



福州大学
FUZHOU UNIVERSITY

第 1 章 第三节

图形绘制系统

基本内容



福州大学
FUZHOU UNIVERSITY

- 图形系统的基本设计
- 流水线体系结构
- 可编程流水线

来自“图像构造”的启示



福州大学
FUZHOU UNIVERSITY

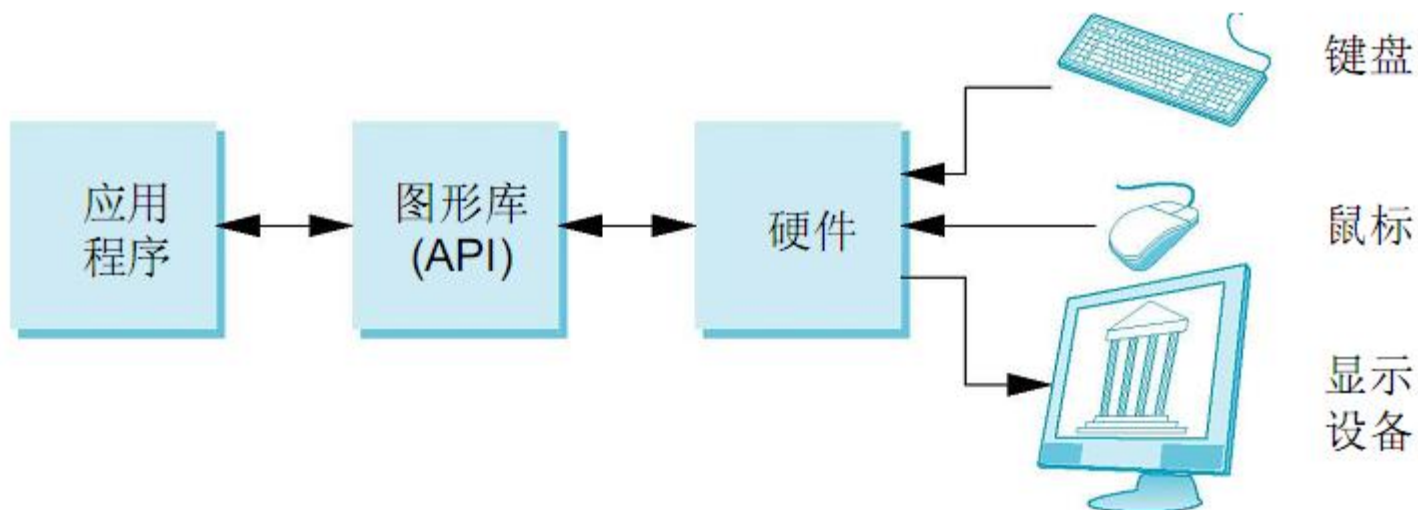
- 能否模拟组合照相机模型设计图形系统中的硬件和软件？
- 应用程序接口(API)
 - ✓ 只需指定
 - 对象
 - 材料
 - 观察者
 - 光源
- 如何实现API？

应用程序接口



福州大学
FUZHOU UNIVERSITY

- 程序设计人员是通过软件接口接触图形系统的，这个接口就是应用程序接口（**API**）。



应用程序开发人员的软件模型图

三维API软件包



- 虚拟照相机模型是OpenGL、Direct3D、Java-3D、WebGL等常用API软件包的基础。
- 定义构成一幅图像应包含以下对象的函数
 - ✓ 对象
 - ✓ 观察者
 - ✓ 光源
 - ✓ 材料属性
- 其它信息
 - ✓ 从鼠标和键盘等设备获取输入
 - ✓ 系统的能力

几何对象的表示



- 绝大多数API支持一些基本几何对象，例如：
 - ✓ 点 **points**（零维对象）
 - ✓ 线段 **line segments**（一维对象）
 - ✓ 多边形 **polygons**（二维对象）
 - ✓ 某些曲线和曲面
 - 二次曲面 **quadrics**
 - 多项式参数曲面
- 所有基本形状都是通过空间中的位置或顶点 (**vertices**) 定义的。



