

iOS实时相机App的实践

2018年4月21日
keyishen@陆家嘴软件园



目录

1

背景
介绍

2

实时
相机

3

3D
实践

4

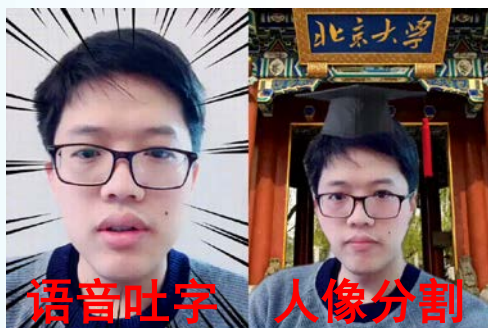
视频
拼接

5

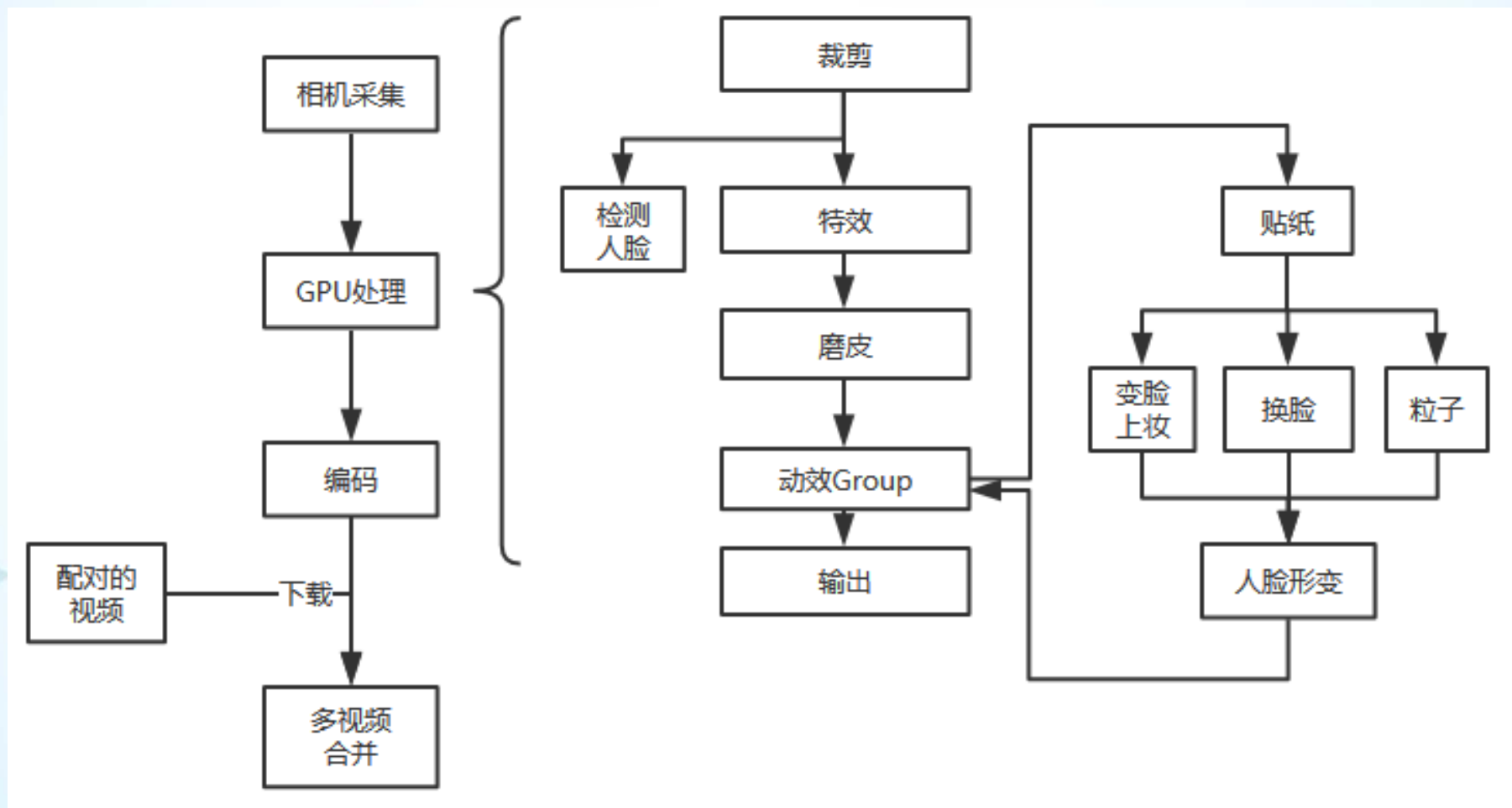
总结

实时相机&视频能力展示

- 丰富的实时动效能力
- 首家支持Face Track的App
- AR功能荣获App Store推荐
- 多人合演（视频拼接）



App流程图

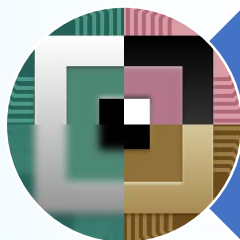


技术方案选型



OpenGL ES2.0

- 优点：平台通用性。OpenGL ES是基于C语言的，与平台无关
- 缺点：可扩展性差，不易于各个功能的组合



GPUImage


- 优点：基于OpenGL ES的封装，方便开发，高可复用和扩展性
- 缺点：性能没有裸写的OpenGL好



Metal

- 优点：基于现代GPU抽象了API，损耗低，性能是OpenGL ES的10倍
- 缺点：仅iPhone 5S iOS 8及以上手机才支持

提问?




