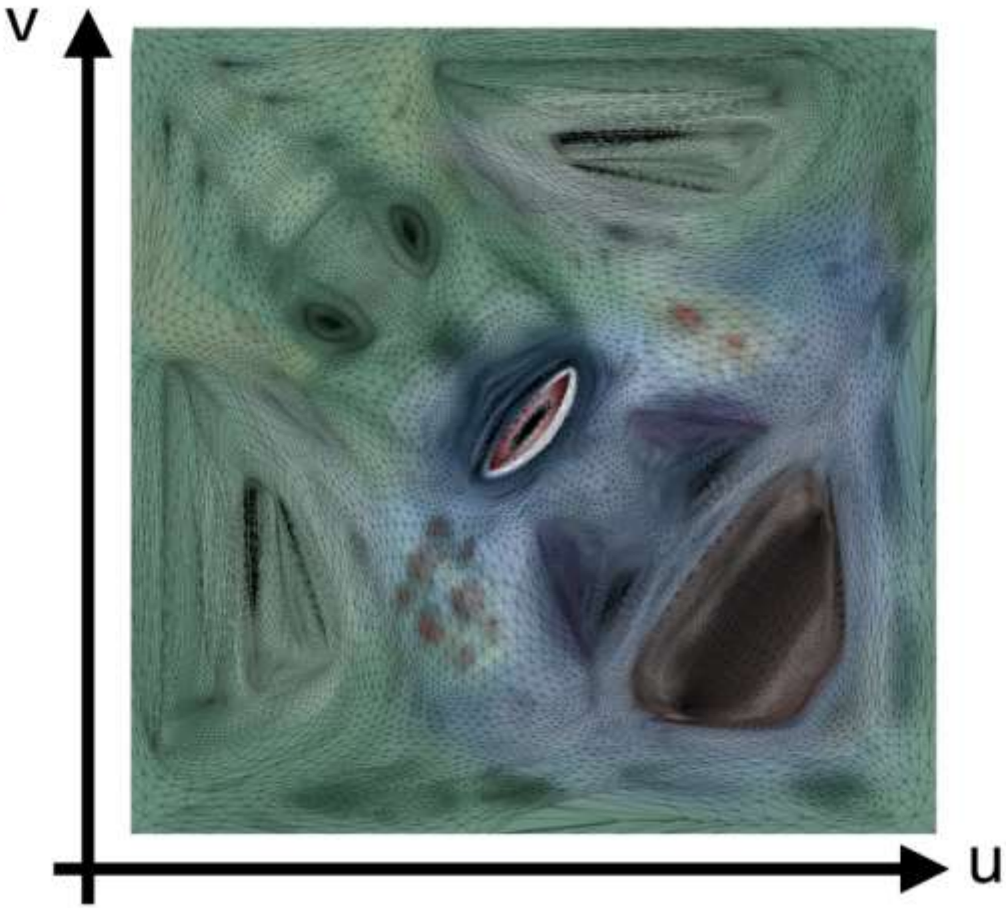


中國海洋大學 OCEAN UNIVERSITY OF CHINA

将纹理应用于表面

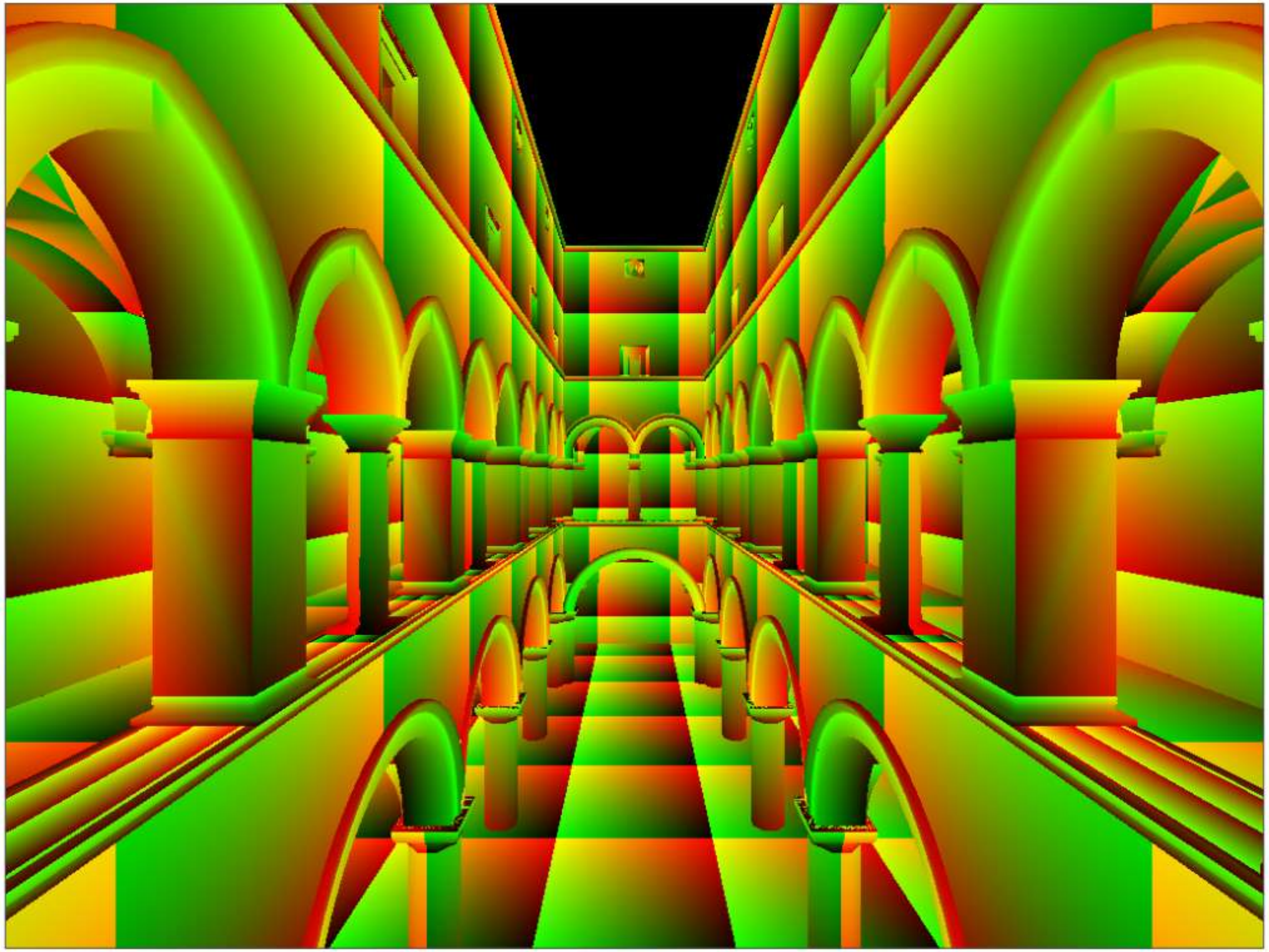
Rendered result

Triangle vertices in texture space

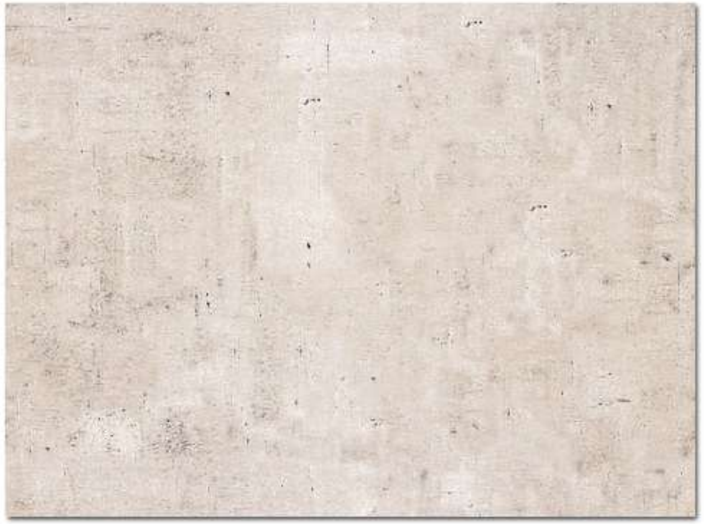
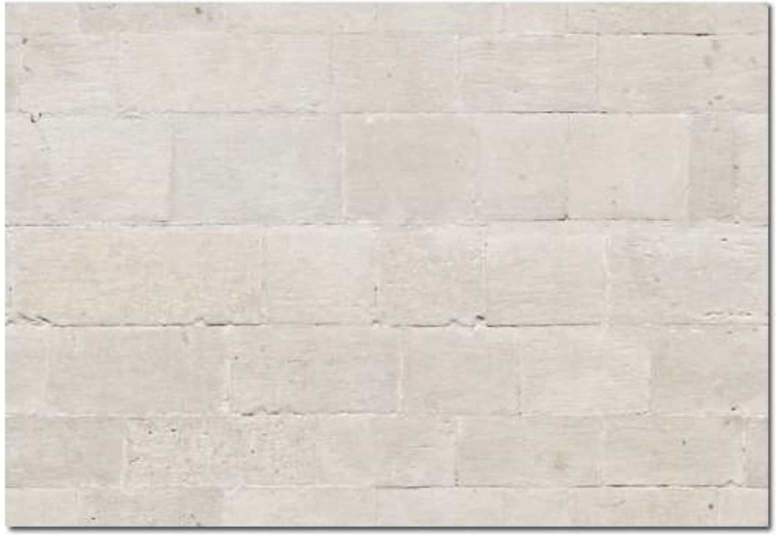


将纹理应用于表面

纹理坐标可视化



纹理可被重复使用


example textures
used / **tiled**

Q&A

在三角形内部进行插值

Q:为什么需要插值?

- 通常在顶点处给定属性的值（离散）
- 需要在三角形内部获得平滑变化的属性值（连续）

Q:要插值些什么属性?

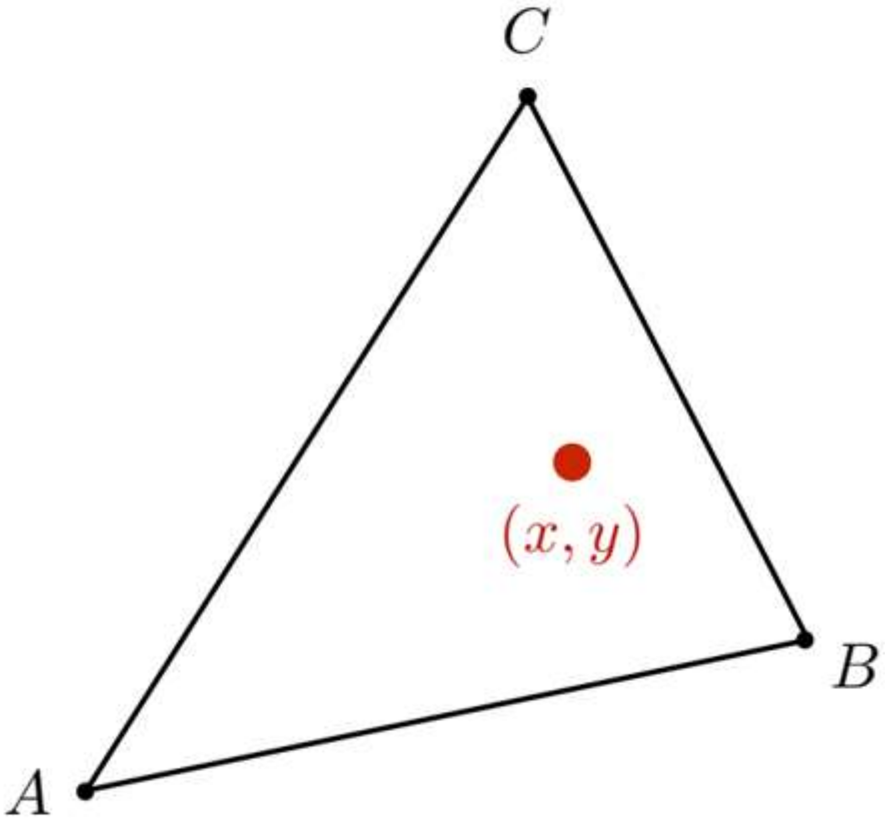
- 纹理坐标、颜色、法向量……

Q:怎么来做插值?