示例--WebGL在JavaScript中指定

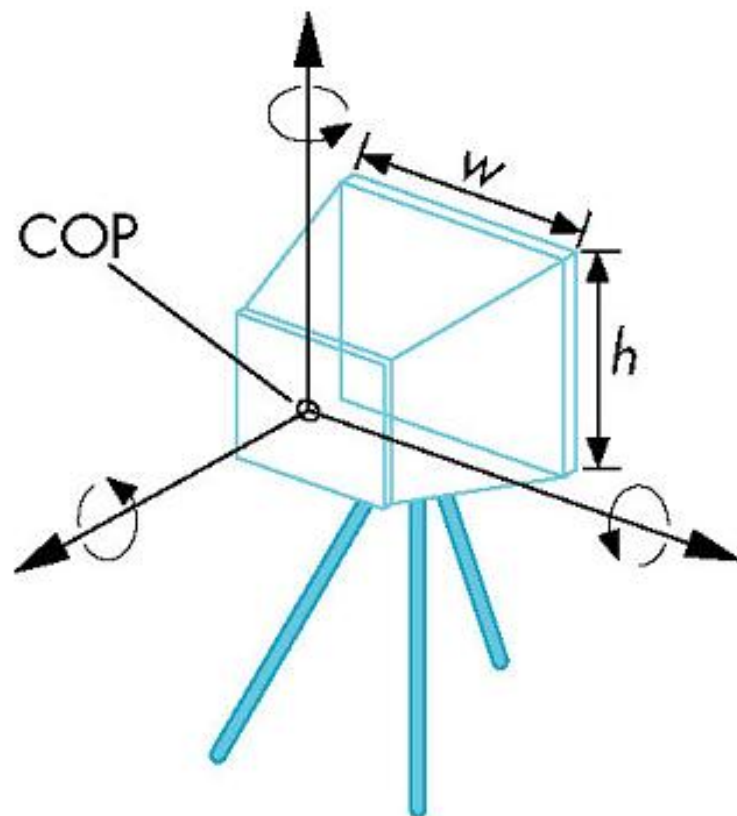
- 几何数据存储在数组中：

```
var vertices=[];  
vertices[0]=[0.0,0.0,0.0];//顶点A  
vertices[1]=[0.0,1.0,0.0];//顶点B  
vertices[2]=[0.0,0.0,1.0];//顶点C  
或者  
var vertices=[];  
vertices.push([0.0,0.0,0.0]);//顶点A  
vertices.push([0.0,1.0,0.0]);//顶点B  
vertices.push([0.0,0.0,1.0]);//顶点C
```

这些数据存储在CPU，然后传送至GPU进行绘制

照相机的指定

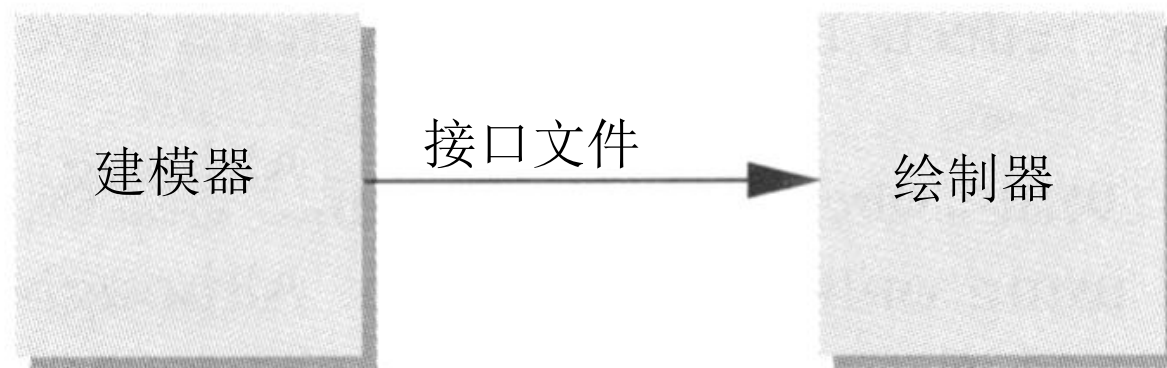
- 定义观察者或照相机需确定4类参数（六个自由度）
 - ✓ 镜头中心的位置，即投影中心（COP）
 - ✓ 方向
 - ✓ 镜头 焦距
 - ✓ 胶卷平面的方向





-
- `gluLookAt(cop_x, cop_y, cop_z,
at_x, at_y, at_z,
up_x, up_y, up_z);`
 - `glPerspective(field_of_view,
aspect_ratio,
near, far);`