在iPhone上如何看GPU调用?



视频如何支持透明通道?



3D人脸如何重建并运用?



两个视频如何快速拼接?

目录

1

背景
介绍

2

实时
相机

3

3D
实践

4

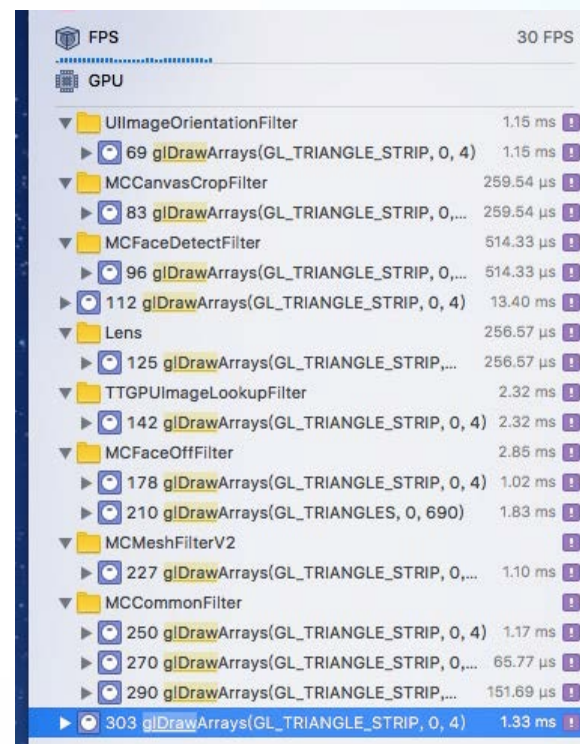
视频
拼接

5

总结

实时相机的具体实现

- 实时相机的逆向调试
 - Demo
- 实时相机的正向设计



The screenshot displays a GPU performance profiler interface. At the top, it shows 'FPS' at 30 FPS and 'GPU' as the active category. Below this, a list of rendering operations is shown, each with a duration and a small icon. The operations are grouped into folders, some of which are expanded to show sub-operations. The bottom-most operation is highlighted in blue.

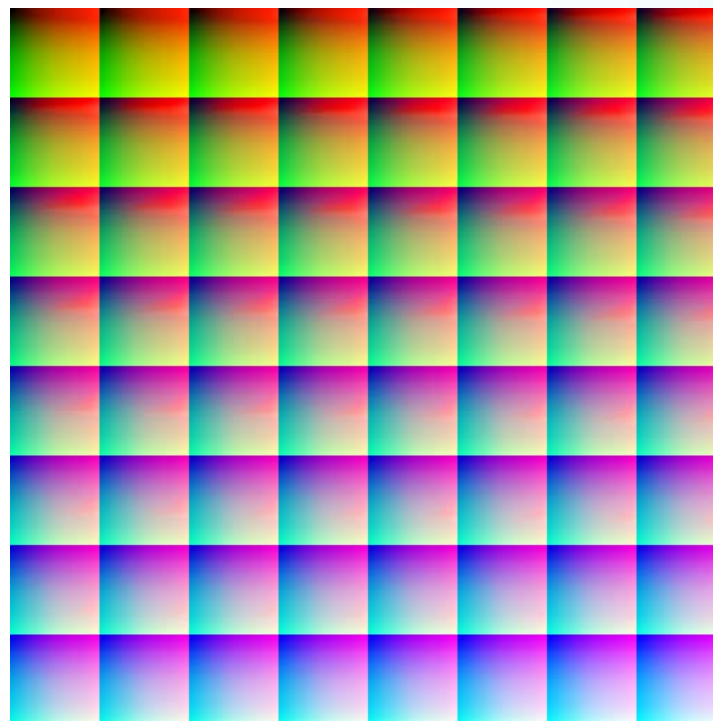
Operation	Duration
UImageOrientationFilter	1.15 ms
69 glDrawArrays(GL_TRIANGLE_STRIP, 0, 4)	1.15 ms
MCCanvasCropFilter	259.54 μs
83 glDrawArrays(GL_TRIANGLE_STRIP, 0, ...)	259.54 μs
MCFaceDetectFilter	514.33 μs
96 glDrawArrays(GL_TRIANGLE_STRIP, 0, ...)	514.33 μs
112 glDrawArrays(GL_TRIANGLE_STRIP, 0, 4)	13.40 ms
Lens	256.57 μs
125 glDrawArrays(GL_TRIANGLE_STRIP, ...)	256.57 μs
TTGPUImageLookupFilter	2.32 ms
142 glDrawArrays(GL_TRIANGLE_STRIP, 0, 4)	2.32 ms
MCFaceOffFilter	2.85 ms
178 glDrawArrays(GL_TRIANGLE_STRIP, 0, 4)	1.02 ms
210 glDrawArrays(GL_TRIANGLES, 0, 690)	1.83 ms
MCMeshFilterV2	1.10 ms
227 glDrawArrays(GL_TRIANGLE_STRIP, 0, ...)	1.10 ms
MCCCommonFilter	1.17 ms
250 glDrawArrays(GL_TRIANGLE_STRIP, 0, 4)	65.77 μs
270 glDrawArrays(GL_TRIANGLE_STRIP, 0, ...)	151.69 μs
290 glDrawArrays(GL_TRIANGLE_STRIP, ...)	1.33 ms
303 glDrawArrays(GL_TRIANGLE_STRIP, 0, 4)	1.33 ms

