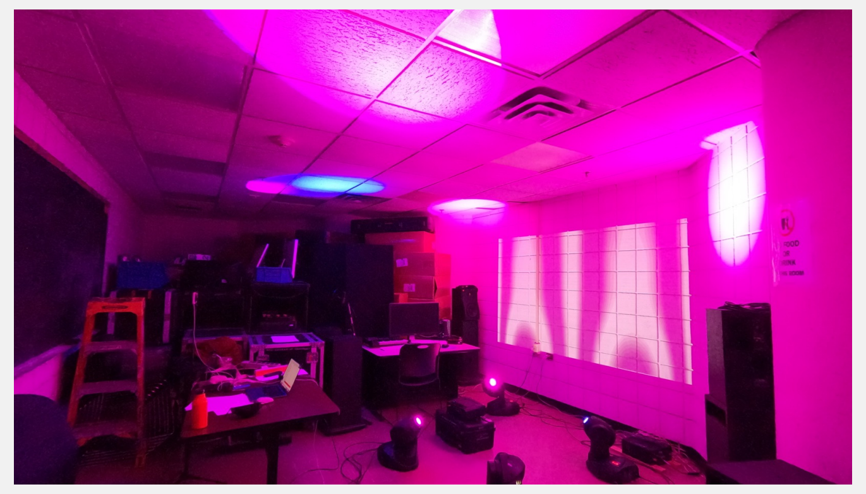
A LIVE AUDIOVISUAL PERFORMANCE SYSTEM

SOUND REACTIVE | SYSTEM DESIGN | VISUAL DESIGN | CODE DRIVEN

YINGLIAN LIU

INTRODUCTION

My project involves creating a live audiovisual performance system focused on *visual effects*. The system features all visual effects, which are driven by *sound* and *code*, and generated in *real-time*. It's designed specifically for live music events in small venues.



OBJECTIVES

- Simplifying the process of programming lighting.
- Utilizing emerging technologies to maintain quality in low-budget live performances.
- Allowing performers to perform freely without being limited by pre-recorded settings.
- Providing a user-friendly interface for show control.

SYSTEM FUNCTIONS

- Motion Graphics
- Live Camera
- Stage Lighting
- User Interface(UI)

MATERIALS AND METHODS

Equipment List:

- Martin MiniMAC Profile
- iColor Cove 12”
- iColor Cove 6”
- ENTTEC DMX USB Pro Interface
- Dell 300 Projector
- LINKSYS Wireless Router
- MacBook Pro
- iPad Air 4
- iPhone 13(Camera)

Software List:

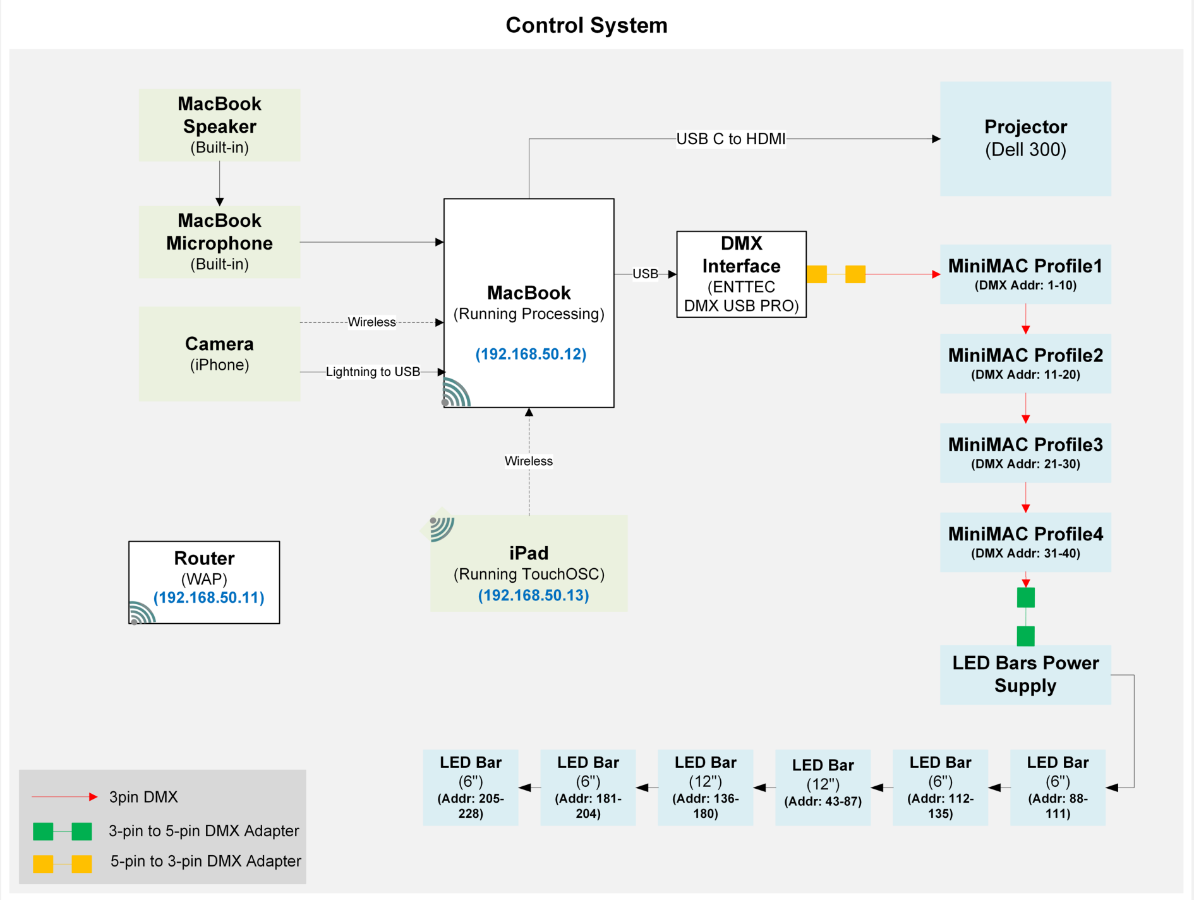
- Processing
- TouchOSC

iColor Cove

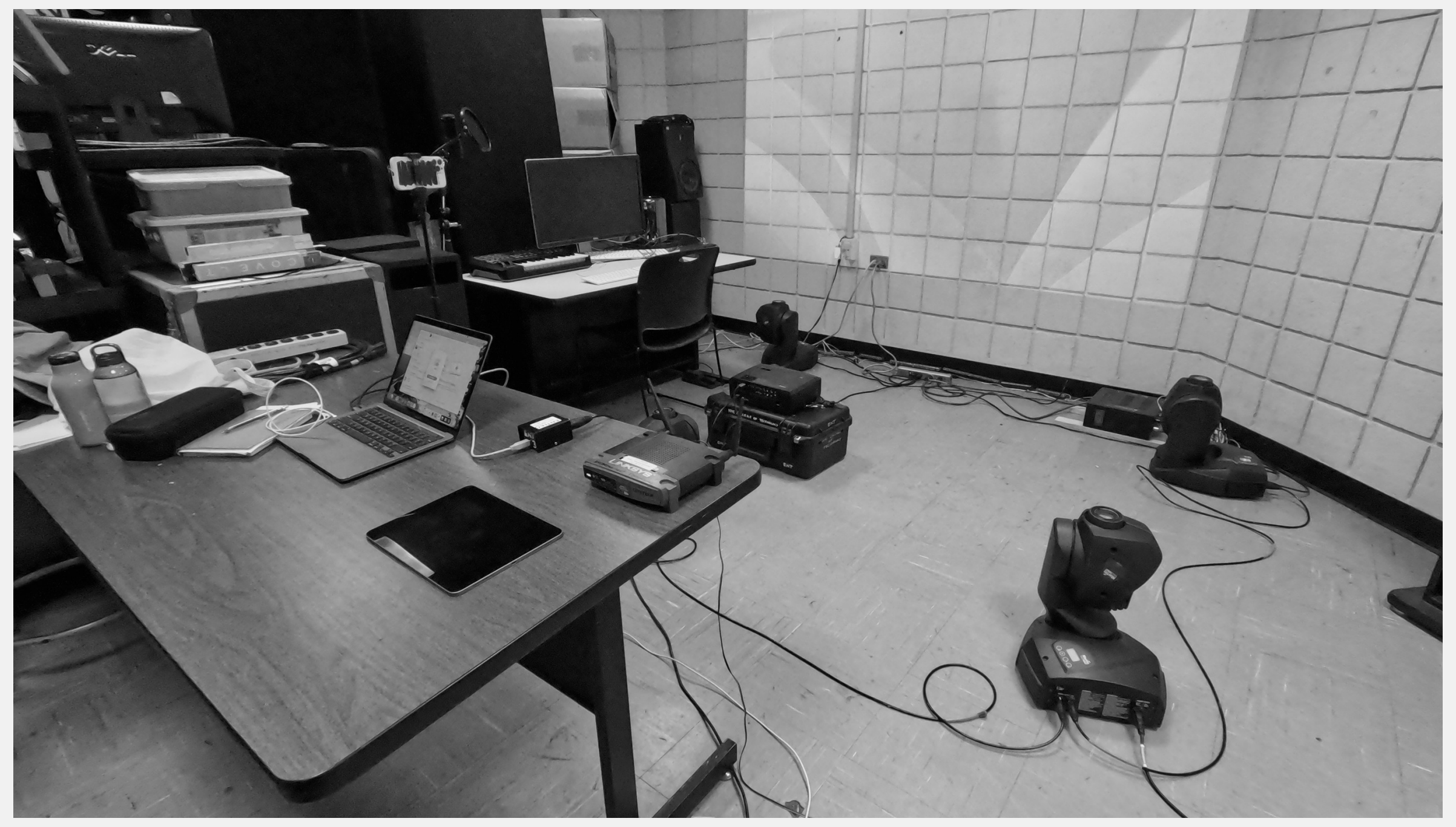
MiniMAC

DMX Interface

SYSTEM DIAGRAM

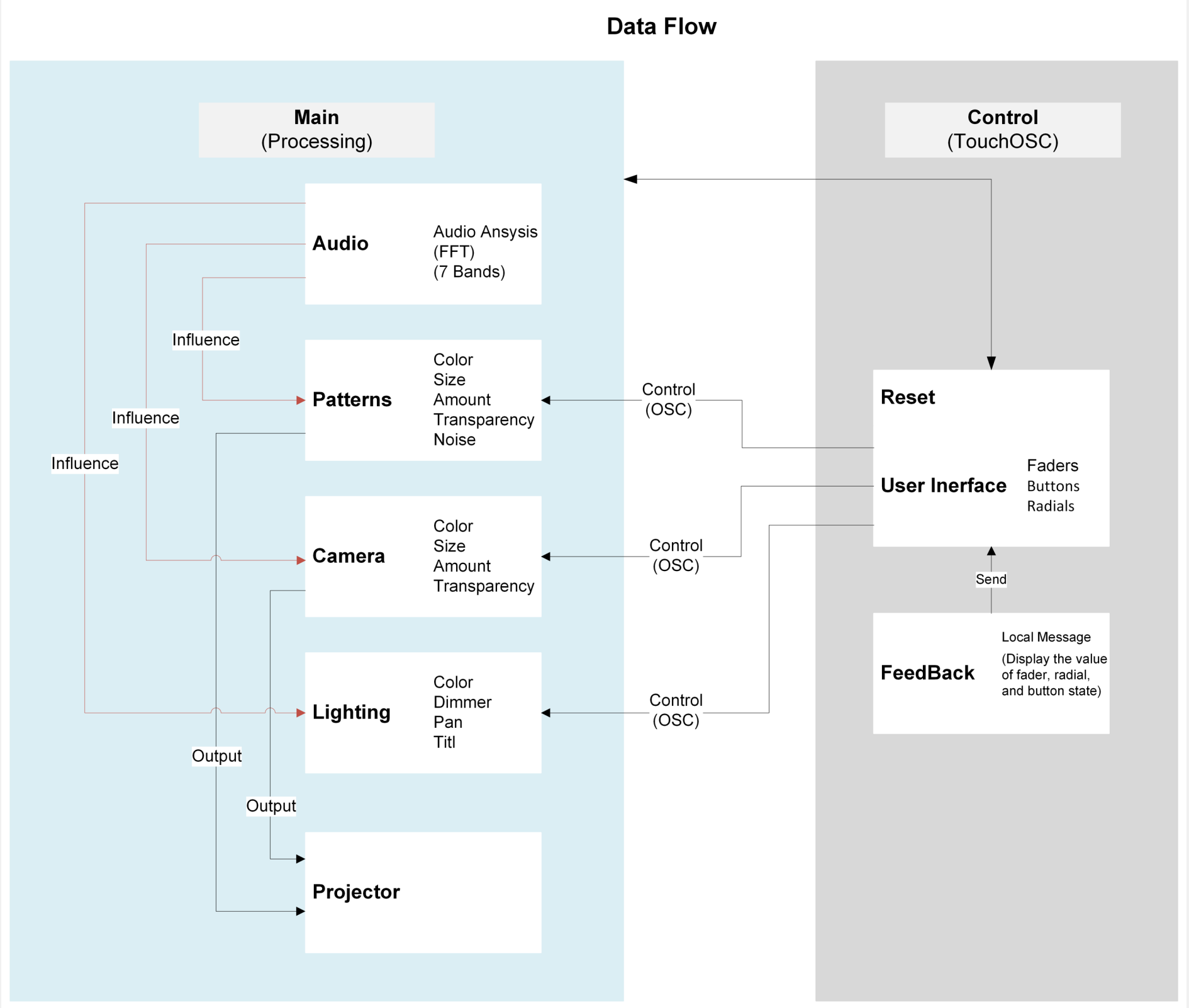


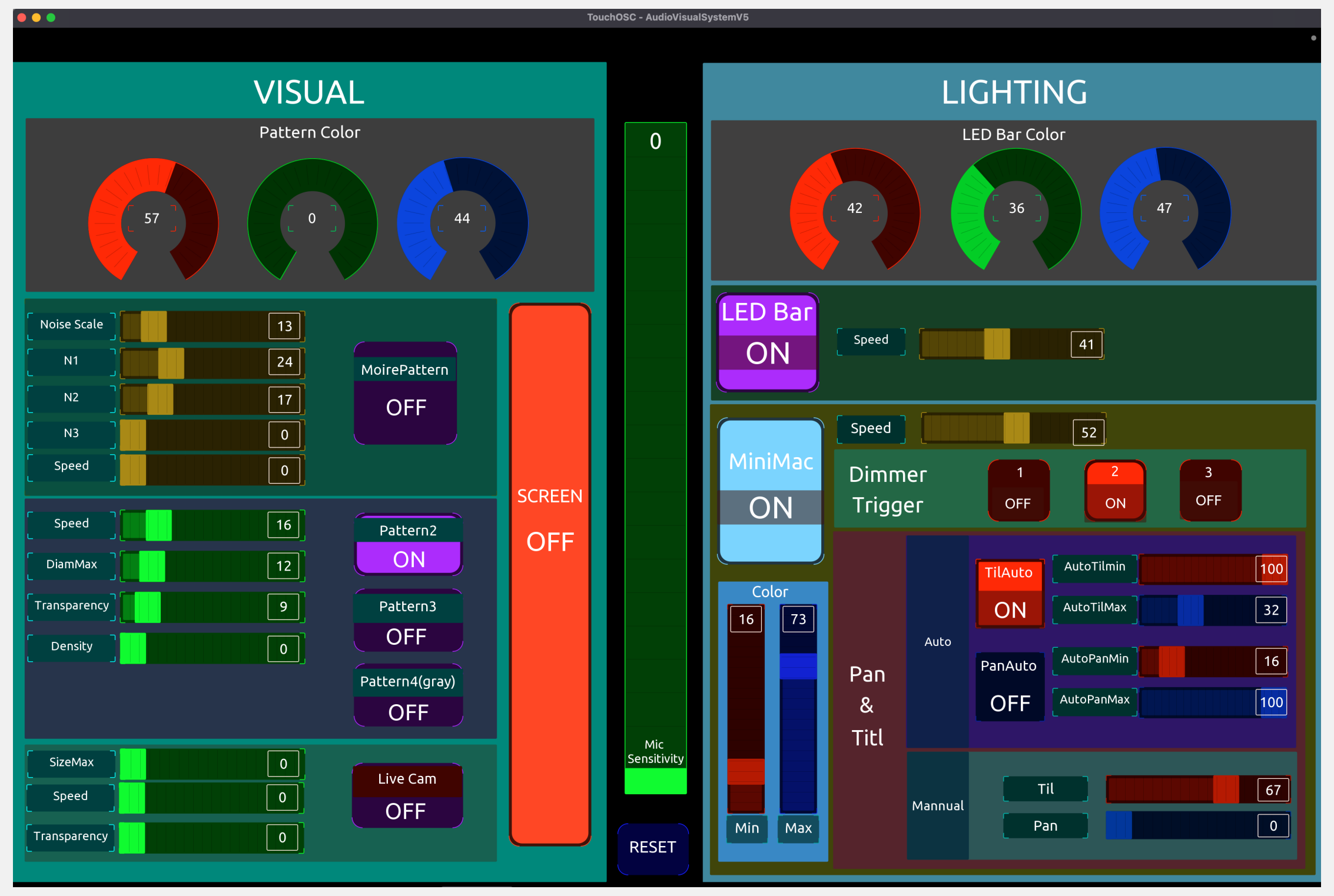
Control System
MacBook Speaker (Built-in)
MacBook Microphone (Built-in)
Camera (iPhone)
Wireless
Lightning to USB
MacBook (Running Processing) (192.168.50.12)
USB C to HDMI
Projector (Dell 300)
USB
DMX Interface (ENTTEC DMX USB PRO)
MiniMAC Profile1 (DMX Addr: 1-10)
MiniMAC Profile2 (DMX Addr: 11-20)
MiniMAC Profile3 (DMX Addr: 21-30)
MiniMAC Profile4 (DMX Addr: 31-40)
LED Bars Power Supply
Wireless
iPad (Running TouchOSC) (192.168.50.13)
Router (WAP) (192.168.50.11)
LED Bar (6") (Addr: 205-228)
LED Bar (6") (Addr: 181-204)
LED Bar (12") (Addr: 136-180)
LED Bar (12") (Addr: 43-87)
LED Bar (6") (Addr: 112-135)
LED Bar (6") (Addr: 88-111)
3pin DMX
3-pin to 5-pin DMX Adapter
5-pin to 3-pin DMX Adapter