- 重心坐标

重心坐标

- 利用三角形三个顶点建立的一套坐标系



$$(x, y) = \alpha A + \beta B + \gamma C$$

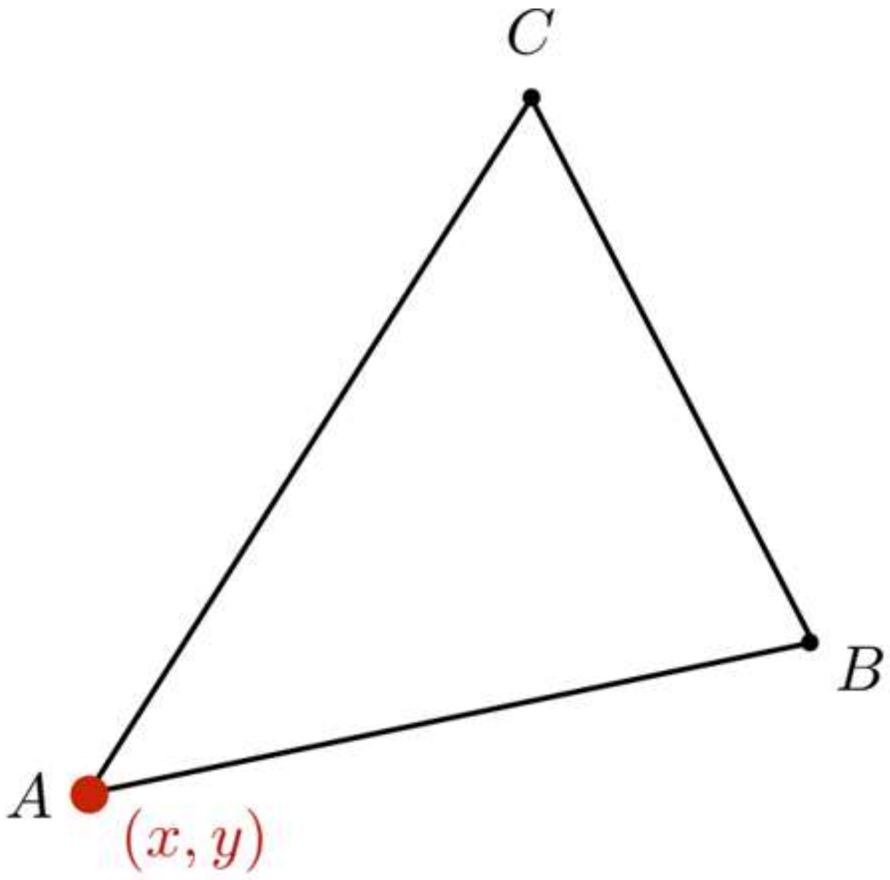
$$\alpha + \beta + \gamma = 1$$

Inside the triangle if all three coordinates are non-negative

OCEAN UNIVERSITY OF CHINA

重心坐标

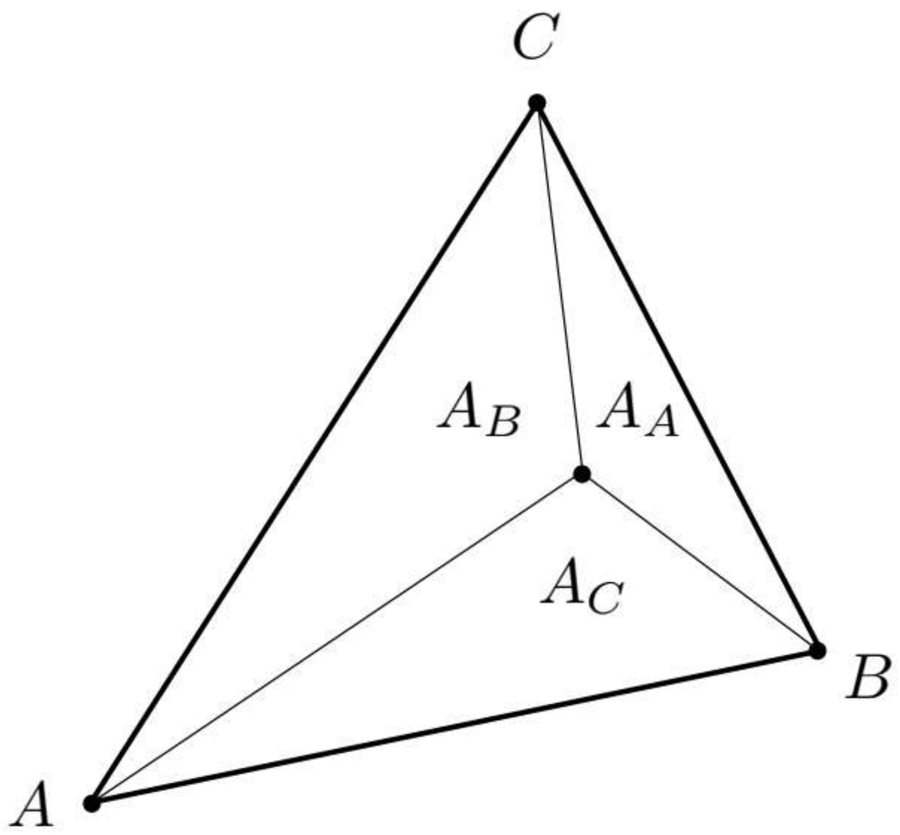
Q：点A的重心坐标是什么？



$$
\begin{aligned}
(\alpha, \beta, \gamma) &= (1, 0, 0) \\
(x, y) &= \alpha A + \beta B + \gamma C \\
&= A
\end{aligned}
$$

重心坐标

Q：如何求解(α，β，γ)?



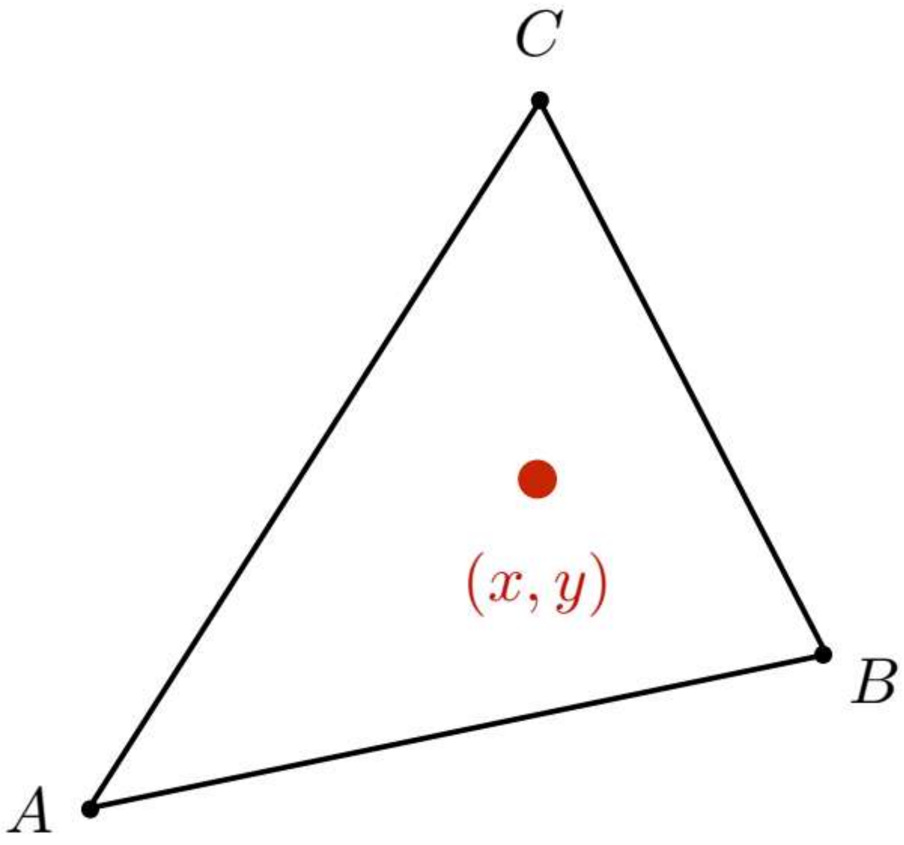
$$\alpha = \frac{A_A}{A_A + A_B + A_C}$$

$$\beta = \frac{A_B}{A_A + A_B + A_C}$$

$$\gamma = \frac{A_C}{A_A + A_B + A_C}$$

重心坐标

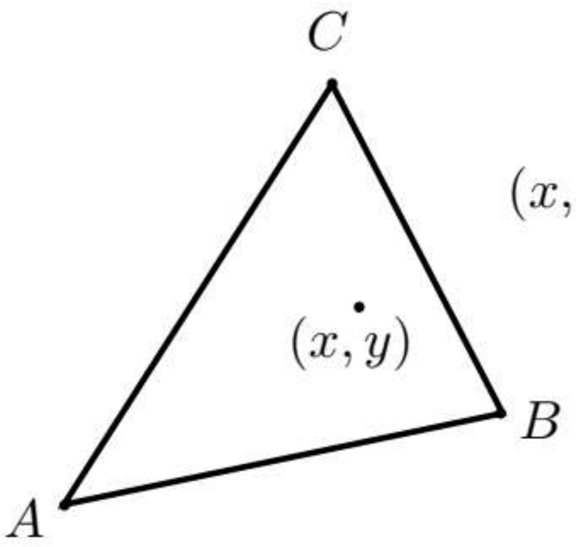
Q：三角形重心的重心坐标是什么？



$$(\alpha, \beta, \gamma) = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$$

$$(x, y) = \frac{1}{3}A + \frac{1}{3}B + \frac{1}{3}C$$

重心坐标



$$(x, y) = \alpha A + \beta B + \gamma C$$
$$\alpha + \beta + \gamma = 1$$

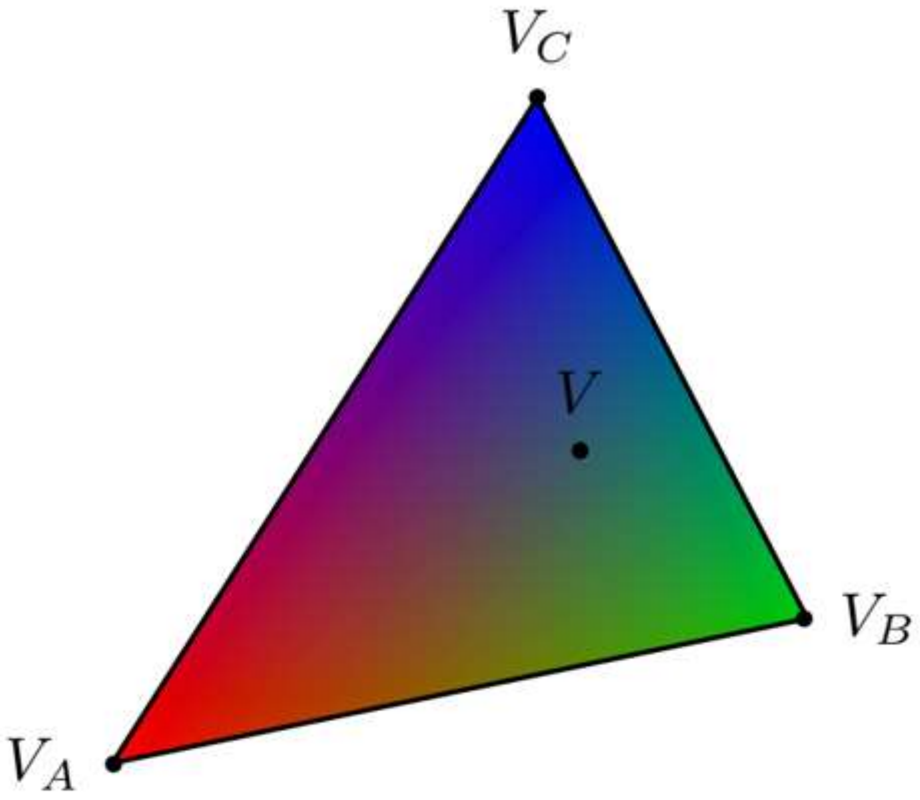
$$\alpha = \frac{-(x - x_B)(y_C - y_B) + (y - y_B)(x_C - x_B)}{-(x_A - x_B)(y_C - y_B) + (y_A - y_B)(x_C - x_B)}$$

$$\beta = \frac{-(x - x_C)(y_A - y_C) + (y - y_C)(x_A - x_C)}{-(x_B - x_C)(y_A - y_C) + (y_B - y_C)(x_A - x_C)}$$

$$\gamma = 1 - \alpha - \beta$$

重心坐标

- 属性值可以是纹理坐标、颜色、法向量、深度、材质属性等等



$$V = \alpha V_A + \beta V_B + \gamma V_C$$

注意：在投影变换下，重心坐标无法保持不变！

Q&A

纹理的应用

```
uniform sampler2D myTexture;    // program parameter
uniform vec3 lightDir;          // program parameter
varying vec2 uv;                // per fragment value (interp. by rasterizer)
varying vec3 norm;              // per fragment value (interp. by rasterizer)

void diffuseShader()
{
  vec3 kd;
  kd = texture2d(myTexture, uv);                   // material color from texture
  kd *= clamp(dot(–lightDir, norm), 0.0, 1.0);     // Lambertian shading model
  gl_FragColor = vec4(kd, 1.0);                    // output fragment color
}
```

简单的纹理映射：漫反射颜色

Usually a pixel's center

for each rasterized screen sample (x,y):

(u,v) = evaluate texture coordinate at (x,y)

texcolor = texture.sample(u,v);

set sample's color to texcolor;

Using barycentric coordinates!

Usually the diffuse albedo Kd
(recall the Blinn-Phong reflectance model)

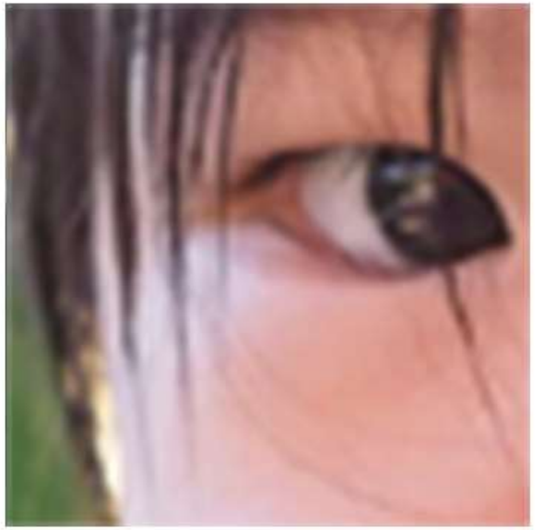
纹理放大

Q:如果纹理很小（分辨率不足）会带来什么问题?

