

车载电子系统中的视频抓拍技术^①



张文兴, 孙庆鹏

(沈阳美行科技有限公司, 沈阳 110000)
通讯作者: 张文兴, E-mail: zwxdlut@163.com

摘要: 目前对视频技术的研究与应用主要集中于利用图像处理与模式识别手段对视频图像进行处理, 视频抓拍都是以抓拍时间点为起始时间点, 所拍摄视频包含的信息缺乏完整性, 进而会影响进一步的分析与处理. 针对此问题, 通过定义时间窗口并利用缓存技术, 提出了一种基于 Android 系统的视频抓拍方案的设计与实现, 其关键在于能够提供抓拍时间点之前到之后一定时间段内的视频. 测试结果表明, 该方案具有较小的时间误差, 并占用较少的系统资源, 已被应用于量产并投入市场的车载电子系统中, 为交通事故责任认定提供了重要依据.

关键词: 视频抓拍; 时间窗口; 缓存技术; Android; 车载电子

引用格式: 张文兴, 孙庆鹏. 车载电子系统中的视频抓拍技术. 计算机系统应用, 2019, 28(11): 233-237. <http://www.c-s-a.org.cn/1003-3254/7121.html>

Video Capture Technology in Vehicle Electronic System

ZHANG Wen-Xing, SUN Qing-Peng
(MXNavi, Shenyang 110000, China)

Abstract: At present, the research and application of video technology mainly focus on the processing of video images by image processing and pattern recognition. The video capture is started from the capture time point, thus the information contained in the captured video lacks integrity, that affects further analysis and processing. Aiming at this problem, design and implementation of a video capture scheme are proposed based on Android system by defining the time window and using the cache technology. The key is to provide video for a certain period of time from before to after the capture time point. The test results show that the capture scheme has a small time error and occupies less system resources, and has been applied to vehicle electronic systems which are mass-produced and put into the market, providing an important basis for traffic accident liability identification.

Key words: video capture; time window; cache technology; Android; vehicle electronic system

随着车辆的普及, 各类汽车电子产品层出不穷, 行车录影作为车载电子领域众多产品中的一项重要功能, 在交通事故责任认定中发挥着不可替代的作用. 众所周知, 车机或是行车记录仪中的录影视频为 3-5 分钟的分段视频, 数量众多. 在事故发生时, 人们往往需要的是事故发生时间点前后十几秒的视频, 并且应对该视频进行安全备份, 便于检索. 因此, 对视频的抓拍以及处理显得尤为重要. 黄凯奇^[1]等人对智能视频监控技

术进行分类, 对目标检测与跟踪、特征提取与分析等经典算法给出了比较全面的综述. 胡博^[2]等人设计了一种对监控视频中的行人进行检测和抓拍的系统. 王崇海^[3]利用多维视频侦查系统检测和抓拍车辆以及人脸, 提取车型、车牌、人脸特征等信息并保存. 吴细老^[4]提出了一种智能交通视频抓拍框架, 对视频图像进行阴影处理与抖动消除, 对违章行为进行判定与抓拍. 高奎贺^[5]提出了采用机动车视频测速进行辅助超速抓拍的

① 收稿时间: 2019-03-29; 修改时间: 2019-04-18; 采用时间: 2019-04-26; csa 在线出版时间: 2019-11-06

技术. 吕正荣^[6]提出了一种对变焦的运动目标进行预测抓拍的方法, 并利用此方法构建了一种交通违法行为车辆智能跟踪抓拍系统. 目前对视频技术的研究与应用要么立足于利用图像处理与模式识别手段对视频图像进行目标检测、信息提取、恢复、重建等^[7-10]; 要么立足于对视频监控或抓拍系统的设计与开发^[11-14]; 而对于视频抓拍本身并没有过多或深入的研究, 涉及到的视频抓拍都是以抓拍时间点为起始时间点, 所拍摄视频包含的信息缺乏完整性, 进而会影响进一步的分析与处理. 由此, 引出一个问题: 当抓拍条件触发时, 能否以抓拍时间点为基准, 保存其之前到之后一定时间段内的视频, 研究目的就是以此展开的.