Lab: V318

DATA FLOW

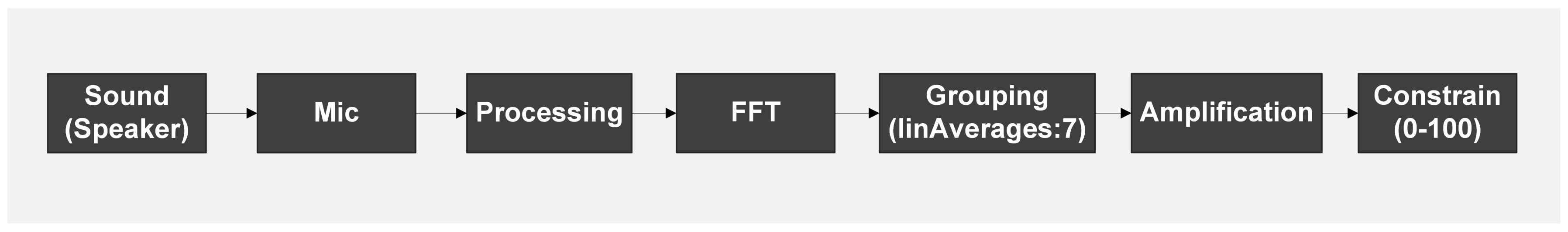




UI Made with TouchOSC

TECHNICAL REALIZATION

- In this project, the visual effects, including stage lighting, motion graphics, and moving images captured by the camera, are controlled using the average amplitude data from different frequency ranges. The audio data is generated by an *FFT analyzer* from a third-party sound library called *Minim* in Processing. Once the raw audio data is gathered, it's grouped into 7 linear frequency bands using the *linAverages()* function. This allows different bands to control different behaviors.
- The user interface communicates with Processing by sending *Open Sound Control (OSC)* messages over the *User Datagram Protocol (UDP)* within an IP network.



Sound (Speaker) → Mic → Processing → FFT → Grouping (linAverages:7) → Amplification → Constrain (0-100)

Audio Signal Analysis Process

RESULTS

- In this experiment, I used audio data to control various parameters (see Fig. Data Flow) of the motion graphics, live camera, and lighting. All these elements responded well to the sound.
- By appropriately processing the audio data and utilizing basic programming techniques, I was able to produce unexpected visual effects.

CONCLUSIONS

- Through this experiment, I've seen that there are numerous possibilities for using these open-source tools to create unique projects.
- The system can generate simple lighting effects, and with manual control, it can produce a variety of other effects.
- This system also provides different types of visual effects using some low-cost equipment, making it suitable for low-budget live shows.
- I'm very excited to have completed this experiment, and I will continue to develop the system, with the hope that it could be used in the future.

FURTHER ITERATIONS

- Adding a function to save settings.
- Using an API or building my own database for different types of lighting fixtures.
- Extending the pattern bank.
- Optimizing the audio analysis process.

ACKNOWLEDGMENTS

ALLISON BERKOY
ADAM WILSON
JOHN ROBINSON

AIYING WU
HOUKIN YING
YANELIS RODRIGUEZ