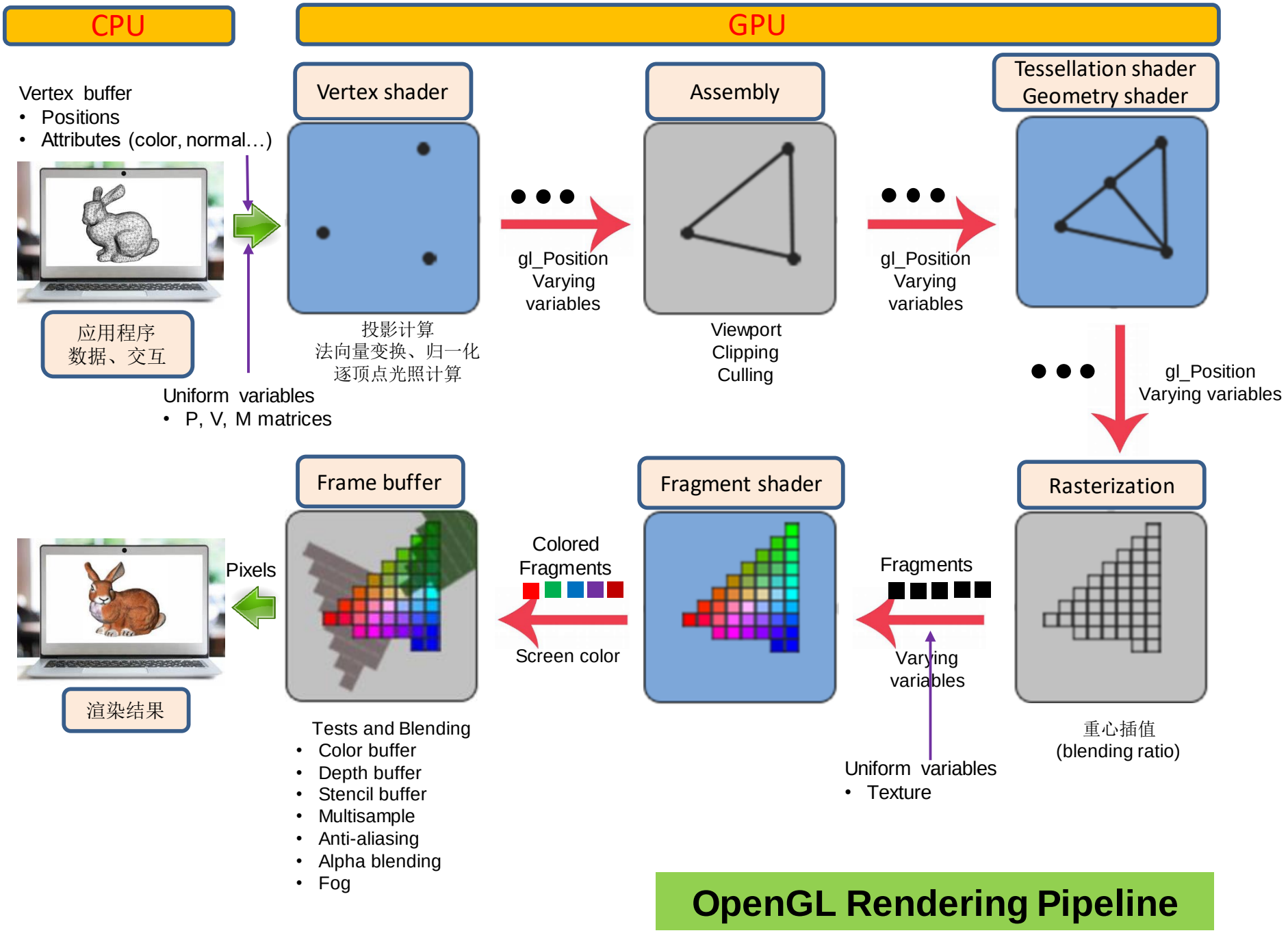
计算机图形学
Computer Graphics

刘利刚
lgliu@ustc.edu.cn
http://staff.ustc.edu.cn/~lgliu

Graphics&Geometric Computing Lab
@USTC

Shader Programming

着色器编程


CPU
GPU
Vertex buffer
• Positions
• Attributes (color, normal…)
应用程序
数据、交互
Uniform variables
• P, V, M matrices
Vertex shader
投影计算
法向量变换、归一化
逐顶点光照计算
gl_Position
Varying variables
Assembly
Viewport
Clipping
Culling
gl_Position
Varying variables
Tessellation shader
Geometry shader
gl_Position
Varying variables
Rasterization
重心插值
(blending ratio)
Fragments
Varying variables
Uniform variables
• Texture
Fragment shader
Colored Fragments
Screen color
Frame buffer
Tests and Blending
• Color buffer
• Depth buffer
• Stencil buffer
• Multisample
• Anti-aliasing
• Alpha blending
• Fog
Pixels
渲染结果
OpenGL Rendering Pipeline

