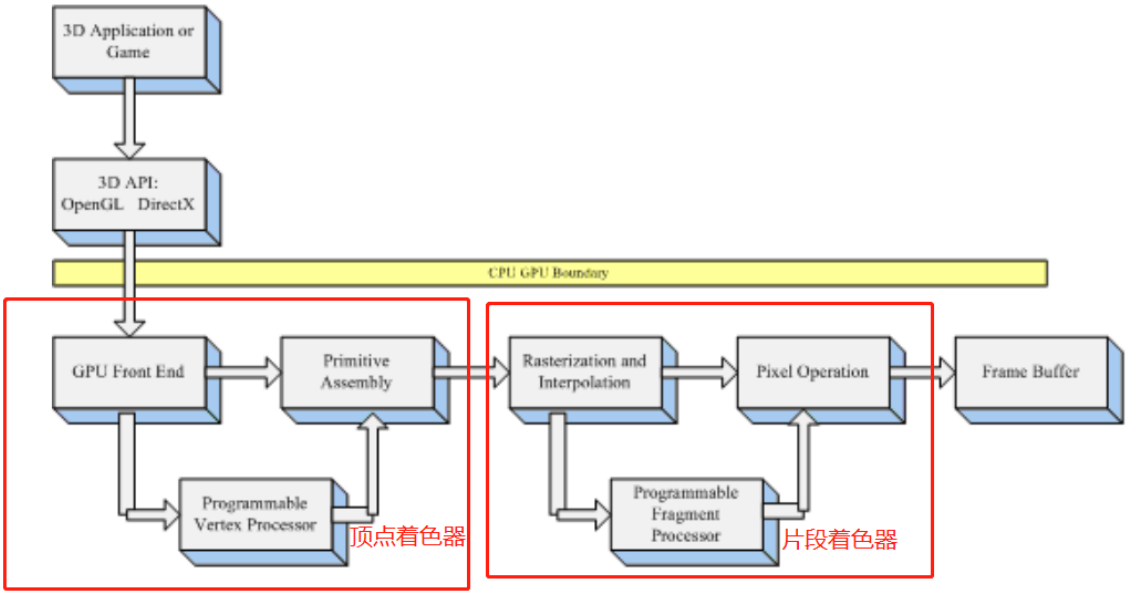
shader language（着色语言）：基于物体本身属性和光照条件，计算每个像素的颜色值。被定位为高级语言，高级语言的一个重要特性是“独立于硬件”，在这 一方面 shader language 暂时还做不到，shader language 完全依赖于 GPU 构架。任意一种 shader language 都必须基于图形硬件，所以 GPU 编程技术的发展本质上还是图形硬件的发展。
3 种语言Shader language：
基于 OpenGL 的 GLSL （OpenGL Shading Language）
基于 Direct3D 的 HLSL（High Level Shading Language）
NVIDIA 公司的 Cg 语言（C for Graphic）

本章的目的是阐述 shader language 的基本原理和运行流程，首先从硬件的角 度对 Programmable Vertex Processor（可编程顶点处理器，又称为顶点着色器）和 Programmable Fragment Processor（可编程片断处理器，又称为片断着色器） 的作用进行阐述，然后在此基础上对 vertex program 和 fragment program 进行具 体论述，最后对 GLSL、HLSL 和 Cg 进行比较。

3.1 Shader Language 原理

shader program（着色程序）：使用 shader language 编写的程序称之为 shader program。
着色程 序分为两类：vertex shader program（顶点着色程序）和 fragment shader program （片断着色程序）。
GPU 上的两个组件：
1.Programmable Vertex Processor（可编程顶点处理器，又 称为顶点着色器）:可编程顶点处理器是一个硬件单元，可以运行顶点程序
2.Programmable Fragment Processor（可编程片断处理器，又 称为片断着色器）:可编程片段处理器则是一个可以运行片段程序的单元。

