Synesthesia

D4: Abhishek Agarwal, Parth Maheshwari, Rachana Murali Narayanan
18-500 Capstone Design, Spring 2023
Electrical and Computer Engineering Department
Carnegie Mellon University

tinyurl.com/synesthesiaD4

Product Pitch

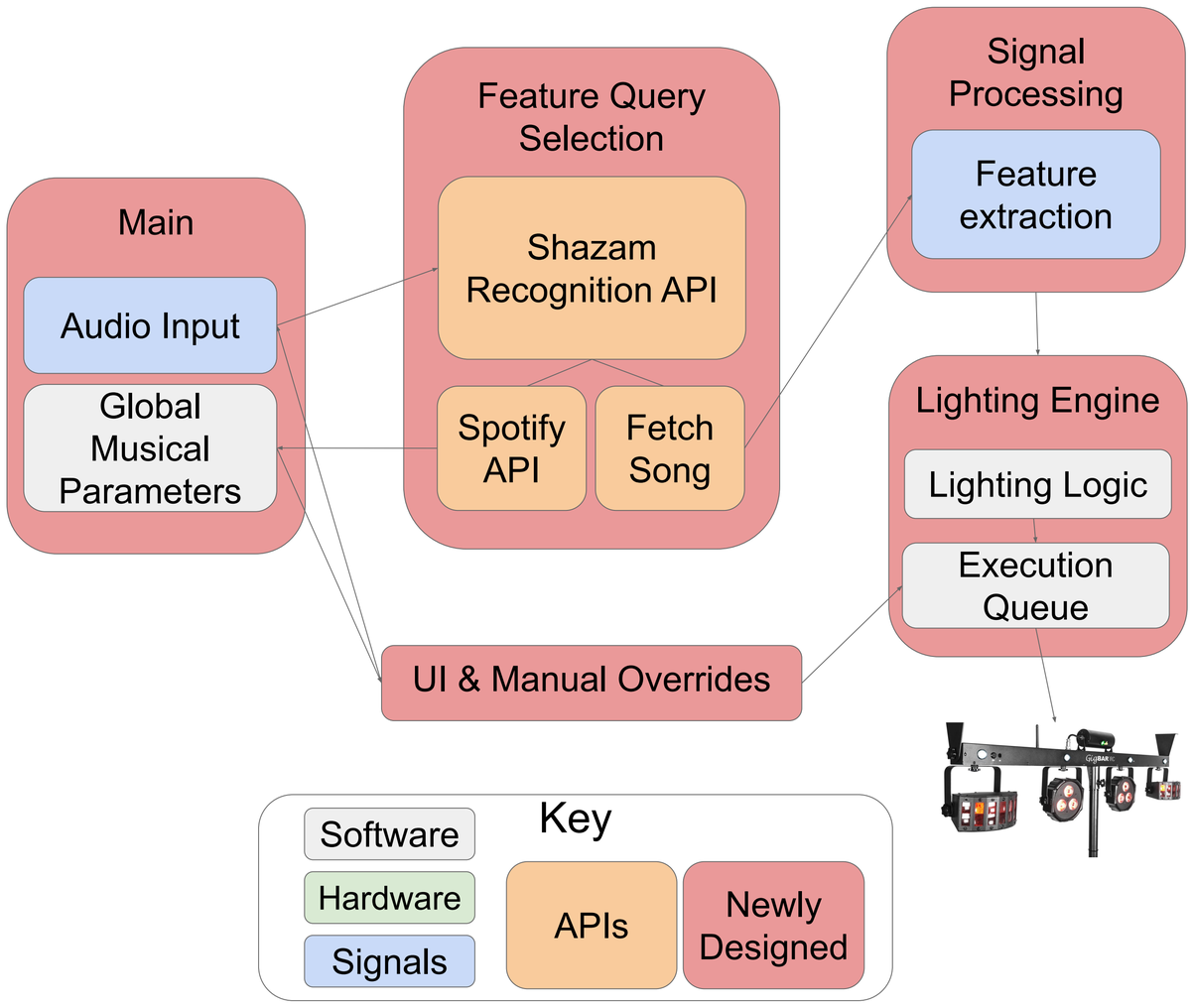
Stage lights are used to engage the audience during musical performances. Good light shows require extensive manual programming and lighting engineers, making it inaccessible to small performers. For this project, we created a dynamic lighting system that automates this process by decomposing audio inputs to control lights. The user can upload songs using the web application and the system then windows the audio to produce an automated light show, aligned with the beats. The user can manually overwrite automated execution to customize the show. Given this goal, we focused on 3 quantitative benchmarks:

