

实验十七 Android 下的摄像头编程

一、实验目的

- (1) 学习 Android 下进行摄像头控制的基本方法。
- (2) 理解回调函数。

二、实验原理

1、android.hardware

android.hardware 包下面包含了如摄像头、传感器等硬件的支持（不是所有的硬件）。如表 1 所示。

表 1 android.hardware 中类[1]

类	类的功能
Camera	The Camera class is used to set image capture settings, start/stop preview, snap pictures, and retrieve frames for encoding for video.
Camera.Area	The Area class is used for choosing specific metering and focus areas for the camera to use when calculating auto-exposure, auto-white balance, and auto-focus.
Camera.CameraInfo	Information about a camera
Camera.Face	Information about a face identified through camera face detection.
Camera.Parameters	Camera service settings.
Camera.Size	Image size (width and height dimensions).
ConsumerIrManager	Class that operates consumer infrared on the device.
ConsumerIrManager.CarrierFrequencyRange	Represents a range of carrier frequencies (inclusive) on which the infrared transmitter can transmit
GeomagneticField	Estimates magnetic field at a given point on Earth, and in particular, to compute the magnetic declination from true north.
Sensor	Class representing a sensor.
SensorEvent	This class represents a Sensor event and holds informations such as the sensor's type, the time-stamp, accuracy and of course the sensor's data .
SensorManager	SensorManager lets you access the device's sensors .
TriggerEvent	This class represents a Trigger Event - the event associated with a Trigger Sensor.
TriggerEventListener	This class is the listener used to handle Trigger Sensors.

2、android.hardware.Camera 类

现在的手机一般都会提供相机功能，有些相机的镜头甚至支持 1000 万以上像素，有些甚至支持光学变焦，这些手机已经变成了专业数码相机。为了充分利用手机上的相机功能，Android APIs 提供了丰富的支持库。如表 2 所示。

使用 Android 设备的摄像头（照相机）需要设备访问权限如下。

```
<uses-permission android:name="android.permission.CAMERA" />
```

```
<uses-feature android:name="android.hardware.camera" />
```

```
<uses-feature android:name="android.hardware.camera.autofocus" />
```

表 2 android.hardware.Camera 的成员函数

	成员函数名称
final void	addCallbackBuffer (byte[] callbackBuffer) Adds a pre-allocated buffer to the preview callback buffer queue.
final void	autoFocus (Camera.AutoFocusCallback cb) Starts camera auto-focus and registers a callback function to run when the camera is focused.
final void	cancelAutoFocus () Cancels any auto-focus function in progress.
final boolean	enableShutterSound (boolean enabled) Enable or disable the default shutter sound when taking a picture.
static void	getCameraInfo (int cameraId, Camera.CameraInfo cameraInfo) Returns the information about a particular camera.
static int	getNumberOfCameras () Returns the number of physical cameras available on this device.
Camera.Parameters	getParameters () Returns the current settings for this Camera service.
final void	lock () Re-locks the camera to prevent other processes from accessing it.
static Camera	open (int cameraId) Creates a new Camera object to access a particular hardware camera.
static Camera	open () Creates a new Camera object to access the first back-facing camera on the device.
final void	reconnect () Reconnects to the camera service after another process used it.
final void	release () Disconnects and releases the Camera object resources.
void	setAutoFocusMoveCallback (Camera.AutoFocusMoveCallback cb) Sets camera auto-focus move callback.
final void	setDisplayOrientation (int degrees) Set the clockwise rotation of preview display in degrees.
final void	set errorCallback (Camera.ErrorCallback cb) Registers a callback to be invoked when an error occurs.
final void	setFaceDetectionListener (Camera.FaceDetectionListener listener) Registers a listener to be notified about the faces detected in the preview frame.
final void	setOneShotPreviewCallback (Camera.PreviewCallback cb) Installs a callback to be invoked for the next preview frame in addition to displaying it on the screen.
void	setParameters (Camera.Parameters params) Changes the settings for this Camera service.
final void	setPreviewCallback (Camera.PreviewCallback cb) Installs a callback to be invoked for every preview frame in addition to displaying them on the screen.
final void	setPreviewCallbackWithBuffer (Camera.PreviewCallback cb) Installs a callback to be invoked for every preview frame, using buffers supplied with addCallbackBuffer (byte[]), in addition to displaying them on the screen.
final void	setPreviewDisplay (SurfaceHolder holder) Sets the Surface to be used for live preview.