GPU并行计算

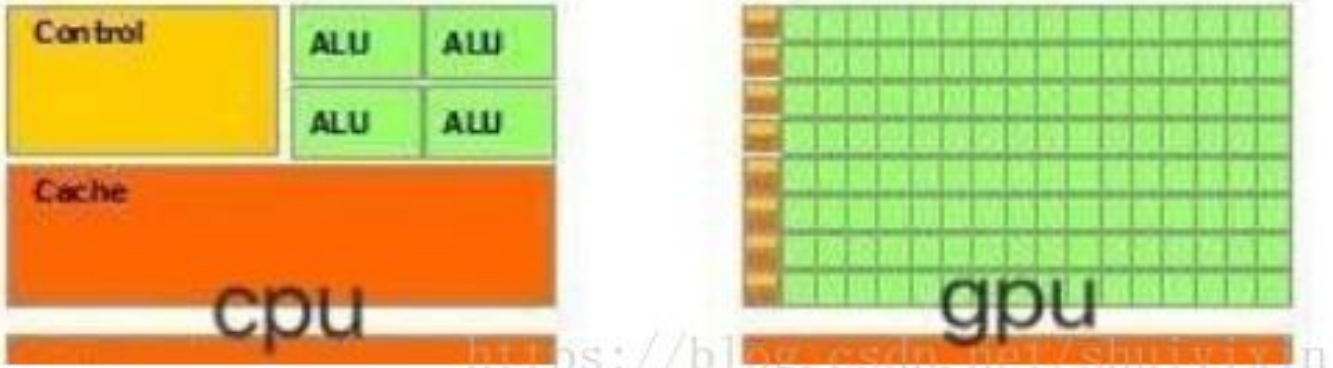
并行计算

- 互相独立（互不依赖）的计算可以并行
- 图形渲染：大量并行运算
  - 顶点变换的计算：逐顶点
  - 片元颜色的计算：逐像素
- 从图形加速卡到GPU

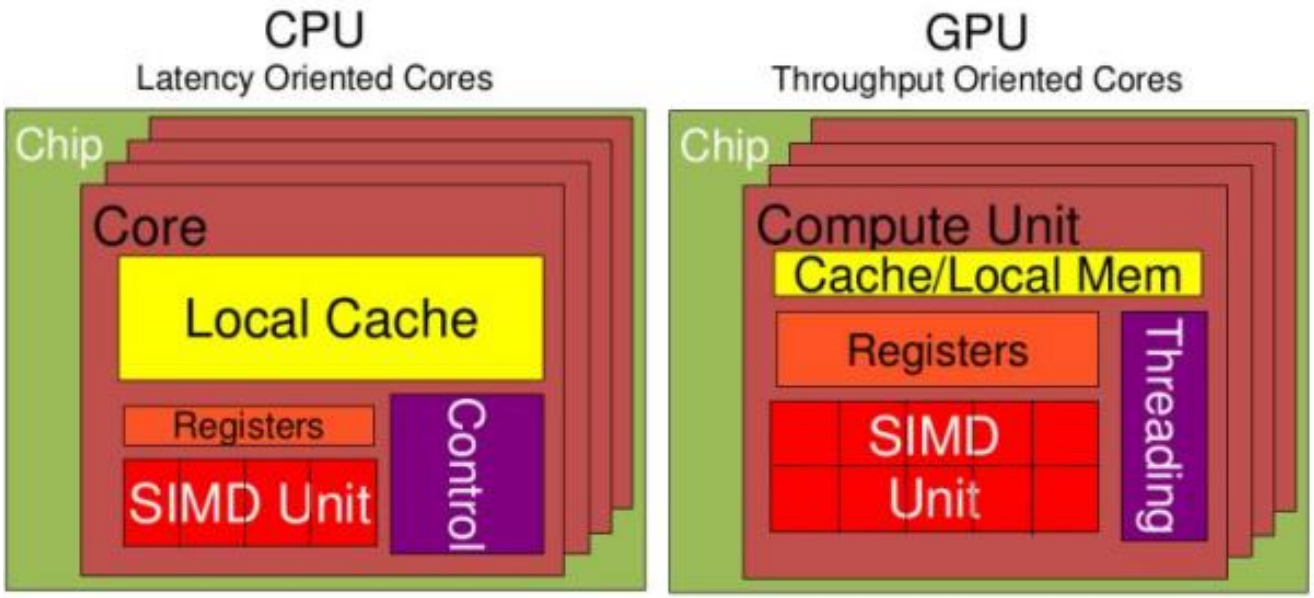
GPU: 基于大吞吐量的设计

- Graphic Processing Unit (GPU)
- 非常多的小的计算单元

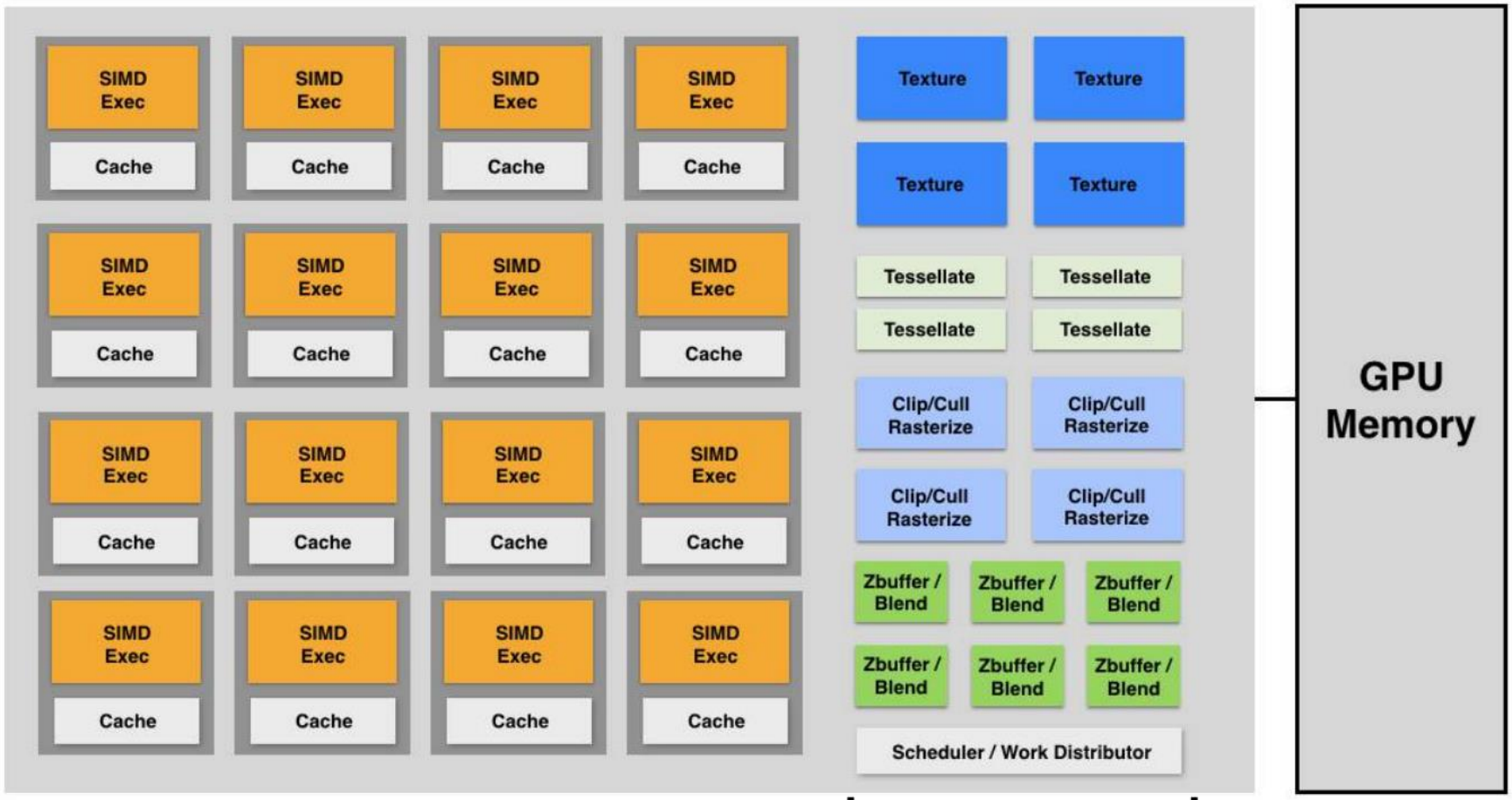
GPU vs. CPU



绿色：计算单元
橙红色：存储单元
橙黄色：控制单元



GPU: Heterogeneous, Multi-Core Processor



Credit: Lingqi Yan

CPU vs. GPU

- 设计架构不一样：目标不同
