

郭鹏

男

1984 年 8 月

北京理工大学 • 计算机科学与技术 • 硕士研究生

13488862189

tony_kwok@126.com

工作经历

2018 年 5 月 ~ 今 北京小米移动软件有限公司

- 项目接口

作为 Mi 8、Mi 9、Redmi Note 8、Mi 10、Mi 11 等项目系统相机应用的负责人，负责系统相机基本功能适配、功能开发、缺陷修复、跨部门协作以及项目进度管理等工作。

- 编译解耦

编写 Shell 脚本实现系统相机应用独立编译（原系统相机应用需要基于系统全代码环境，通过 Make 进行编译）。为了实现增量编译，提高编译速度，后又用 Python 重写了该脚本，最终实现编译耗时从原来的 10 分钟缩减至 20 秒左右：

<https://github.com/tonykwok/mcb>（Shell 版，暂未开源）

<https://github.com/tonykwok/pymcb>（Python 版，暂未开源）

- 资源裁剪

编写 ARPT 工具，在系统进行 RRO 编译时，删除与当前机型无关的资源（比如：字符串，图片等），防止随版本外发导致在研功能泄密：

<https://github.com/tonykwok/arpt>

- 多机联录

借助 WiFi Direct（也叫 WiFi P2P）功能，实现两台手机互联，其中一台作为摄像机，负责将摄制内容实时传输给另外一台与之互联的手机（监视器）上，监视器可以远程操控摄像机，比如切换镜头、画幅、分辨率、调节 EV / AWB 等。期间了解并掌握了 RTSP 协议，并对该协议进行扩展，使其支持基于 json-rpc 规范的远程控制指令；采用ijkplayer进行流媒体播放，优化播放延迟在 100ms 以内；监视器端最多支持两路数据流同时显示，采用 OpenGL ES 实现 PIP 与 PBP 布局绘制。

- 动态照片

实现类似 Apple iPhone 系统相机应用上的 Live Photo 功能。使用 OpenGL ES 将相机预

览数据渲染到 MediaCodec 的 Input Surface 上；采用环队列 CyclicBuffer 缓存 MediaCodec 输出的编码数据（缓存时长大概 4 秒）。当用户按下拍照键时，先从 CyclicBuffer 中提取从该时刻往前 2 秒的历史数据，并向后持续等待 2 秒的数据，然后将这段时间间隔的数据传给 MediaMuxer，从而得到 mp4 格式的视频文件，最后再将该视频文件追加到拍照得到的 JPEG 文件中，并追加相应的 XMP 元数据信息（兼容 Google Photos 应用）。

● HDR 渲染

编写 Fragment Shader 实现文献 ITU-R BT.2446 中提出的 Method C 方法，实现对 HLG 编码的相机预览数据流的 SDR 渲染：

1. 先将 BT.2020 RGB 值转化成 BT.2020 XYZ 值
2. 依据 HLG 规范中的 OETF-1 将 BT.2020 XYZ 值转换成 Scene relative 亮度值
3. 根据 ITU-R BT.2446 Method C 提出的曲线实现 HDR 亮度值到 SDR 亮度值的映射
4. 将位于 BT.2020 颜色空间的 SDR 亮度值裁剪到 Display P3 颜色空间

期间了解并掌握了 ColorSpace、PQ / HLG 曲线、Gammut Clipping、Tone Mapping、OETF、EOTF、OOTF 等相关知识，发现了 [Android/ndk-sample](https://github.com/android/ndk-sample) 项目中的一个 BUG 并提交了 PR：