- 纹理上的一个像素（pixel）称为一个纹理元素/纹素(texel)

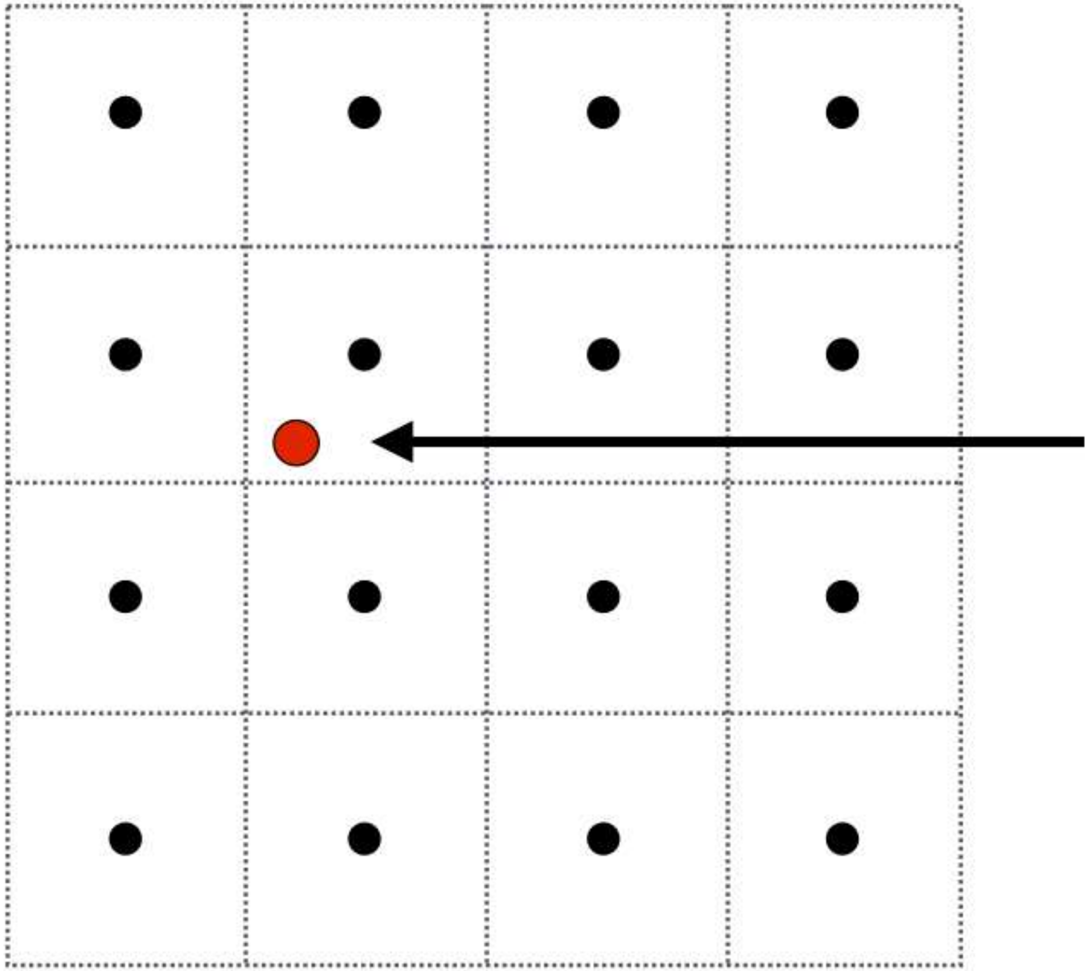
Nearest

Bilinear



Bicubic

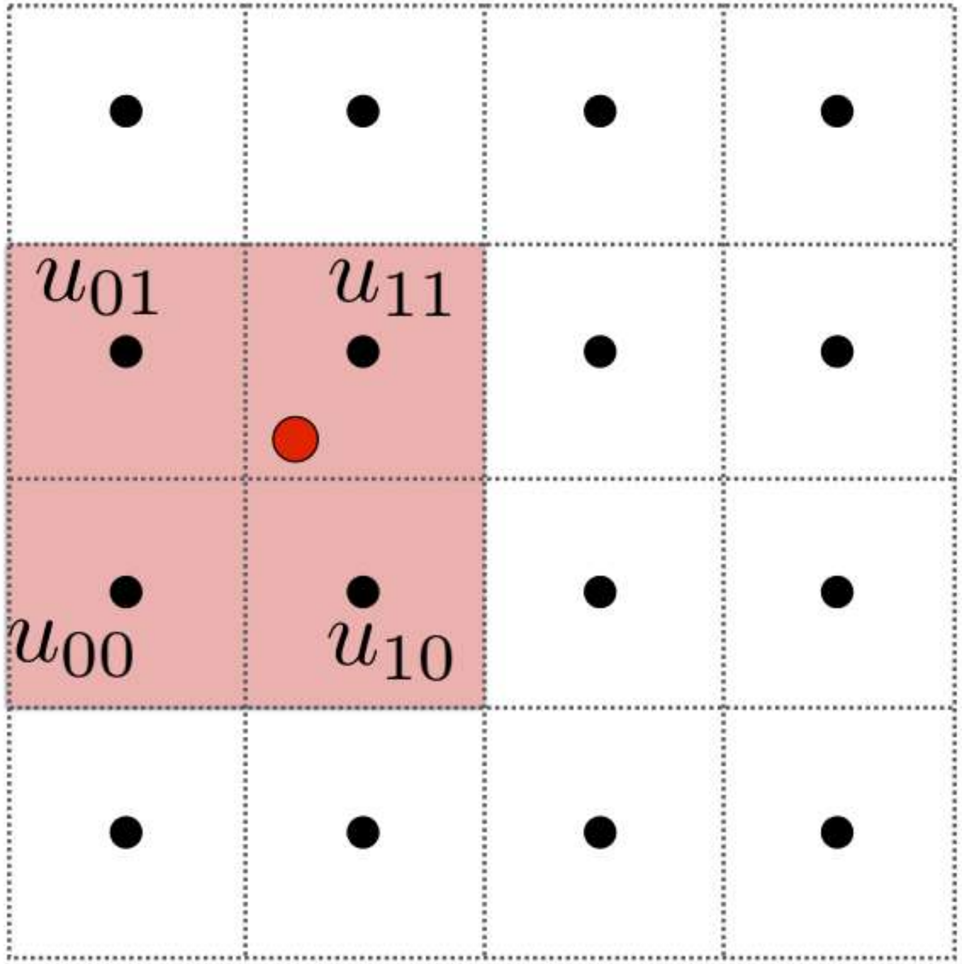
最近邻方法



Want to sample texture value f(x,y) at red point

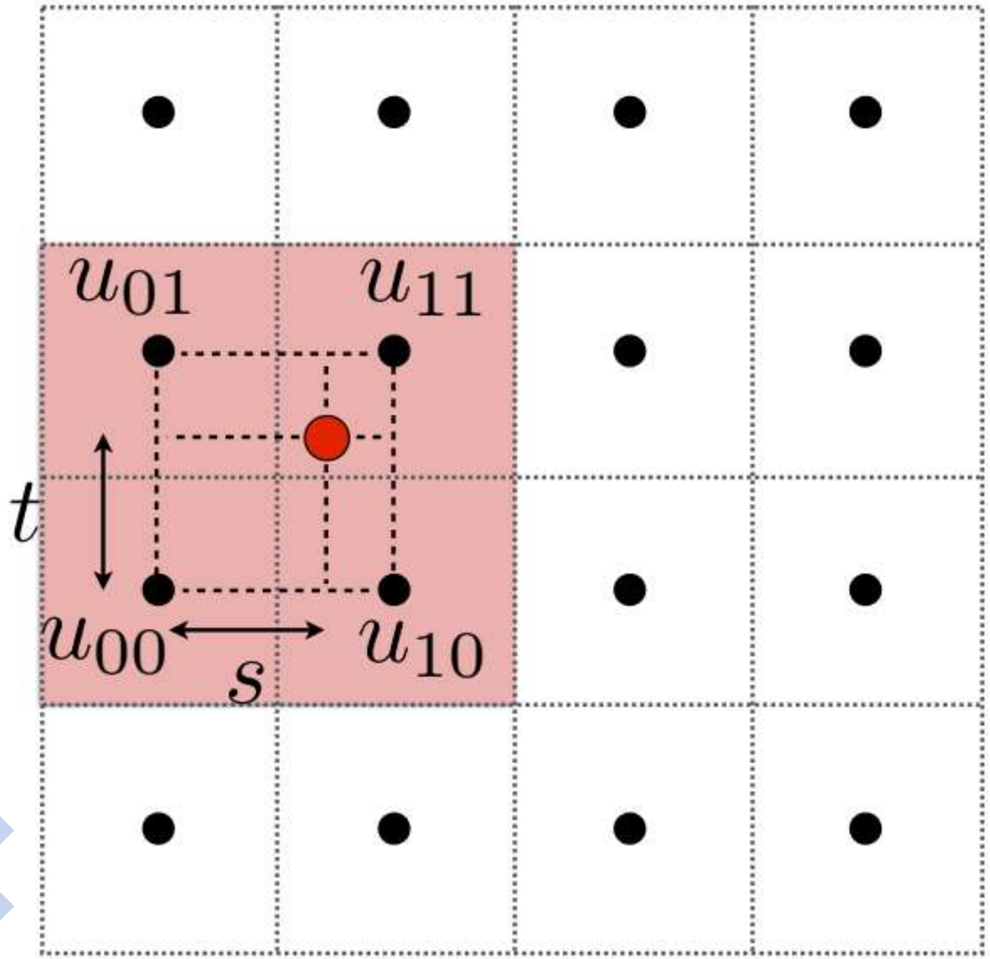
Black points indicate texture sample locations

双线性插值



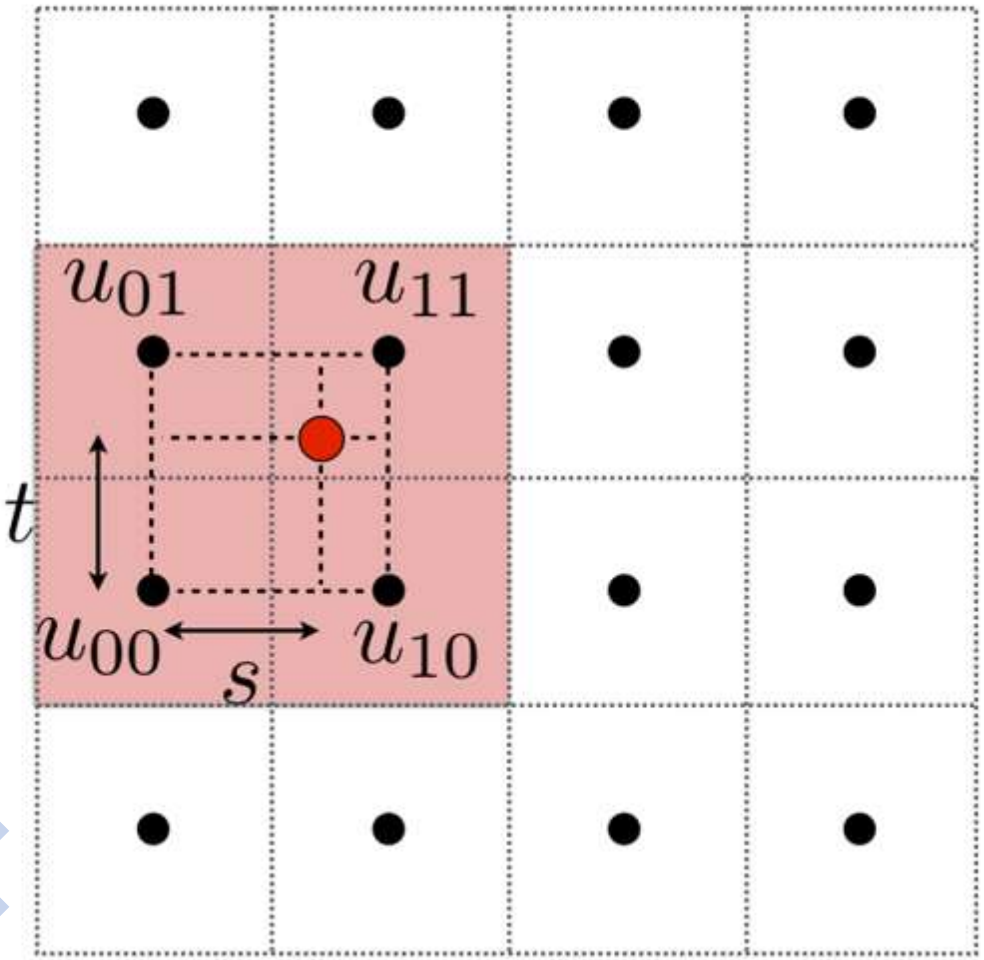
Take 4 nearest sample locations, with texture values as labeled.

