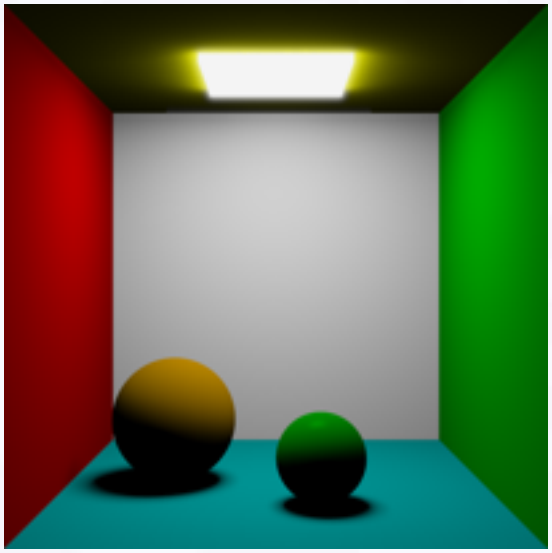
360°全景视频播放器的实现原理

杨显涛

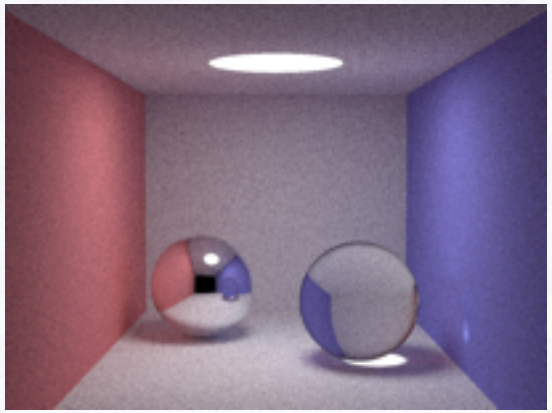
HTY360Player

渲染器



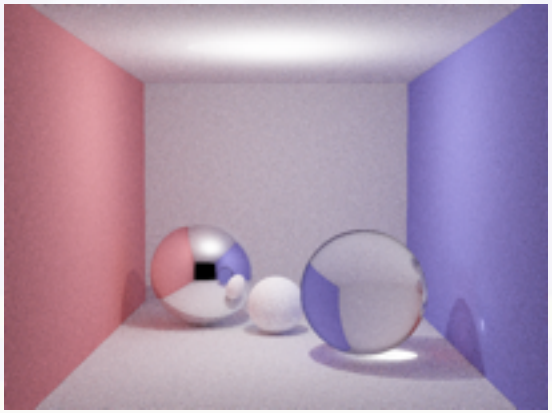
pyTracer

ray tracing



pySmallPT

path tracing



pySmallPPM

Progressive Photon Mapping

实现的核心—OpenGL Shader

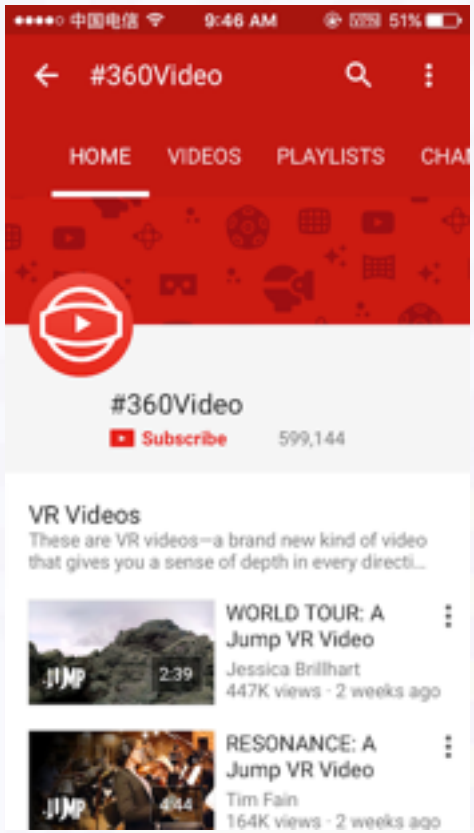
来源: Renderman

实现的核心—OpenGL Shader

来源: Renderman