实时相机的流程

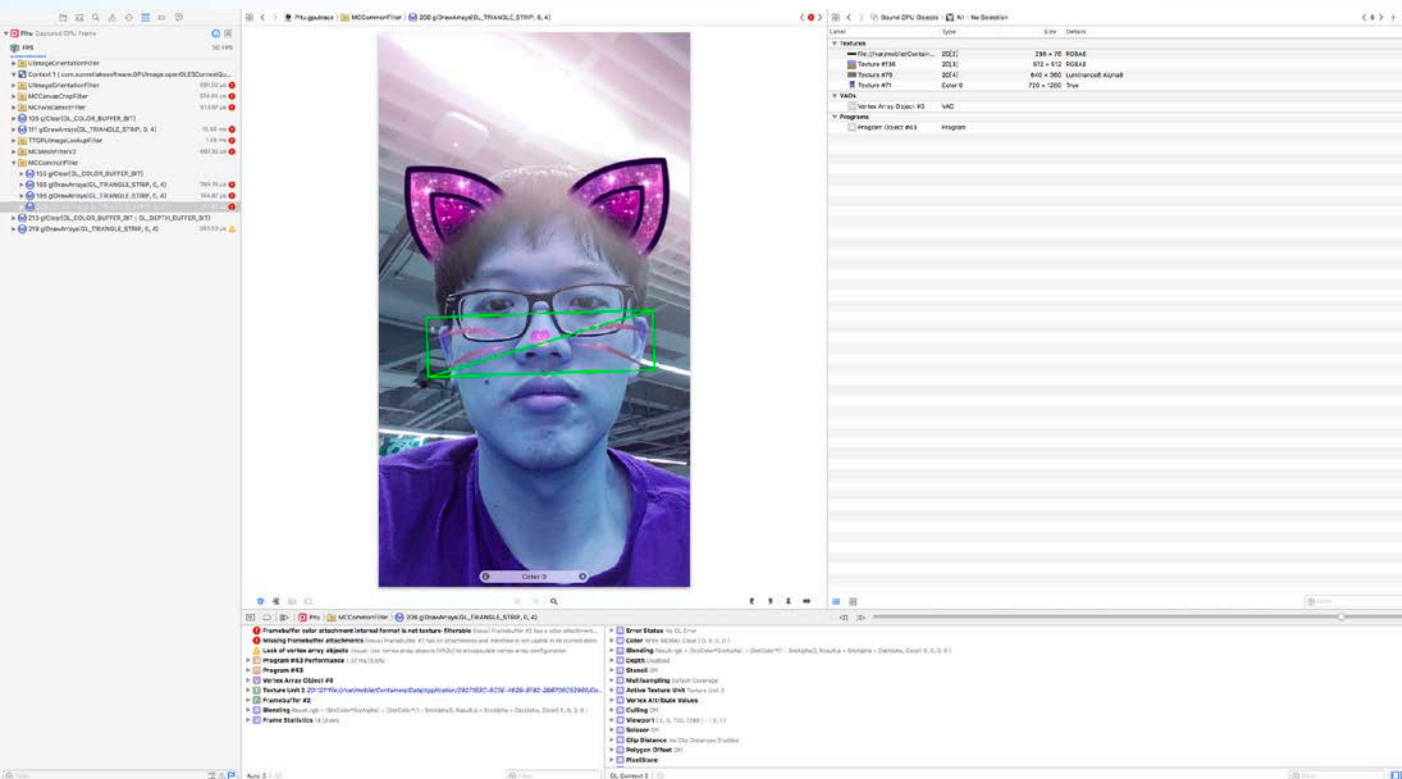


滤镜

- 滤镜LUT (LookUp Table)
 - 优点: LUT速度快
 - 缺点: 400KB一个滤镜

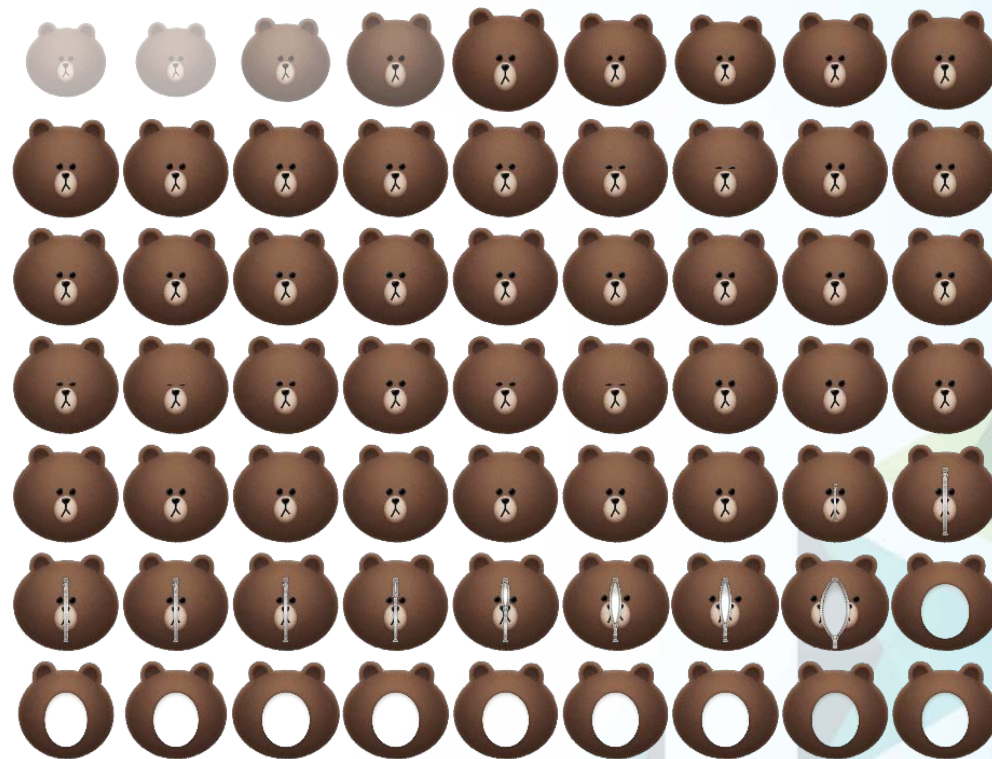


人像贴纸



人像贴纸支持的格式

- 序列帧：Sprite雪碧图
