

2018-2019年度第二学期 00106501

计算机图形学



童伟华 管理科研楼1205室

E-mail: tongwh@ustc.edu.cn

中国科学技术大学 数学科学学院

<http://math.ustc.edu.cn/>





第四节 图像流水线系统及编程接口

成像模型回顾

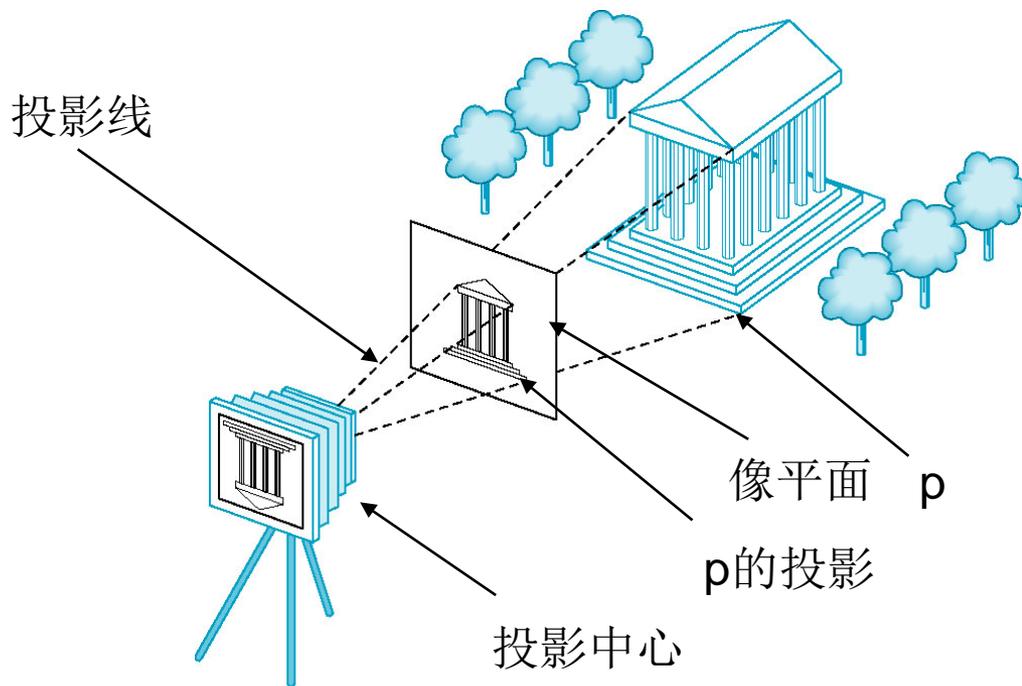


■ 能否模拟合成的照相机模型设计图形系统中的硬件和软件?

■ 应用程序界面

- 只需指定
 - 对象
 - 材料
 - 观察者
 - 光源

■ 如何实现API?



成像模型回顾



■ 全局光照模型

- 光线跟踪方法
- 辐射度方法
- 计算量非常大，目前还不适于实时或交互式系统

■ 局部光照模型

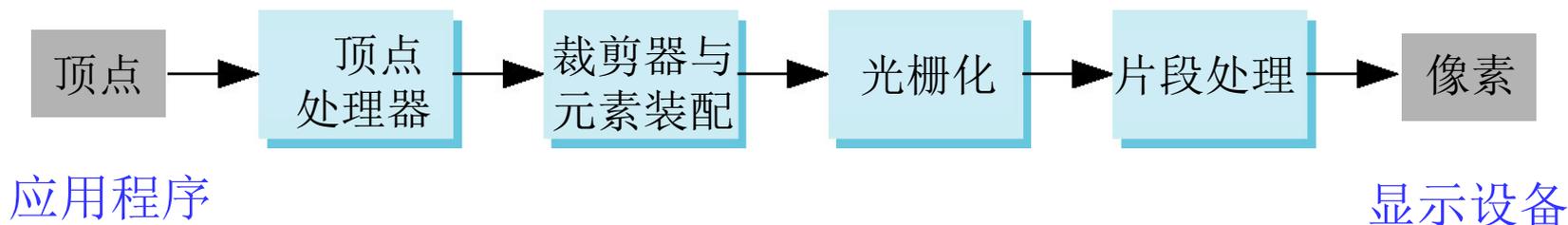
- Phong光照模型
- 计算量适中，有硬件加速支持
- 当前主流API
 - OpenGL: The Khronos OpenGL ARB Working Group, 开放的接口，跨平台
 - Direct3D: Microsoft, Windows操作系统支持较好，更新快（某种程度上也是缺点☹️）

