

多媒体进阶-拍照与录像

字节跳动Android工程师 张明庆



张明庆

字节跳动 Android工程师

- 毕业于浙江大学和中科院
- 14年毕业，先后在网易、得到等工作
- 18年加入字节跳动，目前负责火山小视频
Android 端架构和性能优化工作
- 组件化开源项目 DDComponent 作者

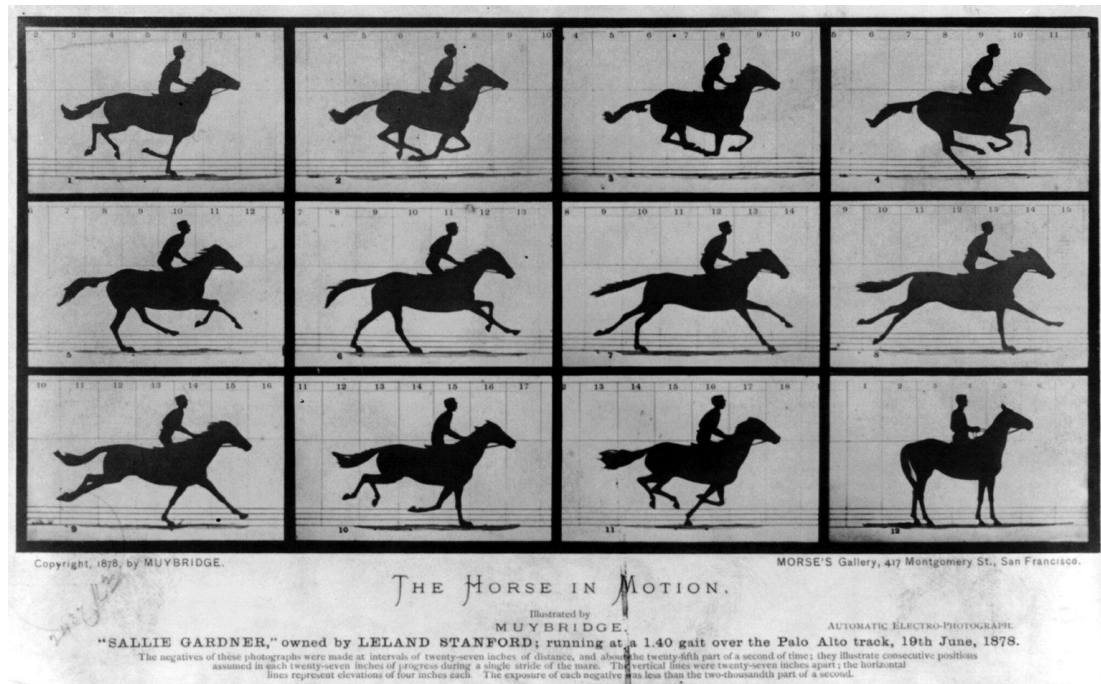


目录

- ❑ 从图片到视频
- ❑ 相机拍照
- ❑ 最简单的录制
- ❑ 自定义录制
- ❑ 课后探索

第一章 从图片到视频

电影的诞生



回顾几个概念

❑ 帧率

每秒的帧数，视觉暂留需要大于24帧，一般都在30帧以上

❑ 分辨率

每帧图像的宽度和长度，例如 $1080 * 1920$

❑ 码率

每秒传送的比特(bit)数，用于视频压缩后得到一个期望的大小

回顾几个概念

□ 视频的编码

相邻图像之间有冗余，可以压缩成一个关键帧+变化差值

编码格式就是不同的压缩算法：MPEG/H264

□ 视频的封装格式

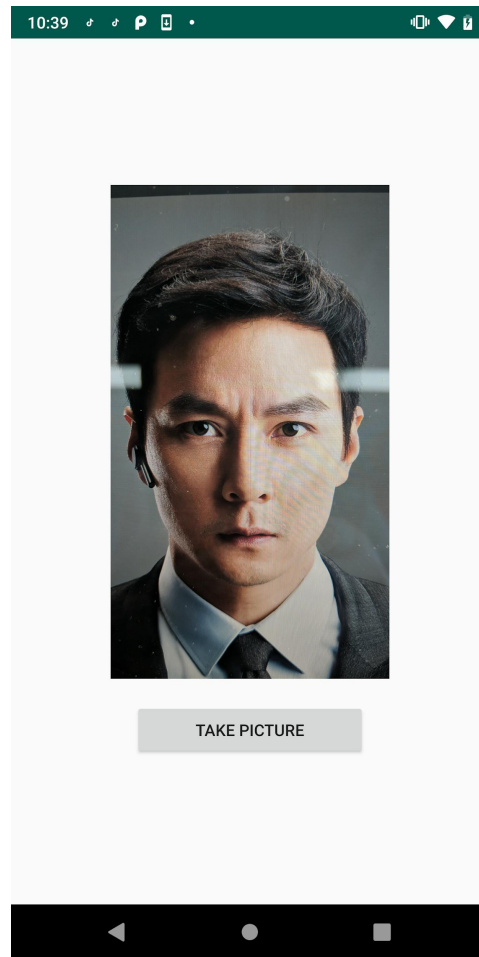
把视频码流和音频码流按照一定的格式存储在一个文件中，与编码格式无关

第二章 相机拍照



效果展示

- ❑ 拍摄一个自拍照
- ❑ 显示在屏幕上
- ❑ 显示图片--ImageView
- ❑ 吊起拍摄--Button



步骤一 调起系统相机

❑ 声明权限

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

❑ 申请相机权限

- ContextCompat.checkSelfPermission
- ActivityCompat.requestPermissions
- onRequestPermissionsResult

步骤二 调起系统相机

□ 如何调起相机

```
1 Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
2 startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
```

步骤三 接收数据

- 接收数据，拿到返回的bitmap，并显示在屏幕上

```
1 @Override
2 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
3     if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
4         Bundle extras = data.getExtras();
5         Bitmap imageBitmap = (Bitmap) extras.get("data");
6         mImageView.setImageBitmap(imageBitmap);
7     }
8 }
```

- 为啥图像这么小？

步骤四 自定义存储路径 (一)

❑ 申请存储权限

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

❑ 创建文件

```
1 public static File getOutputMediaFile(int type) {  
2     File mediaStorageDir = new File(Environment.getExternalStoragePublicDirectory(  
3         Environment.DIRECTORY_PICTURES), "CameraDemo");  
4     // Create a media file name  
5     String timeStamp = new SimpleDateFormat("yyyyMMdd_HHmmss").format(new Date());  
6     File mediaFile = new File(mediaStorageDir.getPath() + File.separator +  
7         "IMG_" + timeStamp + ".jpg");  
8     return mediaFile;  
9 }
```

步骤四 自定义存储路径 (二)

❑ 获取content:// URI，7.0以上手机不允许使用file:// URI跳出应用

```
1 <provider
2     android:name="android.support.v4.content.FileProvider"