- 场景建模与场景绘制分开



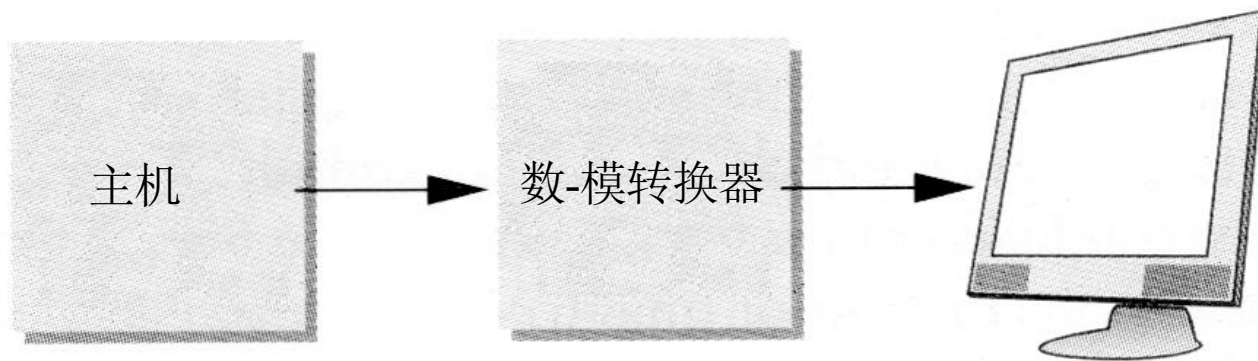
建模-绘制流水结构图

图形系统的体系结构



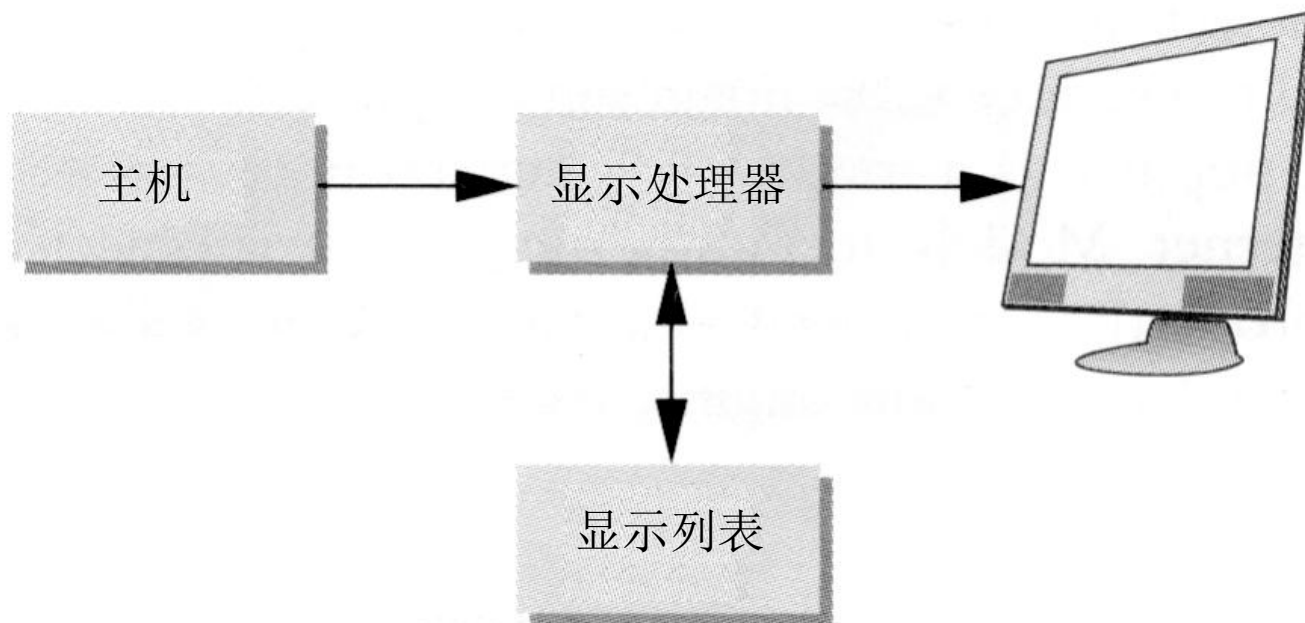
福州大学
FUZHOU UNIVERSITY

- 早期的图形系统



早期的图形系统图

显示处理器

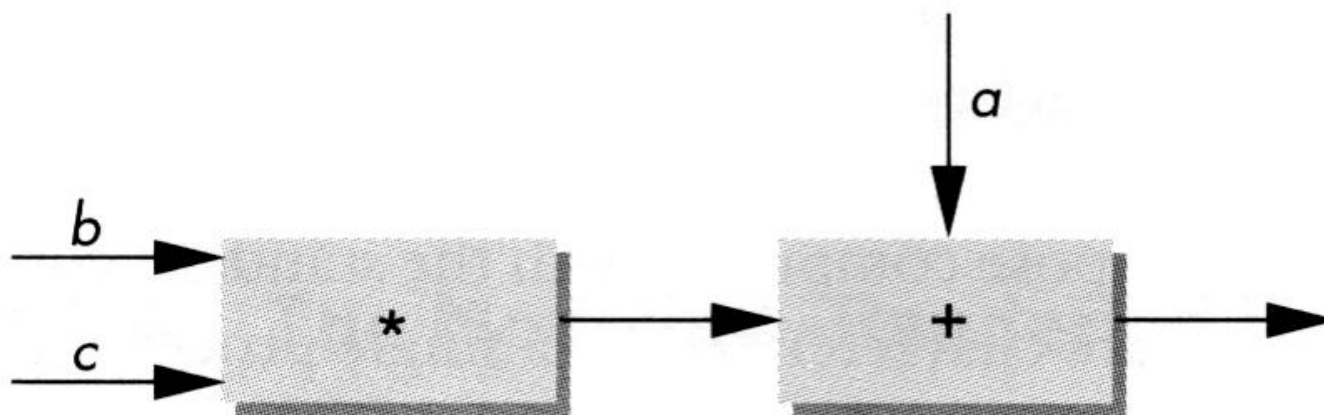


显示处理器的体系结构图

流水线体系结构



- 举例：计算一组数据的 $a + (b * c)$ 运算

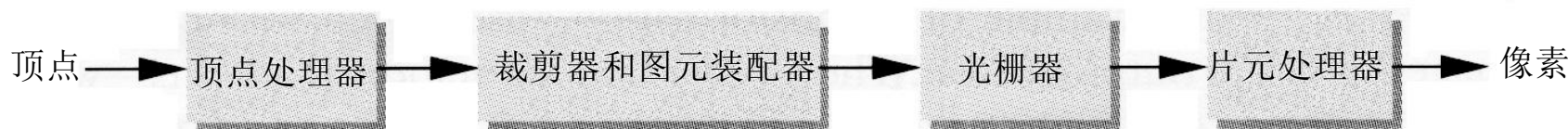


算术运算流水线图

图形处理流水线



- 一个比较复杂的场景，需数十万个，甚至百万个顶点来定义几何对象。
- 按照应用程序定义对象的先后顺序，依次处理每个对象
 - ✓ 只能考虑局部光照
- 流水线体系