当前主流图形系统处理方法

- 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



- 所有步骤都可以通过显示卡的硬件实现

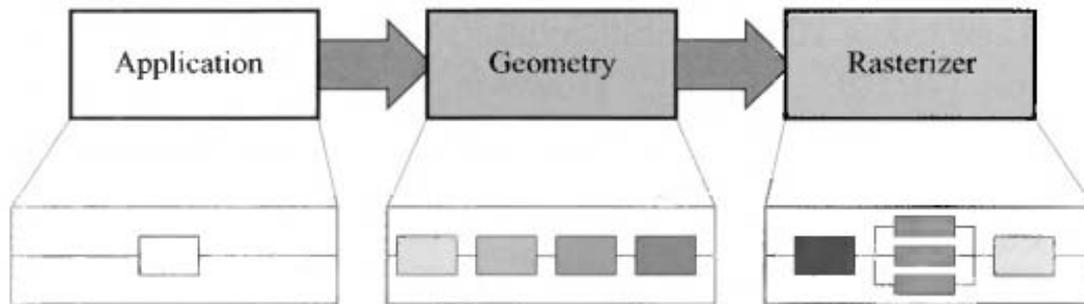
流水线简介

■ 流水线 (pipeline)

- 如果一项操作能分解成 n 个流水线操作, 那么速度能提升 n 倍!
- 如果处理对象能分到 m 个流水线上处理, 那么速度能提升 m 倍! (并行化)

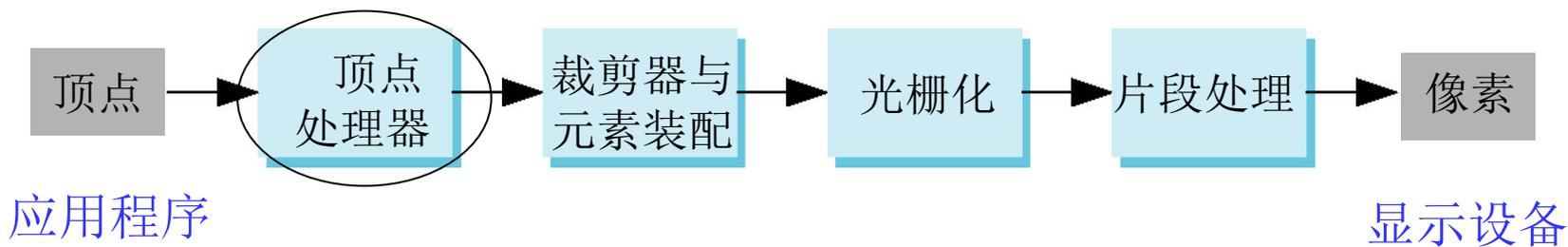
■ 特点

- Step by step: 流水线的速度由最慢的操作决定!