- PVR：减少内存
- 视频：如何支持透明通道？



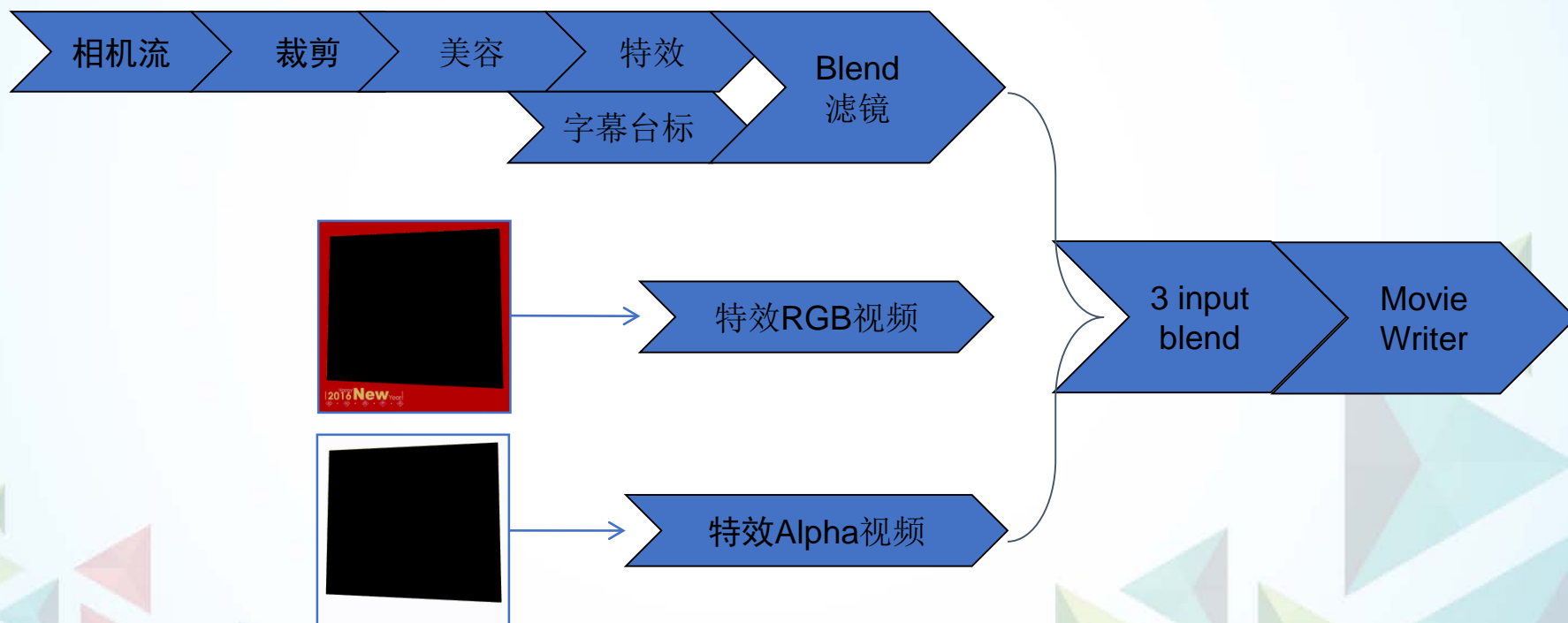
人像贴纸支持的格式

- Q:如何让视频支持透明通道?
- 设计理念
 - 普适
 - 简单
 - 快速
- 支持透明的视频格式
 - mov封装了一组png, 太大
 - webm无硬件解码
- 带透明色的视频
 - 有边缘毛刺