HTY360Player想法的来源

Facebook 360 Videos



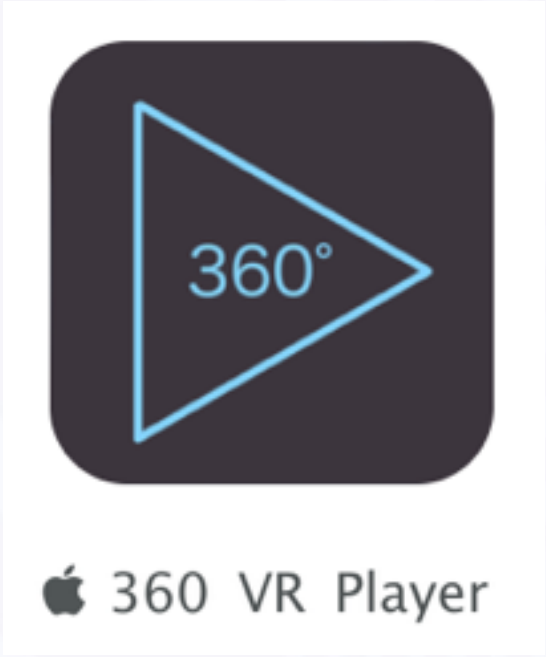
YouTube 360° Videos

全景视频与普通视频

全景视频：互动，视角大

普通视频：固定视角

HTY360Player



http://fir.im/8yln

不恰当的类比

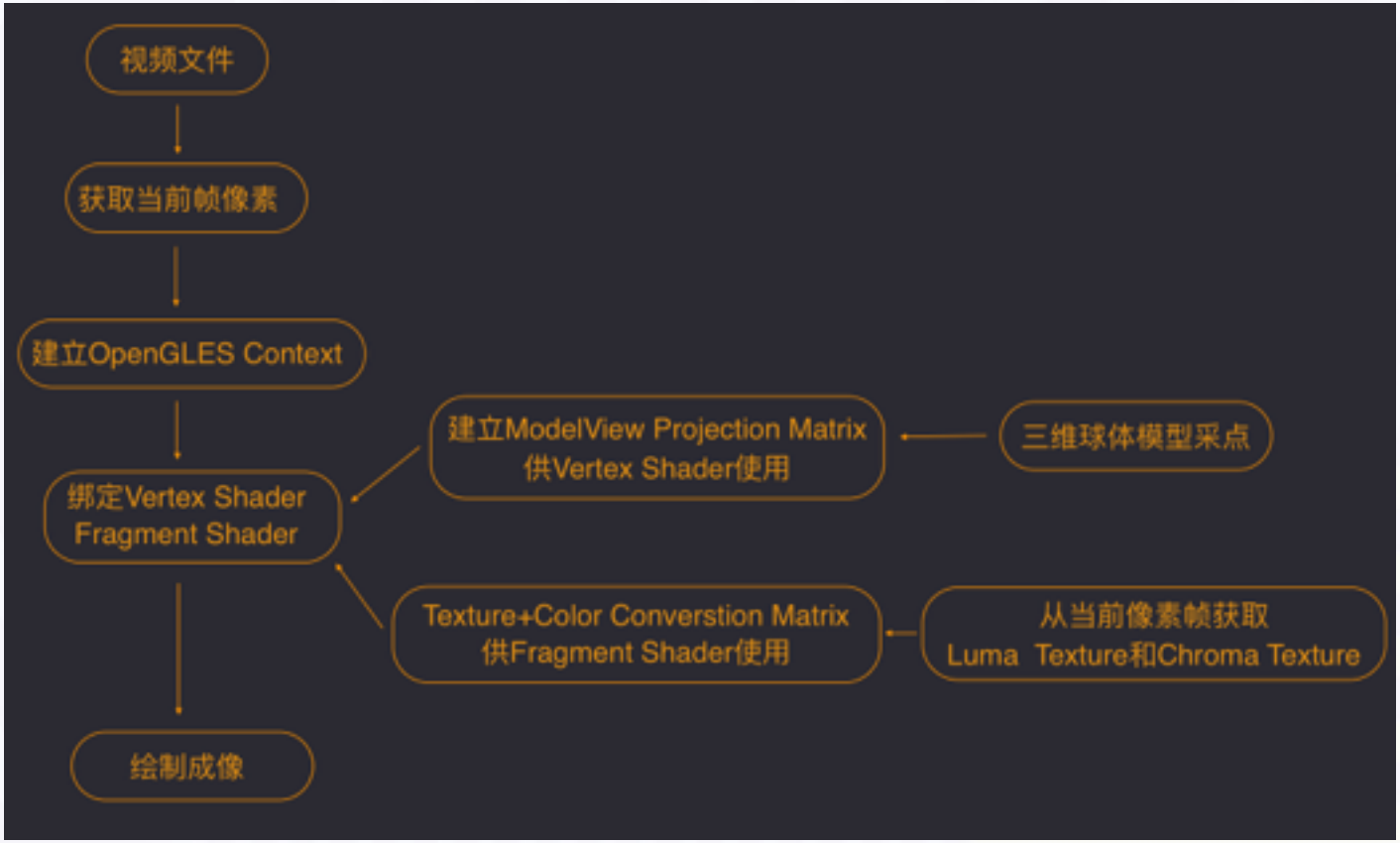
地球 —> 地图

Shader

HTY360Player

- 以iOS平台的开源项目HTY360Player为例子(https://github.com/hanton/HTY360Player)
- 基于OpenGL，实现方式同样适用于Android或Web