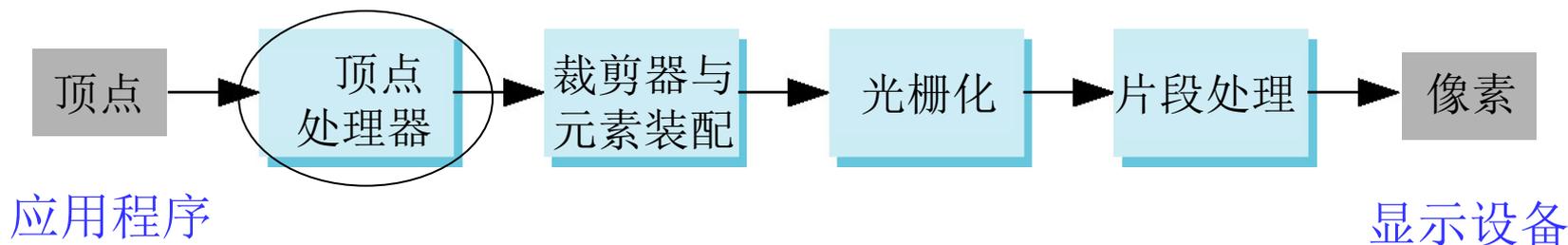
顶点处理

- 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：
 - 世界坐标系
 - 照相机(眼睛)坐标系
 - 屏幕坐标系
- 坐标的每个变换相当于一次矩阵乘法
- 顶点处理器也计算顶点的颜色



