

计算机图形学小白入门

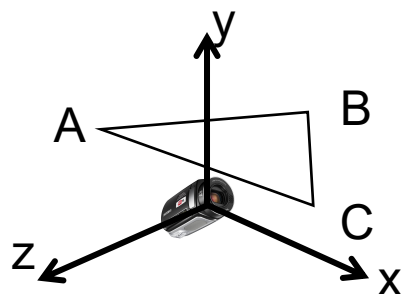
——从0开始实现OpenGL

渲染管线设计

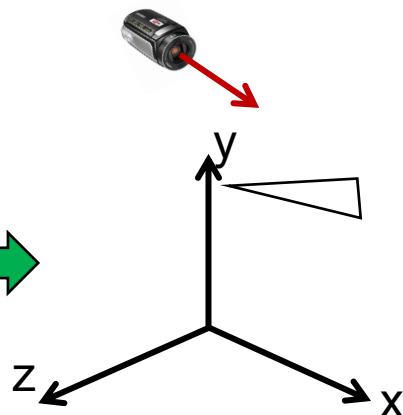


授课：赵新政
资深三维工程师

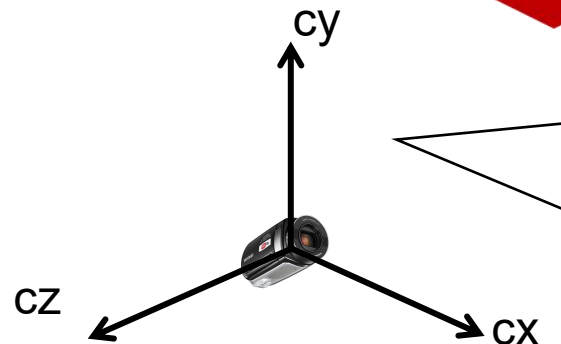
专注3D图形学技术
教育品牌



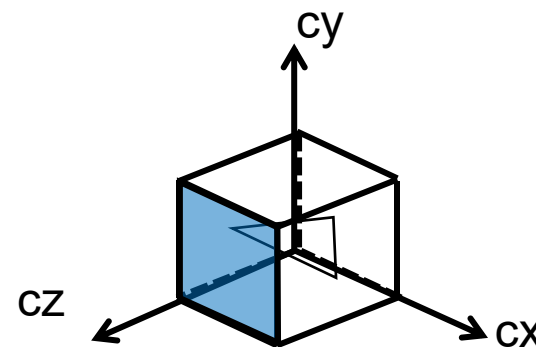
初始化三角形
 • **顶点**位置
 • 每个**顶点**属性
 初始化摄像机



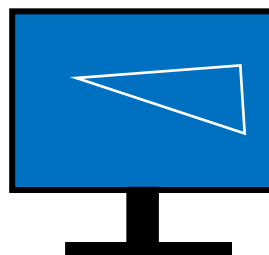
使用三角形的modelMatrix对每个**顶点**进行变换
 使用摄像机的modelMatrix对摄像机进行变换 (求matrix)



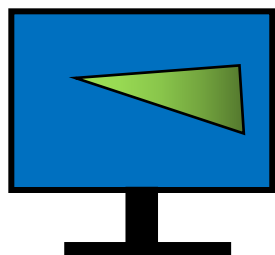
使用摄像机viewMatrix (模型变换逆矩阵) 将每个**顶点**变换到摄像机坐标系下



