

Logic_视觉班课堂笔记[CC022]

- 日期: 2019年7月1日星期一
- 授课: CC老师
- 课程次数: 视觉班第22次课--共计(22次课)
- 主题: GPUImage 主题

课程内容

- GPUImage 框架解析
- GPUImage 框架使用

课程安排

- 08:00 - 09:00 第一节课
- 09:00 - 09:10 课间休息
- 09:10 - 10:00 第二节课
- 10:00 - 10:10 课程总结
- 10:10 - 10:30 课后答疑

课程回顾

- 滤镜回顾.

课程笔记

- GPUImage 简介

开源框架. 图片/视频.

- OpenGL ES/Metal
- AVFoundation 捕捉
- CoreMedia
-

- CoreImage / GPUImage

GPUImage

- 最低支持 iOS 4.0, iOS 5.0 之后就支持自定义滤镜。
- 在低端机型上, GPUImage 有更好的表现。(这个我没用真正的设备对比过, GPUImage 的主页上是这么说的)
- GPUImage 在视频处理上有更好的表现。
- GPUImage 的代码完成公开, 实现透明。
- 可以根据自己的业务需求, 定制更加复杂的管线操作。可定制程度高。

Core Image

- 官方框架, 使用放心, 维护方便。
- 支持 CPU 渲染, 可以在后台继续处理和保存图片。
- 一些滤镜的性能更强劲。例如由 Metal Performance Shaders 支持的模糊滤镜等。
- 支持使用 Metal 渲染图像。而 Metal 在 iOS 平台上有更好的表现。
- 与 Metal, SpriteKit, SceneKit, Core Animation 等更完美的配合。
- 支持图像识别功能。包括人脸识别、条形码识别、文本识别等。
- 支持自动增强图像效果, 会分析图像的直方图, 图像属性, 脸部区域, 然后通过一组滤镜来改善图像效果。
- 支持对原生 RAW 格式图片的处理。
- 滤镜链的性能比 GPUImage 高。(没有验证过, GPUImage 的主页上是这么说的)。
- 支持对大图进行处理, 超过 GPU 纹理限制 (4096 * 4096) 的时候, 会自动拆分成几个小块处理(Automatic tiling)。GPUImage 当处理超过纹理限制的图像时候, 会先做判断, 压缩成最大纹理限制的图像, 导致图像质量损失。

GPUImage 特性

- 丰富的输入组件 摄像头、图片、视频、OpenGL 纹理、二进制数据、UIElement (UIView, CALayer)
- 大量现成的内置滤镜 (4大类)
 - 颜色类 (亮度、色度、饱和度、对比度、曲线、白平衡...)
 - 图像类 (仿射变换、裁剪、高斯模糊、毛玻璃效果...)

- 颜色混合类（差异混合、alpha混合、遮罩混合...）
- 效果类（像素化、素描效果、压花效果、球形玻璃效果...）
- 丰富的输出组件 `UIView`、视频文件、GPU纹理、二进制数据
- 灵活的滤镜链 滤镜效果之间可以相互串联、并联，调用管理相当灵活。
- 接口易用 滤镜和 `OpenGL` 资源的创建及使用都做了统一的封装，简单易用，并且内置了一个 `cache` 模块实现了 `framebuffer` 的复用。
- 线程管理 `OpenGLContext` 不是多线程安全的，`GPUImage` 创建了专门的 `contextQueue`，所有的滤镜都会扔到统一的线程中处理。
- 轻松实现自定义滤镜效果 继承 `GPUImageFilter` 自动获得上面全部特性，无需关注上下文的环境搭建，专注于效果的核心算法实现即可。

总结

`GPUImage` 是一套主流的图像处理框架，很多直播、美图APP都采用此技术，当你的项目业务需要决定使用 `GPUImage` 还是 `Core Image`，它们都是相当成熟的工具。

滤镜链起点:

- `GPUImagePicture`

用来处理静态图片. 本质解压图片->纹理->用滤镜来进行处理.

- `GPUImageRawDataInput`

二进制数据->纹理图片. `CVPixelFormat`

- `GPUImageTextureInput`

用纹理数据.

- `GPUImageUIElement`

`UIView/CALayer` -> 图像数据 -> 纹理.

水印! 图片水印. `label`

- `GPUImageMovie`

视频文件 -> AVAssetReader -> 逐帧读取视频 -> 帧数据转化成纹理 -> 滤镜处理

AVAssetReaderOutput -> CMSamplerBufferRef -> CVImageBufferRef -> CVOpenGLTextureRef -> Texture

- GPUImageVideoCamera

基于 AVFoundation -> didOutputSampleBuffer ->

- 子类(GPUImageStillCamera)

滤镜

GPUImageFilter ;

滤镜终点

1. GPUImageMovieWriter

AVAssetWriter 把每一帧纹理的数据从帧缓存区-> 指定文件.

2. GPUImageRawDataOutput

处理滤镜帧缓存区的二进制数据

3. GPUImageTextureOutput

4. GPUImageView

你要知道GPUImage 和CoreImage已经帮我们实现了将近200个滤镜效果了.GPUImage库中提供的大部分滤镜都是通过片段着色器的一系列操作来实现相应的效果.大部分滤镜都是对图片中像素进行计算产生新的像素颜色处理.

那同学,不理解老师我们为什么要讲滤镜了?我们就理解了GPUImage是如何实现滤镜的,其实滤镜的关键在片元着色器.其实滤镜处理的原理:就是把静态图片或者视频的每一帧进行图形变换后再显示到屏幕上,其本质就是像素点的坐标和颜色的变化.

- GPU:(图形处理单元),手机或电脑用于图像处理和渲染的硬件.
- GPU的工作原理:CPU指定显示器工作, 显示控制器根据CPU的控制到指定

的地方去取数据和指令，目前的数据一般是从显存里取，如果显存里存不下，则从内存里取，内存也放不下，则从硬盘里取。

回忆OpenGL ES 处理图片的步骤

- 初始化OpenGL ES环境,编译,链接到顶点着色器和片元着色器
- 缓存顶点,纹理坐标数据,传送图像数据到GPU
- 绘制图元到特定的帧缓存区
- 在帧缓存区绘制图像.

GPUImage 框架解析

GPUImage 框架采用的 链式(Chain)结构.
主要有一个 GPUImageOutput interface 和 GPUImageInput protocol 串联起来.

GPUImageOutput 负责输出纹理 Texture ;
GPUImageInput 负责输入纹理 Texture ;

整个链式图像数据过程,纹理作为核心载体. 当然 GPUImage 不仅仅适用于静态图片,还适用视频,实时拍摄等. 那这些不同的载体都是继承于 GPUImageOutput 类

基本上每一个滤镜都是继承自 GPUImageFilter ; 而 GPUImageFilter 是整套框架的核心. 接收一个 GPUImageFramebuffer 输入,调用 GLProgram 渲染处理之后,输出一个 GPUImageFramebuffer . 在把输出的 GPUImageFramebuffer 传给通过 targets 属性关联的下一级滤镜. 直到传递给最终的输出组件.

课程总结

课程答疑