■ 把三维观察者位置与三维对象结合在一起，确定二维图像的构成

- 透视投影：所有投影线交于投影中心
- 平行投影：投影线平行，投影中心在无穷远，用投影方向表示

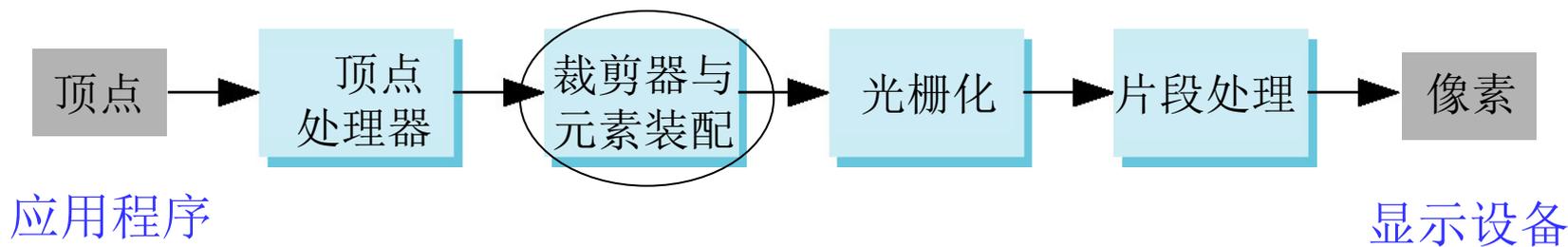


元素集成



■ 在进行裁剪和光栅化处理之前，顶点必须集成为几何对象

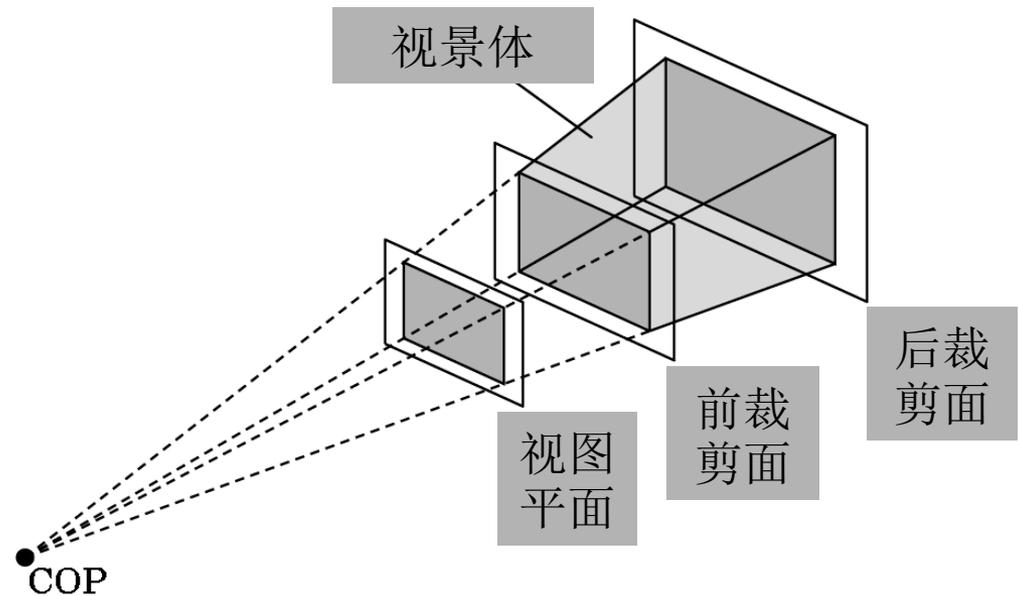
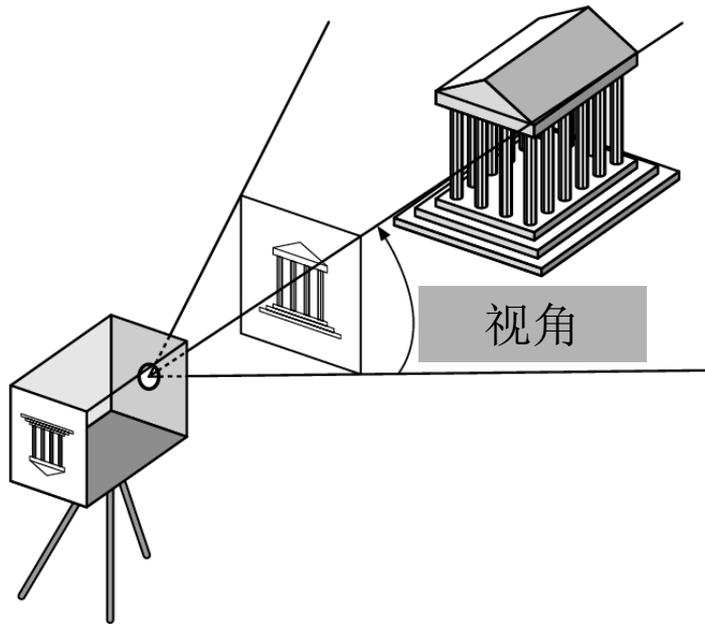
- 线段
- 多边形
- 曲线和曲面



裁剪



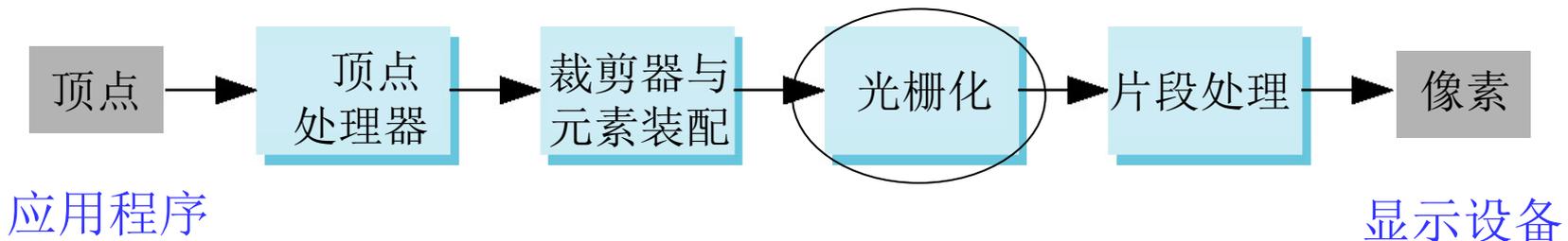
- 真正的照相机不能“看到”整个世界，图形学中虚拟的照相机也只能看到世界的一部分
 - 不在下述视景体中的对象要从场景中裁剪掉（有限计算！）