双线性插值



And fractional offsets, (s,t) as shown

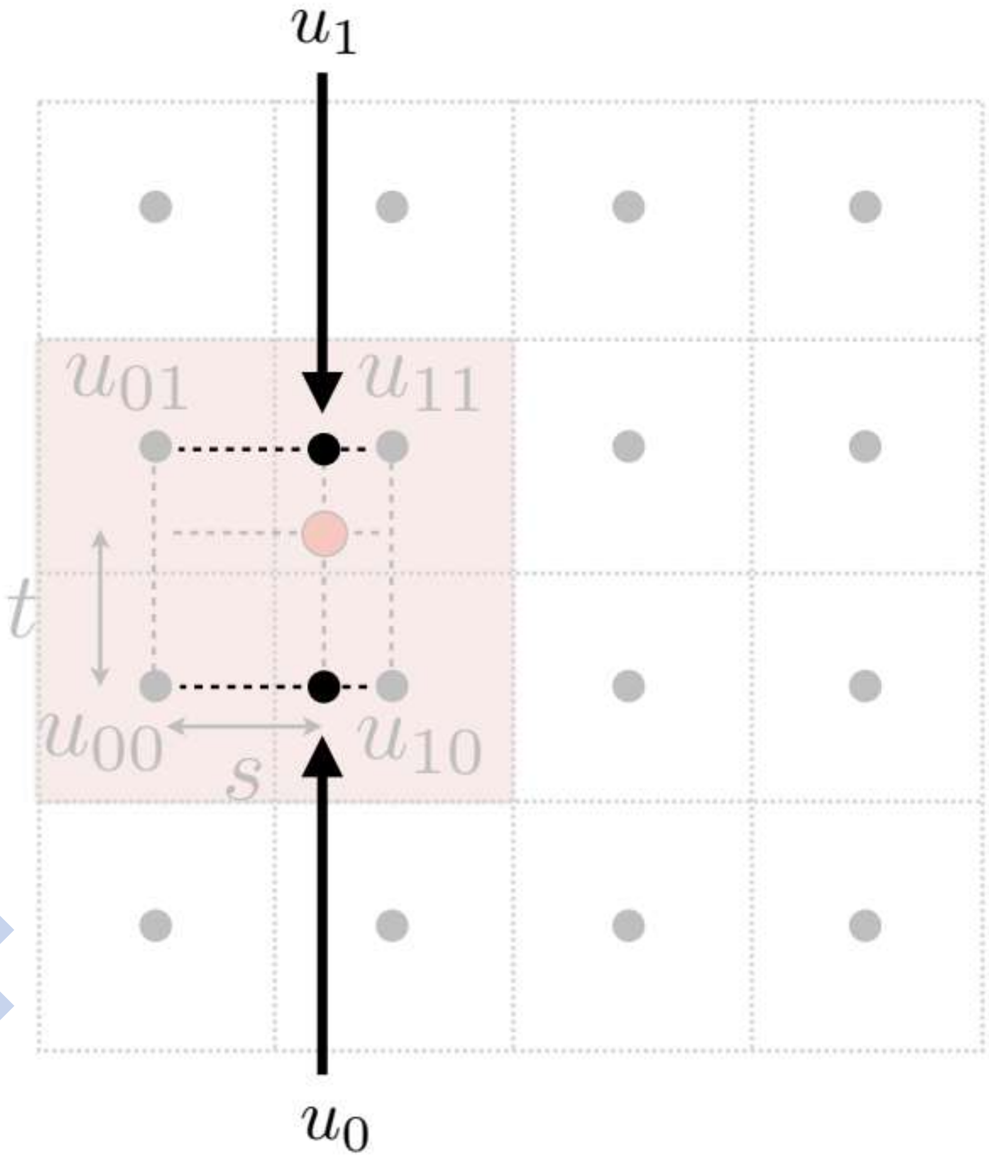
双线性插值



Linear interpolation (1D)

$$\mathrm{lerp}(x, v_0, v_1) = v_0 + x(v_1 - v_0)$$

双线性插值



Linear interpolation (1D)

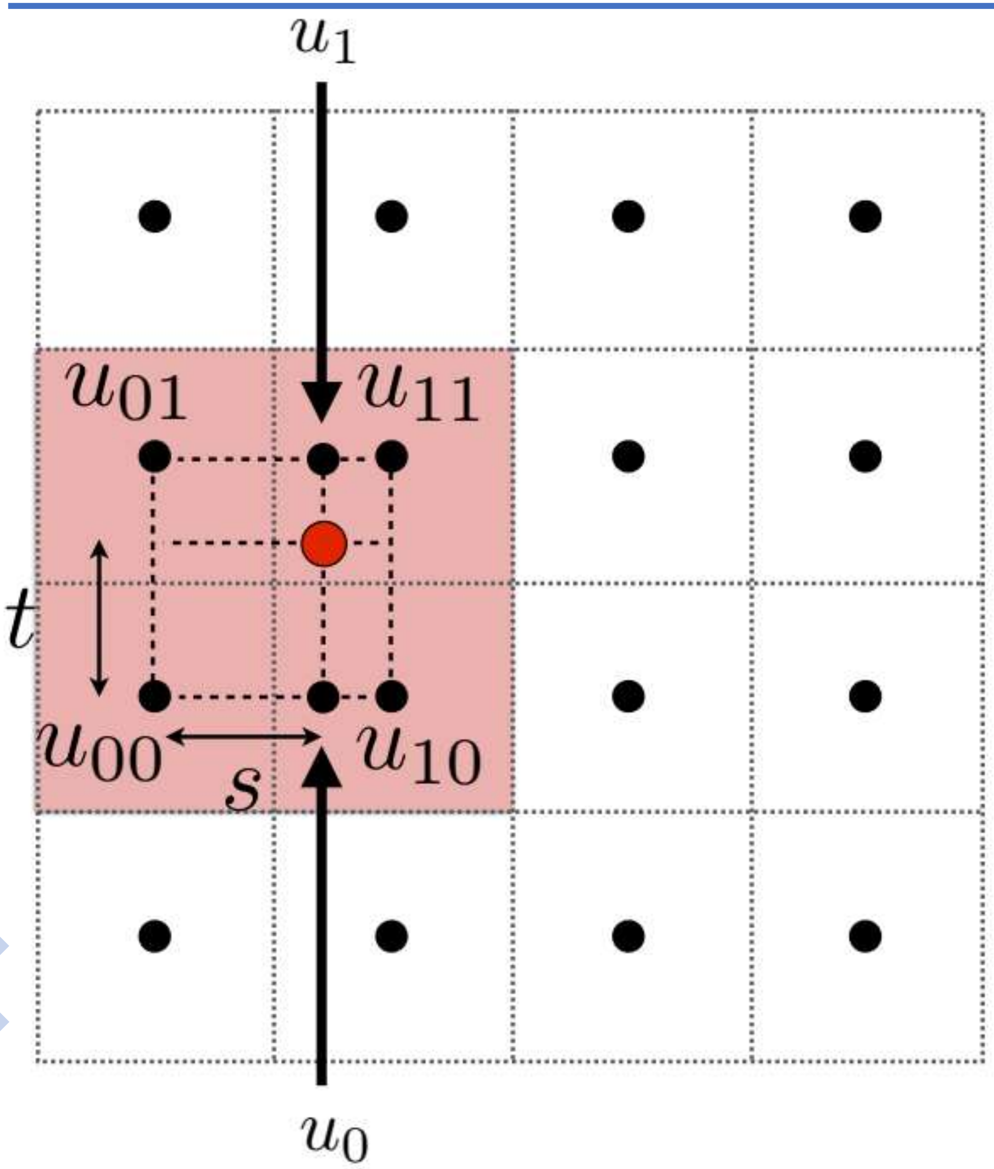
$$\text{lerp}(x, v_0, v_1) = v_0 + x(v_1 - v_0)$$

Two helper lerps (horizontal)

$$u_0 = \text{lerp}(s, u_{00}, u_{10})$$

$$u_1 = \text{lerp}(s, u_{01}, u_{11})$$

双线性插值



Linear interpolation (1D)

$$\text{lerp}(x, v_0, v_1) = v_0 + x(v_1 - v_0)$$

Two helper lerps

$$u_0 = \text{lerp}(s, u_{00}, u_{10})$$

$$u_1 = \text{lerp}(s, u_{01}, u_{11})$$

Final vertical lerp, to get result:

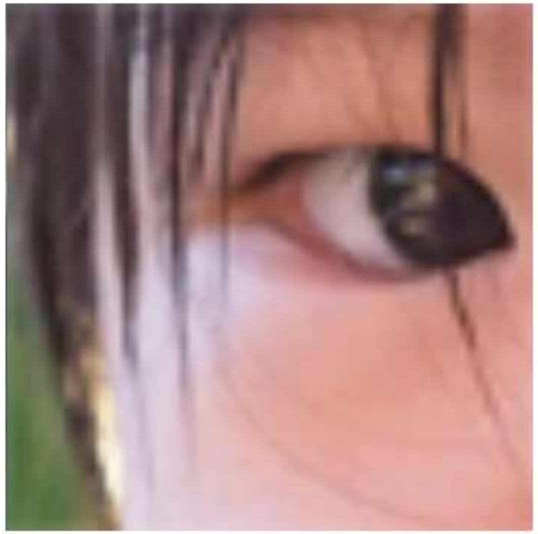
$$f(x, y) = \text{lerp}(t, u_0, u_1)$$

纹理放大

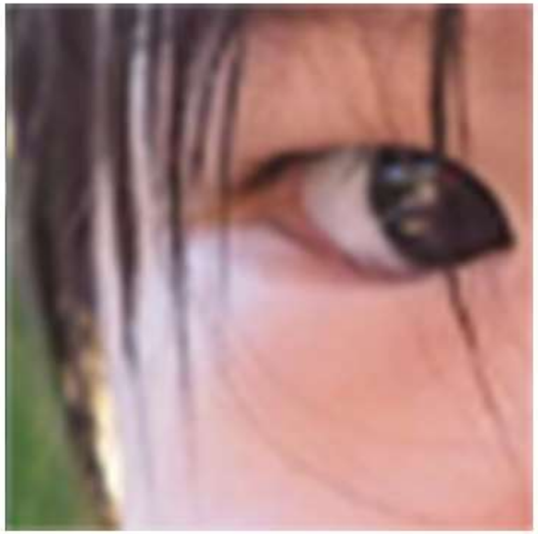
- 纹理放大是一种较简单的情况

➢ 双线性插值可以在可接受的代价下生成不错的结果

Nearest



Bilinear



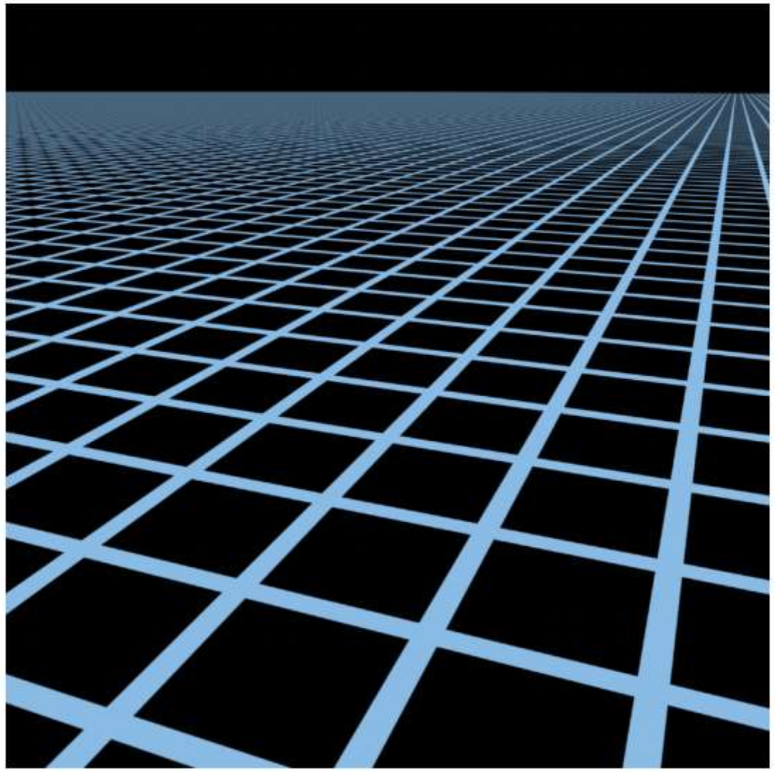
Bicubic

Q&A

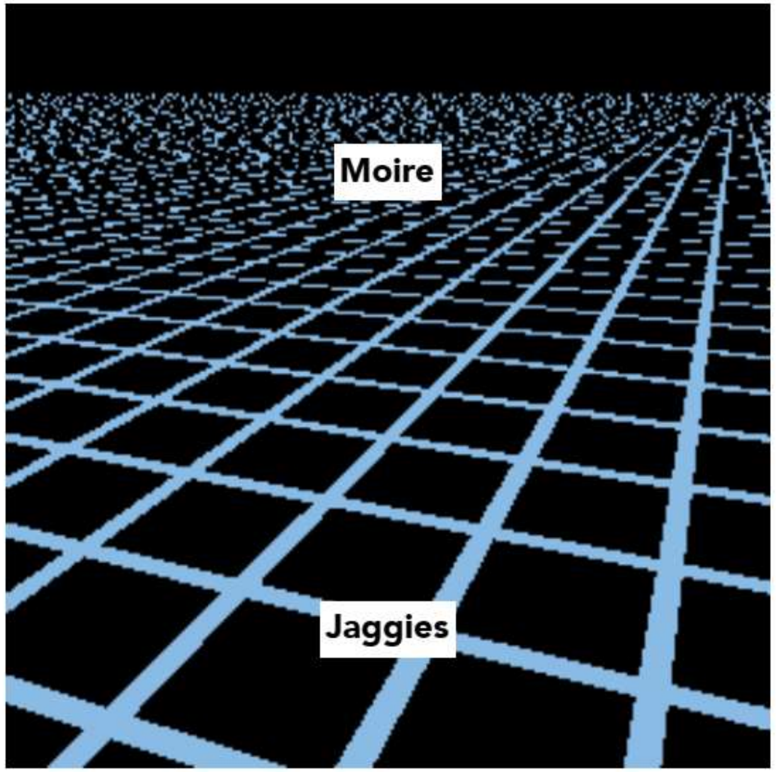
中國海洋大學
OCEAN UNIVERSITY OF CHINA

纹理缩小

Q:如果纹理很大会有问题吗?