图形处理流水线



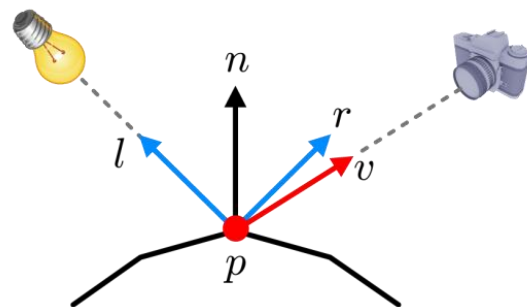
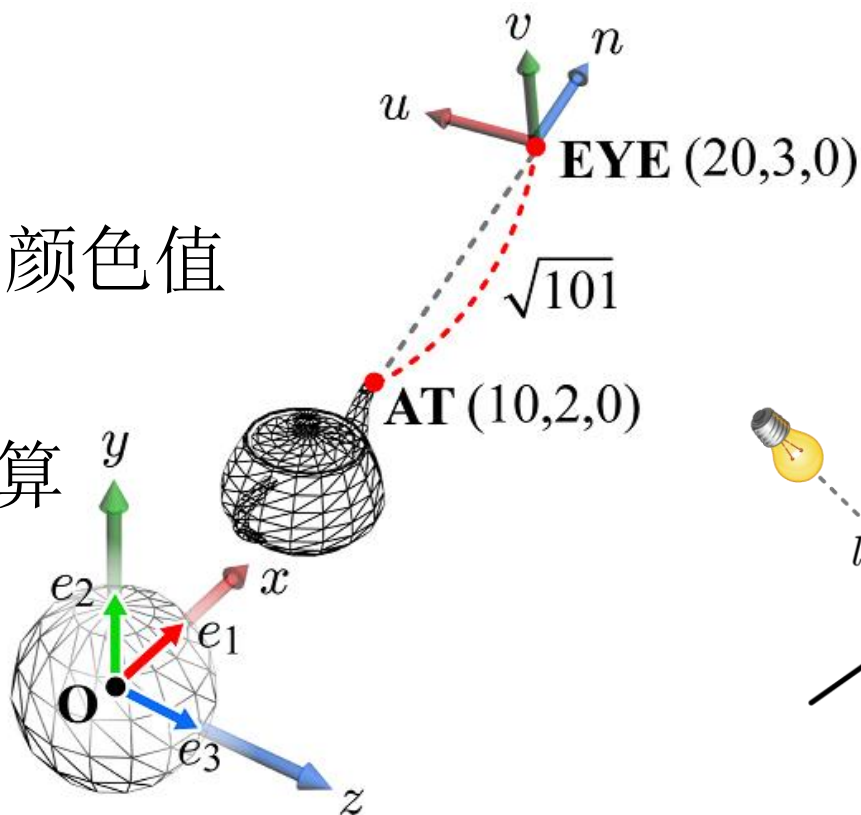
- 流水线中大部分工作是把对象在一个坐标系中表示转化为另一坐标系中的表示：
 - ✓ 对象坐标系或者建模坐标系
 - ✓ 世界坐标系
 - ✓ 照相机坐标系
 - ✓ 裁剪坐标系
 - ✓ 规范化设备坐标系
 - ✓ 屏幕坐标系
- 坐标的每个变换相当于一次矩阵乘法
- 给顶点赋颜色。

顶点处理



福州大学
FUZHOU UNIVERSITY

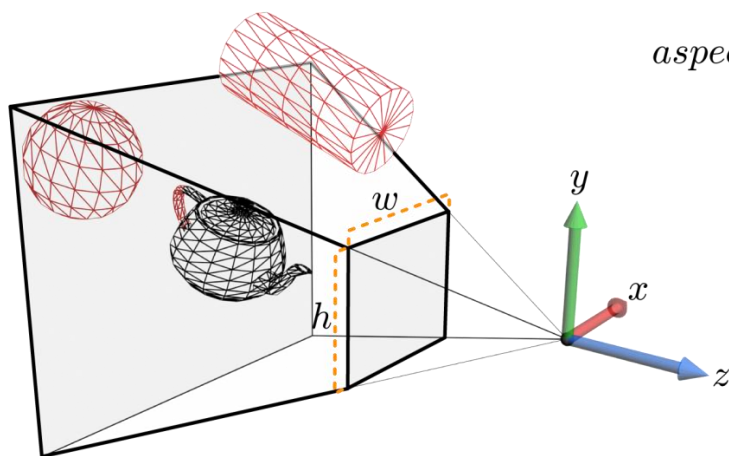
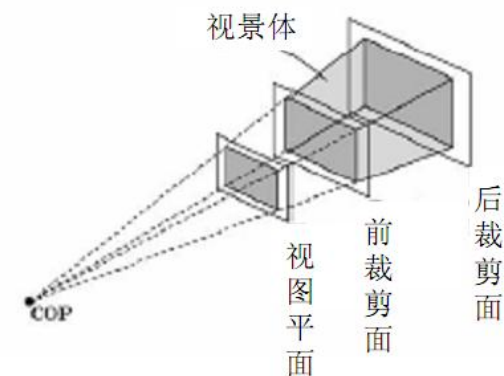
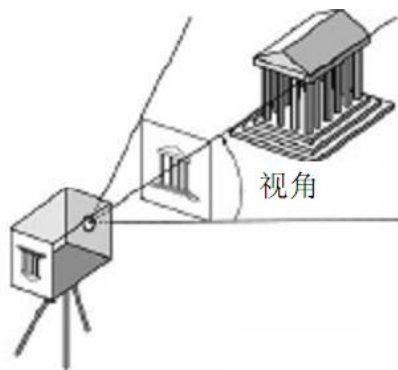
- 执行坐标变换
 - ✓ 世界坐标系到照相机坐标系
 - ✓ 投影变换
- 计算每个顶点的颜色值
 - ✓ 简单指定
 - ✓ 光照模型计算