光栅化

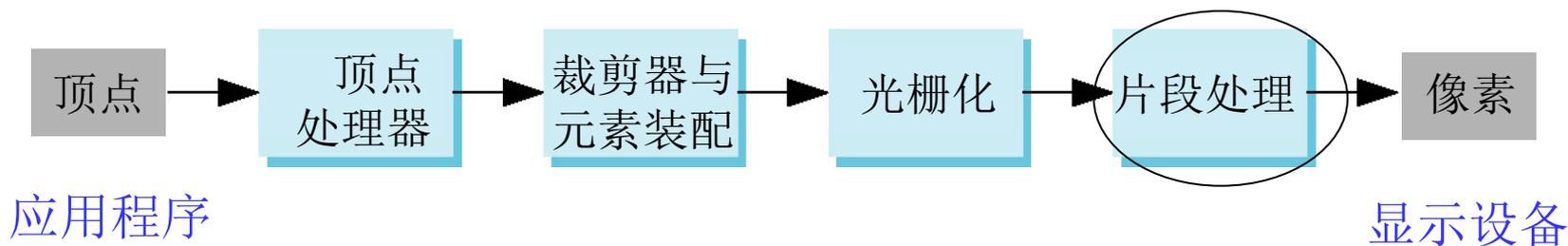


- 如果一个对象不被裁掉，那么在帧缓冲区中相应的像素就必须被赋予颜色（以primitive为单位，依赖于primitive的类型）
- 光栅化程序为每个对象生成一组片段
- 片段是“潜在的像素”
 - 在帧缓冲区中有一个位置
 - 具有颜色和深度属性
- 光栅化程序在对象上对顶点属性进行插值



片段处理

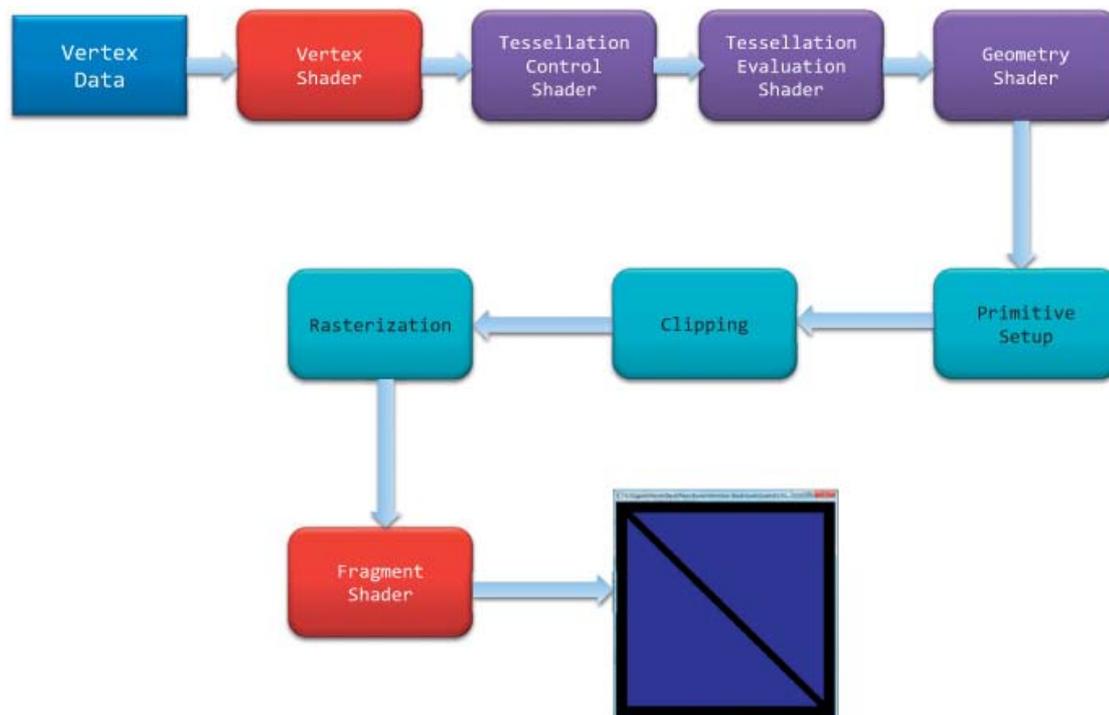
- 对片段进行处理，以确定帧缓冲区中相应像素的颜色
- 颜色可以由纹理映射确定，也可以由顶点颜色插值得到
- 片段可能被离照相机更近的其他片段挡住
 - 隐藏面消除