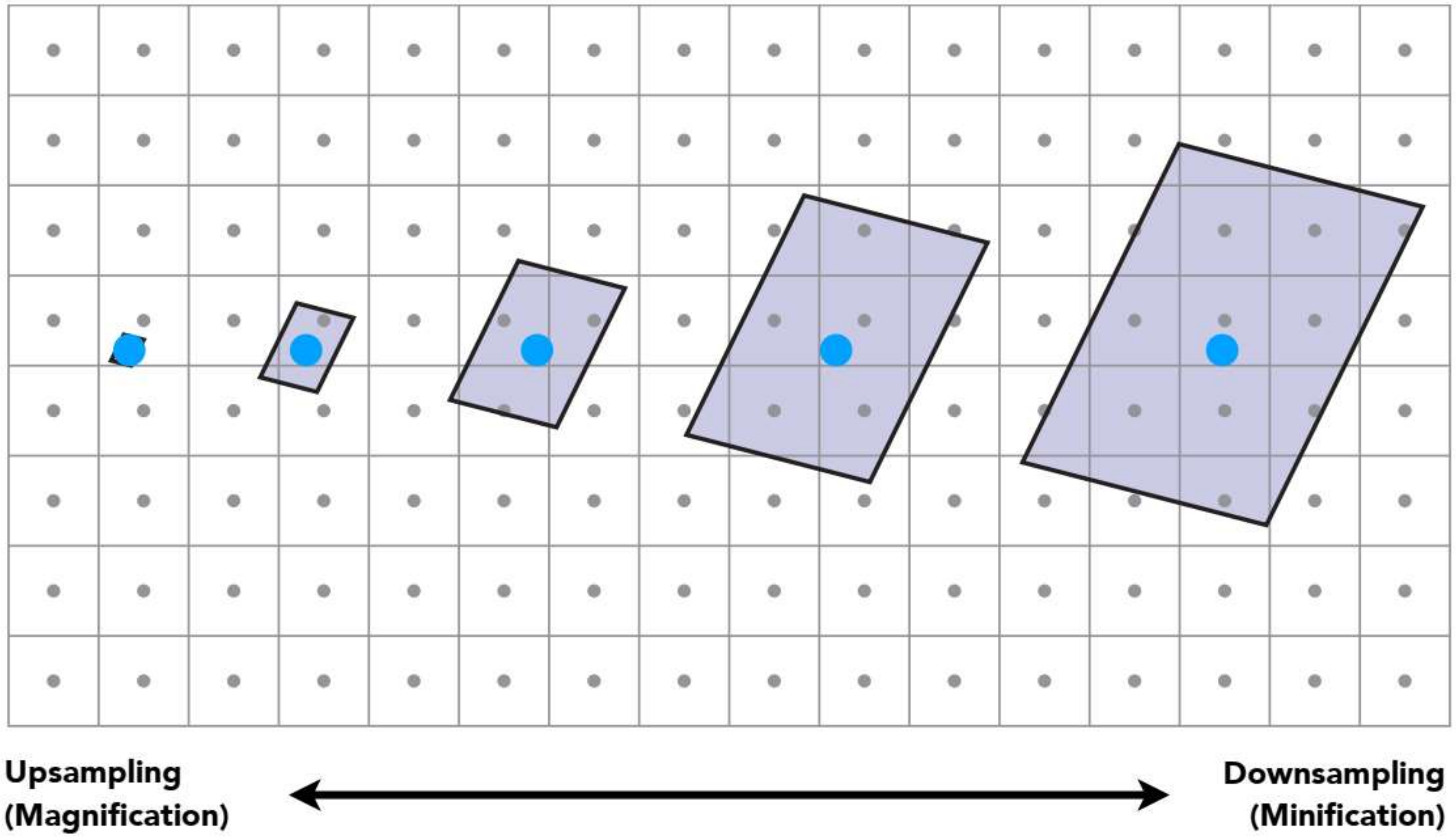
Reference



Point sampled

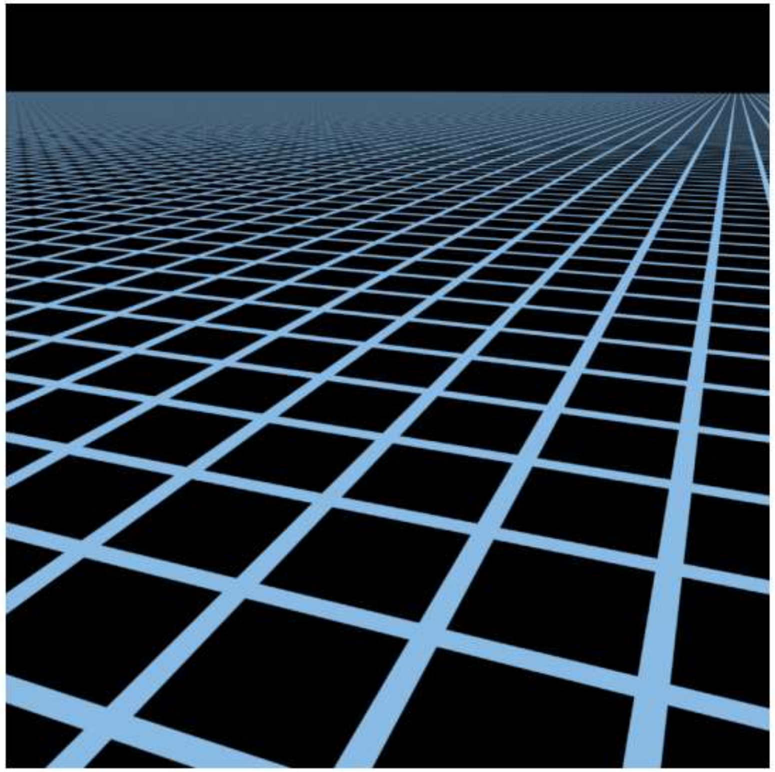
OCEAN UNIVERSITY OF CHINA

像素“覆盖”的纹理

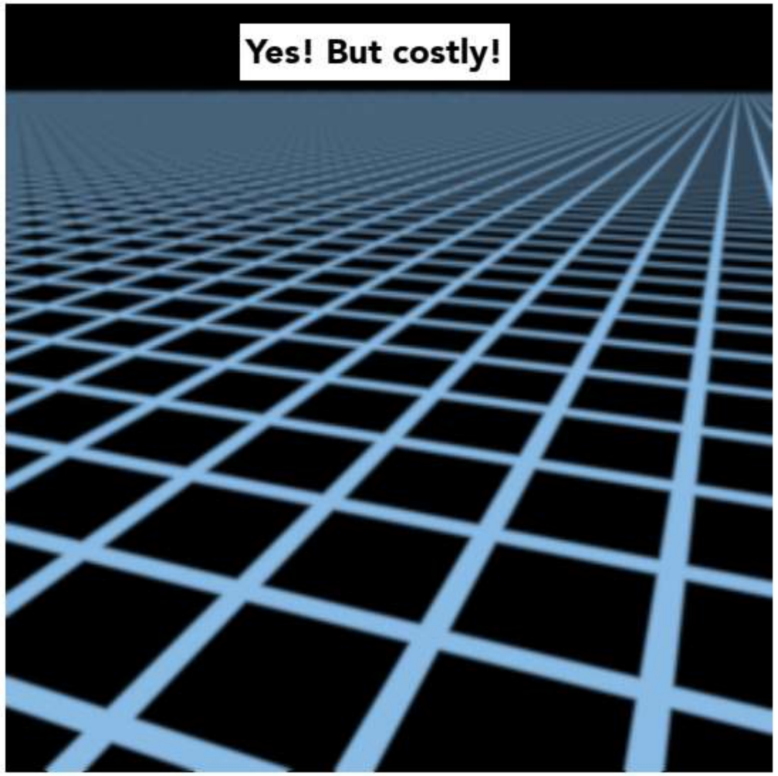


纹理缩小

Q:超级采样可以用来解决这个问题吗?



Reference



512x supersampling

纹理缩小

Q:超级采样可以用来解决这个问题吗?

- 可以，质量高但代价也高
- 当纹理缩小时，一个像素“覆盖”多个纹素
- 像素中的信号频率过高，需要更高的采样频率

Q:有没有其他思路来解决这个问题?