实现的整体流程


视频文件
获取当前帧像素
建立OpenGLES Context
绑定Vertex Shader
Fragment Shader
建立ModelView Projection Matrix
供Vertex Shader使用
三维球体模型采点
Texture+Color Converstion Matrix
供Fragment Shader使用
从当前像素帧获取
Luma Texture和Chroma Texture
绘制成像

实现的整体流程

1. 读取视频流中的图像
2. 搭建OpenGL环境
3. 映射图像到三维球体
4. 从YCbCr颜色域到RGB
5. 绘制转换后的图像到屏幕

如何读取视频流中的图像

1/5

读取视频流中的图像

系统API，如kCVPixelFormatType_420YpCbCr8BiPlanarVideoRange

视频编码格式: YUV (analog encoding)
Y’CbCr (digital encoding)

采样方式: 420

如何搭建OpenGL环境

2/5

搭建OpenGL环境


glCreateProgram()
glCreateShader()
glShaderSource()
glCompileShader()
glGetShaderiv()
glAttachShader()
glDeleteShader()
glBindAttribLocation()
glLinkProgram()
glUseProgram()
glDeleteProgram()

如何映射图像到三维球体

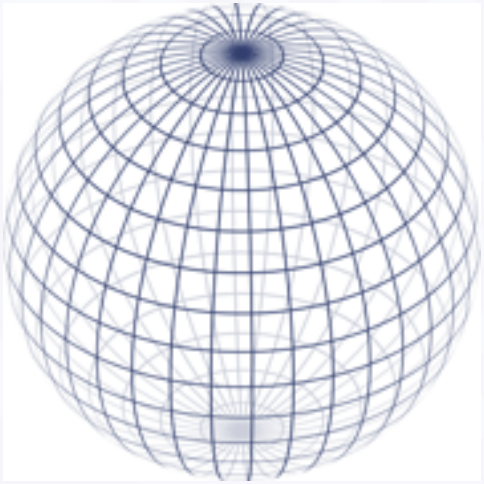
3/5

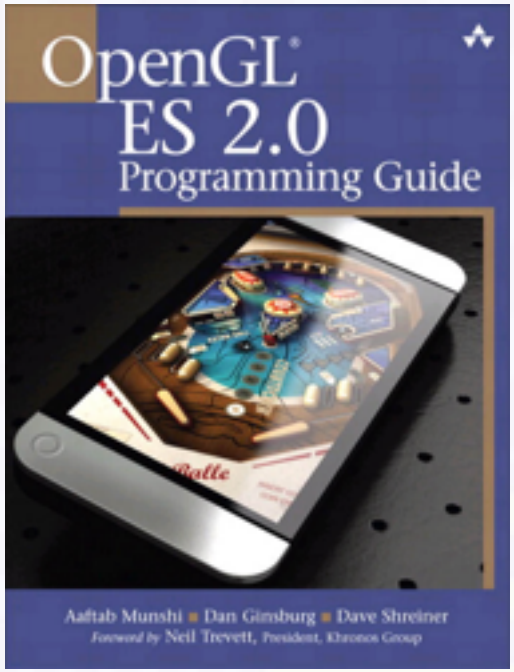
Vertex Shader

The **Vertex Shader** is the programmable Shader stage in the rendering pipeline that handles the processing of individual vertices.
Vertex shaders typically perform transformations to post-projection space, for consumption by the Vertex Post-Processing stage.

来源: WiKi

建立3D球体模型





int ESUTIL_API esGenSphere(int numSlices, float radius, GLfloat ** vertices, GLfloat ** normals, GLfloat ** texCoords, GLuint ** indices)

如何从YCbCr颜色域转到RGB

4/5

Fragment Shader

A **Fragment Shader** is the Shader stage that will processes a Fragment generated by the Rasterization into a set of colors and a single depth value.
The output of a fragment shader is a depth value, a possible stencil value (unmodified by the fragment shader), and zero or more color values to be potentially written to the buffers in the current framebuffers.

