

需求分析:

大致:

ipc 没有存储设备的情况下, 能够保存一些信息 (例如: 有人走过)

分析:

- 1: 信息的类型 对于ipc主要是图片, (文字 不是很有意义, 视频 数据量太大对内存需要太多), 客户端和服务端 可以保存图片和视频,
- 2: 信息的内容 移动侦测, 人脸识别, 手动, 计划 ... 产生的图片和视频
- 3: 保存的方式 ipc (内存), 客户端 (消息推送), 服务器 (上传)
- 4: 保存策略

ipc 保存图片, 内存池实现

客户端 手动操作产生的数据保存在客户端 (手动抓拍, 录像等), 是否有必要同时保存在ipc, 我认为抓拍可以提供用户选择, 录像有必要

服务器 设备有连上服务器的话可以把数据上传给服务器

设计方向:

ipc 端 保存图片, 使用内存池实现

优势: ipc 是数据源, 不会因为没连上服务器, 或者无客户端连接, 无处保存数据

缺点: 只能保存图片 (文字 不是很有意义, 视频 数据量太大对内存需要太多)

图片产生:

移动侦测, 人脸识别, 手动, (计划没必要, 不会有人计划保存图片) 产生的图片

适用场景:

- 1: 对数据保存不是很在意, 但是又希望能够有些。
- 2: 正常情况下数据量不会太大 例如: 家庭用户, 主要是移动侦测, 或者人脸识别

问题:

- 1: 在不正常的情景下, 数据量会很大, 例如: 人流变多,
- 2: 移动侦测 和 人脸识别不准确, 误报
- 3: 保存图片, 图片比较大,

做法:

- 1: 在不正常情况下, 会造成频繁的内存分配和释放, 所以需要使用内存池来实现
- 2: 内存池需要统计, 保存的图片数量 和 释放的图片数量 (来提醒用户购买存储设备)