- 不做采样，而是在一定范围内求平均值
- 点查询vs.范围查询

中國海洋大學 OCEAN UNIVERSITY OF CHINA

不同像素“覆盖区域”可能不同

中國海洋大學
OCEAN UNIVERSITY OF CHINA

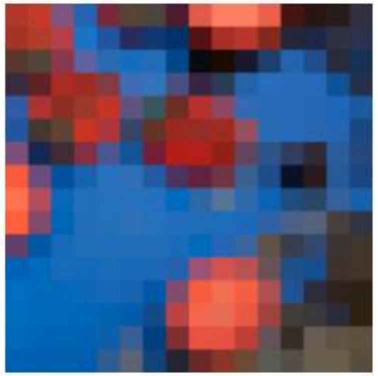
Mipmap（多级贴图）

- 快速、近似、正方形的范围查询

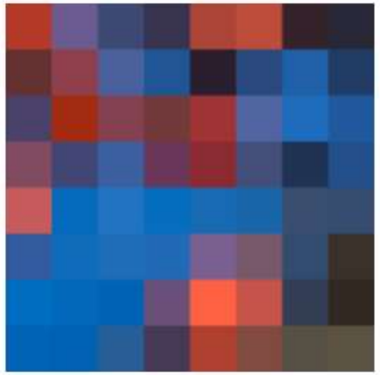
Level 0 = 128x128

Level 1 = 64x64

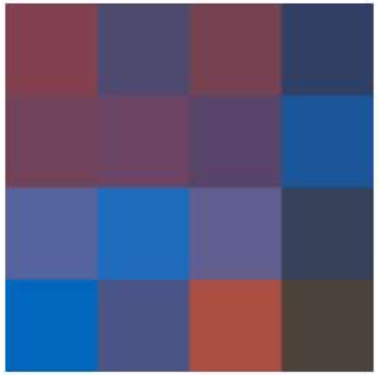
Level 2 = 32x32



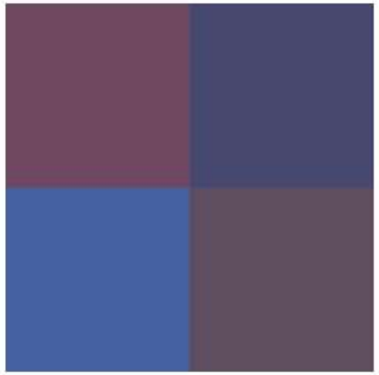
Level 3 = 16x16



Level 4 = 8x8



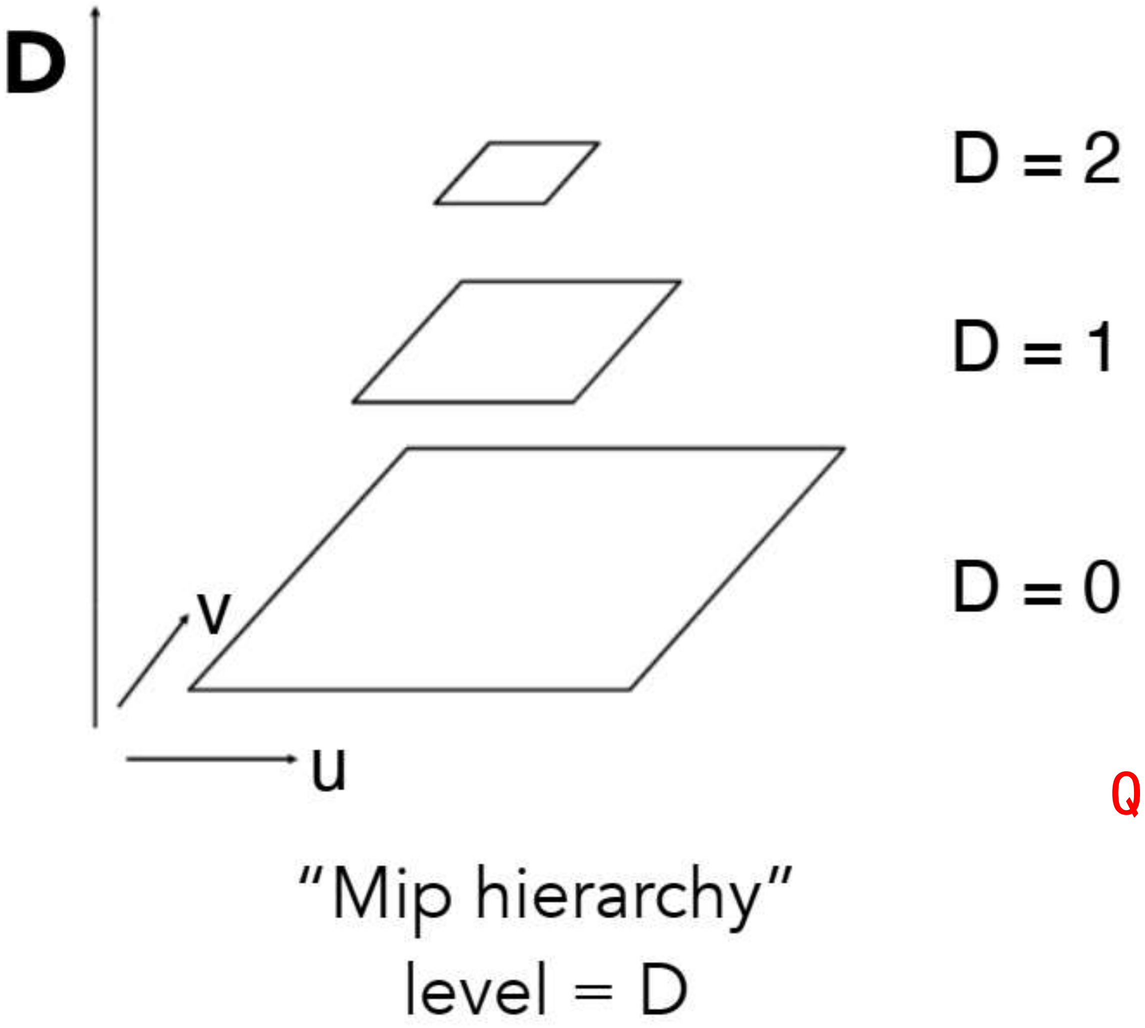
Level 5 = 4x4



Level 6 = 2x2

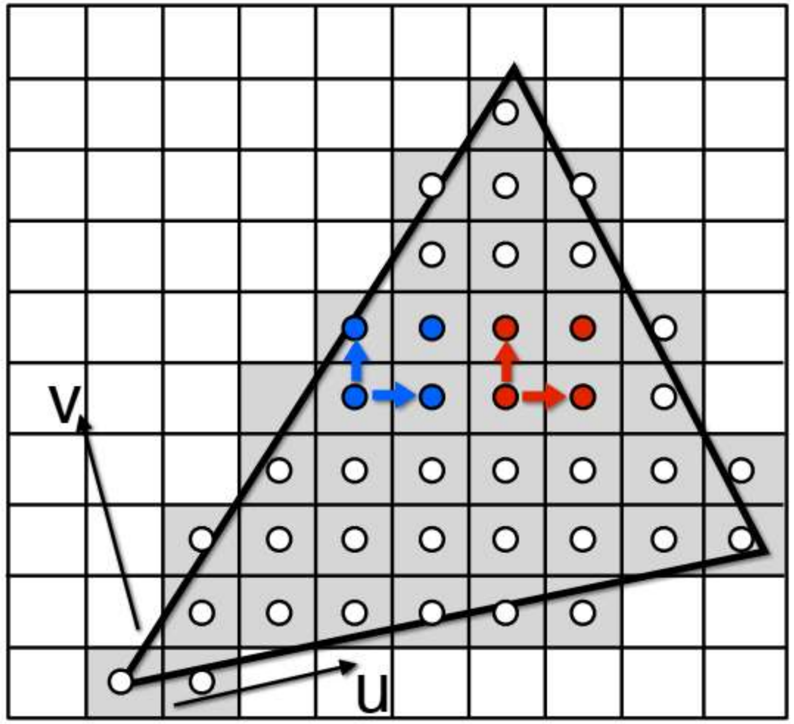
Level 7 = 1x1

Mipmap

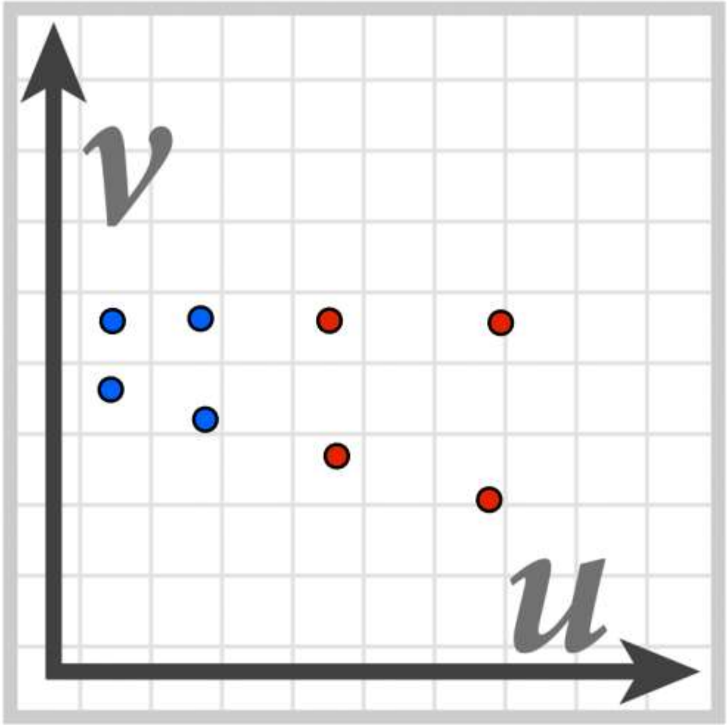


Q:mipmap的额外存储空间是多少？

计算像素的“覆盖”范围

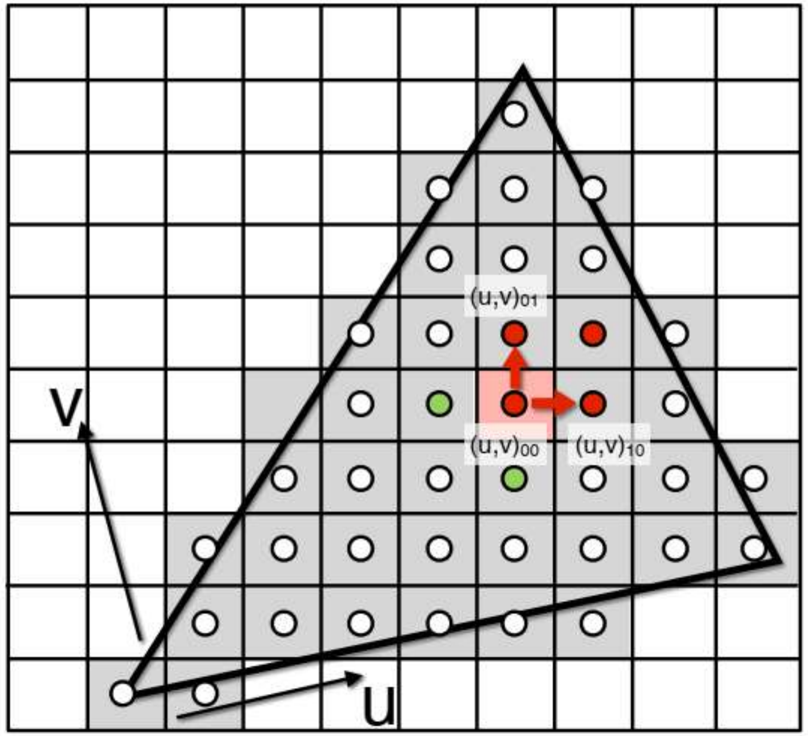


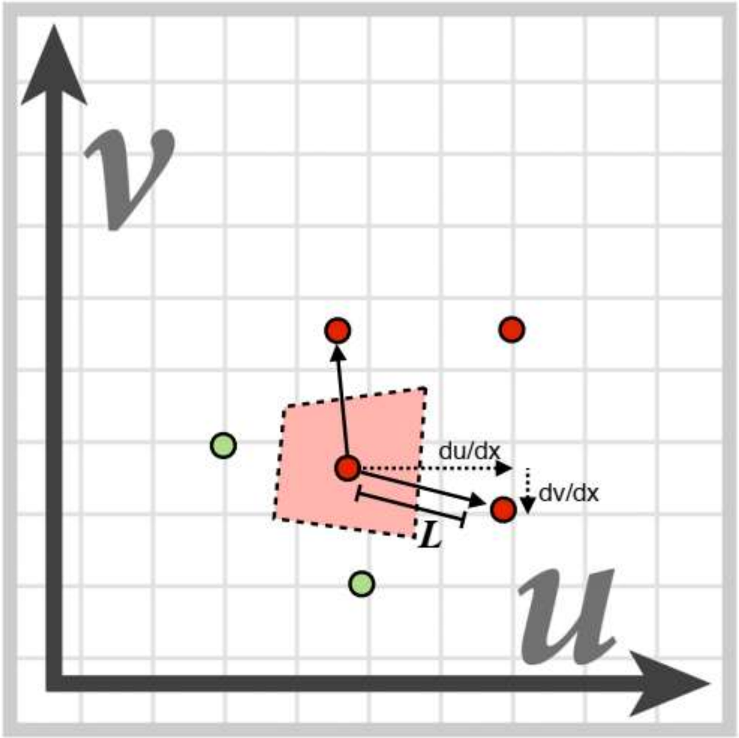
Screen space (x,y)



Texture space (u,v)

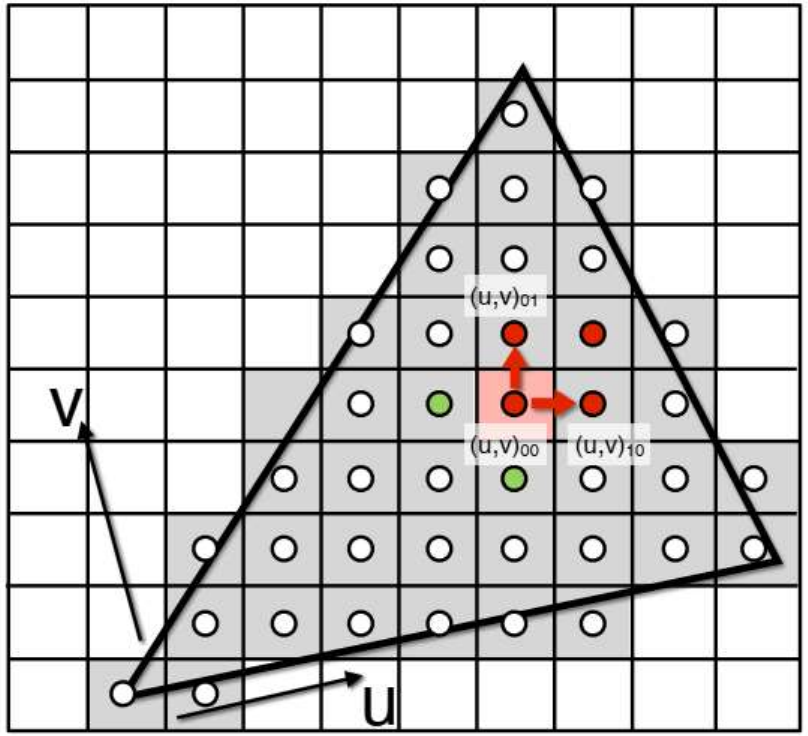
计算像素的“覆盖”范围

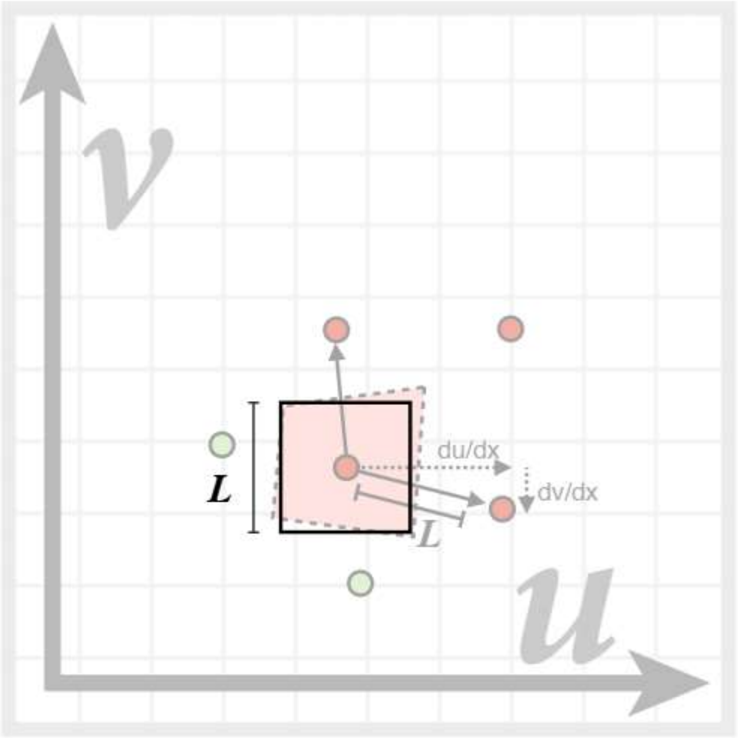




$$L = \max\left(\sqrt{\left(\frac{du}{dx}\right)^2 + \left(\frac{dv}{dx}\right)^2}, \sqrt{\left(\frac{du}{dy}\right)^2 + \left(\frac{dv}{dy}\right)^2}\right)$$

