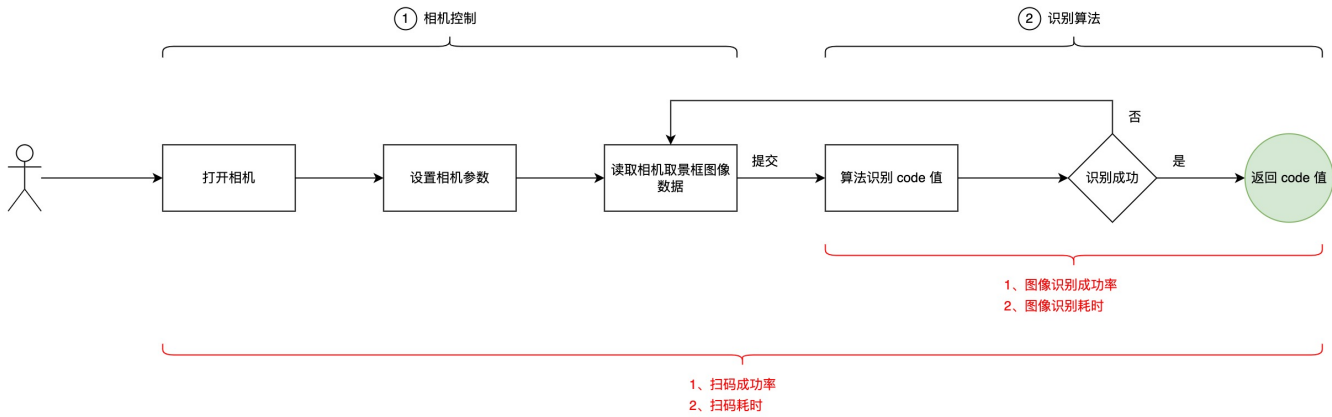


[SCA] 相机扫码方案对比

- 1、相机扫码原理介绍
- 2、识别算法部分
- 3、优化方案调研与对比
- 4、SSC 优化现状

1、相机扫码原理介绍



两个重点阶段：

- **相机控制**：目标是输出一张高质量的图像
- **识别算法**：从图像中通过识别出条形码、二维码的 code 值

扫码场景「业务视角」核心指标：

- **扫码成功率**
 - 单次扫码场景：成功识别到 code 的次数 / 进入扫码页面的次数
 - 连续扫码场景：无法统计
- **扫码耗时**
 - 单次扫码场景：成功识别到 code 的时间 - 进入扫码页的时间
 - **脏数据干扰因素**：打开扫码页，一直不扫码，等一段时间再扫码，占扫码耗时异常的 82% ([SCA] 扫码耗时异常场景个案分类)
 - 连续扫码场景：成功识别到 code 的时间 - 上一次业务处理完结果的时间（首次进入使用进入页面时间）
 - **脏数据干扰因素**：作业人员找包裹的时间。作业员一般是一直开着相机扫码，一个扫码成功后，会再去拿下一个包裹，接着扫，中间这个「找包裹 → 拿起包裹 → 对准二维码」的时间可长可短，从技术手段上暂时没办法去掉

2、识别算法部分

开源条形码、二维码图像识别算法主要有

- **ZBar**：最早的开源二维码识别算法组件，在 ZXing 出来后 2012 年停止迭代
- **ZXing**：谷歌早期的二维码识别算法组件，目前停止迭代，只修改 bug
- **ML Kit**：谷歌最新的二维码识别算法组件，处于持续迭代中

还有一些收费的

- **Scan bot SDK**
- **Barcode Scanner**

除了以上的这些，暂无其他开源的识别算法，国内大厂都是走自研算法或者通过开源算法二次优化。这里我们重点对比 ZXing 和 ML Kit

以下数据是 2023.02.16 全天的数据统计

| 识别算法 | 结果 | 次数 | 均值 |
|--------|----|--------|--------|
| ML Kit | 成功 | 6376 | 114ms |
| | 失败 | 136717 | 52.8ms |

| | | | |
|-------|----|--------|--------|
| ZXing | 成功 | 6255 | 79.5ms |
| | 失败 | 124788 | 144ms |

结论：

- ZXing 在扫描不到结果的时候识别耗时更长，而 ML Kit 在识别不到结果的时候响应更快，可以更快进入下一轮「采集图像 -> 算法识别」，这样在相同的时间内，尝试的次数更多，我们在作业的时候在移动手机对准条形码的过程中检测会更快。

