

面向智慧矿山管控平台的视频点播技术^①



杨 坤^{1,2,3}

¹(煤炭科学技术研究院有限公司, 北京 100013)

²(煤矿应急避险技术装备工程研究中心, 北京 100013)

³(北京市煤矿安全工程技术研究中心, 北京 100013)

通信作者: 杨 坤, E-mail: 772026356@qq.com

摘 要: 针对现有的视频点播技术无法直接在智慧矿山管控平台中直接使用的问题, 基于 HTTP 的自适应码率流媒体传输协议与 FFmpeg 开源库设计一种视频点播技术. 该技术包括客户端模块、Web 请求处理模块、多媒体处理模块. 该技术中客户端模块通过设定的视频源信息向 Web 请求处理模块发送视频请求; Web 请求处理模块利用请求中的视频源信息调用多媒体处理模块获取多媒体流文件, 并将流媒体文件请求地址返回给客户端; 客户端接收到流媒体地址后, 根据流媒体地址请求视频文件并播放. 通过引入 hls.js 开源库, 该技术可以在任何使用 HTML5 技术的客户端中实现视频信息的点播, 实现了智慧矿山管控平台的跨终端浏览器、多业务模块、去插件的嵌入视频信息, 满足了视频信息与煤矿安全生产的各业务信息的融合展示与协同分析的需求.

关键词: 智慧矿山; 视频点播; 去插件; 实时流传输; 自适应码流媒体

引用格式: 杨坤. 面向智慧矿山管控平台的视频点播技术. 计算机系统应用, 2022, 31(3): 318-325. <http://www.c-s-a.org.cn/1003-3254/8383.html>

Video on Demand Technology for Intelligent Mine Management and Control Platform

YANG Kun^{1,2,3}

¹(CGTEG China Coal Research Institute, Beijing 100013, China)

²(Engineering Research Center of Coal Mine Emergency Technology and Equipment, Beijing 100013, China)

³(Beijing Coal Mine Safety Engineering Technology Research Center, Beijing 100013, China)

Abstract: To tackle the problem that the existing video on demand (VOD) technology cannot be directly used in the intelligent mine management and control platform, this study designs a VOD technology based on the adaptive bitrate streaming media transport protocol of HTTP and FFmpeg open-source library. The technology includes client module, Web request processing module and multimedia processing module. In this technology, the client module sends the video request to the Web request processing module through the set video source information. The Web request processing module uses the video source information in the request to call the multimedia processing module to obtain the multimedia streaming file and returns the request address of the streaming file to the client. After receiving the streaming media address, the client requests and plays the video file according to the streaming media address. By introducing hls.js open-source library, the technology can achieve VOD in any client using HTML5 technology, realize the embedded video information of cross-terminal browser, multi-business module and plug-in removal of the intelligent mine management and control platform, and meet the needs of fusion display and collaborative analysis of video information and various business information of coal mine safety production.

Key words: intelligent mine; video on demand; plug-in removal; real time streaming; adaptive bitstream media

① 基金项目: 煤炭科学技术研究院技术创新基金 (2019CX-I-09)

收稿时间: 2021-05-27; 修改时间: 2021-07-01; 采用时间: 2021-07-09; csa 在线出版时间: 2022-01-24

智慧矿山管控平台是实现智慧矿山建设的关键之路,它能够集成、融合、分析与管控各煤矿业务系统,通过“二三维 GIS 一体化”“智能生产”“智慧管理”等业务模块,全面辐射实现煤矿绿色、高效与安全生产的核心业务^[1,2]。视频监控系统作为实现煤矿智能化无人开采的关键系统与煤矿安全生产的多系统协同分析预处理的关键信息源,在智慧矿山管控平台的建设中发挥着重要的作用^[3,4]。视频监控系统作为智慧矿山管控平台的重要信息,其适用场景存在如下特点:

(1) 跨平台:智慧矿山管控平台的部分重要场景需要同时在桌面端与移动端展示运行,因此内嵌的视频监控信息源也必须满足跨平台的特性。

(2) 协同分析:智慧矿山管控平台是集成、融合、分析与管控煤矿各项业务的巨系统,其信息来源包含煤矿安全生产的各个方面,且需要满足多项信息的融、分析与联动功能。例如,预警与报警信息与视频信息的联动就包含:煤矿瓦斯预警系统发出预警、煤矿安全监控系统的测点报警与皮带监测系统预警时,相应预警或报警地点的视频信息自动弹出。因此,内嵌的视频监控信息源无法采用 SDK 开发。

