## 毕业课题论文提纲

## 绪论

**1.1 研究背景与意义**

**背景：**音视频作为目前主流的交流媒介，短视频的火爆导致音视频处理的需求急剧上升。基于服务端的处理通常会给服务端造成过大的压力，其次类似PR等软件对简单音视频处理显得相对复杂。

所以面向 ToB 市场，适配用户各种使用场景，能够在不同浏览器中进行视频播放和剪辑是基本要求。因此需要一套通用的 Web 音视频处理解决方案，从而实现轻量化的Web端音视频剪辑和播放处理来满足大众日常需求。

**意义**：

- 本地即可实现处理，无需上传

- 播放处理兼容性提高，更加通用

- 将音视频处理这类密集型计算移植到浏览器环境，处理速度提升

**1.2 研究现状**

Web音视频处理的研究现状

相关框架技术的研究现状

**1.3 论文的研究目标和内容**

**研究目标**：设计实现Web端的音视频本地处理（播放和剪辑）系统。

**内容：**

1、基于FFmpeg设计实现音视频的解封装-软解码-软编码-封装的流程。

2、基于内容一的基础上，设计对解码后的视频帧数据，即YUV图像数据，进行拼接、截取、编码、转换等操作，进一步实现本地音视频的多种剪辑功能，包括（音视频的截取、音视频格式转换等等）。

3、在针对大部分浏览器内置对音视频的类型解析有限，基于内容一的基础上实现对多种类型(mp4、mov、flv...)的音视频实现本地浏览器兼容性播放。

4、因为整个系统存在两种语言的运行环境，所以播放处理时环境之间的数据的传输的实时性要求更高，还有浏览器渲染yuv图像数据的策略设计。

## 二、相关技术

**2.1 音视频技术**

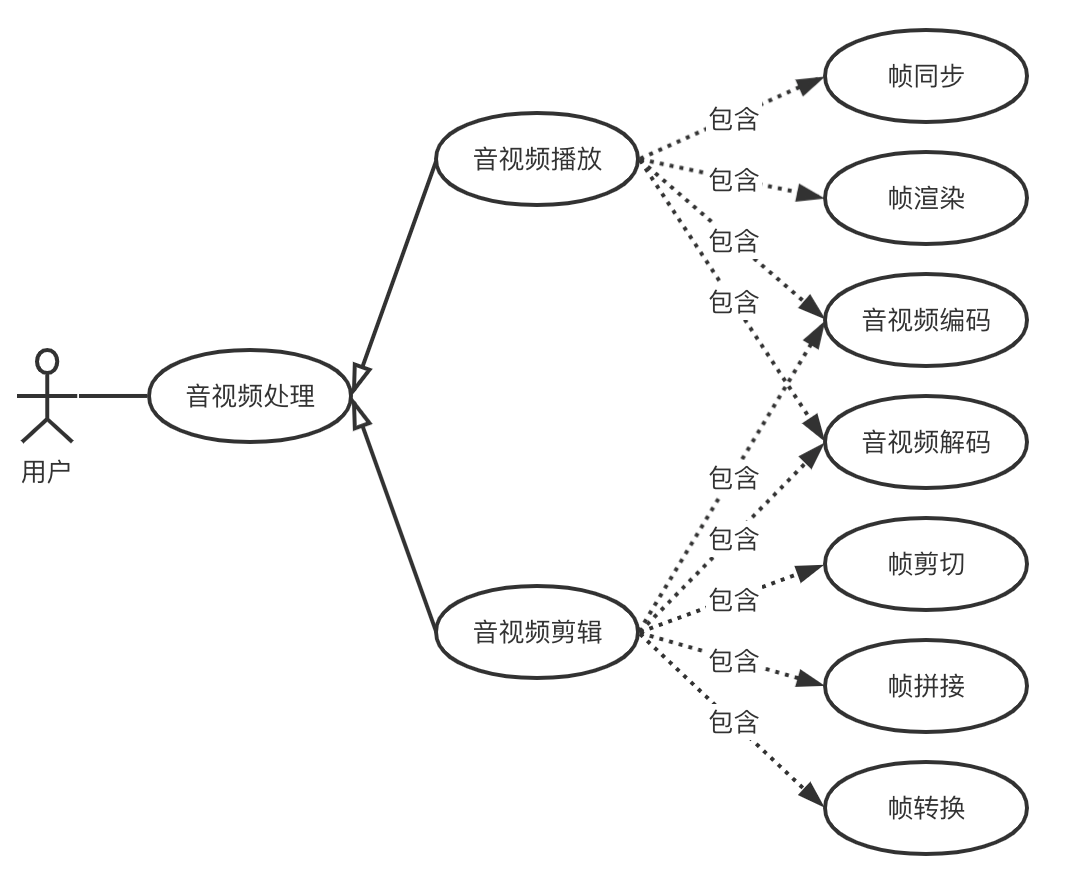
**2.2 FFmpeg**

**2.3 WebAssembly**

(WebWorker)

## 需求分析

**3.1 功能需求**



**3.2 非功能需求**

处理的性能指标（主要是播放时的fps、处理速度）

剪辑前后音视频质量对比

系统自身的代码安全、可维护性

## 详细设计

**4.1 基于FFmpeg的编解码流程设计**

**4.2 剪辑功能中比较有特点的2-3个着重描述**

（增加字幕、截取拼接、格式转换...）

**4.3 播放详细设计**

1、音视频帧的同步解码传递、渲染策略（通过播放性能指标对比）

2、音视频的渲染优化（体现在YUV图像数据渲染）

可补充:（流式的构建剪辑任务，使用DAG拓扑算法）

## 系统实现

同上，暂略

## 六、系统测试

**6.1 功能性测试**

针对各个功能进行灰盒测试，列举测试用例进行测试（包含兼容性测试）

**6.2 非功能性测试**

1. 处理性能指标，通过浏览器自带开发者工具可以监控，（其他工具待调研); 渲染方案和其他市面流行的方案做比较，例如和已有的库ffmpeg.js、flv.js、hls.js做对比实验。

2、剪辑处理前后视频图像质量测试（PSNR指标、SSIM相似度指标），音频质量测试 (PESQ)

3、代码安全、质量测试 （代码安全有专门的工具检测，质量测试可以考虑压测）

## 总结与展望

暂略