视频支持透明通道

• alpha视频+rgb视频方案

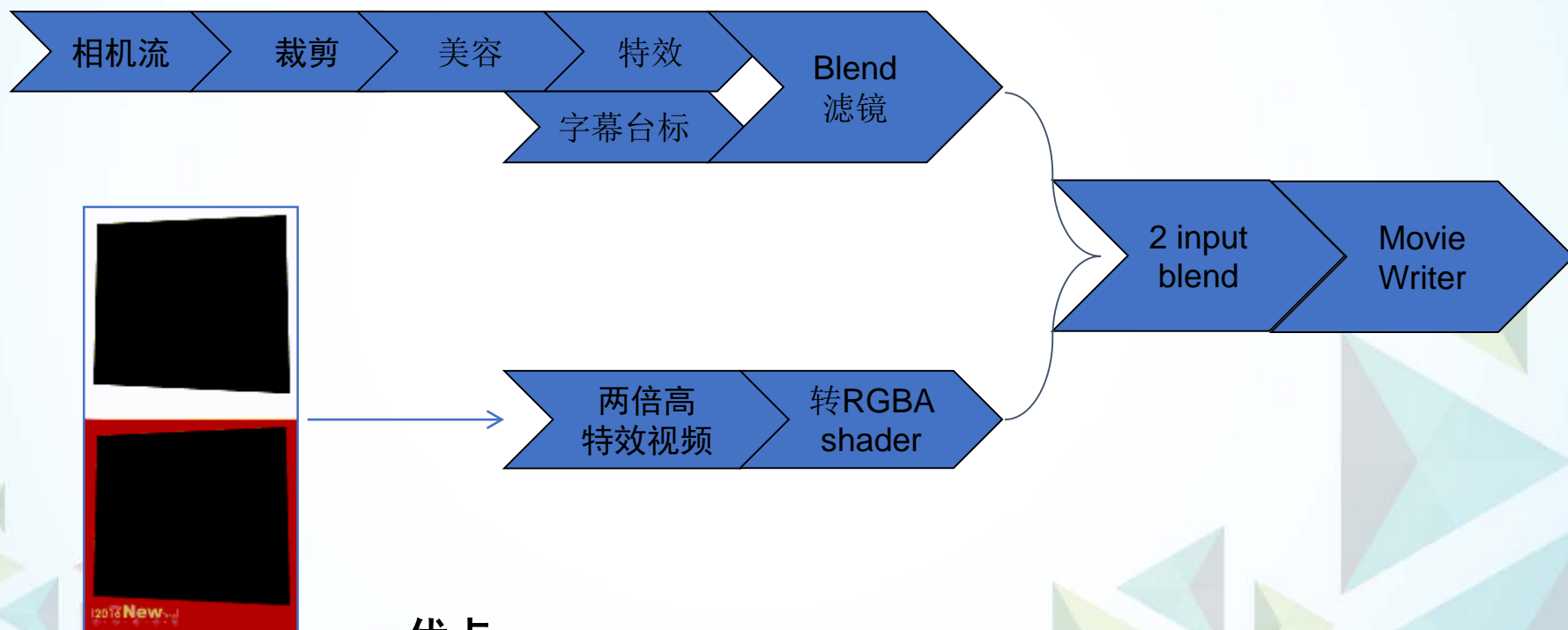


缺点:

- 双路解码压力大 (低端机跑不动)
- 多路之间要同步

视频支持透明通道

- 两倍高alpha视频方案 (480*960)



优点:

- 支持硬件解码, 速度快
- 改动小, 容易实现
- 解码压力小

目录

1

背景
介绍

2

实时
相机

3

3D
实践

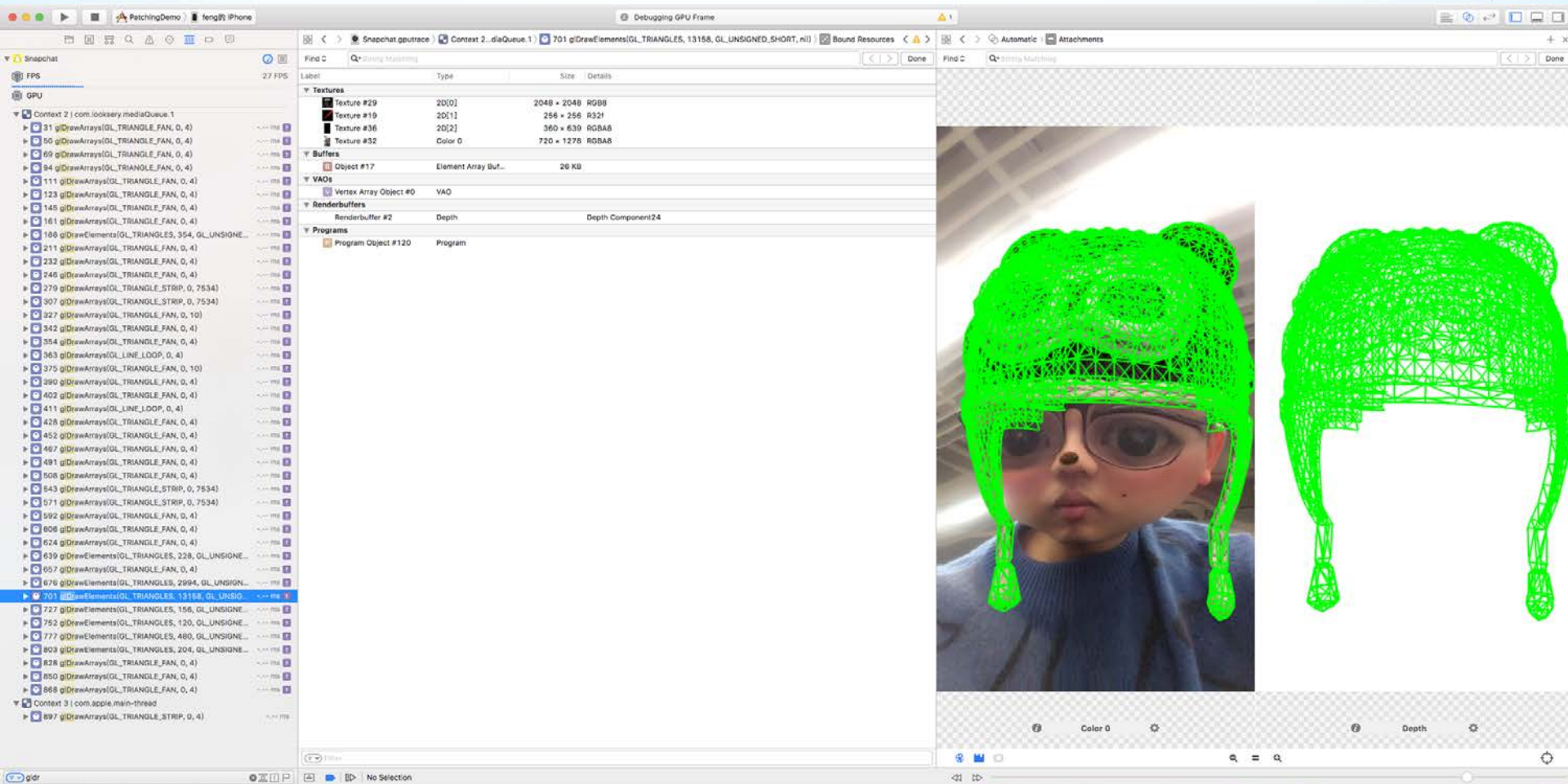
4

视频
拼接

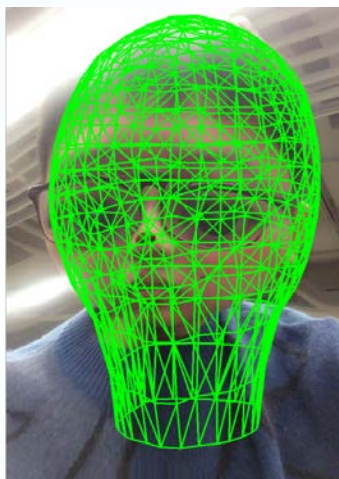
5

总结

人脸的3D模型



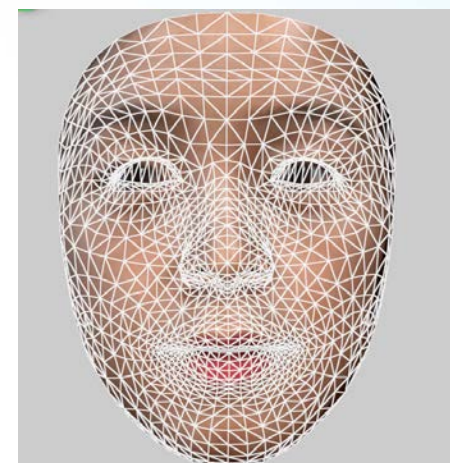
人脸3D建模对比



Snapchat



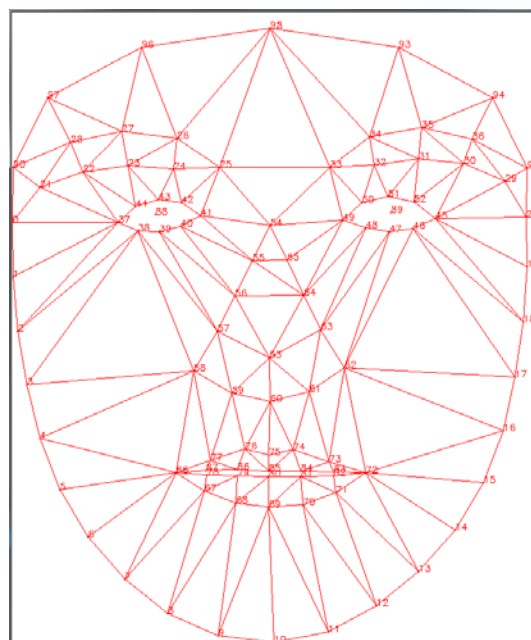
Youtu



iPhone X

换脸网格

- 自动算法+手动调节
- MeshLab

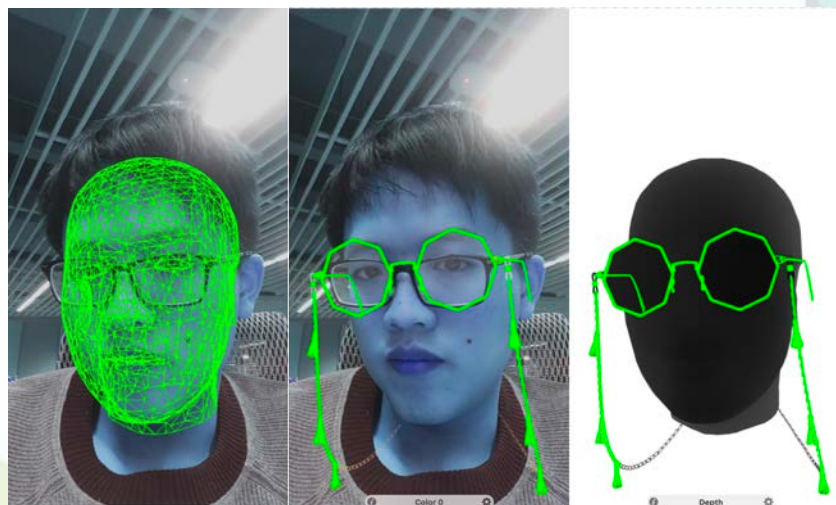
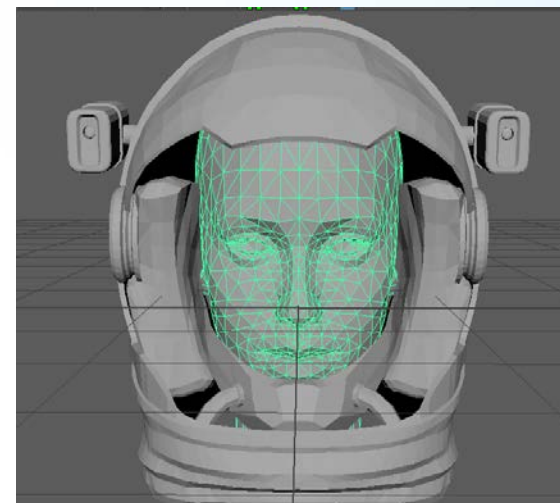


3D难点

• 3D框架选择

- OpenGL
- SceneKit
 - 和现有GPUImage共存
- Untiy

• 克服遮挡问题



目录

1

背景
介绍

2

实时
相机

3

3D
实践

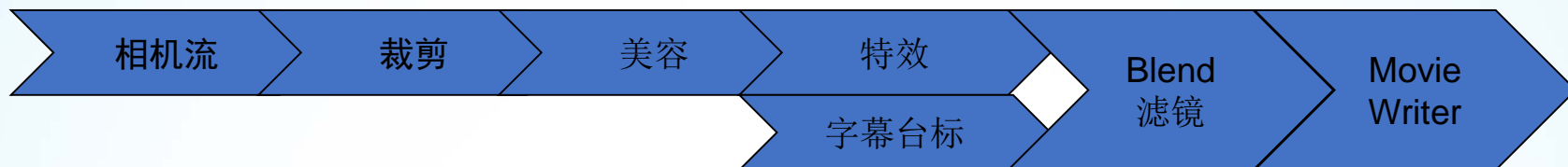
4

视频
拼接

5

总结

视频的录制



- 兼容安卓
 - 25fps, 1280kbps, 480*480, H.264 Baseline Profile 30 (iOS上可调的视频参数少)
- 低端机流畅录制
 - 按机型开启美容磨皮、特效滤镜
 - 非固定帧率

对口型级视频的精准拼接

- 方案1：保证录制视频所有参数一样？
 - 安卓和iOS视频的sps、pps很难完全一致
- 方案2：视频重新解码，再统一编码？
 - 编码速度很慢，体验差
- 方案3：MP4V2快速拼接
 - 多组sps,pps

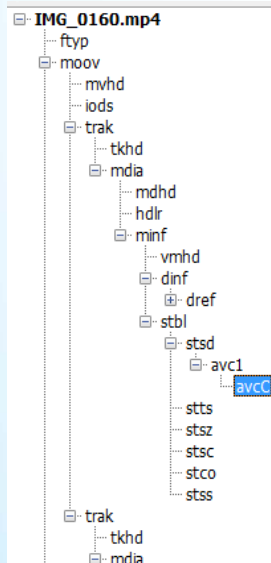
Notes:

- H.264每一个slice中公用的编码参数，决定了如何解码