在众多的智能操作系统中, Android^[15,16]系统以其开放性, 支持硬件多样性, 强大的 SDK 等优势, 迅速成为视频监控、车载电子、移动终端设备中的主流系统. 首先通过定义时间窗口并利用缓存技术, 阐述抓拍的基本原理, 然后简要介绍一下 Android 系统中音视频子系统框架, 接着提出一种基于 Android 系统的视频抓拍方案的设计与实现, 最终给出测试分析以及应用场景.

1 抓拍原理

抓拍所要达到的目标为: 当有事件触发抓拍时, 将当前时间点前 M 秒到后 N 秒的音视频数据写入文件, 其中 M 和 N 为设定好的预期时间. 其基本思路为: 预设一个时间窗口 $M+N$, 假设当前时间点为时间 0 点, 对时间段 $-(M+N)$ 到 0 之间的音视频数据进行缓存, 如图 1 所示. 当触发抓拍时, 创建抓拍文件, 等待 N 秒, 从缓存中取出 $-M$ 到 N 秒时间段内的音视频数据, 对其进行封装并写入文件, 如图 2 所示.

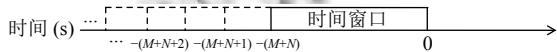


图 1 时间窗口

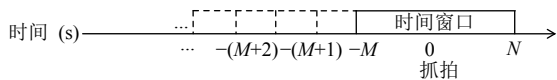


图 2 触发抓拍 N 秒后的时间窗口

2 Android 的音视频子系统

抓拍方案基于 Android 原生系统实现, 所以在阐述抓拍方案之前, 有必要先对 Android 的音视频子系

统框架以及其录制过程做一简单介绍.

Android 的音视频子系统 (录制部分) 如图 3 所示. 摄像头服务 (CameraService) 和音频核心服务 (Audio Flinger) 通过硬件抽象层 (HAL) 从摄像头驱动 (Camera Driver)、音频驱动 (Audio Driver) 中采集音视频数据. 视频录制的核心服务 (MediaPlayerService) 负责音视频数据的编码以及封装. 其中, 录制组件 (StagefrightRecorder) 根据设定的媒体格式类型 (MP4、3GP、...) 创建组合器 (Writer), 视频编码器 (VideoEncoder) 以及音频编码器 (AudioEncoder) 通过共享内存 (Shared Memory) 从 CameraService 和 AudioFlinger 服务中获取音视频数据并进行编码, Writer 将编码后的音视频数据封装成设定的媒体格式, 并将其写入文件. 应用程序 (Applications) 调用框架层 (Framework) 的媒体录制组件 (MediaRecorder) 通过 Java 本地接口 (JNI) 与库层 (Libraries) 的 MediaRecorder 组件进行交互. Libraries 层的 MediaRecorder 通过 Android 独有的绑定 (Binder) 机制与 MediaPlayerService 服务进行进程间通信 (IPC).

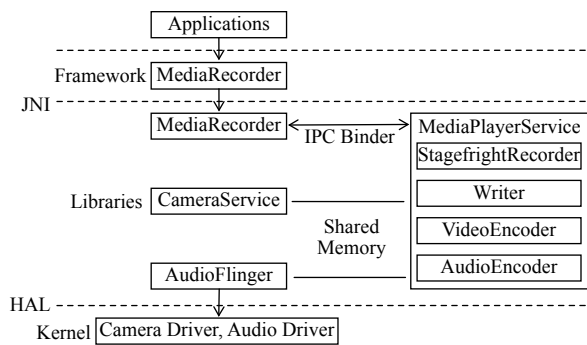


图 3 Android 音视频子系统框架 (录制部分)

3 抓拍方案的设计与实现

编码后的音视频数据的处理以及封装由组合器完成, 其内部包含两种线程, 一种是获取并处理编码后的音视频数据的轨迹线程, 一种是负责将处理后的音视频数据按照预先设定好的媒体格式封装并写入文件的写线程. 轨迹线程一般有两个, 分别对应音频数据和视频数据, 写线程只有一个, 负责对处理后的音视频数据进行封装并写入文件.