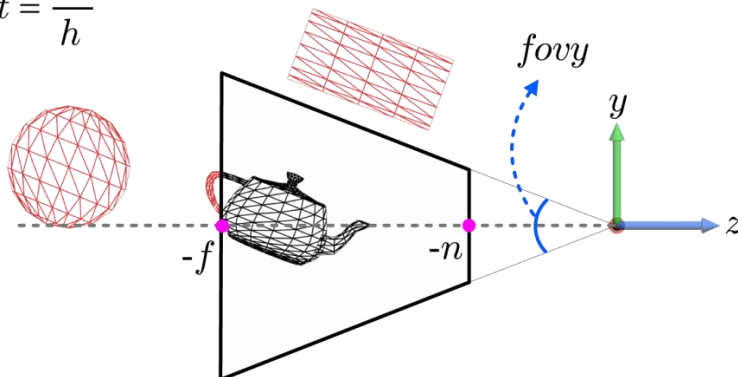
裁剪与图元组装



- 真正的照相机不能“看到”整个世界，图形学中虚拟的照相机也只能看到世界的一部分
 - 不在下述视景体中的对象要从场景中裁剪掉



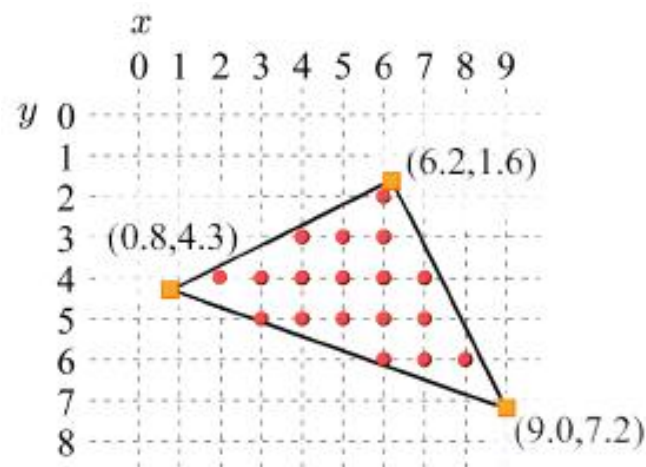
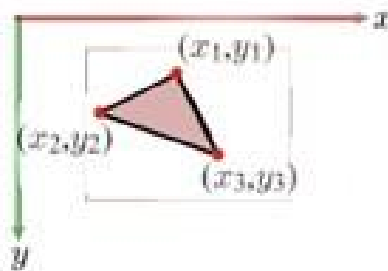
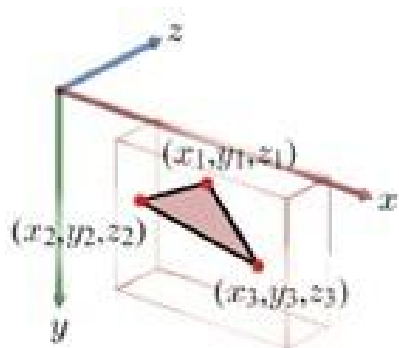
$$aspect = \frac{w}{h}$$



光栅化



- 如果一个对象在图像中可见，那么就需要对帧缓冲区某些的像素赋以指定的颜色，边形内部被指定颜色或模式填充
- 光栅化模块对每个图元输出一组片元（**fragment**）
- 片元可以携带颜色、位置、深度等信息

