多媒体进阶-拍照与录像

字节跳动Android工程师　陶家伟 （Jerry.Tao）

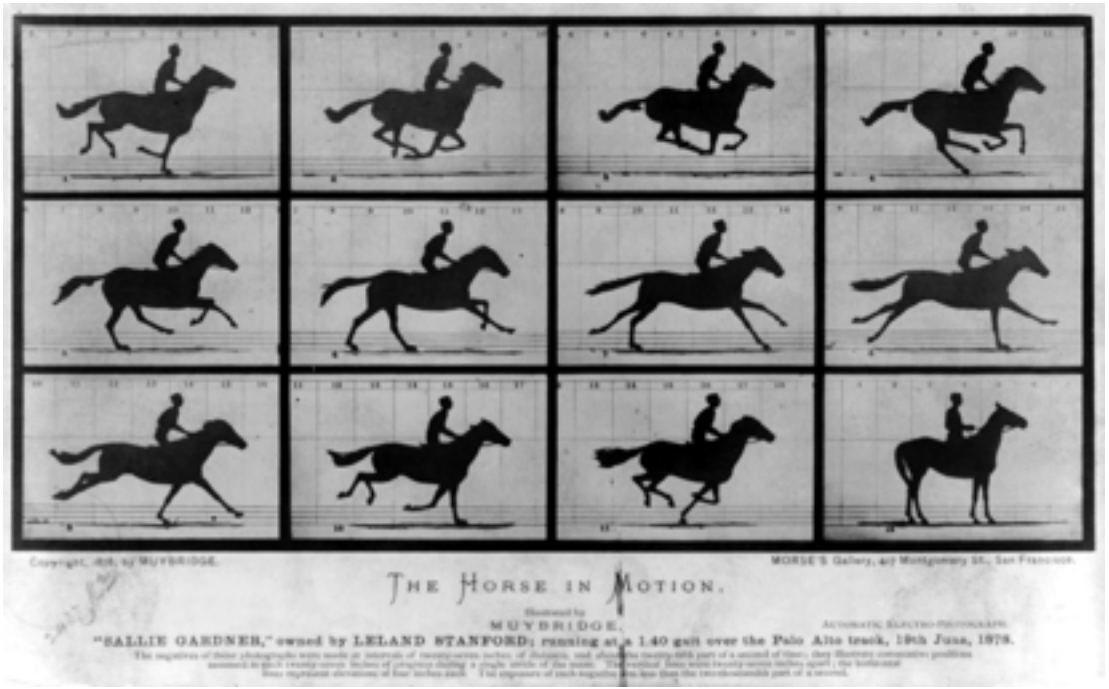
ByteDance字节跳动

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

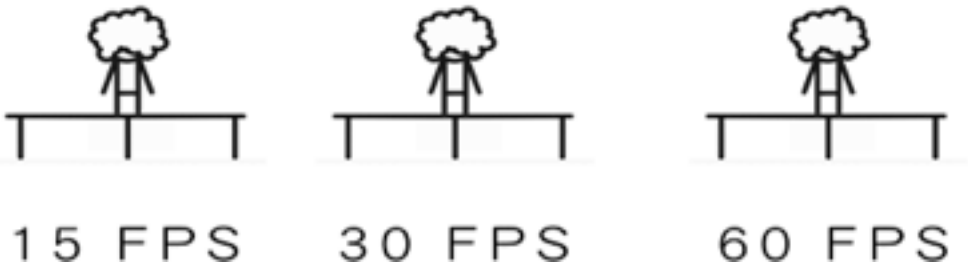


电影的先驱：迈布里奇的马，1878

聊聊几个概念

- 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



聊聊几个概念

- 分辨率

1080P 指的是1920*1080分辨率，也称为“全高清”，表示视频的水平方向有1920个像素，垂直方向有1080个像素。

聊聊几个概念

❑ 码率

视频码率就是数据传输时单位时间传送的数据位数，一般我们用的单位是kbps即千位每秒。通俗一点的理解就是取样率，单位时间内取样率越大，精度就越高，处理出来的文件就越接近原始文件。

