# 工作交接文档

需要交接的内容主要包括三大部分，第一部分是主APP的录音模块(线上项目)，第二部分是主播版录音app的录音模块(预研项目)。两个模块处理流程不一样，主APP是实时采集、处理、编码，主播录音APP是先采集后处理。第三部分是趣配音相关的部分功能。

主APP录音模块包含以下几个项目：AudioProcessor(新版安卓录音底层编解码)、beautify(声音美化)、noise-suppression(降噪)、morph-filter(变声滤镜)、Minions(小黄人变声)、echo-filter(回声混响滤镜)、echo-cancellation（回声消除）。其中，AudioProcessor是专门为安卓实现的；echo-cancellation没有运用在iOS录音模块，iOS使用系统自带的回声消除功能；其他音频的处理，Android和iOS共用。

主APP录音模块(安卓)音频数据流向如图1所示。完全理解整个录音模块，关键在于理解音频数据的处理流程。①背景音乐和音效解码的数据经过混音后分为两路，一路送到扬声器进行音频渲染，另一路与采集到的人声混音并进行编码。②回声消除模块输入数据有两路，分别是扬声器渲染的音频数据（单声道）、麦克风采集的音频数据（单声道、包括人声和外放的音乐）。经过回声消除模块，得到一路数据，即纯净的人声。③采集到的人声需要经过进一步的处理才能与解码的背景音乐混音。处理流程依次是：noise-suppression(降噪)、beautify(声音美化)、echo-filter(回声/混响滤镜)、变声。变声包括Minions(小黄人变声)和morph-filter(变声滤镜)，二者选其一。

各项目内均包含完整的测试样例、测试数据、测试脚本和编译打包脚本。下面详细介绍AudioProcessor项目，其他项目介绍参考Excel表格。

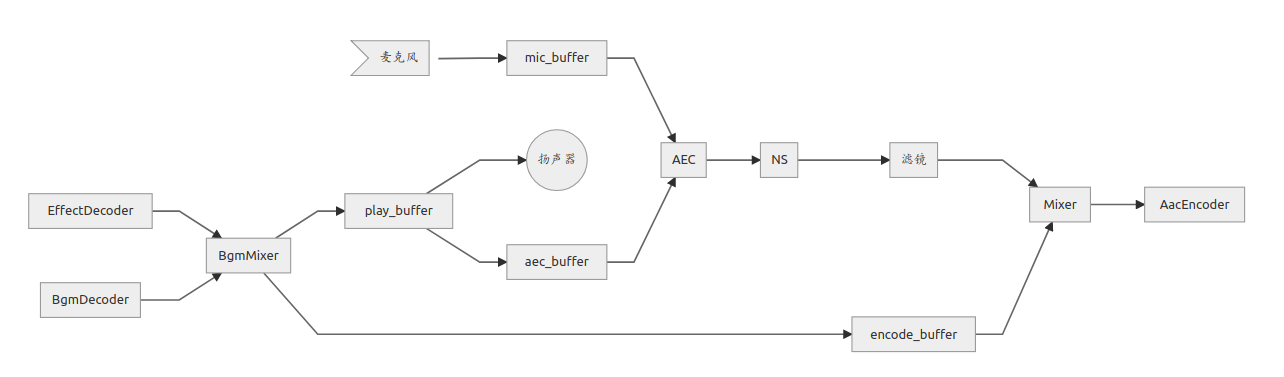


图 1 安卓主app录音模块数据流图

## AudioProcessor

1.1 功能：

①解码背景音乐，支持音量设置，淡入淡出效果。

②解码音效文件。

③混合多路音频。

④实现音频编码封装功能。

⑤实现音频无缝裁剪，支持恢复剪切位置编码器与解码器的状态。

⑥实现完整的日志管理，支持写文件、打印到屏幕或者输入到Android日志机制，支持设置日志级别。

1.2 原理及难点

通过调用ffmpeg的API实现音频的编解码功能。多路混音算法与趣配音人声背景音乐混音算法一致。解码的音频数据与需要编码的音频数据通过AVAudioFifo做缓存。

项目容易理解的是音效解码和多路混音功能。项目的难点在于实现音频的无缝裁剪。如果裁剪后的音频数据不连贯，会出现明显的pop音。AAC编码模块（AacEncoder）记录每一帧的大小是为了精确裁剪。背景音乐解码器(BgmDecoder)记录背景音乐开关、麦克风开关、音量大小等操作也是为了实现精确裁剪。

音频裁剪调用流程可以参考TestSeek.cpp。实现原理：①根据时间剪切编码后的AAC文件。②从AAC编码器获取精确的裁剪时间③将精确的裁剪时间设置到BgmDecoder，恢复裁剪位置的解码器状态。Seek到背景音乐前几帧，解码背景音乐。④用解码的背景音乐数据恢复AAC编码器的状态。AAC编码器刚开始编码时会出现一到两帧的静音帧，需要丢弃。

1.3 背景 音乐裁剪

用户录音过程中可能会出现各种操作，例如：关闭和开启背景音乐，更换背景音乐，开启和关闭麦克风采集。为了实现精确裁剪，恢复裁剪位置的背景音乐状态，需要记录用户的各种操作。