以录制 MP4 文件为例, 对音视频数据的处理和封装过程如图 4 虚线上半部分所示. 轨迹线程以帧为单位获取编码后的音视频数据, 每帧数据会先进入一个缓存队列 (FIFO, 队列个数与轨迹线程个数相同), 当帧

的数量达到一定值时, 轨迹线程会将这些数据帧打包成固定的数据结构 (Chunk), 并通知写线程有新的 Chunk 已准备好. 同时, 轨迹线程将数据帧中的信息及系统环境信息提取汇总存储在一个被称为 track 的表中, 其包含的信息有 Chunk 写入文件的偏移地址 stco (sample to chunk offset)、Sample 与 Chunk 的映射关系 stsc (sample to chunk box)、关键帧 stss (sync sample

box), 每一帧的持续时间 stts (time to sample box) 等. 当录制开始时, 写线程创建文件并封装文件头 ftyp (file type box) 字段, 当写线程接收到 Chunk 已准备好的通知后搜索缓存队列, 将对应的 Chunk 写入文件, 并将文件的偏移地址更新到相应 track 表的 stco 项 (track 表中其它数据是由轨迹线程维护). 当录像结束时, track 表会被写入到文件尾 moov (movie box) 字段.

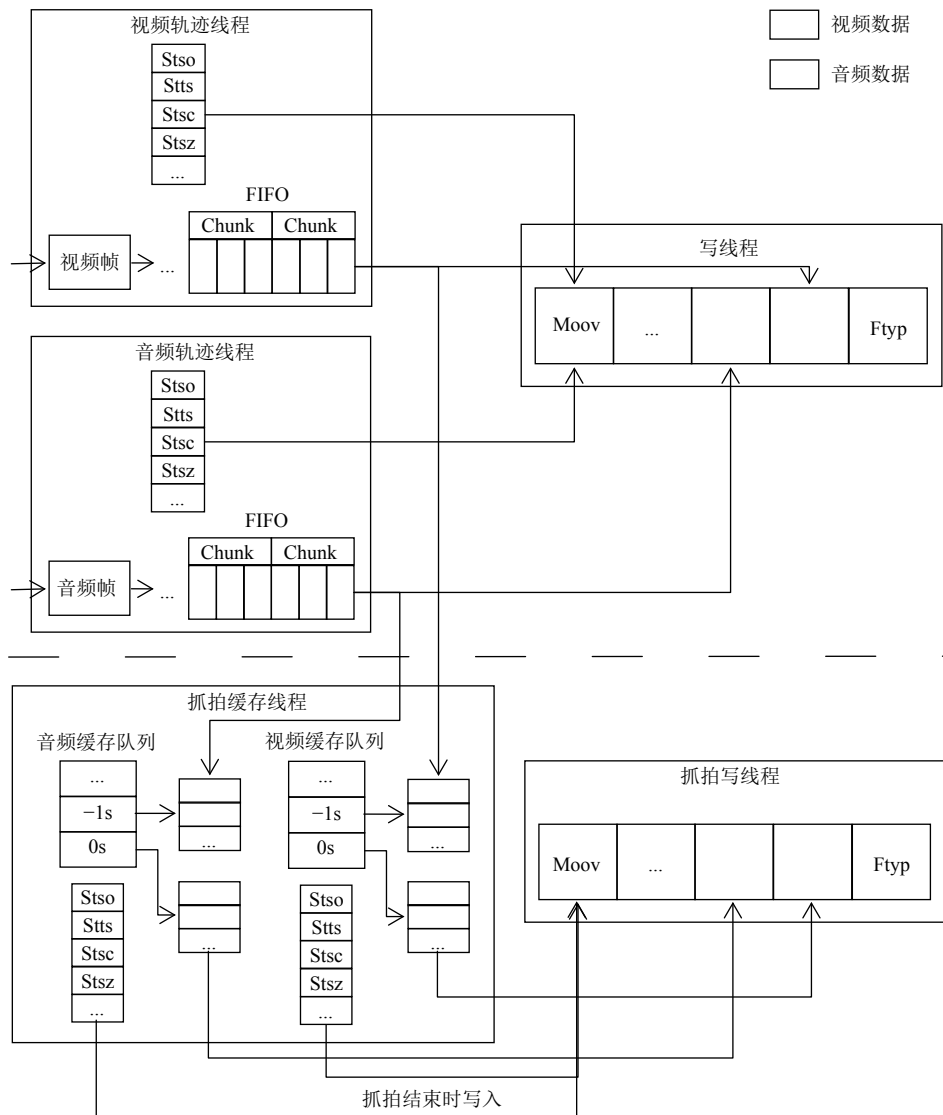


图 4 音视频数据的处理与封装过程

抓拍方案的核心思路为: 为音频和视频分别建立音频缓存队列和视频缓存队列, 并创建一个缓存线程, 按照时间从音频轨迹线程和视频轨迹线程中获取打包后的音视频数据, 同时缓存 $-(N+M)-0$ 秒的音视频数据, 并实时更新缓存数据的 track 表. 当有抓拍请求到来时,