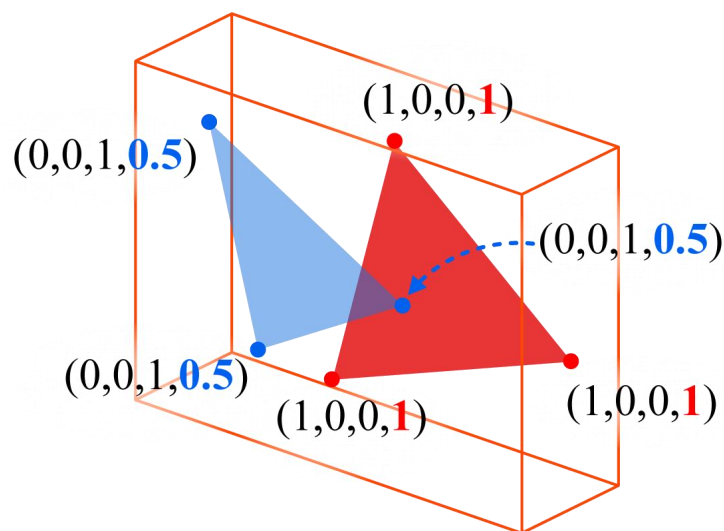


(b)

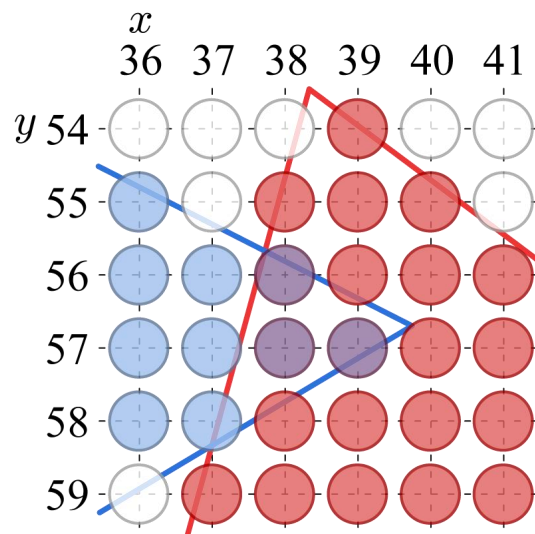
片元处理



- 接收来自光栅器的片元并更新帧缓存的像素。
- 片元的颜色可能受纹理映射或凹凸映射的作用而改变
- 消隐处理



(a)

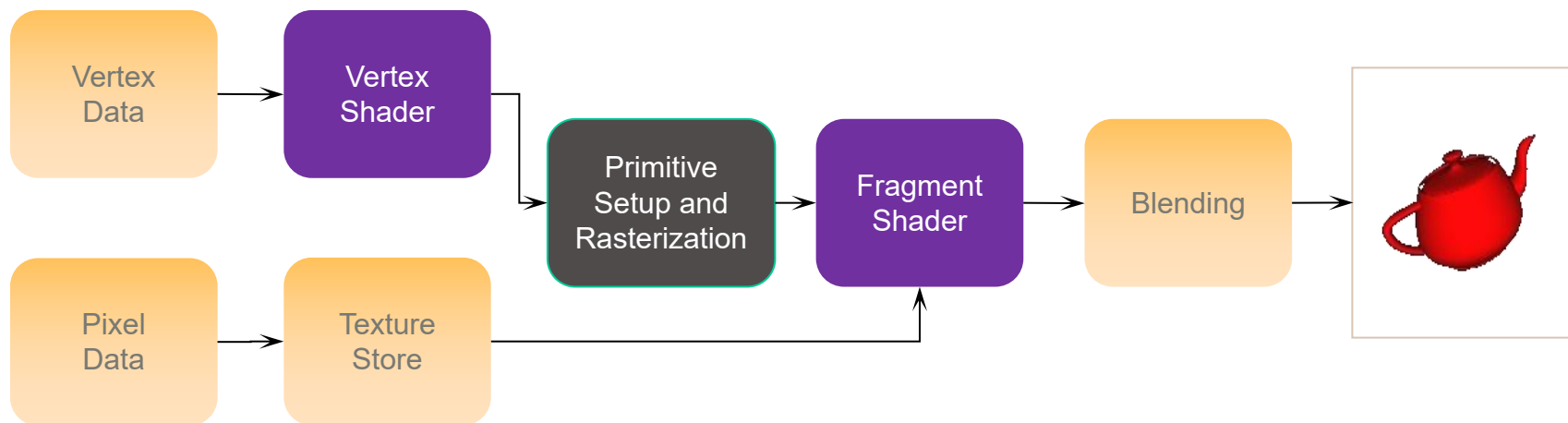


(b)