(3) 无卡顿:智慧矿山管控平台的视频窗口弹出播放时,一般都是相关系统发出预警或报警信息。因此,内嵌的视频监控信息源也必须满足无卡顿播放的特性。

(4) 多客户端:智慧矿山管控平台涉及到煤矿安全生产的各个部门,存在着多个客户端同时弹出视频信息的场景与点播视频信息的场景。因此,内嵌的视频监控信息源也必须满足多客户端的无卡顿播放的特性。

针对矿井视频监控系统在智慧矿山建设与煤矿安全生产领域的应用得到了大量研究。例如,文献^[5]提出了一种地理信息与矿井视频融合展示的理念,文献^[6]提出了一种安全检测、人员定位、语音广播与矿井视频信息协同分析与联动的理念。然而,现有的矿井视频点播技术一般是基于插件或者专用 SDK 来实现,无法满足视频信息与其他信息系统的协同分析与处理的理念,造成了智慧矿山管控平台各业务模块中,无法直接嵌入视频信息,进而无法实现视频信息与其他信息系统的协同分析^[7]。因此,考虑到现有的智慧矿山管控平台一般采用 B/S 架构,本文基于 HTTP 的自适应码率流媒体传输协议与 FFmpeg 开源库设计了一种视频点播技术。通过引入 hls.js 开源库,该视频点播技术可以将视频信息嵌入到智慧矿山管控平台各业务模块,而

且能够满足任意使用 HTML5 技术的客户端。

1 视频点播关键技术

1.1 HLS

HLS (HTTP live streaming, 基于 HTTP 的自适应码率流媒体传输协议) 是苹果公司开发的一种面向 HTTP 渐进下载的代表性协议。该协议将视频流和音频流基于 HTTP 协议发送到客户端,它最初只是引用在苹果公司的终端上,现在大多数桌面应用也使用该协议,例如 HTML5 直接支持该协议^[8]。

如图 1 所示,基于 HLS 的多媒体技术一般由多媒体服务器、分发服务器和客户端组成。多媒体服务器由编码器和分割器组成,编码器主要实现对多媒体信息进行编码,使之符合 HLS 协议的规范,能在相应的客户端播放;分割器主要实现对编码后的多媒体信息进行分割,形成规定长度的多媒体文件片段和对应的 m3u8 索引文件。分发服务器主要是响应客户端的请求,向客户端传输 m3u8 索引文件和多媒体文件片段。客户端主要是请求多媒体信息文件并重组,以多媒体流的形式展现多媒体信息^[9]。

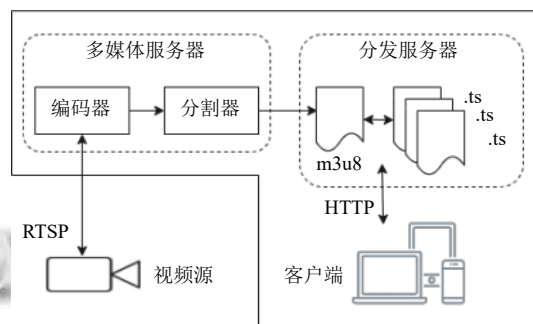


图 1 基于 HLS 的多媒体技术栈图

1.2 m3u8 和 ts

在基于 HLS 的多媒体技术中,多媒体服务器会将采集的音、视频文件分别进行 AAC 和 H.264 编码,并将编码后的数据封装成符合 MPEG-2 格式的 TS 流文件,然后利用分割器对 TS 流文件进行切分,生成一系列 .ts 格式的固定长度的多媒体片段和对应的 m3u8 格式的索引文件^[10]。

m3u8 格式的索引文件包含了一系列的多媒体文件的位置。

.ts 文件是以 MPEG-2 格式封装的多媒体流文件,该文件包含的多媒体信息中必须包含足够的信息以使

得解码器完成初始化工作,且多媒体流文件之间必须包含时间戳与序列号,并在空缺部分加入 DISCONTINUITY 标签,否则会引起视频播放异常.

1.3 RTSP 协议

RTSP (real time streaming protocol, 实时流传输协议) 是实现多媒体串流传输控制的协议,其本身并不进行多媒体串流的传输,但是可以实现基于 RTP (real-time transport protocol, 实时传输协议) 上的流媒体播放、快进、暂停等操作. RTSP 协议的作用是为了选择和控制多个多媒体串流的传输通道,用于在客户端和服务端之间创建和协商实时串流的会话,并且可以同时控制多个连续性的多媒体串流^[11,12].