根据用户的操作，可以将录制状态分为：带背景音乐(Bgm)和不带背景音乐（Margin）。将不同的录制状态定义为片段（Segment），BgmInfo用来记录每个片段的信息，这些信息包括：当前片段的类型（Bgm/Margin）、当前片段背景音乐的文件路径、当前片段第一个样本点在背景音乐中的位置、当前片段播放的样本点个数、当前片段的播放时间。

需要新增片段的场景：①初始状态转为录音状态（Bgm/Margin）②Margin状态转为Bgm状态③Bgm状态转为Margin状态。④更换背景音乐。PS：只有背景音乐的情况下，开关背景音乐不新增片段；同理，Margin状态下，开关麦克风开关不新增片段。

裁剪方法：①获取AAC音频文件的精确裁剪位置（单位为秒）②计算裁剪位置位于哪一个片段③计算裁剪位置位于哪一个样本点④恢复裁剪位置的状态，精确到样本点。

## 趣配音相关

2.1功能：①人声背景音乐混音②视频字幕前添加角色头像

2.2 人声背景音乐混音原理：①根据配置的文件路径及参数，分别解码压缩的背景音乐和人声。将解码的音频数据分别放入各自的fifo。②从两个fifo中分别取等长度的音频数据进行混音，这里需要注意对音频溢出做处理。③将混音后的音频数据放入编码的fifo。④从编码的fifo中取音频数据，送给AAC编码器进行编码。⑤将编码后的音频packet封装到MP4文件中。⑥若有视频，则读取视频packet，以拷贝的形式封装到MP4文件中。

2.3 视频字幕前添加角色头像原理：①加工srt字幕文件，在字幕前添加角色信息(ABCD)。②调用ffmpeg命令，将字幕写入视频时，在命令行中配置头像（图片）的文件路径。③在ffmpeg的subtitles滤镜中解码头像（图片）④修改libass源码，根据字幕前添加的角色信息，给每条track加上角色属性。⑤在subtitles滤镜中，根据libass提供的字幕信息，计算头像的绘制位置。⑥根据overlay滤镜，实现图像的叠加功，将角色头像叠加到视频中。

## 主APP线上音频处理

echo-cancellation（回声消除）：线上使用的传统的回声消除算法，没有使用基于AI的回声消除算法。基于AI的回声消除算法移植到客户端CPU占用率高，需要进一步优化。要保证较好的回声消除效果，需要保证一下三点：①音频从扬声器渲染到被麦克风采集的延时要短，越短越好。②保证扬声器平稳的渲染音频。③保证麦克风平稳的采集音频。

noise-suppression(降噪)：参考Excel文档。

beautify(声音美化)：实现了均衡器、单频段压缩器、多频段压缩器、扩展器、限制器、镶边效果器。主播版录音APP的音效处理可以参考或移植该项目。

echo-filter(回声/混响滤镜):参考Excel文档。

morph-filter(变声滤镜)：参考Excel文档。

Minions(小黄人变声)：参考Excel文档。

## 主APP音频处理优化项目（未上线）

由于线上的多个音频处理模块接口较多，而且每添加一个接口，安卓需要重新封装。另外，音效全选的情况下，音频数据需要在JNI拷贝多次，效率低。xmly-audio-effect(主app录音模块音效库)将主APP录音模块的多个音频处理模块统一在一个框架下，集成了降噪、声音美化、回声、混响、变声等多个音频处理模块。所有的音频处理全部在C层实现，避免了数据在JNI多次拷贝。此外，添加新的功能模块也不用改动接口，简化客户端操作。xmly-audio-effect实现了完整的安卓音频采集、处理和渲染demo，支持日志管理，记录用户的所有操作，方便分析错误。如果需要添加新的模块，参考旧模块实现。

aecm(重构的回声消除)：参考Excel文档。

AecTest(回声消除效果测试)：参考Excel文档。

## 主播版录音APP音频处理

audio-effect(主播版录音app音效处理)框架与主APP录音模块音效库相似，降噪和声音美化功能可以根据产品需求主APP录音模块音效库移植过来。该项目内已经从sox中移植了可调参的混响效果器和回声效果器。

hrtf(3D音效)：参考Excel文档。

主播版录音APP音频处理建议：①优化降噪效果。②考虑到背景音乐本来就有3D环绕的情况，建议背景音乐解码成双声道。③根据人声音量自动调整配乐音量（side-chain）可以根据需求选择闪避处理器或压缩器，如下图所示。闪避处理器的效果会比较突出人声，压缩器的效果人声和音乐更融合。④多轨音频合并不能将音乐作为一个声道，人声作为一个声道，需要混合后分布在两个声道。⑤生成最终的音频压缩文件建议使用各自平台的硬件编码器进行编码，同时支持ffmpeg软件编码，兼容部分机型。⑥设计框架记录音效配置，根据配置渲染音频和处理编码音频⑦了解音效处理原理和场景可以参考《混音指南》，实现音效处理器可以从sox移植。