计算像素的“覆盖”范围

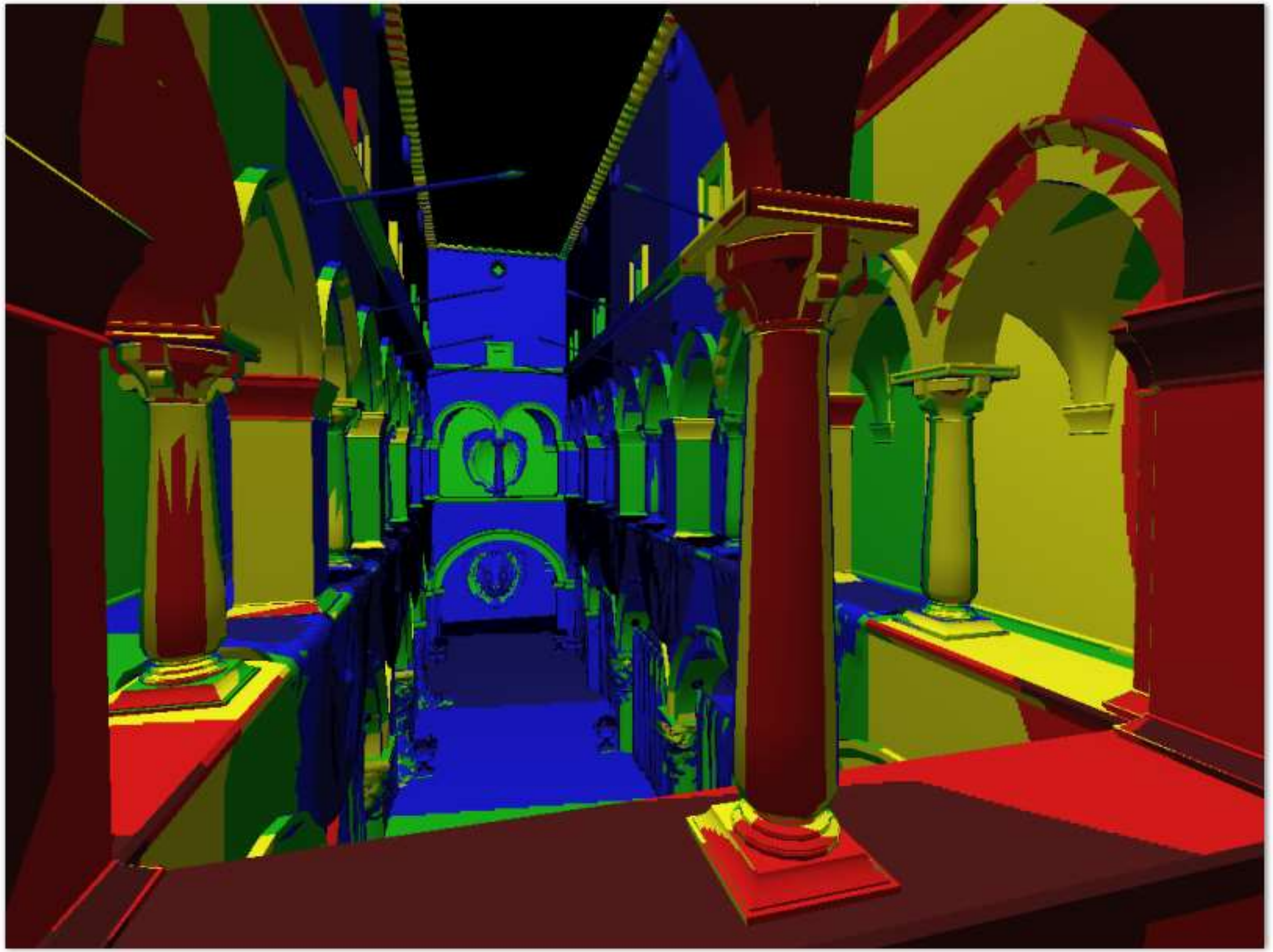




$$D = \log_2 L$$

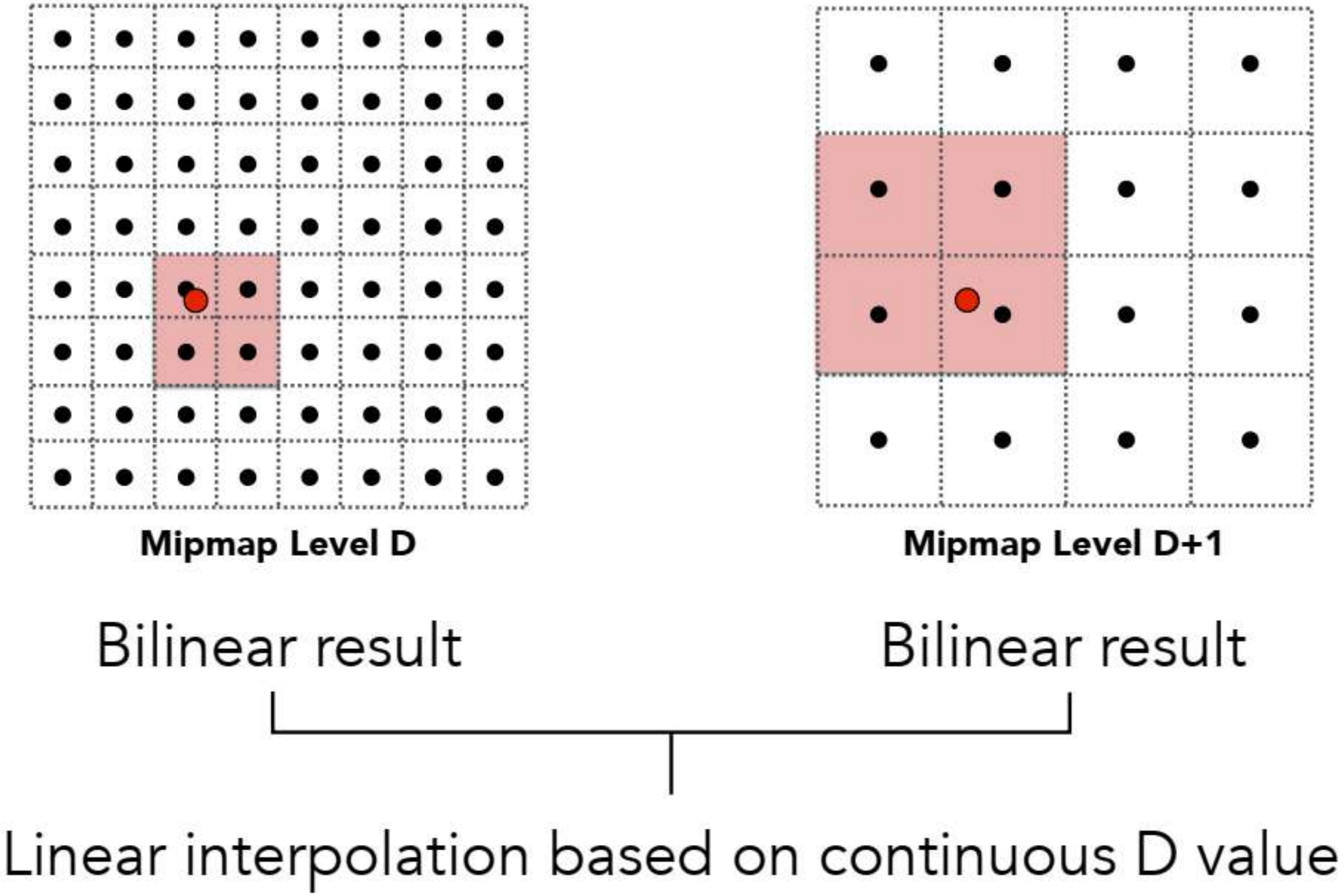
$$L = \max\left(\sqrt{\left(\frac{du}{dx}\right)^2 + \left(\frac{dv}{dx}\right)^2}, \sqrt{\left(\frac{du}{dy}\right)^2 + \left(\frac{dv}{dy}\right)^2}\right)$$

Mipmap level可视化



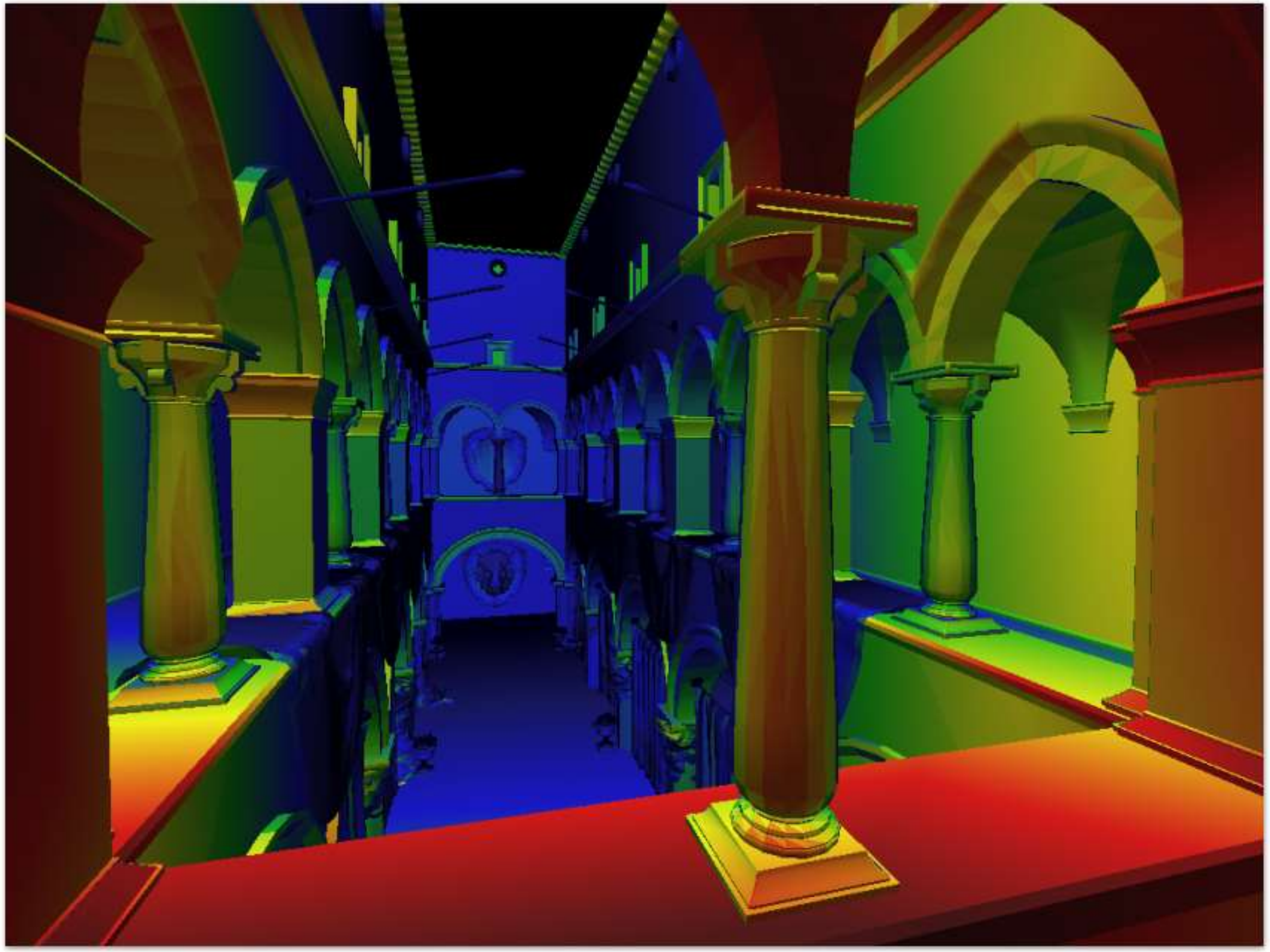
D rounded to nearest integer level

三线性插值



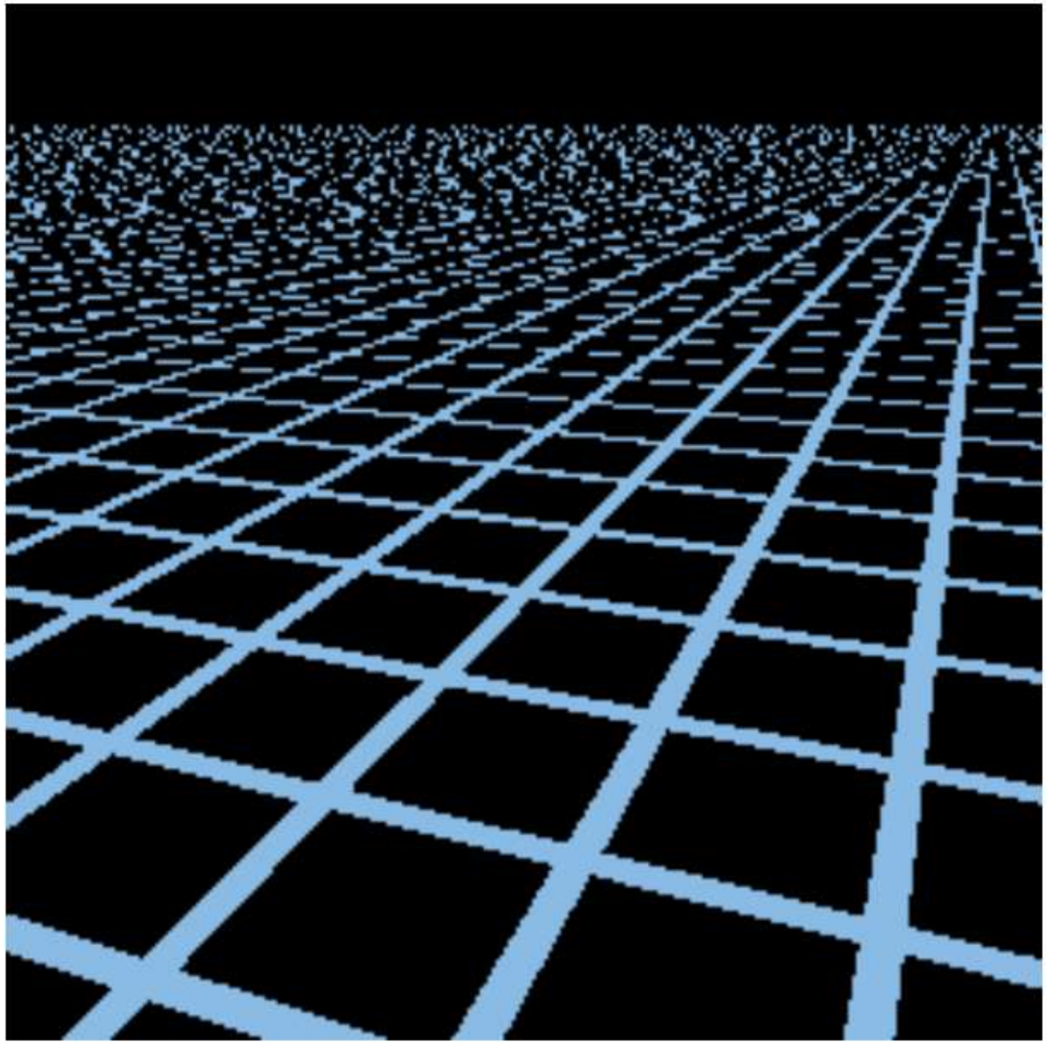
Mipmap Level D
Mipmap Level D+1
Bilinear result
Bilinear result
Linear interpolation based on continuous D value