可编程图形管线



- 固定管线：固定的处理模式（如Phong光照模型），至多有些参数可调，不够灵活
- 可编程管线
 - Vertex shader、fragment shader: programmable
 - tessellation shader、geometry shader: optional



可编程图形管线

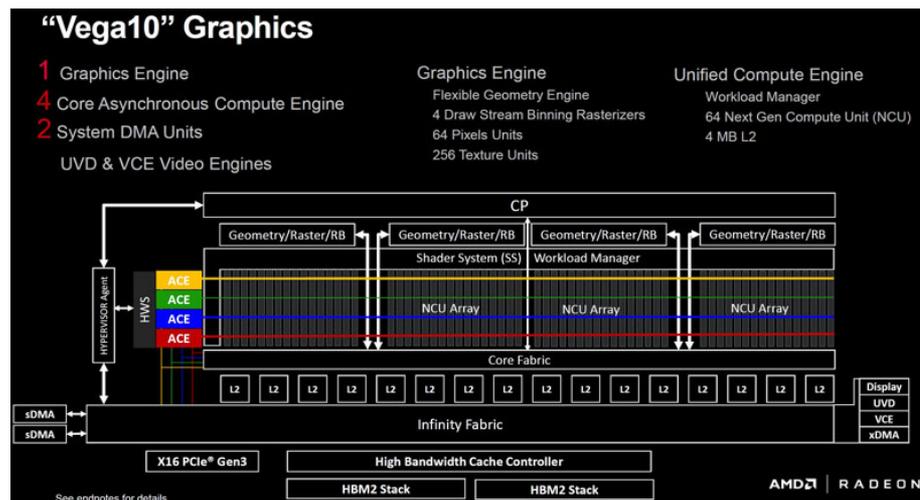


■ NVIDIA 与 AMD 图形硬件架构

GTX 1080 TI OVERVIEW

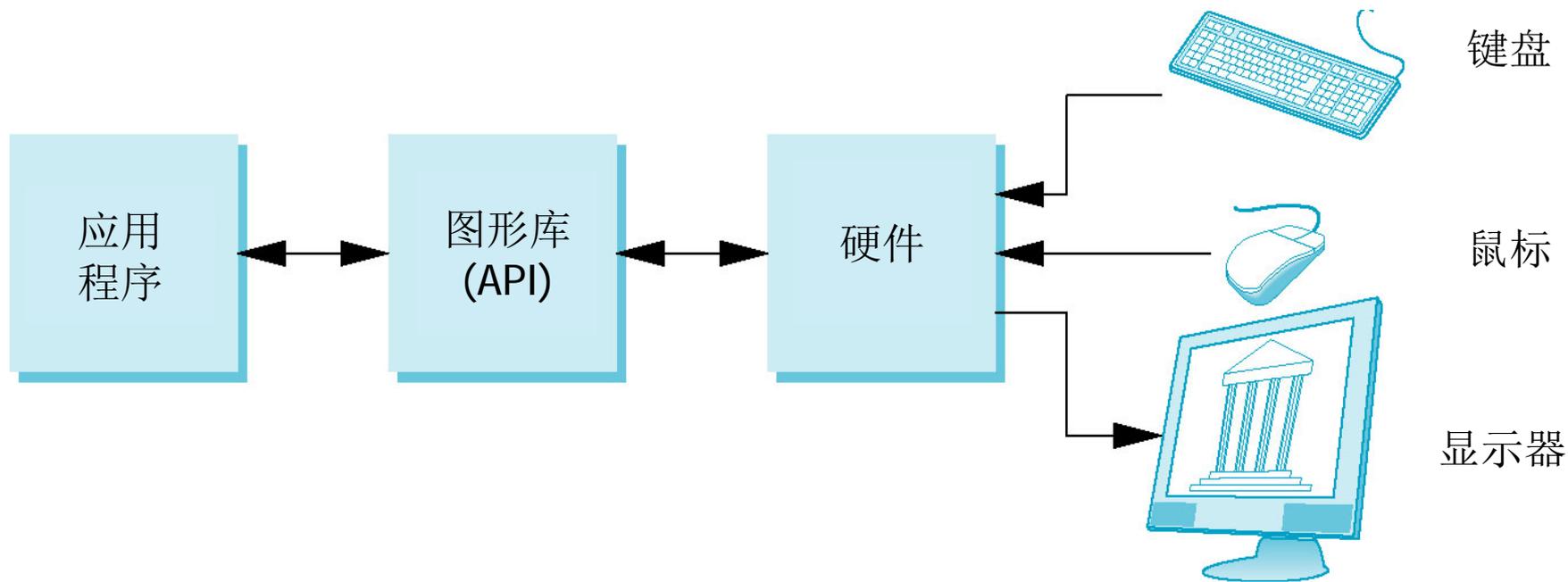


- 12B Transistors
- 1.6 GHz Boost, 2 GHz OC
- 28 SMs, 128 cores each
- 3584 CUDA cores
- 28 Geometry units
- 224 Texture units
- 6 GPCs
- 88 ROP units
- 352 bit GDDR5x



应用程序界面

- 程序设计人员是通过软件界面接触图形系统，这个界面就是应用程序界面 (API)



API的构成



■ 通过调用函数定义构成一幅图像所需要的内容:

- 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

对象的定义



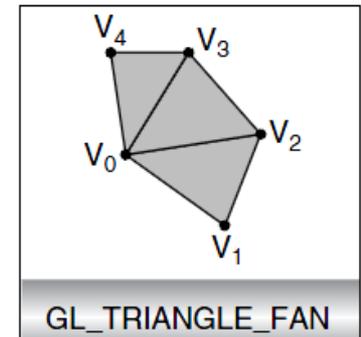
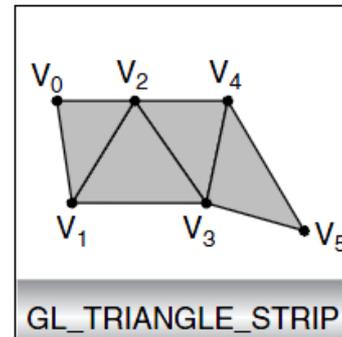
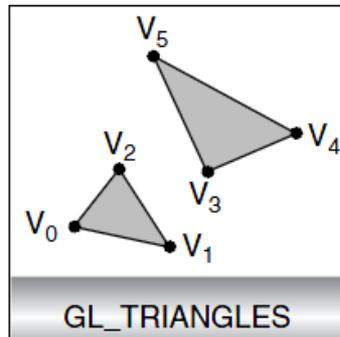
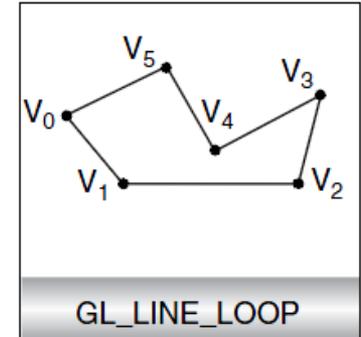
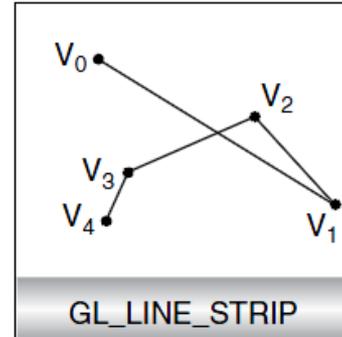
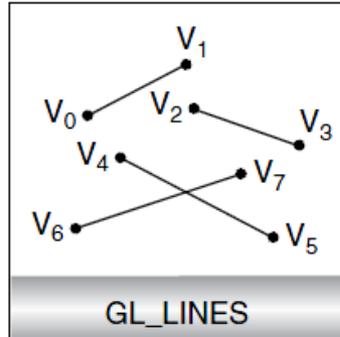
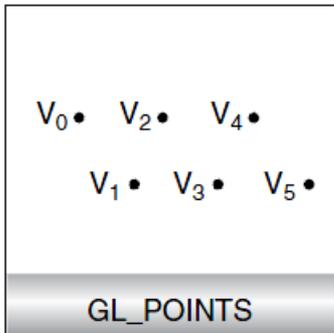
- 绝大多数API支持有限的基本几何对象，例如：
 - 点 (points, 零维对象)
 - 线段 (line segments, 一维对象)
 - 多边形 (polygons, 二维对象)
 - 某些曲线和曲面
 - 二次曲面 (quadrics)
 - 多项式参数曲面 (NRUBS)
- 所有基本形状在空间中的位置通过顶点 (vertices) 来描述。

