

标签: [surface](#) [编码](#) [摄像头](#) [android](#)

2016-06-30 16:44

4201人阅读

[评论\(2\)](#)

[收藏](#)

[举报](#)

≡ 分类:

[android \(2\)](#) ▾ [encode](#)

版权声明: 本文为博主原创文章, 未经博主允许不得转载。
<http://blog.csdn.net/ilotuo/article/details/51791585>

目录(?)

[+]

息屏录像模块

基于以下原理我做出了完整的后头录像功能.不同于其它监控软件的伪后台(伪装窗口运行),此模块完全后台运行,即使锁屏状态也能监测和录像.不仅省电,而且隐蔽性极高,除了你其它人无法察觉.

基于此模块做出一个app.

功能有:

移动侦测录像. 此软件可以侦测摄像头范围内画面.当发生画面变化时自动开始录像,当动作停止一分钟后自动停止录像并保存.

脸部识别侦测录像.侦测到人脸时自动开始录像.

后台录像.这是此软件的特色.不同于其它监控软件的伪后台(伪装窗口运行),此软件完全后台运行,即使**锁屏状态也能监测和录像**.不仅省电,而且隐蔽性极高,除了你其它人无法察觉.

支持高清录像/录音/选择摄像头.

以下是正在补充的功能:- 用usb红外摄像头作为输入源(已测试部分设备可行)- 录像后自动上传到网盘- 录像后短信播放- 多路摄像头录像

使用场景:

场景: 打开软件后识别到人脸时自动后台拍摄.或者移动侦测后拍摄(锁屏下也有效).

可用于室内防盗/宠物记录/隐蔽拍摄等.

后台录像已做成 androidStudio 模块.需要此模块请私信咨询.

演示视频:

[http://v.youku.com/v_show/id_XMTY3NTA1NDk4MA==.html?
spm=a2hzp.8253869.0.0.4C43Bp&from=y1.7-2](http://v.youku.com/v_show/id_XMTY3NTA1NDk4MA==.html?spm=a2hzp.8253869.0.0.4C43Bp&from=y1.7-2)

Android从Surface编码原理

这是api 18之后的功能。下面尝试把手机摄像头preview渲染到Input Surface.然后对该Surface编码.

首先该Surface由MediaCodec创建:

```
1 | mSurface = mMediaCodec.createInputSurface(); // API >= 18
```

然后手动为该Surface初始化EGL后, 渲染GL画面, 编码。

add on 06/30/2016 SurfaceEncode Example .

编码输出

先看编码是怎么输出的.典型的MediaCodec Buffer编码,dequeueOutputBuffer的流水线顺序是这样:

- 1: changeFormat .
- 2: deque sps,pps buffer . BufferInfo 的flags被置位为:BUFFER_FLAG_CODEC_CONFIG.(如果是编码为h264,此时应写入h264文件)
- 3: IFrame (2+3 = IDR Frame)
- 4: many pFrame ...

但是当使用InputSurface时,第二步被跳过,也就是没有BUFFER_FLAG_CODEC_CONFIG buffer,此时应该手动取sps和pps :

```
1 | ByteBuffer sps = newFormat.getByteBuffer("csd-0");  
2 | ByteBuffer pps = newFormat.getByteBuffer("csd-1");
```

如果是H264裸流要先把这两个buffer写到文件头或推流.
详见[SufaceEncoder.md](#) drainAllEncoderMuxer函数.

初始化EGL

本例以手机摄像头录像为例.在preview渲染线程增加录像线程:

```
1 | mSurfaceEncoder = new SurfaceEncoder(thread.mVideoSource.mCols, thread.mVideoSource  
2 | Surface sur = mSurfaceEncoder.getInputSurface();
```

其中 mRendererHolder为preivew主线程.

seRecordingSSurface函数:

```
1 public void setRecordingSurface(final Surface surface) {  
2     RecordSurfaceRenderHandler rh = RecordSurfaceRenderHandler.createHandler();  
3     rh.setEglContext(mMasterEgl.getContext(), mTexId, surface, true);  
4     mClients.add(rh);  
5 }
```

RecordSurfaceRenderHandler 完成创建录制线程,为Surafce初始化EGL,在帧更新回调时切换EGL Context,并绘制到Surface.

[源码 RecordSurfaceRenderHandler.md](#)

关于EGL初始化,首先了解下EGL是什么,我这里看过两篇文章讲的比较好:

[#1 科学网—EGL资源的数据共享应用和底层驱动实现 - 郭叶军的博文](#)

[#2 学习OpenGL-ES: 2 - EGL解析 - kiffa - 博客园](#)

第一篇讲EGL原理,本质.第二篇将EGL应用.

通过以上文章我们知道,我们将要共享一块纹理,那么EGL context 是唯一.任何ELG Surface 渲染前都要和这个Context 绑定.使成为渲染目标.如果你的渲染主线程来自GLSurfaceView ,那么要先获得它的EGLContext,我还没有实践过,也许可以参考这个提问:[android - How can GLSurfaceView use my EGLDisplay, EGLContext and eglSurface? - Stack Overflow](#)

handleSetEglContext 函数为surface初始化EGL环境,EGLBase在grafika工程有,过程和原理结合源码参考上面的#2链接.可以认为其所做的都是为makeCurrent(切换渲染对象)时做准备.

EGLBase#makeCurrent实现就调用了一句:

```
1 EGL14.eglMakeCurrent(mEglDisplay, surface, surface, mEglContext)
```

将共享Context和我们的Surface绑定在一起.

Draw

把编码线程当成主线程的一个client,在主渲染线程帧回调时,向编码线程发送消息:

```
1 handler.sendMessage(handler.obtainMessage(MSG_RENDER_DRAW2, (int) (timestamp >> 32), (int) timesta
```

编码线程的处理: GLDrawer2D 负责编译shader 和调用GL ES接口进行渲染,以及Surface swap交换帧缓存.每次更新自动给MediaCodec输入帧数

据.RencordSurfaceRenderHandler#handleFrameAvailable 完成渲染和drain编码:

```
1 private void handleFrameAvailable(int tex_id, float[] transform, long timestampNanos) {
2     Log.v(TAG, "handleFrameAvailable #0");
3     SurfaceEncoder mVideoEncoder = SurfaceEncoder.getInstance();
4     if(mVideoEncoder==null || !mVideoEncoder.isRecording())
5         return;
6     Log.d(TAG, "handleDrain: #3");
7     mVideoEncoder.drainAllEncoderMuxer(false);
8     mDrawer.draw(tex_id, transform);
9     mTargetSurface.setPresentationTime(timestampNanos);
10    mTargetSurface.swap();
11    Log.v(TAG, "handleFrameAvailable #1");
12 }
```

mVideoEncoder.drainAllEncoderMuxer 编码输出前面讲过了,和BufferInput差不多,去掉Buffer输入和注意sps和pps保存即可.

录制线程的帧回调只是多了mTargetSurface.swap().其作用就是调用

GL14.eglSwapBuffers(mEglDisplay, surface) 将画到EglDisplay上的缓存换到Surface的帧缓存,录制线程的绘制.

reference :

[saki4510t/UVCamera: library and sample to access to UVC web camera on non-rooted Android device](#)

[google/grafika: Grafika test app](#)

- [上一篇](#) 窥探PTAM之基础知识准备
- [下一篇](#) 窥探PTAM之模板搜索

顶
1
踩
1

江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2017, CSDN.NET, All Rights Reserved 