3     android:authorities="com.bytedance.camera.demo"
4     android:exported="false"
5     android:grantUriPermissions="true">
6     <meta-data
7         android:name="android.support.FILE_PROVIDER_PATHS"
8         android:resource="@xml/file_paths" />
9 </provider> |
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <paths xmlns:android="http://schemas.android.com/apk/res/android">
3     <external-path
4         name="sdcard"
5         path="/" />
6 </paths>
```

步骤四 自定义存储路径 (三)

❑ 设置存储地址

```
1 Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
2 imgFile = Utils.getOutputMediaFile(MEDIA_TYPE_IMAGE);
3 if (imgFile != null) {
4     Uri fileUri =
5         FileProvider.getUriForFile(this, "com.bytedance.camera.demo", imgFile);
6     takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, fileUri);
7     startActivityForResult(takePictureIntent, REQUEST_IMAGE_CAPTURE);
8 }
```

步骤五 显示图片

- ❖ 获取view的宽高
- ❖ 获取图片的宽高
- ❖ 计算缩放比例
- ❖ 获取bitmap
- ❖ 显示在屏幕上

```
1 @Override
2 protected void onActivityResult(int requestCode, int resultCode, Intent data) {
3     if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
4         int targetW = imageView.getWidth();
5         int targetH = imageView.getHeight();
6         // Get the dimensions of the bitmap
7         BitmapFactory.Options bmOptions = new BitmapFactory.Options();
8         bmOptions.inJustDecodeBounds = true;
9         BitmapFactory.decodeFile(imgFile.getAbsolutePath(), bmOptions);
10        int photoW = bmOptions.outWidth;
11        int photoH = bmOptions.outHeight;
12        // Determine how much to scale down the image
13        int scaleFactor = Math.min(photoW / targetW, photoH / targetH);
14        // Decode the image file into a Bitmap sized to fill the View
15        bmOptions.inJustDecodeBounds = false;
16        bmOptions.inSampleSize = scaleFactor;
17        bmOptions.inPurgeable = true;
18
19        Bitmap bmp=BitmapFactory.decodeFile(imgFile.getAbsolutePath(), bmOptions);
20        imageView.setImageBitmap(bmp);
21    }
22 }
```