使用投影矩阵将三角形**顶点**变化到剪裁坐标系, 并且进行透视除法, 到达NDC坐标, 即-1到+1



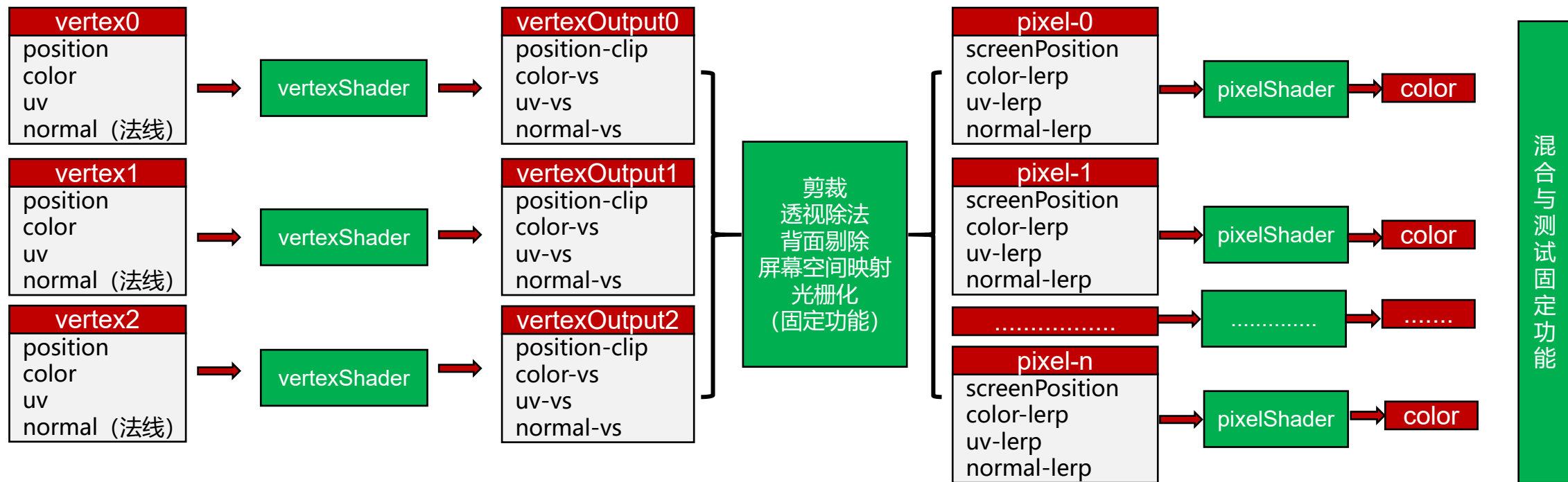
通过屏幕空间变换矩阵 (screenMatrix), 把三角形的**顶点**变换到屏幕空间像素位置



使用顶点屏幕坐标, 对三角形光栅化
 • 得到所有像素, 及属性插值结果
 • 根据每个像素数据, 对应渲染

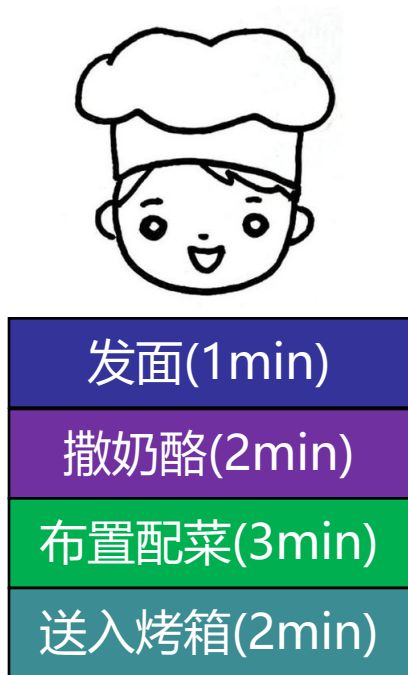
初级架构梳理

- 在进行屏幕光栅化之前，所有的操作都是**针对模型顶点**
- 每一步的操作都是在**前一步得到的顶点数据**基础上进行
- 可以把**对顶点的操作过程（mvp）**设计成一个**单独的模块**，把所有**对像素的操作过程**设计成一个**单独的模块**
- 处理每个顶点的程序我们称为**VertexShader（顶点着色器）**，处理每个像素的过程称为**PixelShader（像素着色器）**，这两个着色器程序可以设计为**可扩展/可编程**的过程，而**中间（剪裁/透视除法/背面剔除/屏幕空间映射/光栅化）**过程我们固化掉