创建并执行抓拍写线程, 创建文件并封装文件头 ftyp 字段, 以抓拍时间点为时间 0 点, 根据预先设定好的 M 和 N (见图 2) 将从 $-M$ 秒开始的缓存数据写入文件, 执行 N 秒后, 抓拍过程结束, 将缓存的音视频数据的 track 表写入文件尾 moov 字段, 过程如图 4 虚线下半

部分所示. 抓拍流程如图 5 所示.

该方案实现于 Android 系统核心服务 MediaPlayer Service 中的 StagefrightRecorder 组件和 Writer 组件中, 并为 MediaRecorder 提供了抓拍接口, 以便对 HMI 支持.

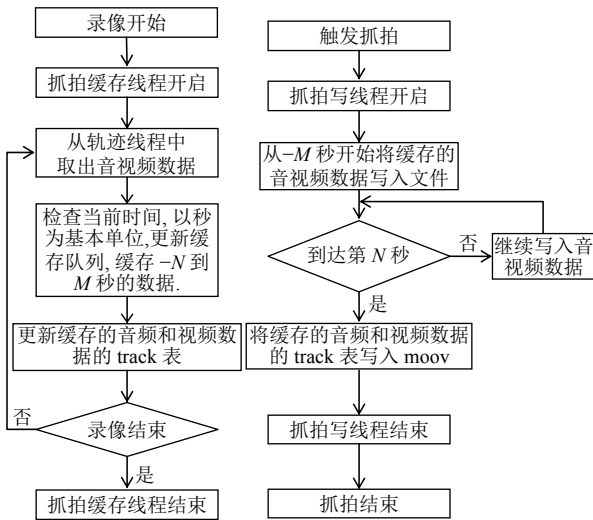


图 5 抓拍流程图

4 测试分析与应用场景

为了测试抓拍的时间精度以及对系统性能的影响, 将其移植进 Android 系统中, 硬件配置为主芯片: MTK6735, 4 core, 1.3 G; CPU: ARM Cortex-A53; 内存: 2 GB; 摄像头分辨率: 1280×800. 该方案实现所在进程为 Android 媒体服务 (mediaserver) 进程, 设置 M 、 N 为不同值 ($M, N \geq 5; M+N < 20$), 多次测试抓拍时长, mediaserver 进程的 CPU 占用率以及内存使用情况, 取平均值, 结果如表 1 所示 (第一行为无抓拍方案的 CPU 占用率与内存使用情况).

实验结果表明, 抓拍时长与预期时长相比误差范围在 500 ms 以内; CPU 占用率与无抓拍相比, 增加范围在 6% 以下, 不同预期时长下 CPU 占用率变化可以忽略不计; 对内存的占用根据预期时长趋于线性变化, 时长每增加 1 s, 内存占用增加约 1 MB.

抓拍的触发条件可以为车辆发生碰撞或是用户通过 HMI 手动触发等, 抓拍方案在车联网系统中的应用场景如图 6 所示.

5 结论与展望

通过定义时间窗口并利用缓存技术, 提出了一种

基于 Android 系统的视频抓拍方案的设计与实现, 并将其应用于车载电子系统中. 从测试结果可以看出, 该方案具有较小的抓拍时间误差, 引起的 CPU 占用率变化几乎可以忽略不计, 并且在高清摄像头 (1280×800) 下, 对内存的影响随抓拍时长呈线性变化, 并且在可接受的范围内. 该方案已被集成进量产并投入市场的 Android 智能轻车机中, 从用户反馈来看, 该方案所实现的功能能够提供抓拍时间点之前一定时间段内的视频, 及时有效, 在事故认定过程中发挥了关键作用. 抓拍方案基于 Android 系统实现, 应用于车载电子系统中, 但并不局限于此, 可以应用于任何有此类需要的智能视频监控系统中.