RTSP 的消息类型有两种:请求消息(客户端对服务器)与回应消息(服务器对客户端),消息格式主要包含起始行、标题行、消息主体等.如图 2 所示,为 RTSP 请求消息格式;如图 3 所示,为 RTSP 回应消息格式.

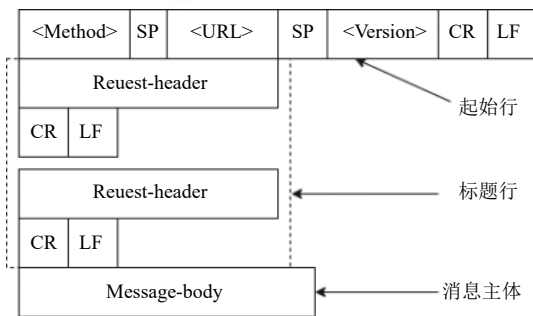


图 2 RTSP 请求消息格式图

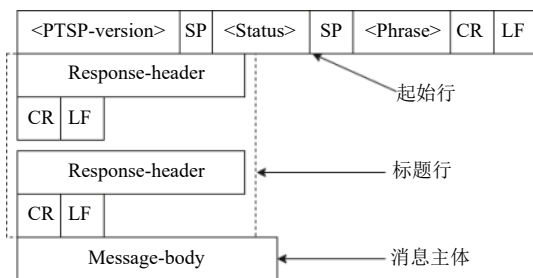


图 3 RTSP 回应消息格式图

1.4 FFmpeg 开源库

FFmpeg 是由 Fabrice Bellard 发起的一个在 Linux 平台上开发的跨平台的开源项目. FFmpeg 是一种比较领先的多媒体框架,能够实现音视频的解码、编码、转码、混合、解密、等功能^[13].

FFmpeg 包含 libavcodec、libavutil、libavformat、

libavfilter、libavdevice、libswscale、libswresample 等库文件,还包含了 ffmpeg、ffplay 和 ffprobe 等可以被终端用户用于转码和播放的工具.

2 视频点播整体架构

视频点播整体架构是基于 HLS 协议,将矿井监控地点的摄像头的多媒体信息或经硬盘录像机传输的多媒体信息编码为 HLS 流媒体文件,并能够响应客户端的多媒体请求,展现实时的多媒体信息.该系统分为客户端模块、Web 请求处理模块、多媒体处理模块、监控视频源,各个模块的交互关系如图 4. 视频点播整体架构如图 5 所示.

3 系统中各模块详细设计

3.1 监控视频源

针对矿井建设的实际情况,矿井下部署的摄像头分为数字式网络摄像头与模拟摄像头,数字式网络摄像头采集的环境信息,既可以直接以 RTSP 视频流的形式直接输出,也可以接入数字式硬盘录像机后,由数字硬盘录像机以 RTSP 视频流的形式直接输出.模拟摄像头须由硬盘录像机转换接入后,以 RTSP 视频流的形式输出多媒体信息.以海康产品为例,数字式摄像头与数字式硬盘录像机的 RTSP 视频流的获取格式如表 1 所示.

3.2 多媒体处理模块

多媒体处理模块以 FFmpeg 框架为核心,包括编码器与分割器.

3.2.1 编码器

编码器主要是基于从矿井部署的数字摄像头或数字式硬盘录像机获取的 RTSP 视频流重新封装成符合 MPEG-2 格式的 TS 流文件,主要分为:解析流信息、编码流信息、添加时间戳、添加附属信息与封装 TS 流.

编码器工作的具体流程如图 6 所示:首先,从部署的数字式摄像头或数字式硬盘录像机中获取的 RTSP 视频流信息中解析出 RTP (real-time transport protocol, 实时传输协议) 数据包,进一步从 RTP 数据包中解析出多媒体 ES (elementary stream, 基本码流);然后对基本的多媒体信息分别进行编码:对视频基本信息进行 H.264 编码,对音频基本信息进行 ACC 编码;然后将多媒体信息的显示时间值和解码时间值添加到编码后的

多媒体基本信息中,形成分组 ES;最后,将节目相关表、节目映射表、网络信息表等解码用的 PSI (pro-

gram specification information, 节目专用信息) 添加到分组 ES,并转成符合 MPEG-2 格式的 TS 流文件^[14].