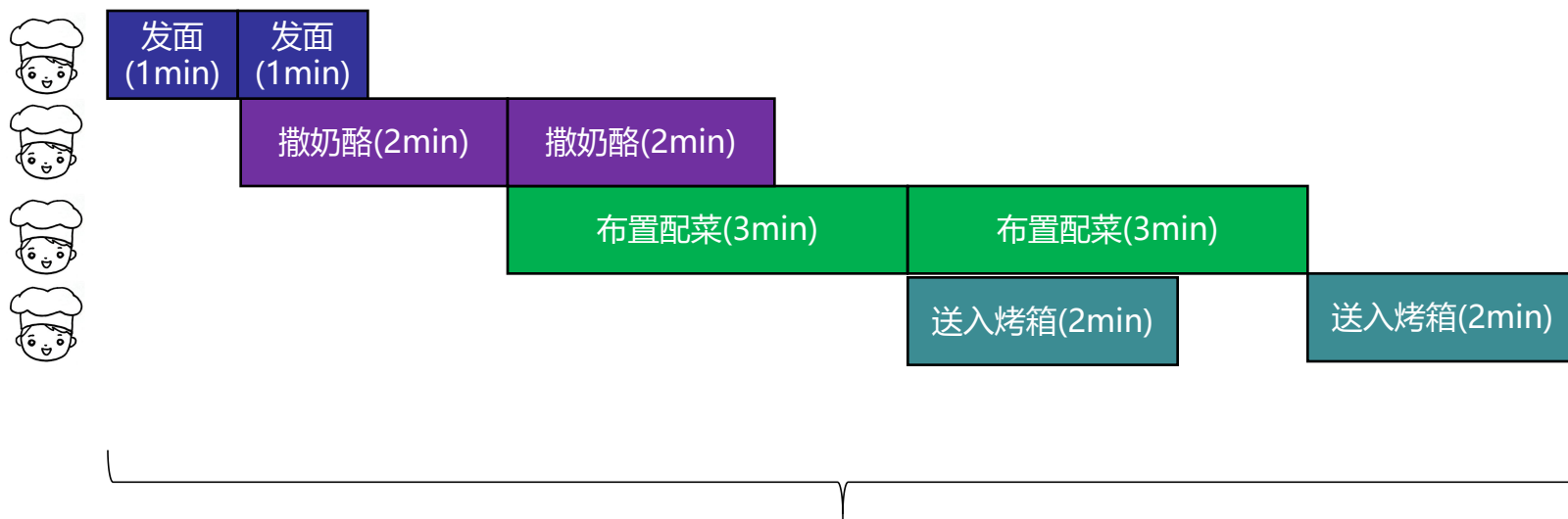


流水线介绍

- 流水线：将可以复杂任务分成独立任务模块，每个执行者专注一个模块，从而形成并行化生产。



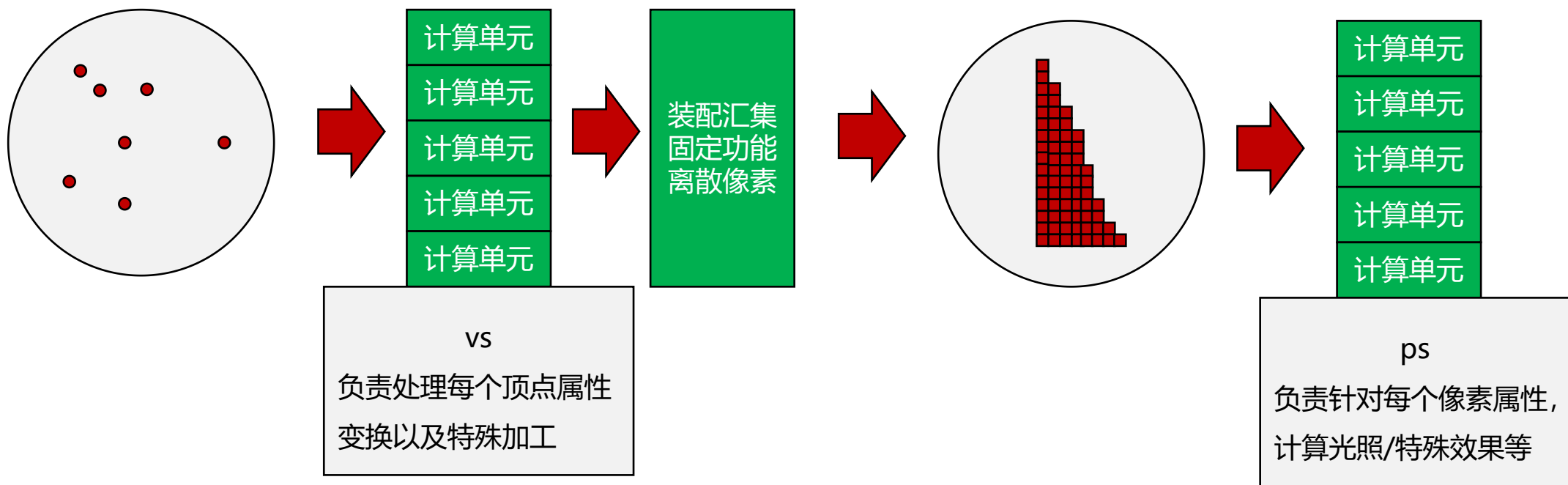
两次任务耗时：
 $8\text{min} * 2 = 16\text{min}$



两次任务耗时：
 $1\text{min} + 2\text{min} + 3\text{min} + 3\text{min} + 2\text{min} = 11\text{min}$

渲染管线

- 在GPU当中，每个顶点都是单独通过VertexShader进行处理；每个像素也会单独通过PixelShader进行处理
- 每个顶点单独处理；每个像素单独处理；则可以进行并行化计算，构成流水线，称为**渲染管线**