步骤六 图片为啥旋转了？

- ❖ 读取图片的旋转角度
- ❖ 在matrix中设置要旋转的角度
- ❖ 旋转图片

```
1 public static Bitmap rotateImage(Bitmap bitmap, String path) throws Exception {
2     ExifInterface srcExif = new ExifInterface(path);
3     Matrix matrix = new Matrix();
4     int angle = 0;
5     int orientation = srcExif.getAttributeInt(ExifInterface.TAG_ORIENTATION, ExifInterface.
    terface.ORIENTATION_NORMAL);
6     switch (orientation) {
7         case ExifInterface.ORIENTATION_ROTATE_90:
8             angle = NUM_90;
9             break;
10        case ExifInterface.ORIENTATION_ROTATE_180:
11            angle = NUM_180;
12            break;
13        case ExifInterface.ORIENTATION_ROTATE_270:
14            angle = NUM_270;
15            break;
16        default:
17            break;
18    }
19    matrix.postRotate(angle);
20    return Bitmap.createBitmap(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(),
    matrix, true);
21 }
```

随堂练习 (30min)

❑ 给自己来个自拍

- ❑ 解决权限申请
- ❑ 存储到sd卡
- ❑ 图片预览方向正确

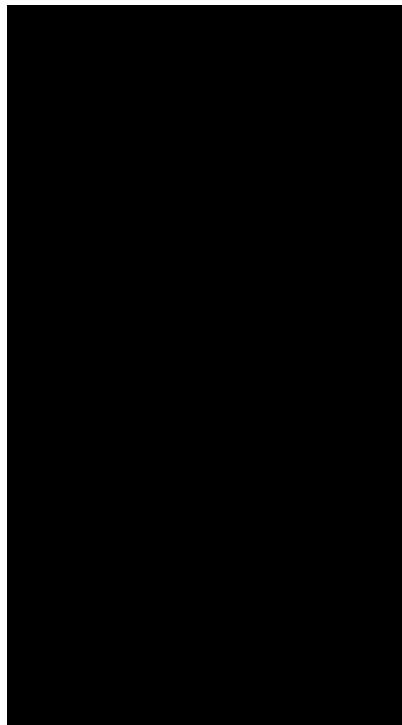
❑ 拓展-在相册中能扫描到自拍照片

第三章 最简单的录制



效果展示

- ❑ 吊起手机相机录一段视频
- ❑ 录制的视频显示到屏幕上
- ❑ 显示视频--VideoView
- ❑ 吊起相机--Button



步骤一 调起系统相机

❑ 申请权限

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

❑ 调起相机的录像页面

```
1 Intent takeVideoIntent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
2 if (takeVideoIntent.resolveActivity(getPackageManager()) != null) {
3     startActivityForResult(takeVideoIntent, REQUEST_VIDEO_CAPTURE);
4 }
```

步骤二 获取返回数据

- 获取拍摄的视频，并显示在页面上，开始播放

```
1 @Override
2 protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
3     if (requestCode == REQUEST_VIDEO_CAPTURE && resultCode == RESULT_OK) {
4         Uri videoUri = intent.getData();
5         videoView.setVideoURI(videoUri);
6         videoView.start();
7     }
8 }
```

步骤三 查看数据

- ❑ 视频的封装格式

.mp4

- ❑ 视频的分辨率是多大？

720 * 1280

- ❑ 视频的文件大小和录制时长

7.15MB / 8秒

- ❑ 计算视频的码率

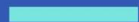
$7.15 * 1024 * 1024 * 8 / 8 = 7497.3 \text{ kbps}$

随堂练习 (15min)