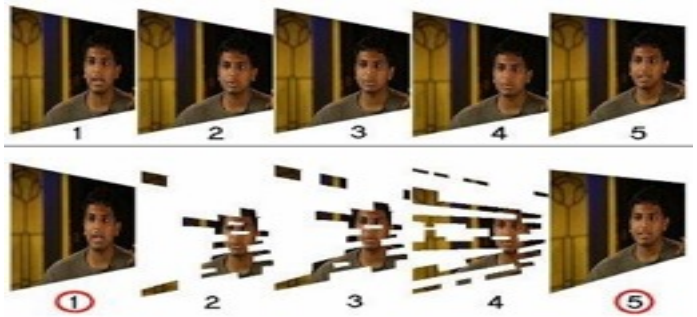
视频大小 = duration 时长(s) x kbps 千位每秒/ 8 = xx MB

聊聊几个概念

- 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 （I(关键帧)、B(双向预测帧) 、P(前向预测帧)帧)



聊聊几个概念

- 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。

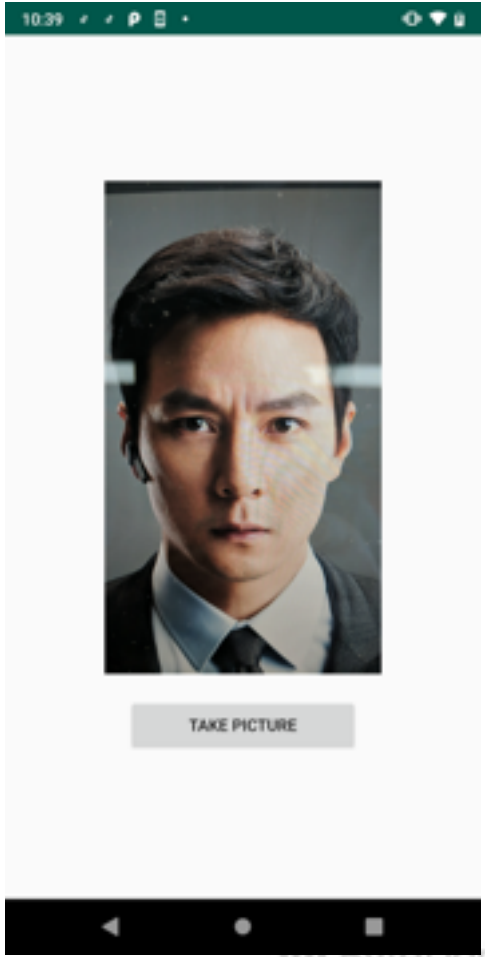
  MP3 MP4 (MPEG 组织-Moving Picture Experts Group,动态图像专家组)

  AVI WAV (Microsoft)

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

- 申请权限

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

- 如何调起相机

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

- 还有那些系统服务
  - 从相册选择
  - 打电话/发短信
  - 查询、新建联系人
  - 打开录音机 ……

步骤二 接收数据

- 接收数据，拿到返回的bitmap，并显示在屏幕上

```java
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        Bundle extras = data.getExtras();
        Bitmap imageBitmap = (Bitmap) extras.get("data");
        mImageView.setImageBitmap(imageBitmap);
    }
}
```

- 为啥图像这么小?

步骤三 自定义存储路径（一）

- 申请存储权限

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

- 创建文件

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

步骤三 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

```
<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="com.bytedance.camera.demo"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/file_paths" />
</provider>
```

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

步骤三 自定义存储路径（三）

- 设置存储地址

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

步骤四 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

```java
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK) {
        int targetW = imageView.getWidth();
        int targetH = imageView.getHeight();
        // Get the dimensions of the bitmap
        BitmapFactory.Options bmOptions = new BitmapFactory.Options();
        bmOptions.inJustDecodeBounds = true;
        BitmapFactory.decodeFile(imgFile.getAbsolutePath(), bmOptions);
        int photoW = bmOptions.outWidth;
        int photoH = bmOptions.outHeight;
        // Determine how much to scale down the image
        int scaleFactor = Math.min(photoW / targetW, photoH / targetH);
        // Decode the image file into a Bitmap sized to fill the View
        bmOptions.inJustDecodeBounds = false;
        bmOptions.inSampleSize = scaleFactor;
        bmOptions.inPurgeable = true;

        Bitmap bmp=BitmapFactory.decodeFile(imgFile.getAbsolutePath(), bmOptions);
        imageView.setImageBitmap(bitmap);
    }
}
```