- 🕙 **Setup Time:** Using the GigBar2, we were able to set up in **< 5 mins** for an hour-long performance
- ⏳ **Latency:** An average song is preprocessed in less than 10 seconds, allowing lights to function in **real-time** with the music
- 🎼 **Signal Processing Accuracy:** Able to **detect 100%** of the songs, and extract signal attributes with more than **90% accuracy**

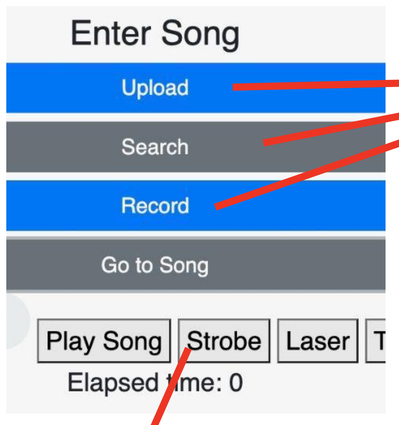
System Architecture

Execution Pipeline:

1. User plays/inputs a song and the relevant audio file is generated
2. Audio file is sent to the signal processor for decomposition and analysis
3. Lighting logic uses audio features to determine lighting behavior and generates function calls
4. Function calls are stored in a queue and executed by the lighting engine
5. The user can interrupt and overwrite the current execution with their own input, which is executed on the subsequent beat

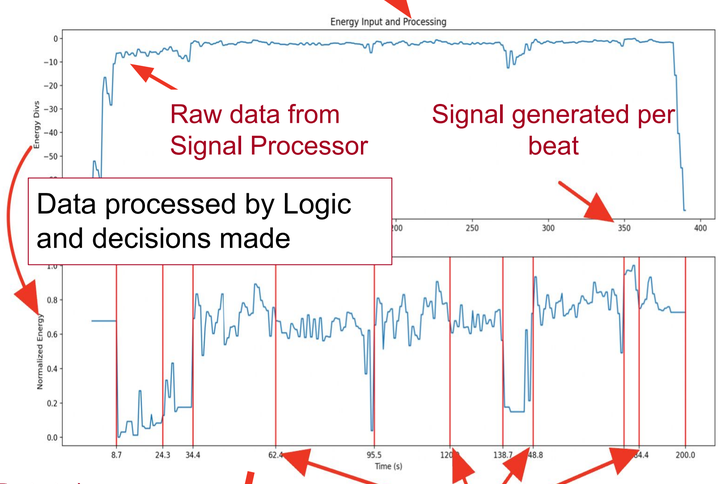


System Description



User overrides sent straight to the lights

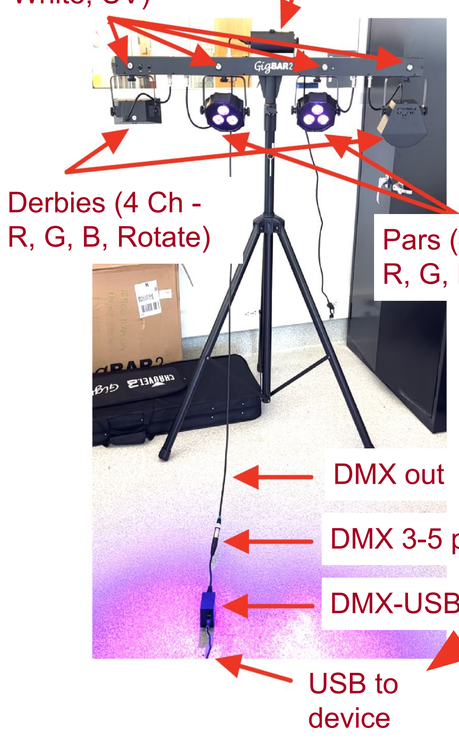
Song is detected and file is fetched, sent to signal processor