Mipmap level可视化



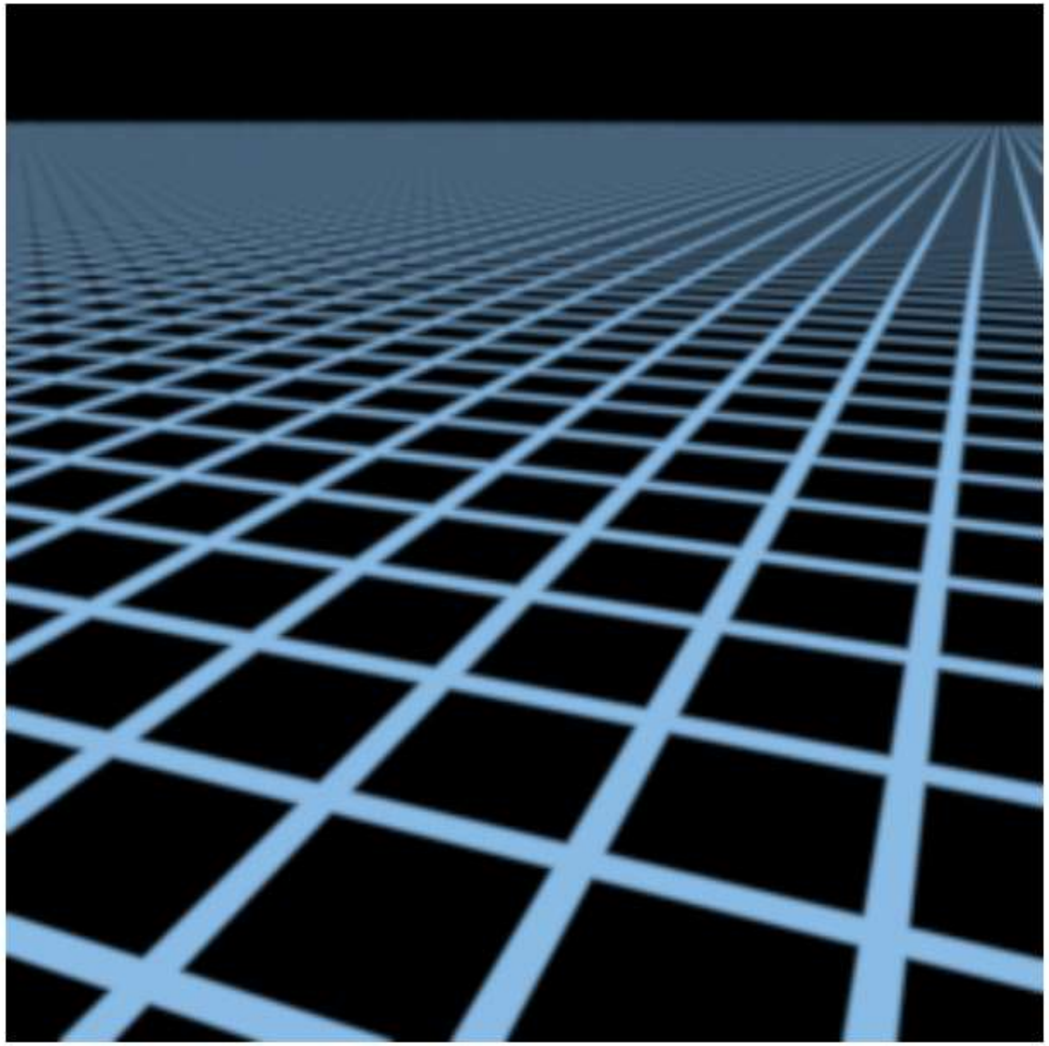
Trilinear filtering: visualization of continuous D

Mipmap的问题



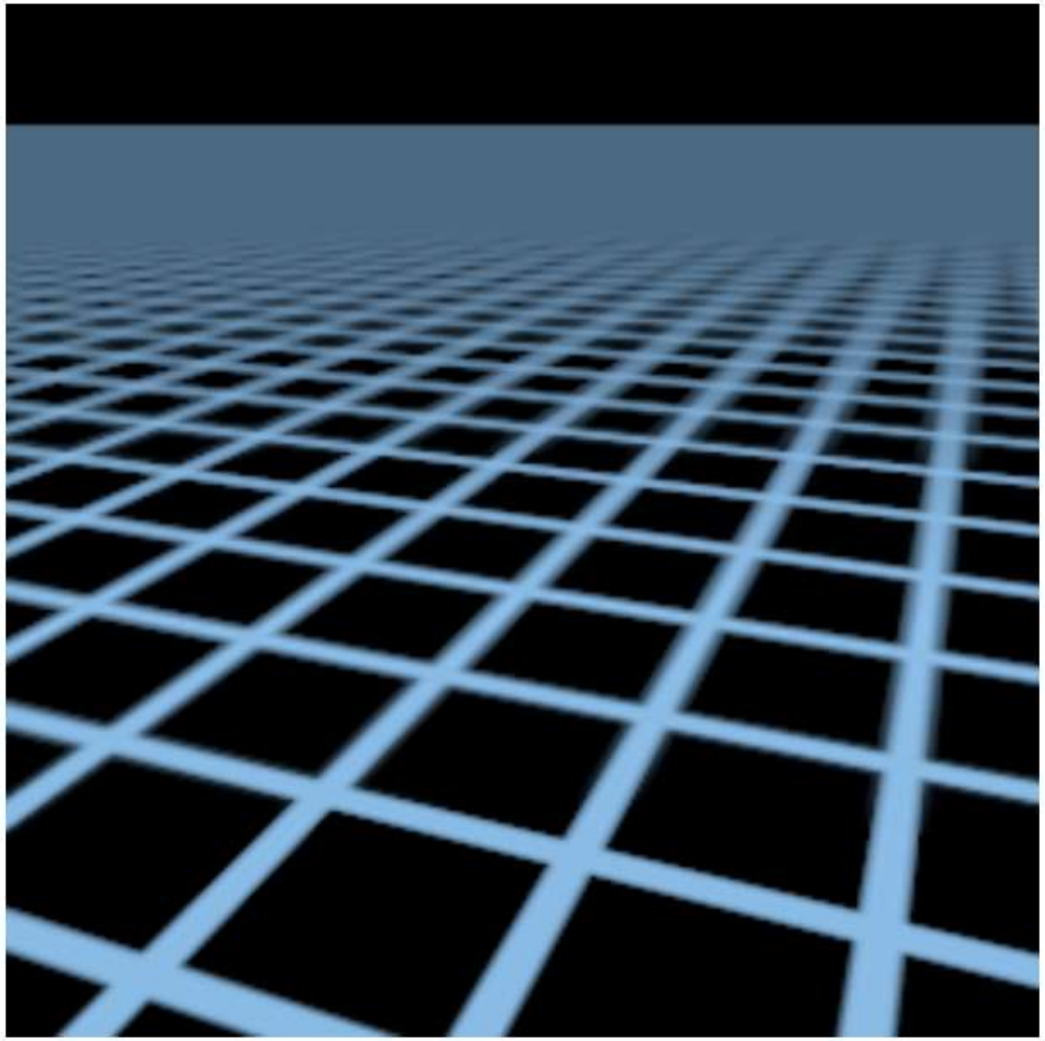
Point sampling

Mipmap的问题



Supersampling 512x (assume this is correct)

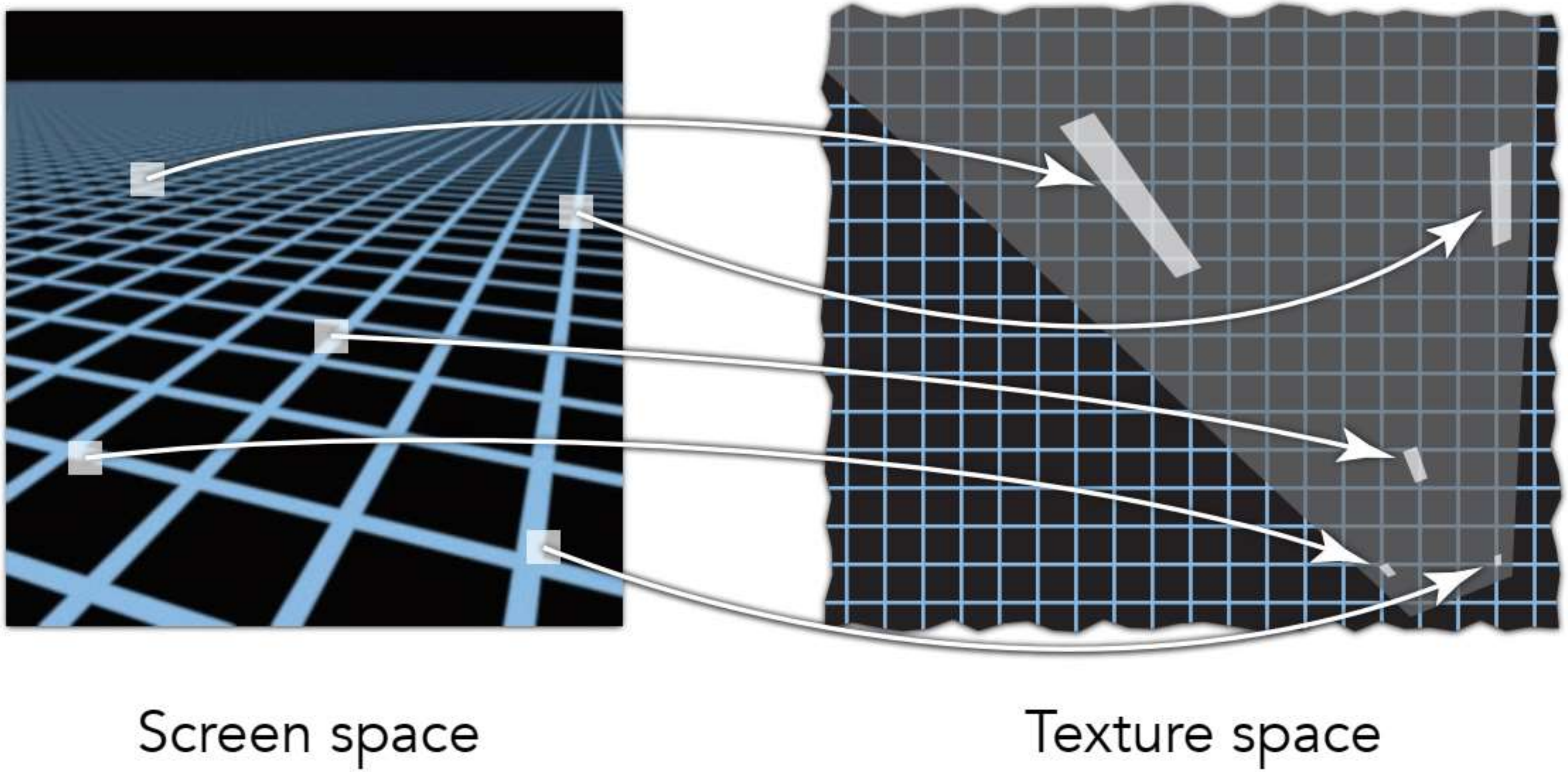
Mipmap的问题



Mipmap trilinear sampling

过度模糊!

Mipmap的问题

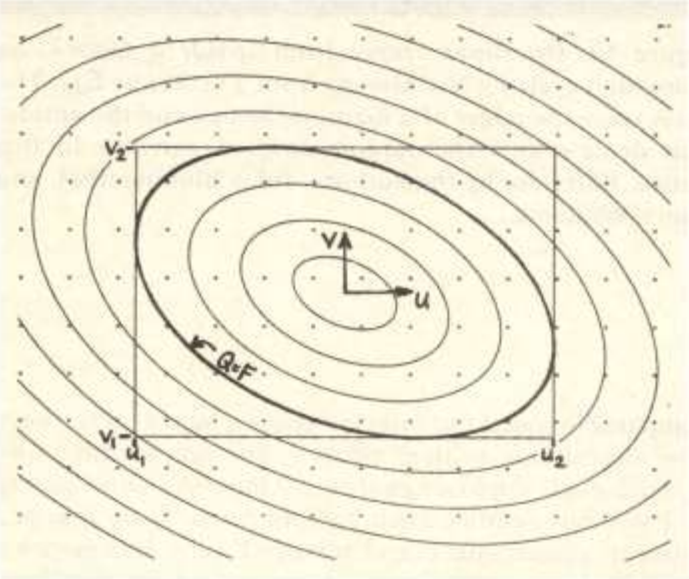


Mipmap的问题

- 各向异性过滤（Anisotropic Filtering）
  - ➢ 可以查找轴对齐的矩形区域
  - ➢ 存储空间增大

Wikipedia

- EWA过滤
  - ➢ 可以处理不规则区域
  - ➢ 多次查询



Greene & Heckbert ‘86

OCEAN UNIVERSITY OF CHINA 中國海洋大學

Q&A

中國海洋大學
OCEAN UNIVERSITY OF CHINA