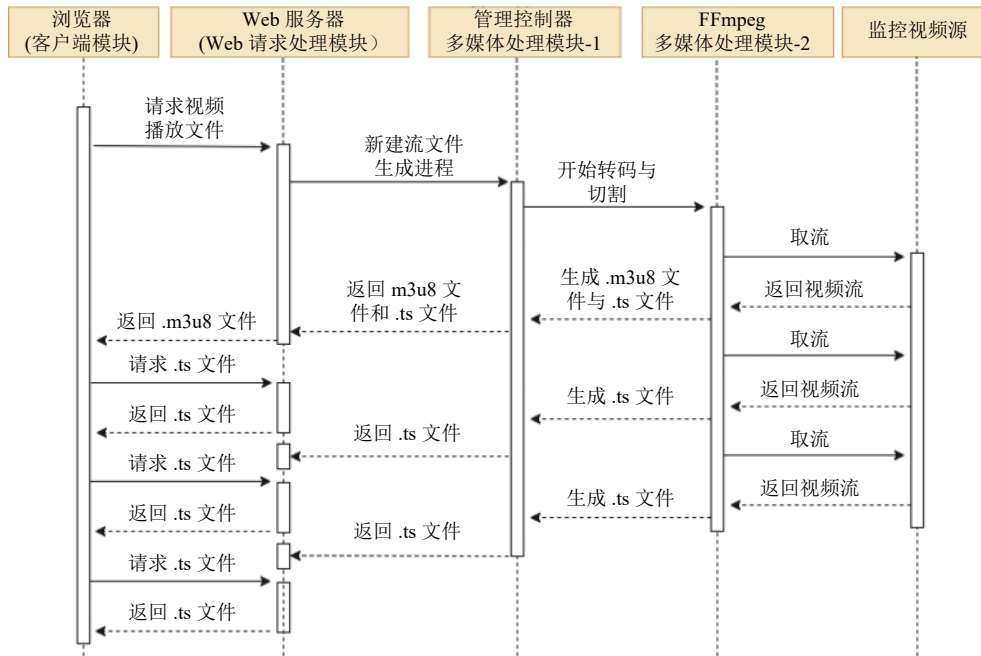


图4 各模块交互图

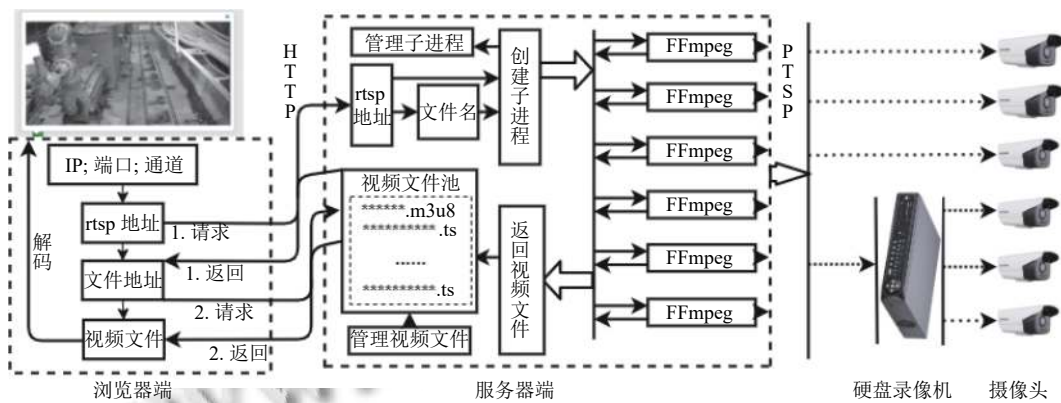


图5 视频点播平台架构图

表1 RTSP 取流格式表

设备类型	地址
摄像头	rtsp://{username}:{password}@{IP}:{port}/h264/ch1/sub/av_stream
硬盘录像机	rtsp://{username}:{password}@{IP}/streaming/channels/{i}01

3.2.2 分割器

分割器主要是实现将符合 MPEG-2 格式的 TS 流文件按照指定的域指切割成固定大小、可以播放的 ts 文件,并在切割的过程中创建一个保存一些列 ts 文

件地址的 m3u8 索引文件。

分割器的工作流程如图 7 所示:首先,获取到符合 MPEG-2 格式的 TS 流文件与预设的分片阈值,查询 TS 传输流中是否还有流数据,若有,则解析 TS 流数据中每一帧的 DTS (decoding time stamp, 解码时间值),当该 DTS 值小于预设的分片阈值时,则写入预定的 ts 文件;当该 DTS 值大于预设的分片阈值时,则关闭当前的 ts 文件,重新打开一个新的 ts 文件并写入。

此外,在分割 TS 流文件时,会生成一个保存 ts 文

件的地址的 m3u8 索引文件. 当产生一个新的 ts 文件时, 会在 m3u8 索引文件的末尾插入一个 ts 地址信息, 并删除首部的一个最旧的 ts 文件地址信息.