顶点和片段处理器都拥有非常强大的并行计算能力，并且非常擅长于矩阵 （不高于4阶）计算，片段处理器还可以**高速查询纹理信息**（目前顶点处理器还不行）

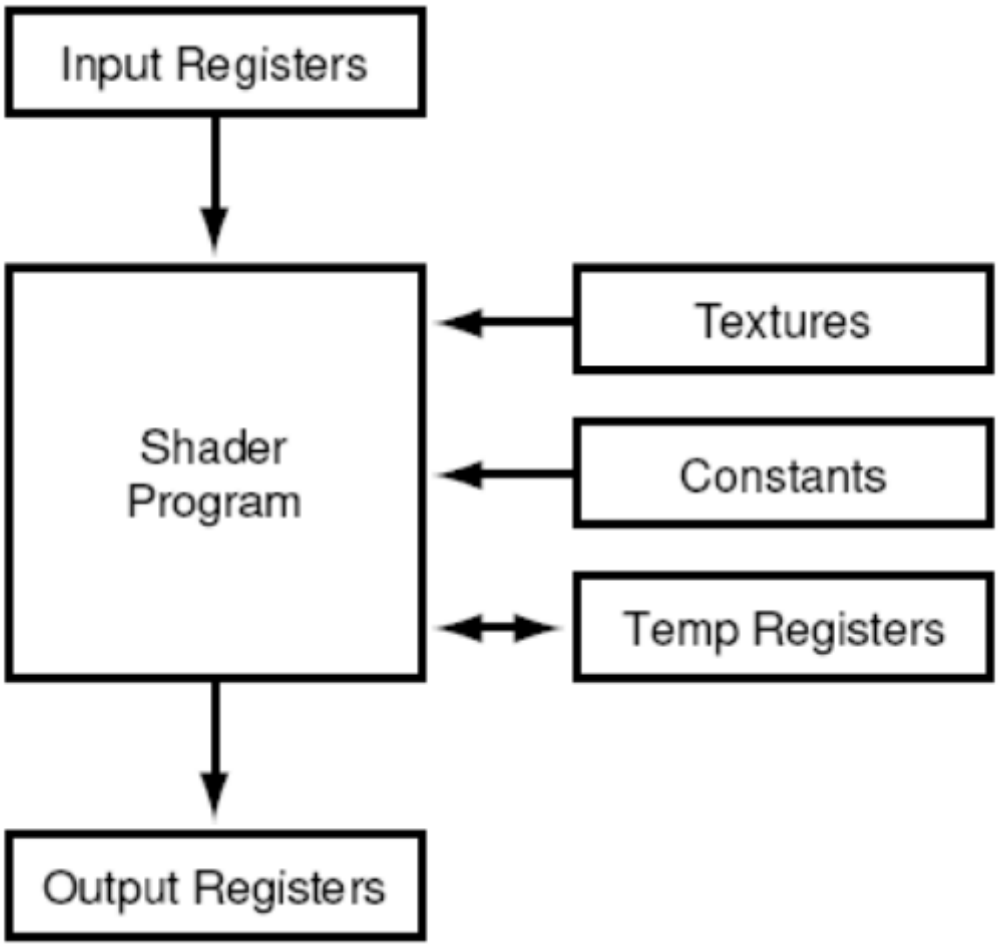


可编程图形渲染管线

顶点着色器控制顶点坐 转换过程；片段着色器控制像素颜色计算过程。

Vertex program 负责顶点坐标变换；Fragment program 负责像素颜色计算。

Vertex program 的输出是Fragment program 的输入。



可编程图形硬件输入\输出

输入寄存器（Input Register）存放输入的图元信息；

输出寄存器（Output Register）存放处理后的图元信息；

纹理buffer存放纹理（Textures）数据，目前大多数的可编程图形硬件只支持片段处理器处理纹理；

从外部宿主程序输入的常量放在常量寄存器（Constants）中；

临时寄存器（Temp Registers）存放着色程序在执行过程中产生的临时数据。

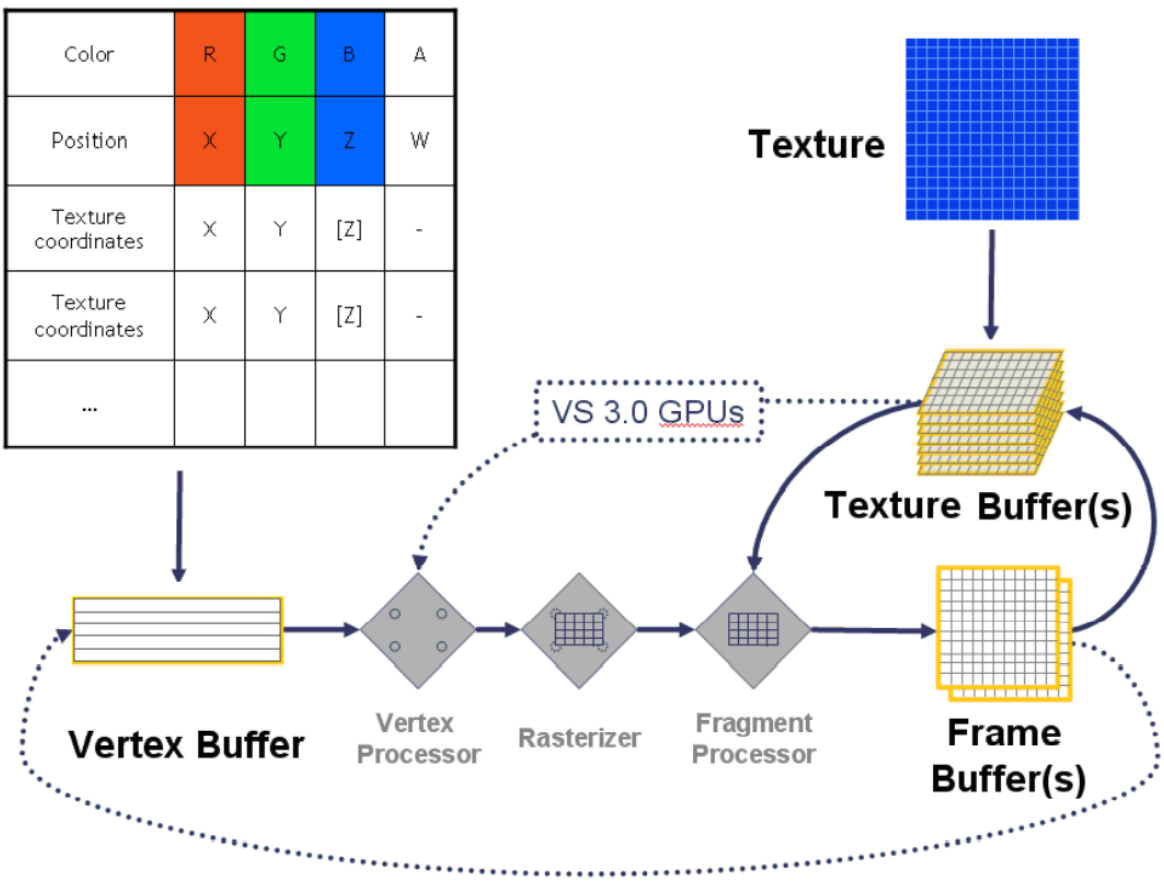
3.2 Vertex Shader Program

顶点着色程序（Vertex shader program）被Programmable Vertex Processor（可编程顶点处理器）所执行。

顶点着色程序从 GPU 前端模块（寄存器）中提取图元信息（顶点位置、法 向量、纹理坐标等），并完成顶点坐标空间转换、法向量空间转换、光照计算等 操作，最后将计算好的数据传送到指定寄存器中；

