CS 4283 Group 5 Final Project Feasibility Report

Introduction

As stated in our proposal, our goal is to build a very basic video streaming application. Our application client would support playback of video files stored on a server, ideally with features like pause, rewind, and fast forward included.

Feasibility Status

Libraries Currently Used:

* OpenCV for video playback, frame-by-frame manipulation, and encoding: Imported and tested out. Seems to be effective if a bit choppy. Much simpler than pipelining with ffmpeg.
* Pyaudio for audio playback and encoding: Lightly tested. Keeping everything synchronized will be the main concern.
* Socket: Just what we’ve been using in class. It seems to work well for our purposes.
* Numpy: Used for part of the encoding/decoding process.
* Possible alternatives: Gstreamer for python, ffmpeg through a pipeline for both audio and video. Compatibility is an issue for both, along with complicated setup processes.

Pending Work

* Sending synchronized audio through the network might make up the bulk of the work on the assignment.
* Ensuring that the data makes it correctly. Might try RTSP or QUIC instead of basic TCP or UDP.
* Playback controls. Pausing, rewind, fast forward.
* Buffering video. The video displays frame by frame immediately as it is sent, which leads to frame pacing issues and choppy playback, not to mention future synchronization issues.
* Video selection: we have a hardcoded video at the moment. User selection could be added as a command line argument.

These tasks are large such that everyone will work some on each one, with subtasks within them being divided evenly.

Conclusion

This project still seems reasonable, especially with the true final submission date of May 1st. Any feedback on our current plans would be welcome.