❑ 录制一段自拍视频

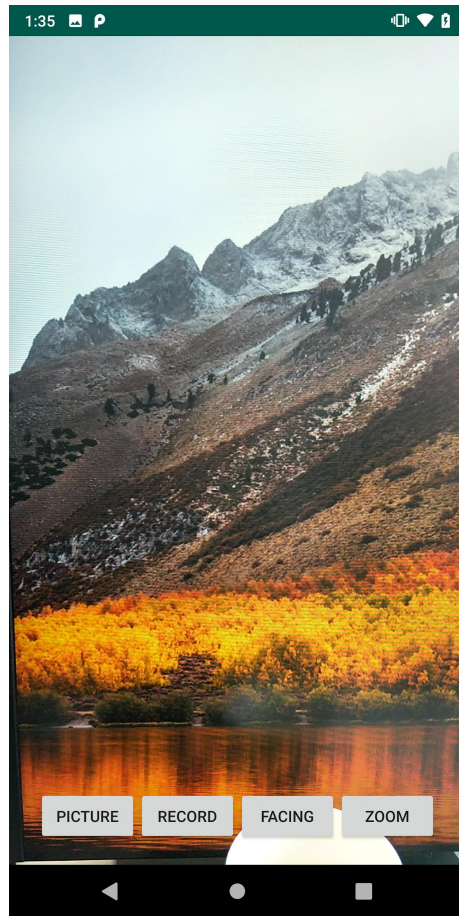
- ❑ 解决权限申请
- ❑ 默认存储
- ❑ 相机拍摄后在页面上播放
- ❑ 点击暂停，再次点击恢复播放

第四章 自定义录制



效果展示

- ❑ 吊起相机
- ❑ 屏幕实时显示画面
- ❑ 拍个照片
- ❑ 录一段视频
- ❑ 实时显示-SurfaceView



步骤一 获取Camera实例

❑ 申请权限

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

❑ 一共几个摄像头

Camera.getNumberOfCameras

❑ 怎么获取后置摄像头

```
1 releaseCameraAndPreview();
2 Camera cam = Camera.open(Camera.CameraInfo.CAMERA_FACING_BACK);
3 rotationDegree = getCameraDisplayOrientation(position);
4 cam.setDisplayOrientation(rotationDegree);
```

步骤二 摄像头数据实时显示

❑ 用什么控件？

SurfaceView or TextureView

❑ 几个关键类

Camera

SurfaceView

SurfaceHolder

SurfaceHolder.Callback

```
1 Camera mCamera = getCamera();
2 SurfaceView mSurfaceView = findViewById(R.id.img);
3 SurfaceHolder surfaceHolder = mSurfaceView.getHolder();
4 surfaceHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
5 surfaceHolder.addCallback(new SurfaceHolder.Callback() {
6     @Override
7     public void surfaceCreated(SurfaceHolder holder) {
8         mCamera.setPreviewDisplay(holder);
9         mCamera.startPreview();
10    }
11    @Override
12    public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {}
13    @Override
14    public void surfaceDestroyed(SurfaceHolder holder) {
15        mCamera.stopPreview();
16        mCamera.release();
17        mCamera = null;
18    }
19 });
```