片元处理--融合图

可编程流水线

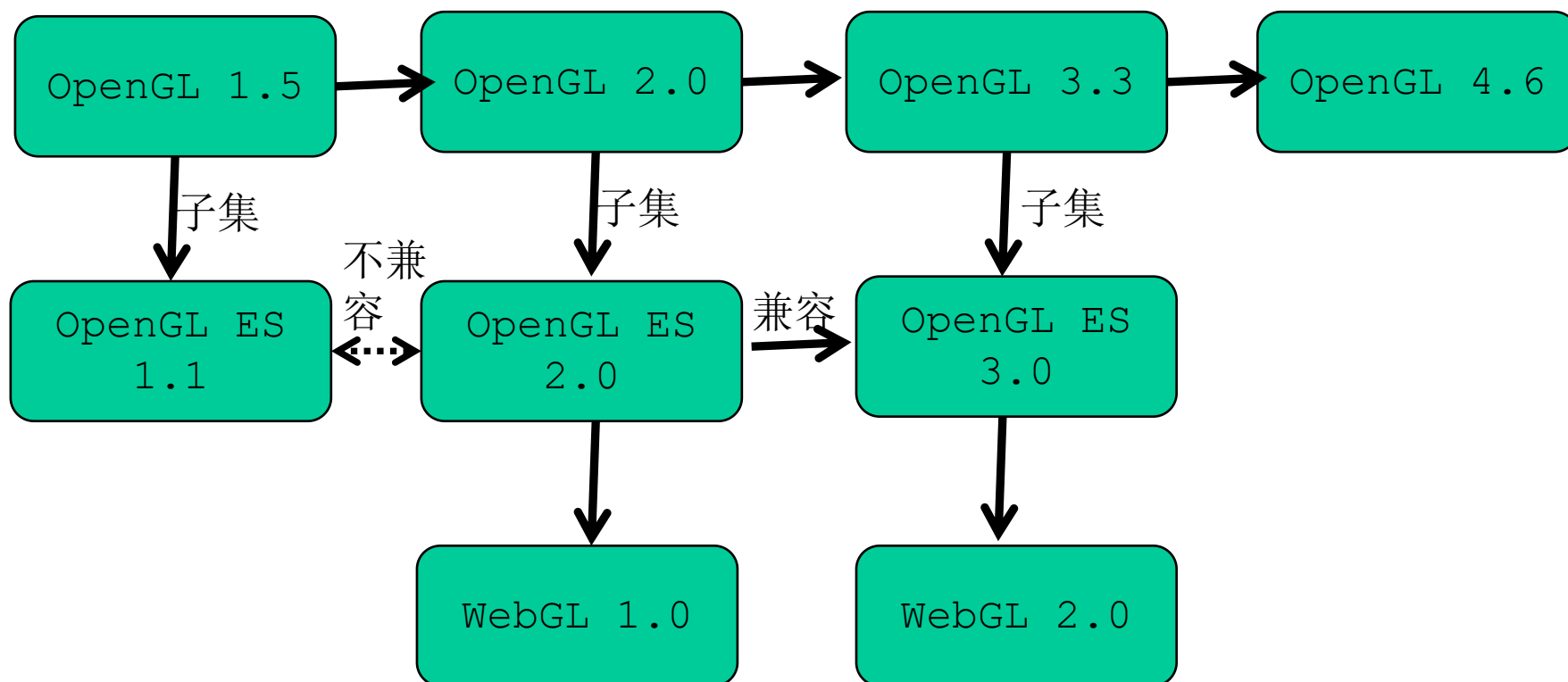
- 很多年来，流水线体系结构都具有固定功能
 - 近年来，流水线体系结构有一个较大进展：**顶点处理模块和片元处理模块可编程**
- 进而，许多技术可以实时实现！



OpenGL与WebGL



福州大学
FUZHOU UNIVERSITY



WebGL程序的结构



福州大学
FUZHOU UNIVERSITY

网页

HTML

JavaScript

HTML
渲染引擎

浏览器

使用WebGL的网页

HTML5

JavaScript

GLSL ES

HTML
渲染引擎

WebGL

OpenGL/OpenGL ES

浏览器

传统的动态网页软件结构

WebGL网页软件结构图