  - CPU: 复杂逻辑运算
  - GPU: 成百上千个计算核，无复杂逻辑
- CPU擅长
  - 逻辑控制和通用类型数据运算不同
- GPU擅长：大规模并发计算
  - 类型高度统一的、相互无依赖的大规模数据
  - 不需要被打断的纯净的计算环境

GPU vs CPU

GPU

CPU

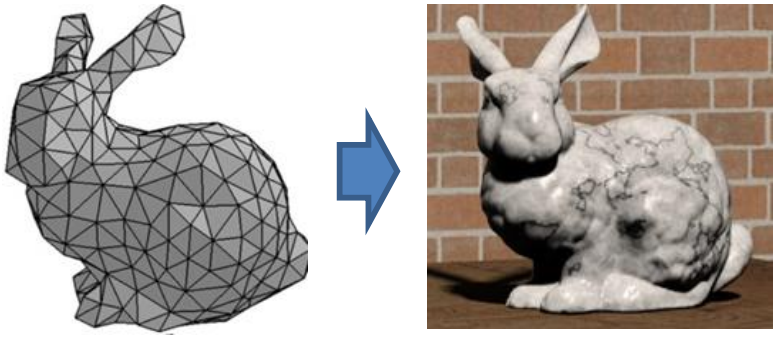
GPGPU（通用GPU计算）

- 将GPU的高并行性能用于其他需要高密度计算的领域
- OpenGL利用纹理存储器在GPU中计算以及把结果取回内存，流程：
  - 创建FBO（帧缓存对象），设置纹理参数
  - 然后将纹理绑定到帧缓存对象，传输数据到纹理
  - 接着用片元着色器对数据进行处理
  - 取回数据到内存用于其他程序
- OpenCL: Open Computing Language，开放运算语言

可编程 管线/流水线
Programmable Pipeline

OpenGL渲染管线（流水线）

- 是一套固定的计算流程
  - 3D数据→2D图像
- 渲染管线的工作方式
  - 从数据输入开始
  - 流程中的每一步获取前一步的生成的数据
  - 经过自己的计算/加工
  - 再传输给下一步作为输入



Rendering Pipeline

- 逐顶点计算
  - 视点变换，法向量变换，纹理坐标，光照计算
- 图元装配
  - 背面剔除，投影变换，裁剪，透视除法，视口变换
- 光栅化
  - 插值顶点属性：颜色，深度值，纹理坐标
- 片元操作
  - 纹理生成，雾化
- 光栅操作
  - 融合，反走样，Alpha测试，深度测试

固定管线

- 用户能控制
  - 管线中的某子流水线的参数设置
- 用户不能控制
  - 管线中的某子流水线的具体实现（计算）方法
- 不足：
  - 参数的类型与数目都是有限的，可能的结果也仅仅是有限多种，无法创造新的渲染结果