步骤三 拍摄一张实时照片

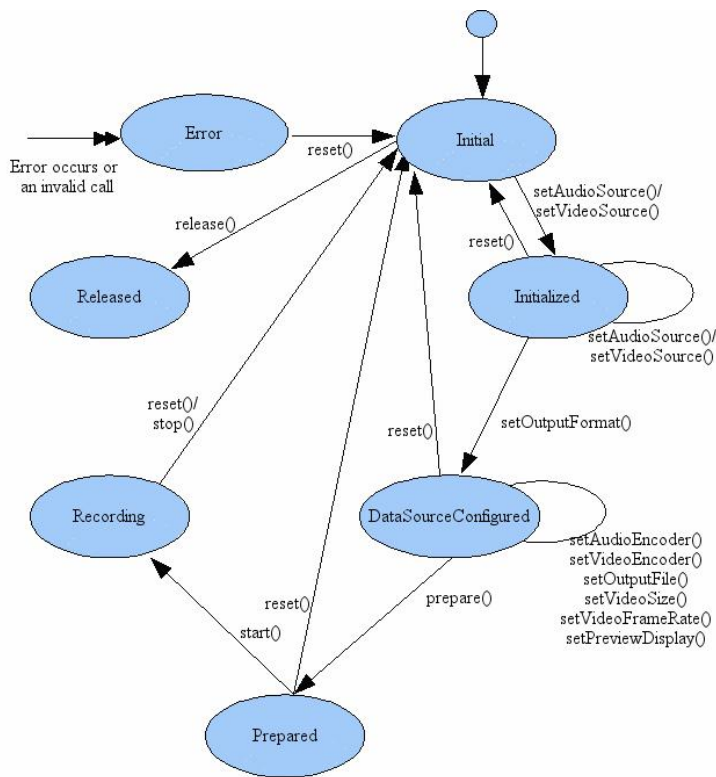
❑ 怎么用 camera api 拍照

`mCamera.takePicture(null, null, mPicture)`

❑ 拍照后继续预览

```
1 private Camera.PictureCallback mPicture = (data, camera) -> {  
2     File pictureFile = getOutputMediaFile(MEDIA_TYPE_IMAGE);  
3     try {  
4         FileOutputStream fos = new FileOutputStream(pictureFile);  
5         fos.write(data);  
6         fos.close();  
7     } catch (IOException e) {  
8         Log.d("mPicture", "Error accessing file: " + e.getMessage());  
9     }  
10    mCamera.startPreview();  
11 };
```

步骤四 认识MediaRecorder



MediaRecorder state diagram

步骤五 开始录制（按部就班）

❑ Unlock the Camera

❑ Configure MediaRecorder

- ❑ `setCamera()`
- ❑ `setAudioSource()`
- ❑ `setVideoSource()`
- ❑ `setProfile`
- ❑ `setOutputFile()`
- ❑ `setPreviewDisplay()`

❑ Prepare MediaRecorder

❑ Start MediaRecorder

```
1 mMediaRecorder = new MediaRecorder();
2 // Step 1: Unlock and set camera to MediaRecorder
3 mCamera.unlock();
4 mMediaRecorder.setCamera(mCamera);
5 // Step 2: Set sources
6 mMediaRecorder.setAudioSource(MediaRecorder.AudioSource.CAMCORDER);
7 mMediaRecorder.setVideoSource(MediaRecorder.VideoSource.CAMERA);
8 // Step 3: Set a CamcorderProfile (requires API Level 8 or higher)
9 mMediaRecorder.setProfile(CamcorderProfile.get(CamcorderProfile.QUALITY_HIGH));
10 // Step 4: Set output file
11 mMediaRecorder.setOutputFile(getOutputMediaFile(MEDIA_TYPE_VIDEO).toString());
12 // Step 5: Set the preview output
13 mMediaRecorder.setPreviewDisplay(mSurfaceView.getHolder().getSurface());
14 mMediaRecorder.setOrientationHint(rotationDegree);
15 // Step 6: Prepare configured MediaRecorder
16 try {
17     mMediaRecorder.prepare();
18     mMediaRecorder.start();
19 } catch (Exception e) {
20     releaseMediaRecorder();
21     return false;
22 }
```

步骤六 结束录制（按部就班）

- ❑ Stop MediaRecorder
- ❑ Reset MediaRecorder
- ❑ Release MediaRecorder
- ❑ Lock the Camera

```
1 mMediaRecorder.stop();  
2 mMediaRecorder.reset();  
3 mMediaRecorder.release();  
4 mMediaRecorder = null;  
5 mCamera.lock();
```


随堂练习 (30min)

❑ 录制一段教室内的视频

- ❑ 解决权限申请
- ❑ 存储到sd卡
- ❑ 视频预览正确
- ❑ 视频存储后。预览正确

❑ 拓展-在相册中能扫描到该视频

第五章 课后探索





课后探索

- ❑ 探索-相册中选择照片
- ❑ 探索-实时变焦
- ❑ 探索-开启闪光灯
- ❑ 探索-延时拍摄
- ❑ 探索-录制暂停和恢复，分段录制



参考文献

- ❑ <https://developer.android.com/training/camera/videobasics>
- ❑ <https://developer.android.com/guide/topics/media/camera>
- ❑ https://blog.csdn.net/feiduclear_up/article/details/51968975



作业上交

- 使用 github 托管你的项目
- 发邮件
 - 发给: zhangmingqing@bytedance.com
 - 标题: 北理Android课设-多媒体进阶
 - 内容: 你的姓名、学号和项目地址

THANKS.