3、优化方案调研与对比

| 内 / 外部 | 公司 / 团队 | 采用技术 | 优化工作 | 技术资料 / 数据 | 其他说明 |
|--------|--------------|--|---|---|--------------------------------------|
| 公司内部 | Shopee Infra | 无公共技术组件 | |  | |
| | Shopee App | 主要采用开源技术 CameraX + ML Kit <ul style="list-style-type: none">• 相机控制：CameraX，参数配置、图像采集• 识别算法：ML Kit，条形码、二维码图像识别算法 | | 优化 Jira Task 记录 | 未做什么优化 |
| | Shopee Pay | 主要采用开源技术 BGAQRCode-Android + ZXing <ul style="list-style-type: none">• 相机控制：BGAQRCode-Android，参数配置、图像采集• 识别算法：ZXing，条形码、二维码图像识别算法 | <ul style="list-style-type: none">• 调度流程优化<ul style="list-style-type: none">• 扫码任务调度顺序• 相机控制优化<ul style="list-style-type: none">• 对焦优化• 调整相机分辨率、裁剪区域• 识别算法优化<ul style="list-style-type: none">• 减少识别码类型• 升级ZXing 算法 | scan优化技术文档.pdf | |
| 公司外部 | 阿里 - 支付宝 | 自研二维码图像识别算法，提高识别成功率 <ul style="list-style-type: none">• 相机控制：不详• 识别算法：自研，主要针对二维码 | <ul style="list-style-type: none">• 识别算法优化：自研，重点是优化<ul style="list-style-type: none">• 二维码识别 | 支付宝二维码扫描优化 <ul style="list-style-type: none">• 扫码成功率：未披露• 扫码耗时：未披露 | 算法部分有自研团队，支付宝是C端场景，重点在于优化二维码而非条形码 |
| | 腾讯 - 微信扫码 | 主要是在 ZXing 的基础上进行二次改造，优化二维码图像识别算法 <ul style="list-style-type: none">• 相机控制：不详• 识别算法：自研（ZXing 二次封装） | <ul style="list-style-type: none">• 识别算法优化：基于开源框架 ZXing 二次改造，重点是优化<ul style="list-style-type: none">• 二维码识别 | QBarAI 引擎提高识别率 <ul style="list-style-type: none">• 扫码成功率：未披露• 扫码耗时：未披露 | 算法部分有自研团队，C端场景，重点在于二维码，而非条形码 |
| | 美团 - App 平台 | 主要是在 ZXing 的基础上进行二次改造，优化条形码、二维码图像识别算法 <ul style="list-style-type: none">• 相机控制：BGAQRCode-Android• 识别算法：自研（ZXing 二次封装） | <ul style="list-style-type: none">• 相机控制优化<ul style="list-style-type: none">• 自动放大• 夜间曝光补偿• 对焦优化• 调整相机分辨率• 识别算法优化：通过视觉智能部门 AI SDK-Edu 框架提供，重点优化<ul style="list-style-type: none">• 条形码识别• 二维码识别 | [SCA] 扫码方案调研 - MT <ul style="list-style-type: none">• 扫码成功率：92.68%• 扫码耗时<ul style="list-style-type: none">• P50：2.9s | 算法部分有自研团队，C & B 端场景都有，所以二维码、条形码都有做优化 |

| | | | | |
|-----|---|---|--|-------------------------------|
| 爱奇艺 | 主要是在 ZXing 基础上做二次改造，优化二维码图像识别算法 <ul style="list-style-type: none">相机控制：不详识别算法：自研（ZXing 二次封装） | <ul style="list-style-type: none">调度流程优化<ul style="list-style-type: none">扫码任务调度顺序相机控制优化<ul style="list-style-type: none">调整相机分辨率对焦优化：连续对焦、点击区域对焦自动放大识别算法优化：基于开源框架 ZXing 二次改造，重点优化<ul style="list-style-type: none">二维码减少识别码类型 | 二维码扫描优化及爱奇艺App的实践 <ul style="list-style-type: none">扫码成功率：未披露扫码耗时：未披露 | 算法部分有自研团队，C 端场景，重点在于二维码，而非条形码 |
|-----|---|---|--|-------------------------------|

4、SSC 优化现状

| 时间 | 采用技术 | 优化工作 | 上线市场 / 模块 | 技术资料 / 数据 | 已实现 / 待实现 |
|---------|--|--|---|--|--|
| 2022 Q3 | 主要采用 BGAQRCode-Android + ZXing <ul style="list-style-type: none">相机控制：BGAQRCode-Android识别算法：ZXing | <ul style="list-style-type: none">调度流程优化<ul style="list-style-type: none">优化扫码库初始化固有延迟扫码任务调度顺序相机控制优化<ul style="list-style-type: none">调整相机分辨率、裁剪区域 | <ul style="list-style-type: none">市场：Sea场景：<ul style="list-style-type: none">PickupDelivery | [SCA] Camera scan 优化总结 2022Q3 <ul style="list-style-type: none">Pickup 模块<ul style="list-style-type: none">ID 市场扫码耗时：3965ms → 3156ms，减少 809msDelivery 模块<ul style="list-style-type: none">全扫码耗时：3299ms → 2008ms，减少 1291ms 2022Q4 <ul style="list-style-type: none">Delivery 模块<ul style="list-style-type: none">全扫码耗时：2008ms → 1634ms，减少 374ms | <ul style="list-style-type: none">调度流程优化<ul style="list-style-type: none">扫码任务调度顺序相机控制优化<ul style="list-style-type: none">调整分辨率、裁剪区域对焦优化：连续对焦、点击区域对焦暗光补偿（目前有闪光灯，可能作用不大）自动放大识别算法优化<ul style="list-style-type: none">升级识别算法到 ML Kit调整码的识别顺序，优先条形码减少码的格式 |
| 2022 Q4 | 主要通过 CameraX + ML Kit <ul style="list-style-type: none">相机控制：CameraX识别算法：ML Kit | <ul style="list-style-type: none">相机控制优化（BGAQRCode-Android 改为 CameraX）<ul style="list-style-type: none">对焦优化调整相机分辨率、裁剪区域识别算法优化<ul style="list-style-type: none">从 ZXing 升级到 ML Kit | <ul style="list-style-type: none">市场：Sea场景：Delivery | | |

BR 目前的 Pickup 场景扫码是没有任何优化的，如果 Sea 的优化都上线到 BR，预期可实现的提升为

- Q3 的数据提升：减少 500ms 左右（参考 ID 市场 800ms 的数据，因为数据各市场会有一些差异，保守点估计 500ms）
- Q4 的数据提升：预计减少 300ms 左右（参考 Delivery 的数据提升，不过 Delivery 与 Pickup 场景统计口径不同，只能作为参考）