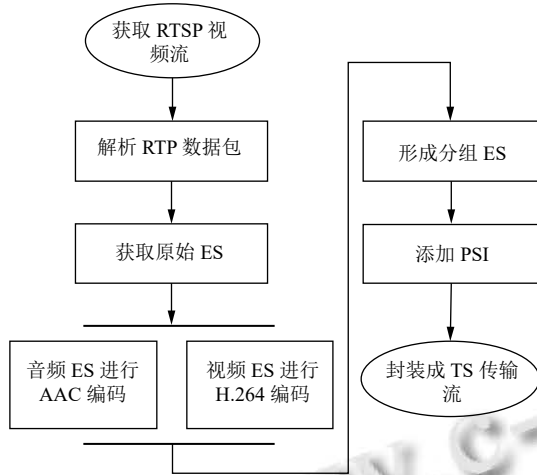


图6 编码器工作流程图

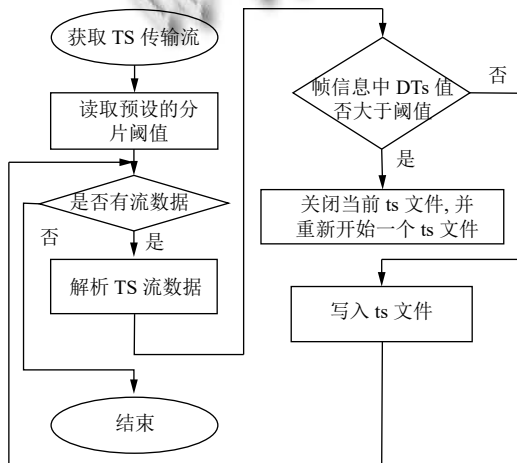


图7 分割器工作流程图

3.3 Web 请求处理模块

Web 请求处理模块主要是基于 HTTP 协议响应客户端的请求, 在本文设计的视频点播平台中主要实现: 1) 根据视频源信息, 调用多媒体处理模块, 生产请求的视频源信息的 m3u8 文件与 ts 文件, 并返回视频信息的 m3u8 索引文件; 2) 根据 m3u8 索引文件, 返回 ts 文件.

Web 请求处理模块的工作流程如图 8 所示, Web 请求处理模块一直监听客户端的请求, 当监听到客户端的处理视频请求就时, 首先获取到 src 参数, 即 `rtsp://admin:admin123@192.168.21.5/streaming/channels/10` (以某硬盘录像机, 第 1 通道的视频源信息为例), 将 src 参数做一个 sha256 加密运算, 生成一个固定长度

的 m3u8 文件名, 如: “e79e68f070cdedcfe63eaf1a2e-92c83b4cfb1b5c6bc452d214c1b7e77cdfd1c7-playlist.m3u8”, 以 src 值与 m3u8 文件名为参数调用多媒体处理模块, 然后检测文件夹中是否生成了预设名称的 m3u8 文件, 若生成了, 则将 m3u8 文件信息返回给客户端. 当客户端获取到 m3u8 文件信息中 ts 文件地址后, 则根据文件地址返回 ts 文件信息.