示例



■ OpenGL中的基本图元:

- 点: GL_POINTS
- 线: GL_LINES, GL_LINE_STRIP, GL_LINE_LOOP
- 面: GL_TRIANGLES, GL_TRIANGLE_STRIP, GL_TRIANGLE_FAN
- 片: GL_PATCHES



示例



- Put geometric data in an array

```
vec3 points[3];  
points[0] = vec3(0.0, 0.0, 0.0);  
points[1] = vec3(0.0, 1.0, 0.0);  
points[2] = vec3(0.0, 0.0, 1.0);
```

- Send array to GPU
- Tell GPU to render as triangle

合成的照相机位置指定



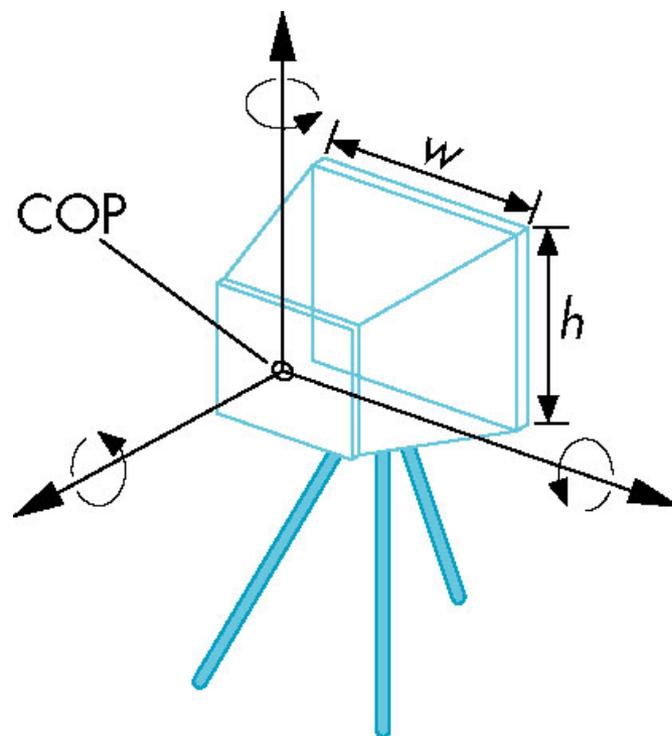
■ 六个自由度

- 镜头中心的位置，即投影中心 (COP)
- 方向

■ 镜头、焦距

■ 胶卷尺寸

■ 胶卷平面的方向



■ 光源类型

- 点光源与分布式光源
- 聚光灯 spot lights
- 光源的远与近
- 光源的颜色属性

■ 材料属性

- 吸收性：颜色属性
- 反射性：漫反射、镜面



Thanks for your attention!