固定管线功能

剪裁 (Clip) : 输入剪裁空间坐标, 考察所有三角形与可视区域的关系, 并根据是否相交/包含进行剪裁

VertexShader

剪裁阶段

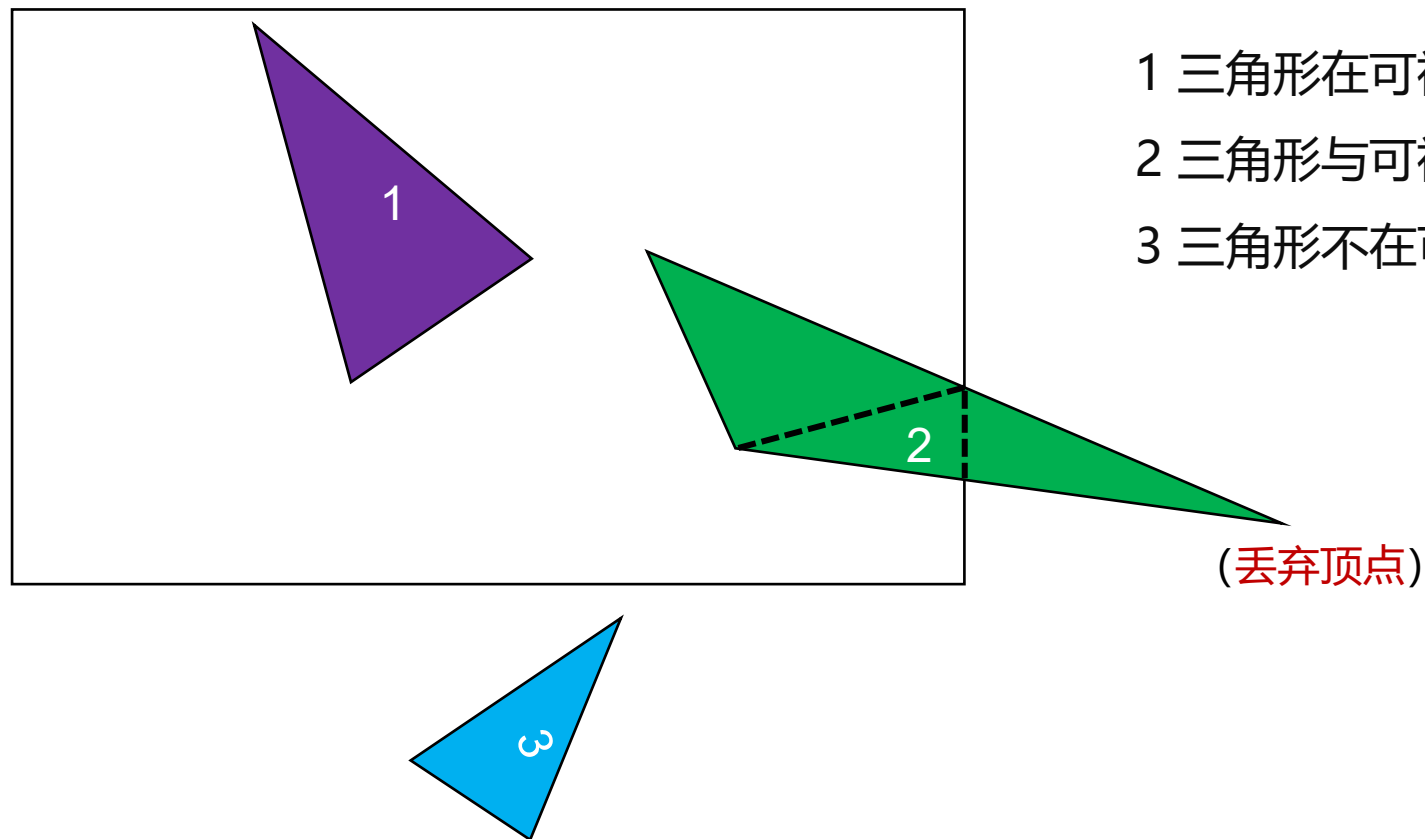
剔除阶段

图元装配

光栅化

PixelShader

混合与测试



固定管线功能

剔除 (Cull)： 输入透视除法后的NDC，去除背面朝向我们的三角形，留下正面朝向我们的（或者反过来）

VertexShader

剪裁阶段

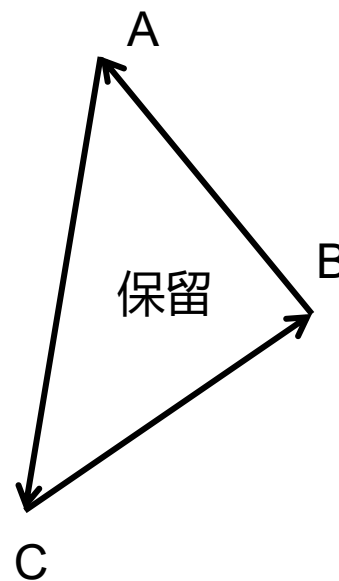
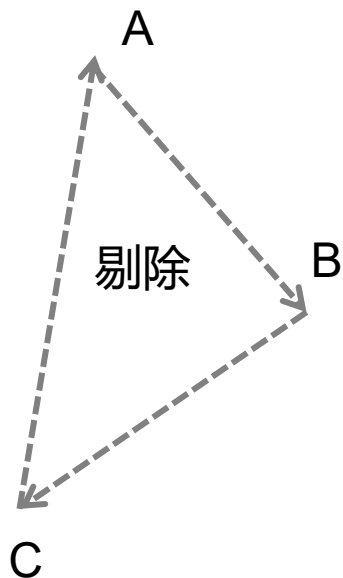
剔除阶段