可编程管线

- 随着硬件的发展，用户已经可以针对图形处理器（GPU）进行编程
- OpenGL提供了相应的机制支持这一点：用户可以编写专门运行于GPU的小程序，而非只能为函数设置有限的参数
  - 通过编写着色器(Shader)，用于替代固定流水线中的各个子流水线
  - 着色器，是一种针对GPU编写的，由GPU直接运行的，高度并行化的小程序
    - 顶点着色器、细分着色器、几何着色器、片元着色器、通用的计算着色器
    - …

可编程管线

- 大部分显卡都支持
- 软件支持
  - OpenGL Shading Language (GLSL)
    - 作业U-Engine中用的是OpenGL 3.3
    - 最新版本OpenGL 4.6 (GLSL 4.6)
  - Direct X 10, 11, 12 (HLSL)

```
const GLubyte* glGetString(GLenum name);
```

返回OpenGL实现相关的信息，name可以是

- GL_VENDOR：厂商信息
- GL_RENDERER：硬件平台
- GL_VERSION：OpenGL版本
- GL_EXTENSIONS：扩展信息

固定流水线VS可编程流水线

- 固定流水线
  - 只支持Blinn-Phong光照模型
  - 无法实现
    - 物理真实感的光照效果：折射、区域光、软阴影
    - 非真实感效果：卡通着色效果、水墨画
- 可编程流水线
  - 把顶点和片元处理中那些固定的功能用可编程处理器代替，可定制化编写效果
  - 须实现固定管线的功能

生产线上的工人都是机器人，只有有限种工作方式

生产线上以人工智能代替机器人，能够听懂你的指令

顶点着色器应用

- 顶点的位置计算
  - 偏置 (Displacement)
  - 扰动
- 顶点的光照计算
  - 更真实的模型
  - 卡通着色器

片元着色器应用

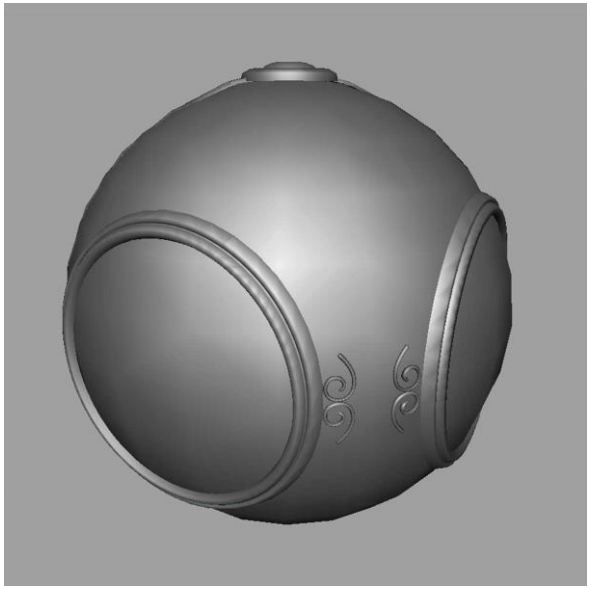
- 逐片元进行光照计算

逐顶点光照计算

逐片元光照计算

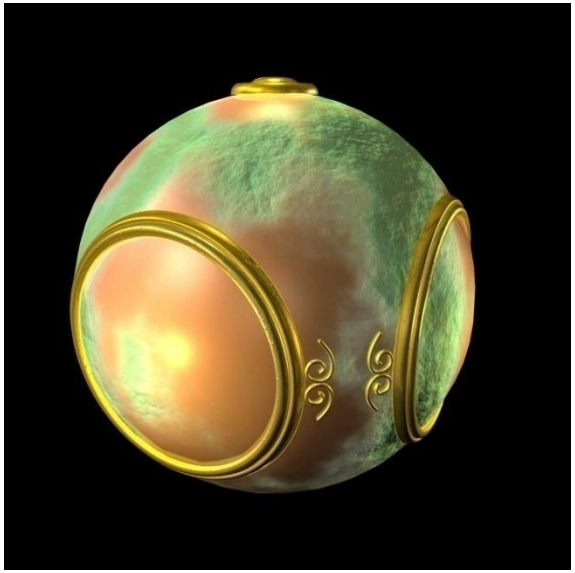
片元着色器应用

- 纹理映射



光滑明暗处理

环境映射



凹凸映射

Q&A