- Sequence Parameter Set
 - Profile, level, 帧率等
- Picture Parameter Set
 - 图像相关参数，如量化初始值等

FTYP

MDAT

MOOV



AVC Decoder Configuration Record	
Start offset	511 (0X000001FF)
Box size	73 (0X00000049)
Box type	avcC (0X61766343)
Detailed Information	
Configuration version	1 (0X00000001)
AVC profile indication	Baseline = 66 (0X00000042)
AVC profile compatibility	0 (0X00000000)
AVC level indication	30 (0X0000001E)
NAL Unit length size	4 (0X00000004)
Num sequence parameter sets	2 (0X00000002)
Sequence parameter set (0)	0x67 0x42 0x00 0x1e 0xab 0x40 0xf0 0x7b 0x4d 0x40 0x40 0x41 0x80 0x80
Sequence parameter set (1)	0x67 0x42 0xc0 0x1e 0x56 0x40 0x78 0x3d 0xa6 0xa0 0x20 0x20 0x28 0x00 0x00 0x03 0x00 0x08 0x00 0x00 0x03 0x01 0x90 0x78 0xb1 0x72 0x40
Num picture parameter sets	2 (0X00000002)
Picture parameter set (0)	0x28 0xce 0x3c 0x30
Picture parameter set (1)	0x68 0x48 0xb8 0x3c 0xb2