图元装配

光栅化

PixelShader

混合与测试



固定管线功能

装配与光栅化： 输入屏幕空间坐标，把顶点按照图元（三角形/直线）进行组合，然后离散成栅格点的过程

顶点处理阶段

剪裁阶段

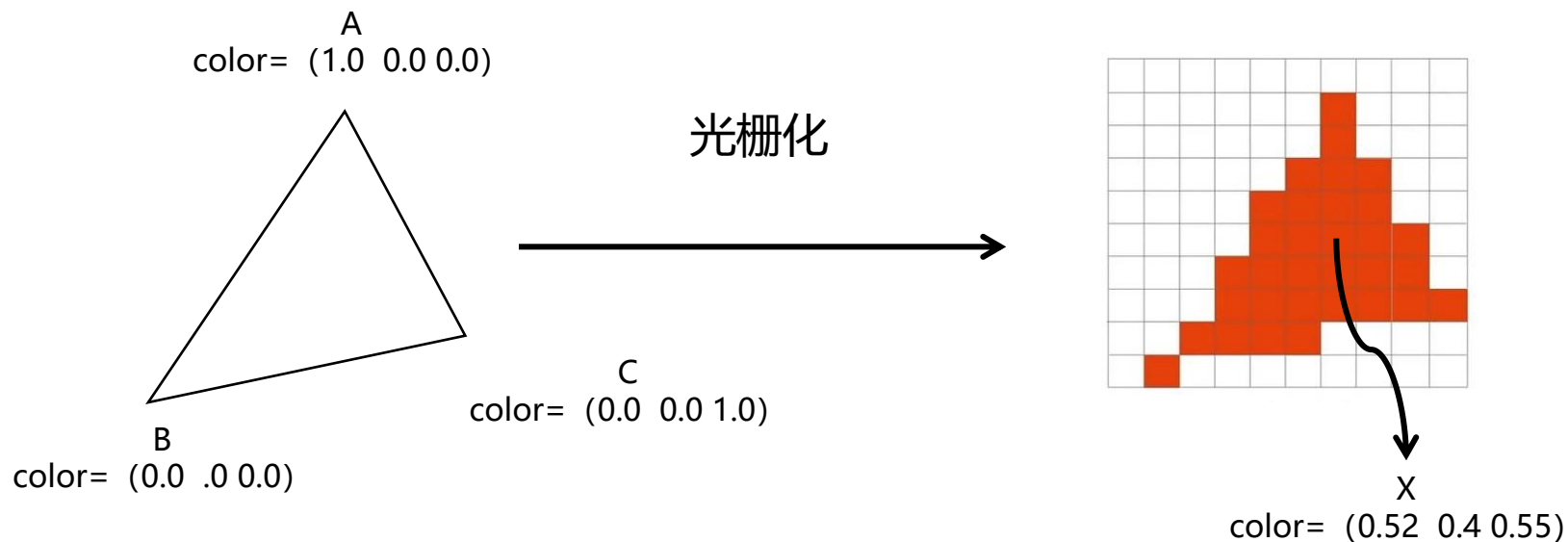
剔除阶段

图元装配

光栅化

PixelShader

混合与测试



注意：

每个顶点，除了位置信息，还有可能携带比如**颜色/法线/UV**等属性信息；在光栅化过程中，会通过“**插值**”得到每个栅格点内部的各个属性值（**透视矫正**）