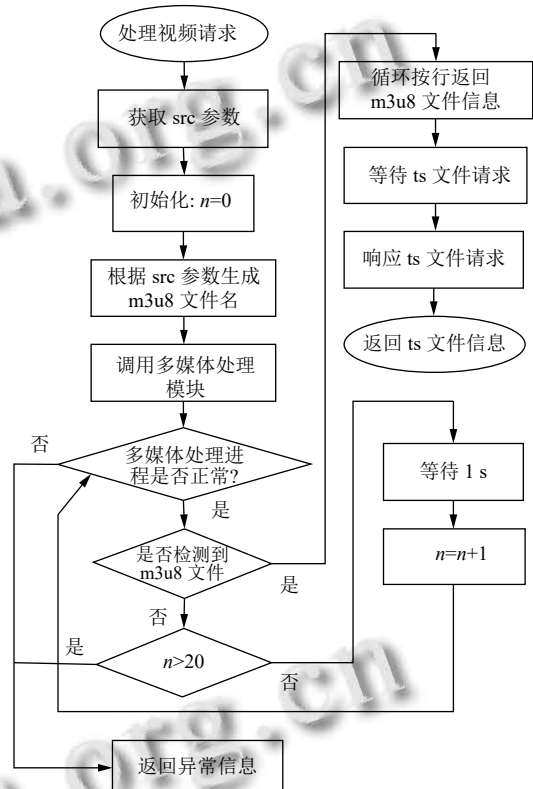


图8 Web 请求处理模块的工作流程图

此外, 针对 Web 处理模块中多媒体处理进程与 m3u8 文件检测异常的处理流程为: 当检测到多媒体处理进程工作异常时, 则返回给客户端异常信息; 当检测不到预设的 m3u8 文件名时, 等待 1 s, 继续检测; 当检测 m3u8 文件名的次数大于 20 次时, 则抛出异常信息.

3.4 客户端

结合现有的智矿山管控平台的客户端场景, 本文设计的视频点播平台的客户端以浏览器为主. 现有的一些浏览器还不支持基于 HLS 协议的传输多媒体的方式, 为使本文设计的视频点播平台兼容所有的浏览器, 本文在视频点播场景中使用 hls.js 开源库.

开源的 hls.js 是一个 JavaScript 库, hls.js 基于 HTML5 和 MSE 技术, 实现无插件的方式将 MPEG-

2 格式的 TS 流文件转换成 MP4 视频流,使得 TS 流文件可以在 HTML5 中的<Video>标签和<audio>标签中播放,从而实现了视频信息无插件的播放的目的。

基于开源的 hls.js 库实现视频信息的播放的工作流程如图 9 所示,启动播放窗口后,首先获取视频源信息,其格式为: rtsp://admin:admin123@192.168.21.5/streaming/channels/101 (以某硬盘录像机,第 1 通道的视频源信息为例),将视频源信息添加到 Web 服务器请求路径形成组合请求地址,其格式为: http://192.168.21.15:8000/streams/?src=rtsp://admin:admin123@192.168.21.5/streaming/channels/101,随后获取到返回的最新的 m3u8 索引文件,并对其进行解析;然后根据 m3u8 索引文件获取 ts 文件,将获取到的 ts 文件转码存入 ArrayBuffer,并在 Media Source 中将 ts 文件合流,转换成视频信息输出。

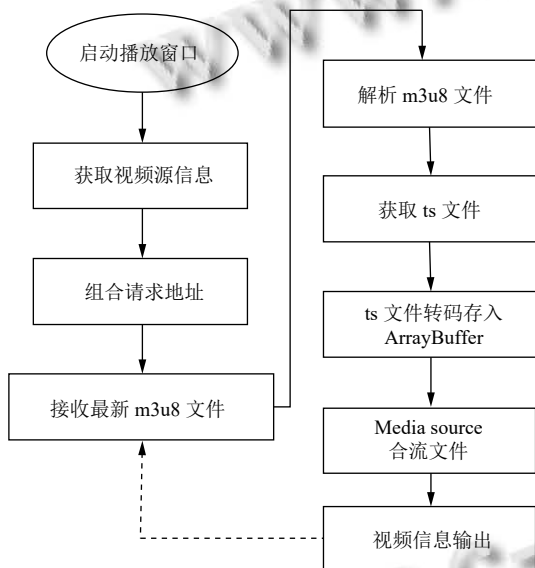


图 9 客户端工作流程图

4 方案设计验证

本文设计视频点播平台实际的开发环境选用 Ubuntu 20.04 操作系统,数字式硬盘录像机选用海康威

视某型号的设备,测试摄像头连接的硬盘录像机的通道号为 01,其获取多媒体信息流的 RTSP 地址为: rtsp://admin:admin123@192.168.21.5/streaming/channels/101.