片断着色程序（Fragment shader program）被Programmable Fragment Processor（可编程片断处理器）所执行。

片断着色程序从指定寄存器（顶点着色程序计算好的数据）中获取需要的数据，通常为“纹理坐标、光照信息等”，并根据这些信息以及从应用程序传递的纹理信息（如果有的话）进行每个片断的颜色计算，最后将处理后的数据送往光栅操作模块。



顶点着色器和像素着色器的数据处理流程

在应用程序中设定的图元信息（顶点位置坐标（Position）、颜色（Color）、纹理坐标（Texture Coordinates）等）传递到 Vertex Buffer 中；
纹理信息（Texture）传递到 TextureBuffer 中。
(图中虚线表示目前还没有实现的数据传递)当前的顶点程序还不能处理纹理信息，纹理信息只能在片断程序中读入。
GPU 对数据进行并行处理，所以每个数据都会执行一次 shader 程序程序（每个顶点数据都会执行一次顶点程序；每个片段都会执行一次片段程序）。

顶点着色程序：只有顶点着色程序，则只对输入的顶点进行操作，而顶点内部的点则按照硬件默认的方式自动插值。
片断着色程序：对每个片断进行独立的颜色计算，并且算法由自己编写， 不但可控性好，而且可以达到更好的效果。

3.3 Fragment Shader Program

片断着色程序对每个片断进行独立的颜色计算，最后输出颜色值的就是该片段最终显示的颜色。
片段着色程序：拥有检索纹理的能力。对于 GPU 而言，纹理等价于数组，这意味着，如果要做通用计算，例如数组排序、字符串检索等， 就必须使用到片段着色程序。

片断和像素有什么不一样?
片断：是所有的三维顶点在光栅化之后的数据集合，还没有经过深度值比较。
像素：屏幕显示的像素都是经过深度比较的。