<https://github.com/android/ndk-samples/pull/839>

2011 年 6 月～2018 年 5 月 索尼移动通讯产品（中国）有限公司

- 集成 Arcsoft® 公司的美颜算法，实时预览美妆效果
- 使用 EXIF 规范中的 APP2 数据段保存语音数据，实现可以拍摄包含语音的图片
- 根据人脸检测的结果，语音播报手机角度是否合适，帮助用户使用后置摄像头进行自拍
- 设计并实现 CameraExtensionService 模块，其主要用来实现索尼相机所特有的功能，比如快速连拍，目标跟踪对焦以及自动场景识别等。CameraExtensionService 被设计成独立于系统原生 CameraService，运行在自己的进程空间的系统服务，从而降低了对 AOSP 代码的耦合
- 负责对 Treble 架构进行调研并实现 CameraExtensionService 对底层 HAL 调用的 HIDL 化

2009 年 7 月～2011 年 5 月 金蝶软件（中国）有限公司

从事基于 JAVA EE 架构和 JAVA Swing 的 EAS 企业管理应用套件的开发工作。该软件所使用的应用服务器为金蝶自己研发的 Apusic，该应用服务器目前已通过 SUN 公司的 JAVA 5.0 EE 认证。软件开发环境为 Kingdee Business Operating System，BOS 是金蝶多年积累的用于快速开发企业级应用程序的开发环境，通过可视化的配置和基于元数据的描述，发布策略实现绝大部分业务对象、业务逻辑（如：增、删、改、查）代码的自动生成。在此期间，主要负责生产计划模块的开发任务，主要包括：供料计算、生产线日能力计划维护、需求计划维护、生产订单分割改制、物料替代以及相关业务单据的套打、报表设计等工作。

业余学习

- **JAVA Swing Alloy Look and Feel 研究与改进**

Alloy 是 INCORS® 公司开发的一款面向 JAVA Swing 的 Pluggable Look and Feel 产品，无论是在 Windows 平台还是 macOS 平台都有很好的用户体验，比如默认按钮的“脉动（pulse）”效果以及玻璃风格的进度条等。在对这款软件进行研究与改进的过程中对 JAVA Swing 体系中的 MVC 结构设计有了更好的认识，同时也对如何利用 JAVA 2D 绘制漂亮的用户界面，比如：渐变、动画、转场等，有了一定的实践经验。

- **基于 JNI 的 JAVA Bytecode 文件加密**

目前，对于 JAVA 软件的加密，通常采用的都是使用混淆软件（Obfuscator）对编译生成的 Class 文件进行混淆，比如开源的 ProGuard。混淆降低了程序的可读性，提高了循环结构的复杂度，比如：混淆后的代码较多地使用了 break 语句等，但是使用反编译工具（比如：JAD）还是能够看出个所以然来。使用 JNI 技术结合自定义 ClassLoader 可以避免这个问题，因为 Class 文件本身已经被加密了所以反编译软件无法识别。当需要加载该 Class 到 JVM 时，自定义的 ClassLoader 会解密并将对应的 Byte 数组转化成相应的 Class 对象。

- **基于纹理特征的图像检索系统**

该项目的客户是一家从事布料销售的外贸公司，为了能及时快速地查询到目前数据库中是否有某一类花纹的布料，系统需要根据输入的样本图片，对数据库进行查询，按相似度从大到小返回匹配结果。显然，颜色不是客户关心的重点，图像纹理特征才是图像是否匹配的判定标准，同时考虑到特征提取效率等问题，决定采用相对简单 Haar 小波特征，最终采用 JAVA 语言并参考网上的一篇文章（Fast Multi-resolution Image Querying）实现了该需求。

开源项目

- **regex4j**

regex4j 是一个移植自微软的 .Net Core，运行在 JAVA 平台上的正则表达式引擎。该引擎能够避免 JDK 内置 java.util.regex.Pattern 类在处理复杂模式匹配串时所导致的 StackOverflowError 异常（JDK-6337993、JDK-6882582、JDK-8078476）。

- **primitive**

primitive 是一个移植自 Michael Fogleman 的 Primitive for macOS，运行在 JAVA 平台上的应用程序。该程序可以使用最基本的几何图元（比如椭圆形、圆形、三角形、矩形等）对输入图片进行重绘，并最终实现对原始图片的矢量化输出。

- 更多请访问 <https://github.com/tonykwok>

last updated: 2022.06.01