final void	<code>setPreviewTexture(SurfaceTexture surfaceTexture)</code> Sets the <code>SurfaceTexture</code> to be used for live preview.
final void	<code>setZoomChangeListener(Camera.OnZoomChangeListener listener)</code> Registers a listener to be notified when the zoom value is updated by the camera driver during smooth zoom.
final void	<code>startFaceDetection()</code> Starts the face detection.
final void	<code>startPreview()</code> Starts capturing and drawing preview frames to the screen.
final void	<code>startSmoothZoom(int value)</code> Zooms to the requested value smoothly.
final void	<code>stopFaceDetection()</code> Stops the face detection.
final void	<code>stopPreview()</code> Stops capturing and drawing preview frames to the surface, and resets the camera for a future call to <code>startPreview()</code> .
final void	<code>stopSmoothZoom()</code> Stops the smooth zoom.
final void	<code>takePicture(Camera.ShutterCallback shutter, Camera.PictureCallback raw, Camera.PictureCallback jpeg)</code> Equivalent to <code>takePicture(shutter, raw, null, jpeg)</code> .
final void	<code>takePicture(Camera.ShutterCallback shutter, Camera.PictureCallback raw, Camera.PictureCallback postview, Camera.PictureCallback jpeg)</code> Triggers an asynchronous image capture.
final void	<code>unlock()</code> Unlocks the camera to allow another process to access it.

3、Camera2 类说明

Android 5.0(SDK 21)之后 `android.hardware.Camera` 就被废弃了，取而代之的是全新的 `android.hardware.Camera2`。Android 5.0 对拍照 API 进行了全新的设计，新增了全新设计的 Camera v2 API，这些 API 不仅大幅提高了 Android 系统拍照的功能，还能支持 RAW 照片输出，甚至允许程序调整相机的对焦模式、曝光模式、快门等[2]。

- **CameraManager**: 摄像头管理器。这是一个全新的系统管理器，专门用于检测系统摄像头、打开系统摄像头。除此之外，调用 **CameraManager** 的 `getCameraCharacteristics(String)` 方法即可获取指定摄像头的相关特性。
- **CameraCharacteristics**: 摄像头特性。该对象通过 **CameraManager** 来获取，用于描述特定摄像头所支持的各种特性。
- **CameraDevice**: 代表系统摄像头。该类的功能类似于早期的 **Camera** 类。
- **CameraCaptureSession**: 这是一个非常重要的 API，当程序需要预览、拍照时，都需要先通过该类的实例创建 Session。而且不管预览还是拍照，也都是由该对象的方法进行控制的，其中控制预览的方法为 `setRepeatingRequest()`；控制拍照的方法为 `capture()`。
- **CameraRequest** 和 **CameraRequest.Builder**: 当程序调用 `setRepeatingRequest()` 方法进行预览时，或调用 `capture()` 方法进行拍照时，都需要传入 **CameraRequest** 参数。**CameraRequest** 代表了一次捕获请求，用于描述捕获图片的各种参数设置，比如对焦模式、曝光模式……总之，程序需要对照片所做的各种控制，都通过

`CameraRequest` 参数进行设置。`CameraRequest.Builder` 则负责生成 `CameraRequest` 对象。

4、`android.hardware.ImageReader` 类

与播放声音不同，在使用摄像头的时候，我们需要把摄像头捕获的图像进行显示。示例程序中使用 `ImageReader` 类完成摄像头捕获的图像数据到界面上显示控件的显示。类 `ImageReader` 支持图像数据的着色（The `ImageReader` class allows direct application access to image data rendered into a `Surface`）。

表 3 `android.hardware.ImageReader` 类的成员函数[3]