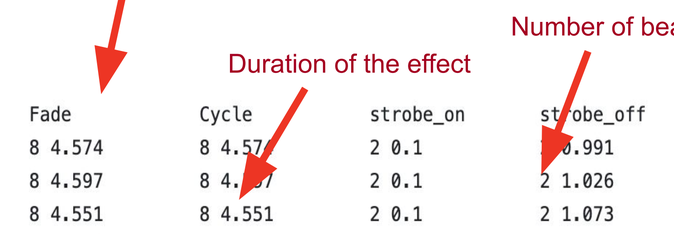
Change times for commands detected

Strobe (2 Ch - White, UV)

Laser (3 Ch - R, G, Rotate)



Commands generated per beat.
Ex: Effect times for first 3 beats



Commands sent to the light from the device for each individual light.

Multiple commands and overrides are handled using threads

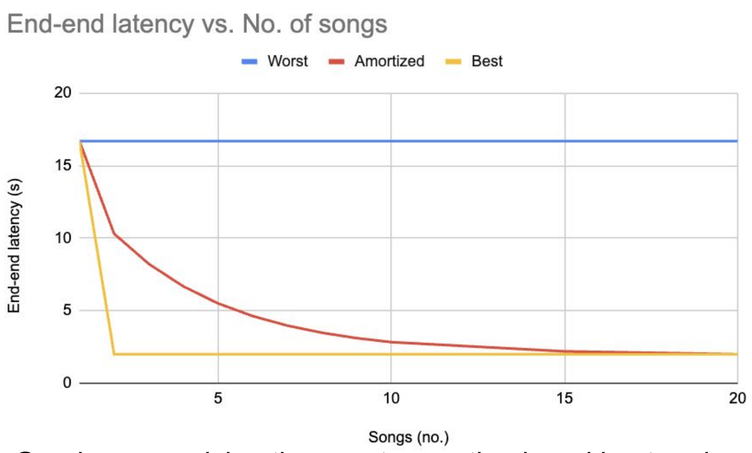
System Evaluation

Song	Change Points	Accuracy
Teenage Dream	7	100 %
Jiya Jale	13	82 %
Flowers	11	100 %
Rap God	17	80 %
Dandelions	14	95 %
Pepas	9	96 %
	Average	**93 %**

Table summarizing accuracy of feature detection as compared to hand label audio waveforms, tested across songs of different genres.

Table summarizing 4 qualitative metrics of the system as averages measured across 10 users

Metrics	Avg Rating (out of 5)
Setup time	5
Recommendation effectiveness	4.1
User Interface Intuitiveness	3.5
Aesthetics	4.7
Average rating	**4.3**



Graph summarizing the worst, amortized, and best end to end latencies over a period of 20 songs

Stretch goals

1. Train a model to detect choruses better
2. Reduce preprocessing latency
3. Allow additional customization parameters
4. Allow users to re-adjust weights on different signal parameters

Conclusions & Additional Information

Originally, we planned to do real-time processing of microphone input as opposed to preprocessing audio. However, that reduced the accuracy of the system and increased the latency to over 100 ms per beat. Switching to a preprocessing system with a recommendation engine, we were able to meet all our use-case requirements with **real time playback,** a **~90% accuracy**, and an amortized latency of ~2 seconds, making our project successful.

Over the course of the project, we were able to apply our technical knowledge from classes in signals, web applications, and statistics. We also learned a lot about software development and testing. In the future, we hope to see this project extended to real time inputs and more granular signal processing. We expect the project being used by musical performers across the world, engaging different audiences with captivating light shows.

Electrical & Computer ENGINEERING
Carnegie Mellon