多媒体处理模块的编码器和分割器采用基于 FFmpeg 的开源框架,根据 rtsp 地址参数与对应的 m3u8 文件地址参数调用多媒体处理模块,其主要格式为:“ffmpeg -copytb 1 -r 100 -crf 25 -preset faster -maxrate 500k -bufsize 1500k -codec copy -hls_time 2 -hls_list_size 5 -hls_flags delete_segments -start_number 1 619004890 -y/static/streams/e79e68f070cdedcfe63eaf1a2e92c83b4cfb1b5c6bc452d214c1b7e77cdfd1c7-playlist.m3u8 -i rtsp://admin:admin123@192.168.21.5/streaming/channels/101”,其中,设定分频阈值的格式为: -hls_time 2; m3u8 索引文件存储 ts 文件地址最大个数的指定方式为: -hls_list_size 5; m3u8 文件地址指定方式为: -y/static/streams/e79e68f070cdedcfe63eaf1a2e92c83b4cfb1b5c6bc452d214c1b7e77cdfd1c7-playlist.m3u8, ts 文件的存放文件夹为/static/streams/;码流原始文件的获取地址指定方式为: -i rtsp://admin:admin123@192.168.21.5/streaming/channels/101.

Web 请求处理模块是基于 Tornado 框架来实现的,其中, Tornado 是一个基于 Python 语言编写的 Web 服务框架.该模块接收视频信息源请求,并返回对应的 m3u8 索引文件;接收 ts 流文件的请求,返回对应的流文件内容。

4.1 方案效果

视频点播平台搭建完毕后,通过引入 hls.js 开源库,能够满足任意使用 HTML5 技术的客户端,并且可以嵌入智能矿山平台中任意使用场景.如图 10 所示,3D-GIS、2D-GIS、视频集中展示场景中都可以嵌入本文设计的视频点播窗口。

4.2 视频延时优化

考虑到 HLS 协议实现方式的限制,基于 HLS 的视频播放技术中存在着播放延时,其理论延时时间如式(1)所示。

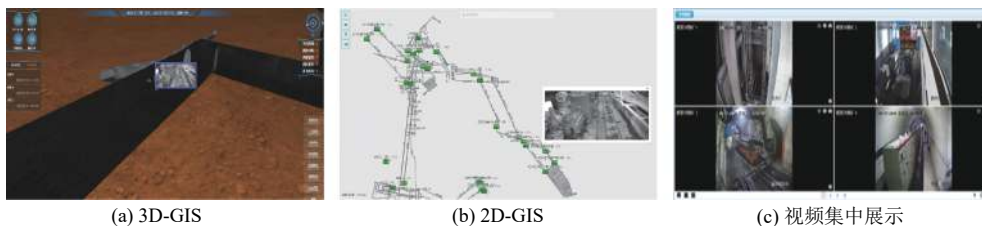


图 10 示例场景图

为了进一步减少延时的影响,针对理论延时的限制,从以下3个方面对该点播技术在智慧矿山管控平台中的场景做了部分优化:

(1) 视频源设置

视频分辨率强制为1280×960:针对现有智慧矿山建设情况的调研,现有的矿井的摄像头的分辨率最低为1280×960,将其强制设置成固定分辨率,可以优化多媒体处理模块动态化扩展视频分辨率所占用的资源。

帧数设置为25 fps:考虑到人眼的识别率,将现有的矿井的摄像头的播放帧设置为25。

I帧间隔设置为50 fps:关键帧I帧设置为每2s包含1帧,可以对容错与压缩进行最大程度的优化。

(2) 切片优化

如式(1)所示, T_1 越小, T 越小,因此,理论上 T_1 设置的越小越好。但是,考虑到矿井井下网络的有波动时, T_1 过小会造成监视图像中出现灰色轮廓或绿色区域。根据关键帧I设置为2s包含,最终确定 T_1 为2s。

(3) 播放数量优化

如式(1)所示, num_1 越小, T 越小,因此,理论上 num_1 设置的越小越好。考虑到分段音视频文件的无缝连接,最终确定 num_1 为2。

(4) 客户端拉流优化

针对Web服务端的响应无法立即在Header中确定响应内容大小时,服务器一般不会提供Content-Length的头信息,而是会采用HTTP1.1的Transfer-

Encoding: chunked。考虑到HLS协议的m3u8和ts文件是动态生成,在响应的过程中可以设置Transfer-Encoding: chunked,此时,对于分片ts文件来说,它的生产和发送就可以实现同步。从而进一步降低播放延时,此时的播放延时时长如式(2)所示。当我们把满足播放的最小切片个数设置为1时,即可实现生产、发送和播放同步。

$$T = num_1 + T_1 + T_0 \quad (1)$$

$$T = (num_1 - 1) \times T_1 + T_0 \quad (2)$$

其中, T 表示延时时间, T_1 表示生成1个切片的时长, num_1 表示满足播放的最小切片个数, T_0 表示其他影响因素,且此部分的影响因素一般无法优化。