对口型视频的精准拼接

MP4V2快速拼接

iOS视频
pps_id=0

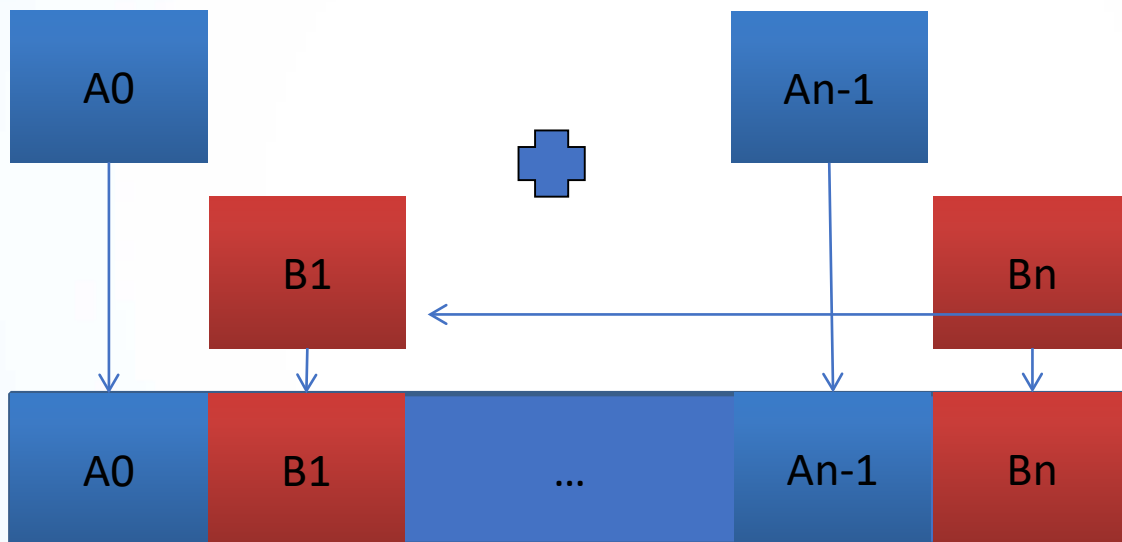
安卓视频
pps_id=1

拍档视频
下载模块

素材解析

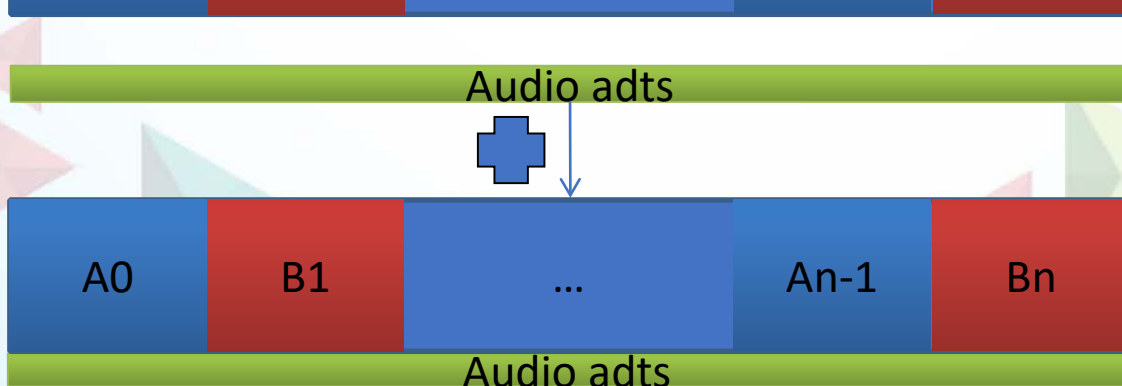
Stage1:

MP4v2合并
分段视频



Stage2:

MP4v2合并
音视频



对口型视频的精准拼接

iPhone6 iOS9录制的MP4
sps_id=0, pps_id=0

iPhone5s iOS9录制的MP4
sps_id=0, pps_id=0

FTYP | MOOV | MDAT

FTYP | MOOV | MDAT

Slice ppsid:0 | Slice ppsid:0 | Slice ppsid:0 | Slice ppsid:0

Slice ppsid:0 | Slice ppsid:0 | Slice ppsid:0 | Slice ppsid:0

Decoder: sps_table[32] Sequence Parameter Set

sps[0] | sps[1] | ...

pps_table[256] Picture Parameter Set

pps[0] | pps[1] | ...

sps
0x6742...
spsid:0

pps
0x28ce1f0c
ppsid:0
spsid:0

sps
0x6742...
spsid:0

pps
0x28ce1f0c
ppsid:0
spsid:0



对口型视频的精准拼接

iPhone6 iOS9录制的MP4
sps_id=0, pps_id=0

iPhone5s iOS9录制的MP4
sps_id=1, pps_id=1

FTYP

MOOV

MDAT

FTYP

MOOV

MDAT

Slice
ppsid:0

Slice
ppsid:0

Slice
ppsid:0

Slice
ppsid:0

Slice
ppsid:1

Slice
ppsid:1

Slice
ppsid:1

Slice
ppsid:1

Decoder: sps_table[32] Sequence Parameter Set

sps[0]

sps[1]

...

pps_table[256] Picture Parameter Set

pps[0]

pps[1]

...

sps
0x6742...
spsid:0

pps
0x28ce1f0c
ppsid:0
spsid:0

sps
0x6742...
spsid:1

pps
0x28ce1f0c
ppsid:1
spsid:1

目录

1

背景
介绍

2

实时
相机

3

3D
实践

4

视频
拼接

5

总结

回顾



在iPhone上如何看GPU调用？



视频如何支持透明通道？



3D人脸如何重建并运用？



两个视频如何快速拼接？



IT大咖说
知识共享平台

livevideoStack
— 音视频技术社区 —

Thank You

