简易视频会议软件(YauMeeting)技术报告

黄添豪 2024013355

一、技术概述

视频会议是现代互联网办公最为重要的一项功能，在许多领域具有重要价值。市场上有腾讯会议、Zoom等多种产品。视频会议服务涉及了服务器的架设、网络传输管理、网络安全防范、用户操作界面等诸多方面。本项目旨在实现一个简易的视频会议软件（称作YauMeeting，致敬了清华大学**丘成桐**教授），其中涉及了多方面的C++程序设计知识与面向对象的程序设计思想，作为《面向对象程序设计基础》课程的大作业。

二、整体框架

（一）综述

本应用使用C++编写，采用C/S架构，分为服务端Server和客户端Client。本应用适应Windows平台，使用了WinAPI。外部库的使用包括asio（non-boost）,OpenCV, SFML，PortAudio，分别用于网络、视频、客户端窗口、音频等模块的实现。

网络部分采用asio这个网络库，采用UDP协议传输，相较TCP具有连接便捷、传输方便的优势，缺点是传输可能不稳定，这在本应用中进行了一些优化的尝试。

视频部分采用OpenCV获得摄像头以及一些处理，OpenCV是比较流行的一个图像处理库，通过它与SFML配合实现窗口的绘制。

窗口部分采用SFML，这是一个轻量级的多媒体库，可以用来绘制我们的窗口，且使用相较Windows自带的WinAPI更加方便。

音频部分采用PortAudio，主要用来获得麦克风声音。

其余还有C++的thread、mutex等多线程库，以及一些基础的控制库。

（二）框架

以下简要介绍客户端与服务端的运行逻辑。

客户端采用多线程，包括主线程、视频源获取线程、音频获取线程、网络接收线程。主线程每一帧内执行窗口更新、数据发送，视频源获取线程持续地捕捉视频源并按帧发送数据，音频获取线程同理，网络接收线程持续读取数据并更新数据。结束时向服务器广播下线。

服务端采用多线程，每个房间单独用一个线程来管理，服务器读取数据后依次转发到房间内各个用户，以及执行一些控制指令的转发。结束时向所有在线用户广播下线。

三、使用方法

客户端运行通过命令

yauclient.exe connect HOST\_IP USER\_NAME USER\_PWD ROOM\_ID (ROOM\_PWD)

来连接至对应的服务器并登录、加入对应房间。

yauclient.exe create HOST\_IP USER\_NAME USER\_PWD (ROOM\_PWD)

来连接至对应服务器并登录、创建指定房间。房间号会在随后窗口中展示，房间密码为可选项。

服务端运行通过命令

yauserver.exe run

来开启服务器服务。通过命令

yauserver.exe register USER\_NAME USER\_PWD

来注册用户。

服务端收发端口为60114，客户端收发端口为60514。服务器的数据存在($USERPROFILE)\AppData\YauMeeting\Server目录下。

四、技术细节

这个项目具有颇多的技术细节，下面我分别讲讲我的实现过程。

（一）视频渲染

我了解到视频渲染有一个“双缓冲”的做法，也即提前渲染下一次要呈现的帧，然后直接替换。通过front\_frame和back\_frame可以缓解传输过程分片导致帧渲染不完全的问题，确保了每一帧完整后才被渲染，保证了视频的流畅度。

（二）网络传输

项目采用UDP协议传输数据。首先需要有一个标头（Header）记录数据类型、用户信息、帧序号、片数、片编号等，然后是待传输的数据。

考虑到UDP的丢包率与传输安全性，所以每次传输尽量限制在1KB的大小，并且采用交叉储存的方式，每次发送相邻两片数据，然后步进一片，这样每片数据会被接受两次，可以降低丢包导致的损失。

最终一并拼接完毕后作为完整帧再进行之后的渲染操作。

（三）音频混合

多人视频会议中的音频混合是一个重要的问题，我采用混音后输出，每次将对应同时的音频先行储存，统一将采样数据叠加后合并再播放，从而达到混音的效果。

（四）SFML与OpenCV配合

由于OpenCV获得的图像数据格式和SFML并不一样，因此需要一步数据处理将cv::Mat的BGR转化为sf::Image的RGB。

五、参考资料

本项目用ChatGPT、Gemini等人工智能应用辅助我进行设计，过程中涉及的AIGC代码在充分理解原理后自行编写，对话记录见reference目录。