来源: WiKi

像素域转换（YUV420 -> RBG）

Single Frame YUV420:

视频编码格式: YUV (analog encoding)
Y'CbCr (digital encoding)

采样方式: 420

Position in byte stream:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0.000 & 1.793 \\ 1.164 & -0.213 & -0.533 \\ 1.164 & 2.112 & 0.000 \end{bmatrix} \cdot \begin{bmatrix} (Y - 16) \\ (Cb - 128) \\ (Cr - 128) \end{bmatrix}$$

Ranges:
Y [16 ... 235]
Cb/Cr [16 ... 240]
R/G/B [0 ... 255]

如何绘制转换后的图像到屏幕

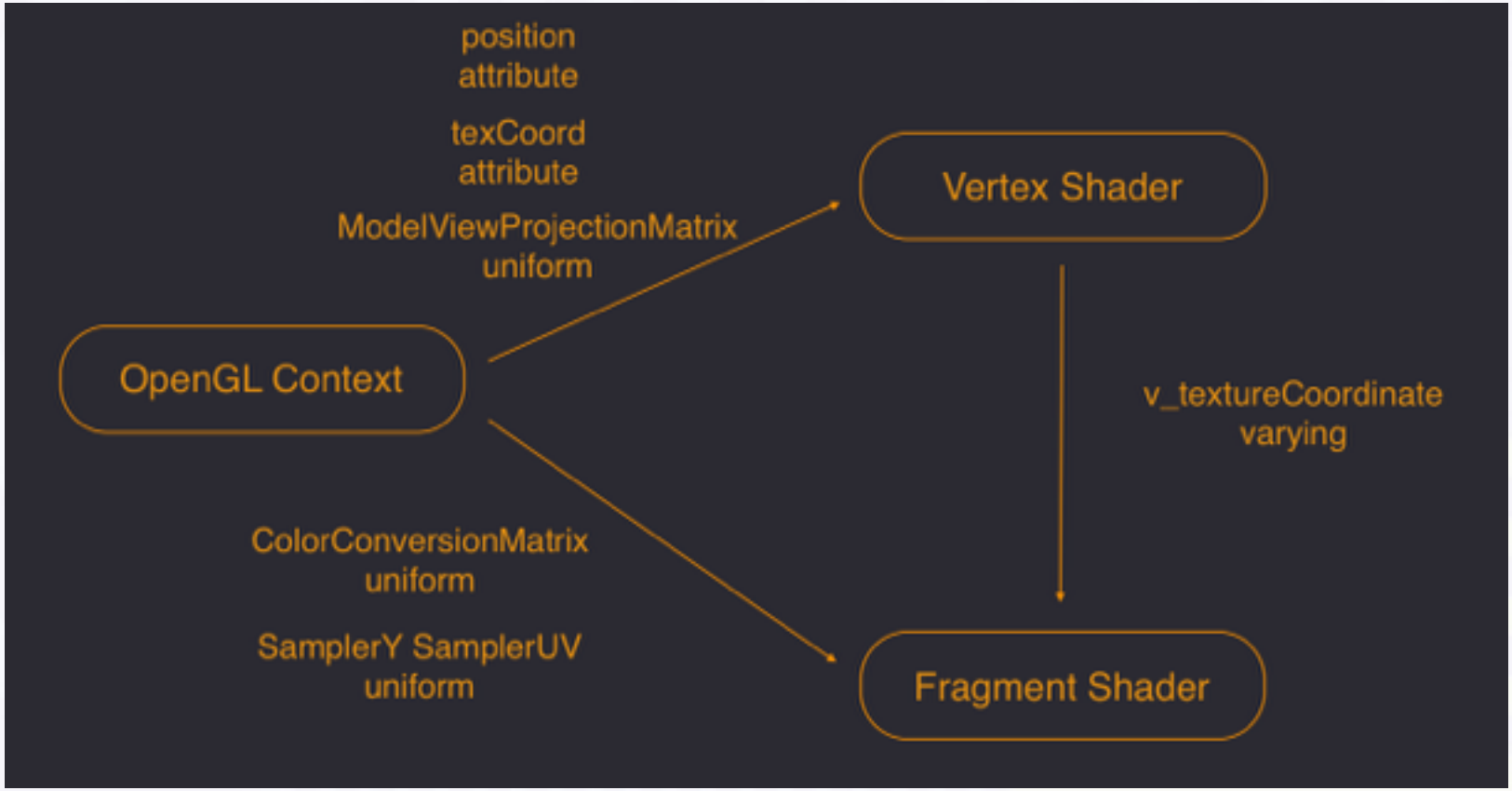
5/5

绘制转换后的图像到屏幕

glDrawElements

GLKViewController
- (void)update
- (void)glkView:(GLKView *)view drawInRect:(CGRect)rect

打通OpenGL和Shader的数据流 (Attributes & Uniform)



当前的发展情况

Nokia OZO

Daydream Cardboard

Google VR

Facebook Surround 360

Gear 360

Q&A

GMTC 全球移动技术大会
GLOBAL MOBILE TECH CONFERENCE

THANKS

聚焦前沿技术　传递实践经验

主办方 Geekbang. 极客邦科技　InfoQ