固定管线功能

片元着色：输出的每个栅格点即片元，本步骤决定每个片元最终输出到屏幕的颜色

顶点处理阶段

剪裁阶段

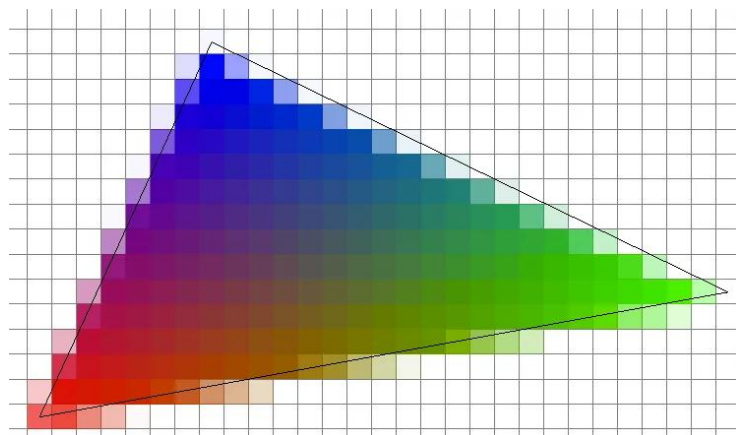
剔除阶段

图元装配

光栅化

PixelShader

混合与测试



原理：

每个栅格（**片元**）都记录着自己携带的属性信息（颜色/法线/uv等；每个栅格都会通过片元着色程序（**Pixel/Fragment Shader**）

固定管线功能

混合与测试：在每个三角形（or直线）的片元输出后，跟画布上已有的进行对比，进行深度测试/模板测试/透明颜色混合等

顶点处理阶段

剪裁阶段

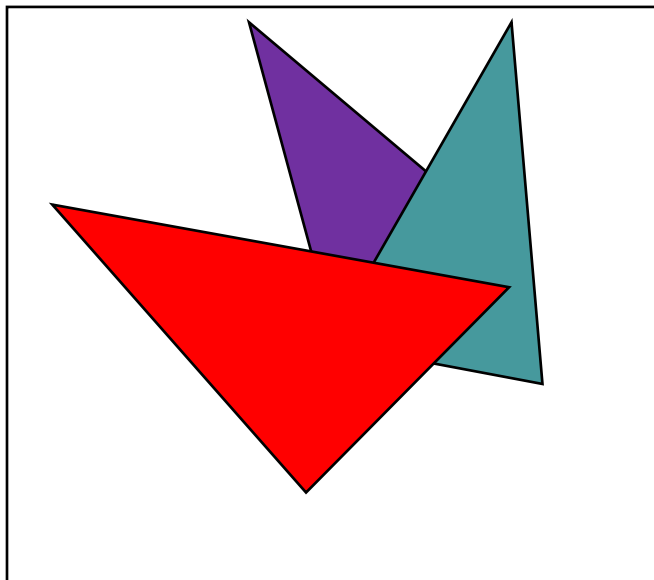
剔除阶段

图元装配

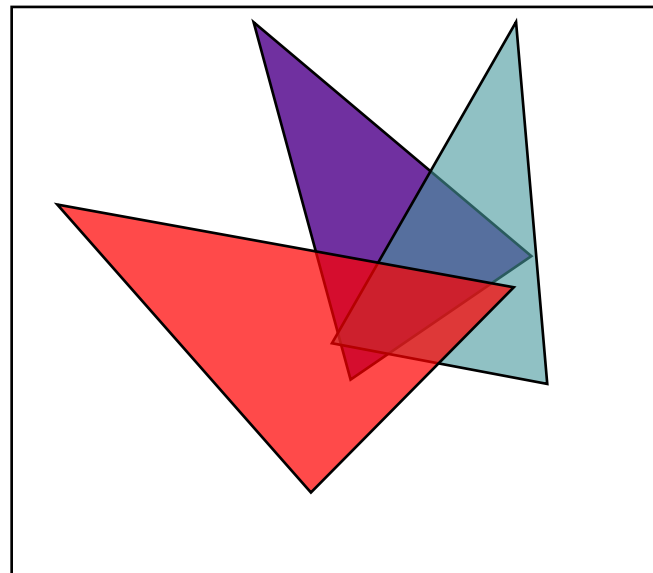
光栅化

片元着色

混合与测试



被挡住的片元会被丢弃



透明与下方的颜色混合显示