

备忘录_几何着色器

笔记本: DirectX 12

创建时间: 2022/8/27 14:19

更新时间: 2022/9/3 10:02

作者: handsome小赞

- **几何着色器**阶段位于顶点着色器与像素着色器阶段之间

```
[maxvertexcount(4)]
void ShaderName (
    PrimitiveType InputVertexType InputName [NumElements]
    inout StreamOutputObject<OutputVertexType> OutputName )
{
    ... // 几何着色器的具体实现
}
```

- **[maxvertexcount (N)]**: 几何着色器单次调用所输出的顶点数量最大值 (每个图元都会调用一次几何着色器)

N: 1~20 (最佳)

N: 27~40 (性能下降50%)

- **输入参数**

输入参数必须是一个定义有特定图元的顶点数组

输入顶点类型即为顶点着色器输出的顶点类型 (例如 VertexOut)

输入参数前缀:

1. point: 输入图元为点 (1个顶点)
2. line: 输入图元为线列表或线条带 (2个顶点)
3. triangle: 输入的图元为三角形列表或三角形带 (3个顶点)
4. lineadj: 输入的图元为线列表及其邻接图元, 或线条带及其邻接图元 (4个顶点)
5. triangleadj: 输入的图元为三角形列表及其邻接图元, 或三角形带及其邻接图元 (6个顶点)

- **输出参数**

流类型:

1. PointStream<OutputVertexType> : 一系列顶点所定义的点列表
 2. LineStream<OutputVertexType> : 一系列顶点所定义的线条带
 3. TriangleStream<OutputVertexType> : 一系列顶点所定义的三角形带
- 向输出流列表添加单个顶点**

```
void StreamOutputObject<OutputVertexType>::Append (OutputVertexType  
v) ;
```

线条列表和三角形列表的实现

RestartStrip 函数表示结束当前三角形带的绘制

```
void StreamOutputObject<OutputVertexType>::RestartStrip() ;
```

- 精灵 sprite

一种不经渲染流水线而直接绘制到渲染目标的2D位图。教材中的广告牌实为应用于 3D环境中的精灵，是具有alpha通道且面向摄像机的图像。

- SV_PrimitiveID

```
[maxvertexcount(4)]  
void GS(point VertexOut gin[1],  
        uint primID:SV_PrimitiveID,  
        inout TriangleStream<GeoOut> triStream)
```

- 若指定了该语义，输入装配阶段会自动为每个图元生成图元ID。0 ~ n-1。每次绘制调用，图元的ID都是唯一的。
- 如果代码中不存在几何着色器，可以将图元ID参数加入像素着色器的参数列表

- SV_VertexID

输入装配阶段也能生成顶点ID。需要向顶点着色器签名额外添加一个由 SV_VertexID 修饰的 uint 类型的参数

- 纹理数组

由 ID3D12Resource 接口来表示

DepthOrArraySize 属性指定元素个数
对于3D纹理来说，此项设定则为深度值

在 HLSL中，纹理数组是通过 Texture2DArray 类型来表示

- **注意** 若使用 Texture2D TexArray[4] 这种形式，在 Direct3D 12之前是不行的，而且像这样的索引纹理可能会依具体硬件而产生少量开销。

- 纹理数组的采样需要三个坐标值，UVW

- **加载纹理数组**

利用 DDSTextureLoader.h/.cpp 中的代码，可以加载纹理数组的DDS文件。

利用微软的 texassemble 工具可以创建含有纹理数组的DDS文件

- 纹理子资源

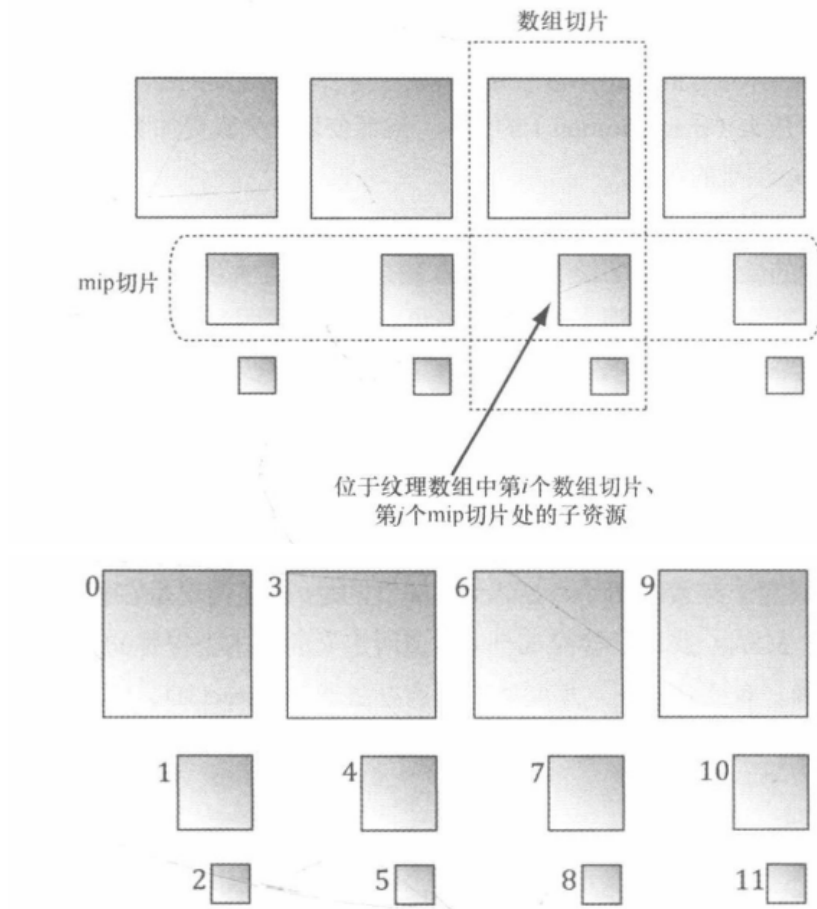


图 12.9 在纹理数组中由线性索引来标记的子资源

- alpha-to-coverage

传统MSAA并不会检测 alpha 通道所定义的树木公告牌的剪裁边缘，而只是关注纹理所映射到的四边形的边缘。

开启 MSAA 和 alpha-to-coverage 后，硬件就会检测像素着色器返回的alpha值，并将其用于确定覆盖的情况。

D3D12_BLEND_DESC::AlphaToCoverageEnable = true