表 1 预设不同时间下的抓拍测试结果

$M(s)$	$N(s)$	抓拍视频长度 (s)	CPU(%)	内存 (KB)
-	-	-	16	29 884
5	5	10.334	21	39 376
5	6	11.462	21	40 328
5	7	12.264	21	41 272
5	8	13.372	20	42 224
5	9	14.466	20	43 172
5	10	15.246	20	44 120
6	5	11.520	20	40 416
6	6	12.368	20	41 188
6	7	13.574	19	42 172
6	8	14.620	22	43 268
6	9	15.376	22	44 032
6	10	16.558	21	45 072
7	5	12.486	20	41 376
7	6	13.548	20	42 332
7	7	14.384	19	43 376
7	8	15.439	19	44 052
7	9	16.436	21	45 128
7	10	17.448	20	46 224
8	5	13.342	19	42 280
8	6	14.554	22	43 448
8	7	15.444	21	44 040
8	8	16.488	22	45 244
8	9	17.366	22	46 356
8	10	18.388	22	47 280
9	5	14.388	20	43 588
9	6	15.446	19	44 320
9	7	16.328	22	45 160
9	8	17.426	21	46 284
9	9	18.484	20	47 184
9	10	19.428	21	48 280
10	5	15.488	22	44 280
10	6	16.364	22	45 392
10	7	17.386	19	46 192
10	8	18.484	21	47 240
10	9	19.368	21	48 488
10	10	20.444	22	49 440

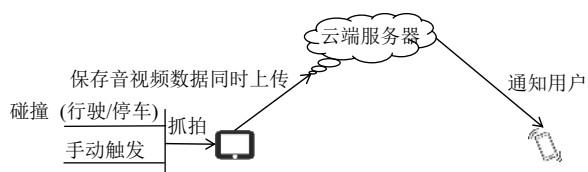


图6 抓拍方案在车联网系统中的应用

参考文献

- 黄凯奇, 陈晓棠, 康运锋, 等. 智能视频监控技术综述. 计算机学报, 2015, 38(6): 1093-1118. [doi: 10.11897/SP.J.1016.2015.01093]
- 胡博, 连捷. 基于监控视频的行人抓拍系统的设计. 电子技术与软件工程, 2018, (9): 60.
- 王崇海. 多维视频侦查系统车辆和人脸检测识别设计与实现技术. 数字通信世界, 2018, (4): 102-103. [doi: 10.3969/J.ISSN.1672-7274.2018.04.073]
- 吴细老. 多场景复杂环境智能交通视频处理关键技术研究及实现[硕士学位论文]. 南京: 东南大学, 2017.
- 高奎贺. 视频测速在机动车测速仪抓拍过程中辅助判断的应用. 轻工标准与质量, 2016, (5): 50-51, 64.
- 吕正荣. 智能交通系统中视频处理关键技术研究[硕士学位论文]. 南京: 东南大学, 2016.
- 楚翔宇. 基于深度学习的交通视频检测及车型分类研究[硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2017.
- 张亚男. 基于车载摄像机的前方车辆测距测速方法研究[硕士学位论文]. 大连: 大连海事大学, 2018.
- 黄丹. 基于监控视频的交通信息提取技术研究[硕士学位论文]. 成都: 电子科技大学, 2017.
- 郭林波. 基于视频监控的复杂场景目标运动分析[硕士学位论文]. 南京: 东南大学, 2017.
- 柯常志, 徐鹏. 基于视频检测的高清摄像机在智能交通监控系统中的应用. 工业控制计算机, 2015, 28(1): 43-44. [doi: 10.3969/j.issn.1001-182X.2015.01.019]
- 普锋. 道路车辆图像智能抓拍与管理系统的实现[硕士学位论文]. 大连: 大连理工大学, 2015.
- 孙笑. 交通违章抓拍系统的设计与实现[硕士学位论文]. 长春: 吉林大学, 2015.
- 董越刚. 车辆图像智能抓拍系统的设计与实现[硕士学位论文]. 成都: 电子科技大学, 2014.
- 林学森. 深入理解 Android 内核设计思想. 2 版. 北京: 人民邮电出版社, 2017.
- 何俊林. Android 音视频开发. 北京: 电子工业出版社, 2018.