通过以上的优化,实验表明该视频点播技术的播放延时可以优化到3~4s,基本可以满足智慧矿山管控平台的视频点播要求。

4.3 方案对比

为了进一步说明该方案在智慧矿山管控平台的适用特点,如表2所示为本文设计的视频点播技术与现有的主流的Web视频点播技术的相关指标的测试对比。其中,本文设计的视频点播技术满足智慧矿山管控平台的跨平台、无插件、协同分析、无卡顿的适用场景;基于海康SDK开发的视频点播技术不支持跨平台与协同分析,并且在客户端数量过多时,存在卡顿现象;基于插件开发的Web视频点播技术依赖插件,且不满足协同分析的要求,并且在客户端数量过多时,存在卡顿现象。

表2 不同技术方案指标对比表

视频点播技术	跨平台	支持度	无插件	协同分析	卡顿现象
本文设计	支持	主流浏览器都支持	满足	满足	无
基于海康SDK	不支持	—	—	不满足	有
基于插件	依赖插件	依赖插件	不满足	不满足	有

表3的测试环境为:矿井网络带宽为千兆,摄像头数量50个,在正常生产环境(其他系统正常运行)下,选取16路测试视频集中播放。其结果如表3所示,其中,在客户端数量超过一定限制后,由于基于海康SDK开发的Web视频点播技术与基于插件开发的Web视频点播技术采用直接从摄像头获取视频流的方式播放,存在着卡顿现象,不太适合与智慧矿山管控平台的适

用场景。

5 结论

本文结合现有的智慧矿山管控平台的架构特点,基于HLS协议与FFmpeg开源库,设计一种面向智慧矿山管控平台建设的视频点播技术。通过引入hls.js开源库,该技术可以在任何使用HTML5技术的客户端

中实现视频信息的点播,实现了智慧矿山管控平台的跨终端浏览器、多业务模块、去插件的嵌入视频信息,满足了视频信息与煤矿安全生产的各业务信息的融合展示与协同分析的需求。

表3 不同方案的卡顿现象对比表

客户端数量	本文设计	基于海康SDK	基于插件
1	无	无	无
3	无	少量卡顿	少量卡顿
6	无	部分卡顿	部分卡顿

参考文献

- 孙继平. 煤矿信息化与自动化发展趋势. 工矿自动化, 2015, 41(4): 1-5.
- 王国法, 赵国瑞, 任怀伟. 智慧煤矿与智能化开采关键核心技术分析. 煤炭学报, 2019, 44(1): 34-41.
- 毛善君, 杨乃时, 高彦清, 等. 煤矿分布式协同“一张图”系统的设计和关键技术. 煤炭学报, 2018, 43(1): 280-286.
- 张科学. 综掘工作面智能化开采技术研究. 煤炭科学技术, 2017, 45(7): 106-111.
- 毛善君. “高科技煤矿”信息化建设的战略思考及关键技术. 煤炭学报, 2014, 39(8): 1572-1583.
- 孙继平, 钱晓红. 煤矿重特大事故应急救援技术及装备. 煤炭科学技术, 2017, 45(1): 112-116, 153.
- 黄瑶. 基于 HLS 的视频点播系统的设计与实现 [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2017, (7): 169-173.
- 王林, 石焘. 基于 HLS 协议视频监控加密系统优化设计与实现. 计算机测量与控制, 2019, 27(7): 169-173, 179.
- 王豪赞. 基于 SIP 的信令处理与音视频切片传输机制的设计与实现 [硕士学位论文]. 杭州: 浙江工业大学, 2020.
- 罗淑贞, 耿恒山, 徐祥男, 等. 基于 HLS 协议的流媒体直播系统的研究和改进. 郑州大学学报(工学版), 2014, 35(5): 36-39.
- 徐华龙. 基于三维视频融合的矿井实时监测技术. 煤矿安全, 2021, 52(1): 136-139.
- 李文峰, 高原. 基于 ONVIF 的本安型手持式信息记录仪. 煤矿安全, 2019, 50(1): 92-95.
- 金艳楠, 邹波, 韩杰. 地下矿山移动视频监控系统的的设计. 煤矿安全, 2015, 46(6): 104-106, 110.
- 朱秀昌, 陈杰. 基于 HTTP 的视频流网络传输. 南京邮电大学学报(自然科学版), 2016, 36(3): 1-9.