步骤五 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

```
public static Bitmap rotateImage(Bitmap bitmap, String path) throws Exception {
    ExifInterface srcExif = new ExifInterface(path);
    Matrix matrix = new Matrix();
    int angle = 0;
    int orientation = srcExif.getAttributeInt(ExifInterface.TAG_ORIENTATION, ExifInterface.ORIENTATION_NORMAL);
    switch (orientation) {
        case ExifInterface.ORIENTATION_ROTATE_90:
            angle = NUM_90;
            break;
        case ExifInterface.ORIENTATION_ROTATE_180:
            angle = NUM_180;
            break;
        case ExifInterface.ORIENTATION_ROTATE_270:
            angle = NUM_270;
            break;
        default:
            break;
    }
    matrix.postRotate(angle);
    return Bitmap.createBitmap(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(), matrix, true);
}
```

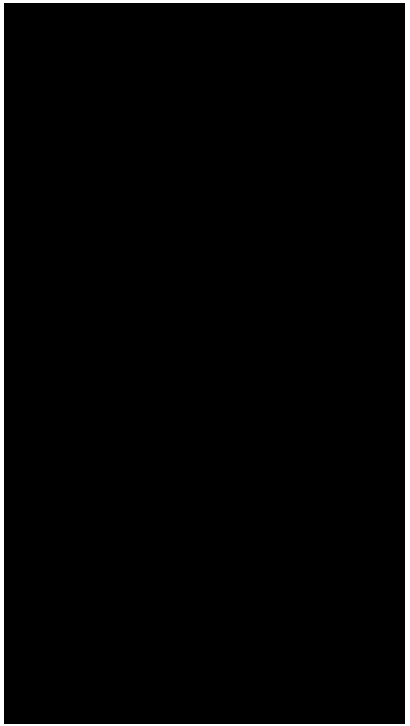
随堂练习 （15min）

- 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

- 吊起手机相机录像
- 录像显示到屏幕上
- 显示视频--VideoView
- 吊起相机--Button



步骤一 调起系统相机

- 申请权限

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

- 调起相机的录像页面

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

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

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    if (requestCode == REQUEST_VIDEO_CAPTURE && resultCode == RESULT_OK) {
        Uri videoUri = intent.getData();
        videoView.setVideoURI(videoUri);
        videoView.start();
    }
}
```

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 kbps

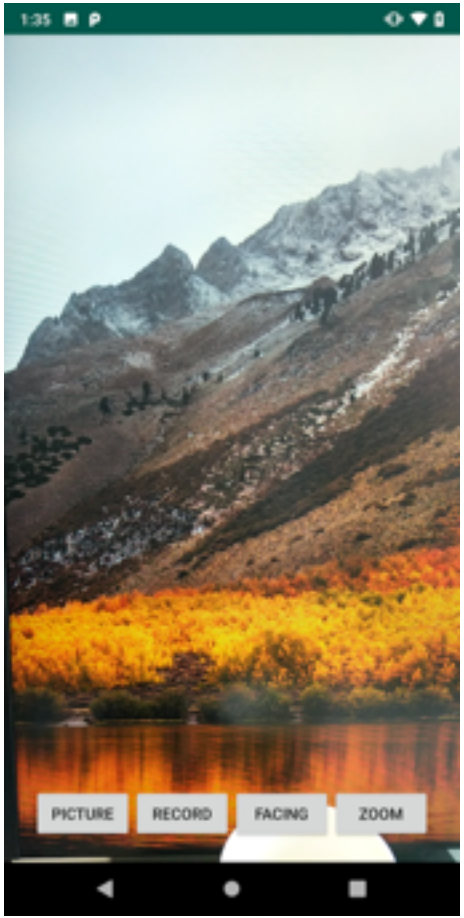
随堂练习 （15min）

- 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

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

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

步骤二 摄像头数据实时显示

- 用什么控件?

  SurfaceView

- 几个关键类

  Camera

  SurfaceView

  SurfaceHolder

  SurfaceHolder.Callback

```
Camera mCamera = getCamera();
SurfaceView mSurfaceView = findViewById(R.id.img);
SurfaceHolder surfaceHolder = mSurfaceView.getHolder();
surfaceHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
surfaceHolder.addCallback(new SurfaceHolder.Callback() {
    @Override
    public void surfaceCreated(SurfaceHolder holder) {
        mCamera.setPreviewDisplay(holder);
        mCamera.startPreview();
    }
    @Override
    public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {}
    @Override
    public void surfaceDestroyed(SurfaceHolder holder) {
        mCamera.stopPreview();
        mCamera.release();
        mCamera = null;
    }
});
```

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

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

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-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：jerry.tao@bytedance.com
  - 标题：浙大Android课设-多媒体进阶
  - 内容：你的姓名、学号和项目地址

THANKS.