函数返回值类型	成员函数名称
<code>Image</code>	<code>acquireLatestImage()</code> Acquire the latest <code>Image</code> from the <code>ImageReader</code> 's queue, dropping older <code>images</code> .
<code>Image</code>	<code>acquireNextImage()</code> Acquire the next <code>Image</code> from the <code>ImageReader</code> 's queue.
<code>void</code>	<code>close()</code> Free up all the resources associated with this <code>ImageReader</code> .
<code>int</code>	<code>getHeight()</code> The height of each <code>Image</code> , in pixels.
<code>int</code>	<code>getImageFormat()</code> The <code>image format</code> of each <code>Image</code> .
<code>int</code>	<code>getMaxImages()</code> Maximum number of images that can be acquired from the <code>ImageReader</code> by any time (for example, with <code>acquireNextImage()</code>).
<code>Surface</code>	<code>getSurface()</code> Get a <code>Surface</code> that can be used to produce <code>Images</code> for this <code>ImageReader</code> .
<code>int</code>	<code>getWidth()</code> The width of each <code>Image</code> , in pixels.
<code>static ImageReader</code>	<code>newInstance(int width, int height, int format, int maxImages)</code> Create a new reader for images of the desired size and format.
<code>void</code>	<code>setOnImageAvailableListener(ImageReader.OnImageAvailableListener listener, Handler handler)</code> Register a listener to be invoked when a new image becomes available from the <code>ImageReader</code> .

表 4 `android.hardware.ImageReader` 类的回调接口[3]

<code>interface</code>	<code>ImageReader.OnImageAvailableListener</code>	Callback interface for being notified that a new image is available.
------------------------	---	--

5、回调函数

回调函数就是一个通过函数指针调用的函数。如果你把函数的指针（地址）作为参数传递给另一个函数，当这个指针被用来调用其所指向的函数时，我们就说这是回调函数。回调函数不是由该函数的实现方直接调用，而是在特定的事件或条件发生时由另外的一方调用的，用于对该事件或条件进行响应。

多媒体操作往往需要回调函数的支持，例如，我们通过摄像头捕获图像，启动摄像头后，会通知摄像头数据采集有关的函数，等采集完一帧图像后，你要做一些处理（回调函数中就是这些处理的具体代码）。

三、 示例说明

示例代码用到的主要的类如下。

private CameraManager **mCameraManager**;

** The ImageReader class allows direct application access to image data rendered into a Surface */*

private ImageReader **mImageReader**;

*/*系统向摄像头发送 Capture 请求，而摄像头会返回 CameraMetadata，这一切都是在由对应的 CameraDevice 创建的 CameraCaptureSession 会话完成，当程序需要预览、拍照、再次预览时，都需要先通过会话*/*

private CameraCaptureSession **mCameraCaptureSession**;

*/*CameraDevice 是 Camera2 中抽象出来的一个对象，直接与系统硬件摄像头相联系*/*

private CameraDevice **mCameraDevice**;

*/*CameraRequest 代表了一次捕获请求，而 CameraRequest.Builder 用于描述捕获图片的各种参数设置，包含捕获硬件（传感器，镜头，闪存），对焦模式、曝光模式，处理流水线，控制算法和输出缓冲区的配置*/*

private CaptureRequest.Builder **pictureBuilder**;

四、 实验要求

- (1) 编译示例程序，运行结果截图放入实验报告。
- (2) 读懂示例程序 VideoCapture 的操作步骤，在实验报告中画出访问摄像头的流程示意图。

五、 思考题

- (1) 示例代码中用到的 TextureView 与一般的 View 有什么不同？
- (2) 阐述一下个人关于 Callback 函数用途的理解。

参考文献

- [1] AndroidAPIs. android.hardware [Online]. Available: <http://www.android-doc.com/reference/android/hardware/package-summary.html>
- [2] 简书 . Android Camera2 使用总结 [Online]. Available: <https://www.jianshu.com/p/73fed068a795>
- [3] AndroidAPIs. ImageReader [Online]. Available: <http://www.android-doc.com/reference/android/media/ImageReader.html>