

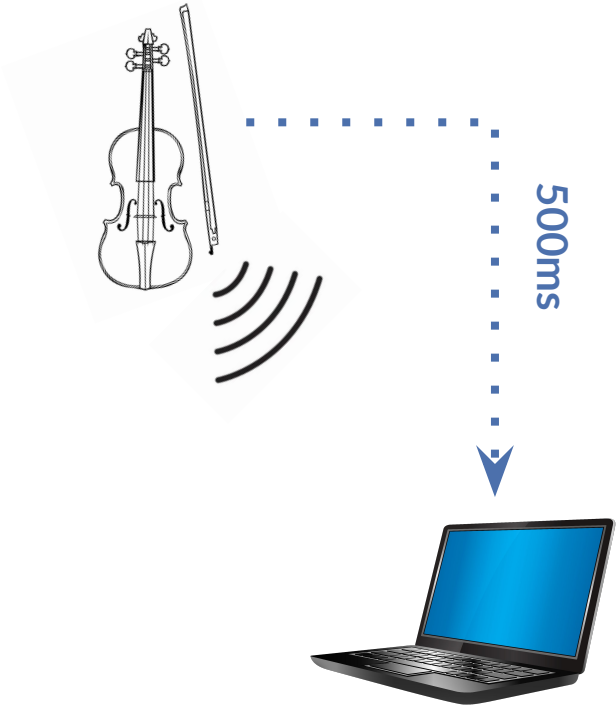
SoundSync



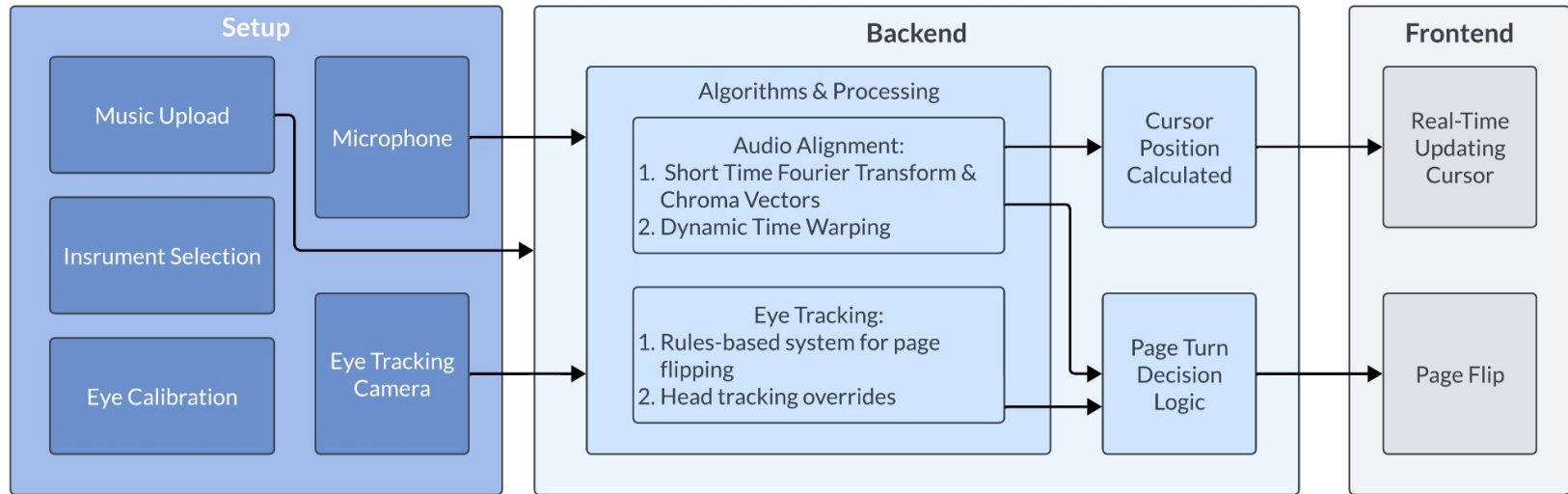
Caleb Lille, Rohan Raavi, Sanjana Shriram



Use Case Requirements

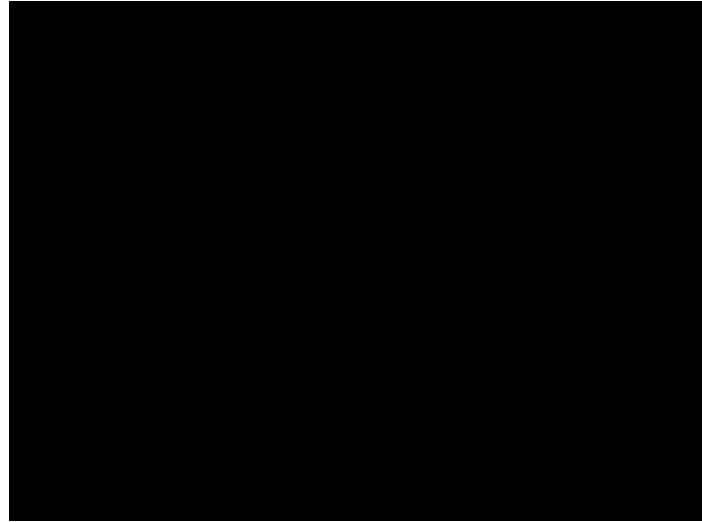
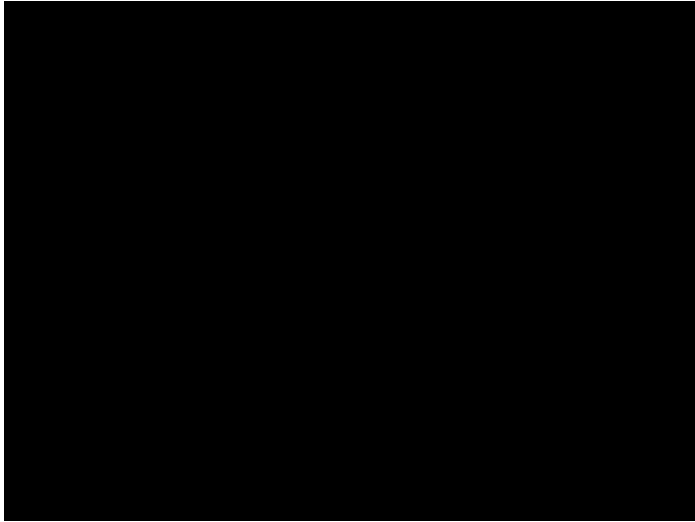


Solution Approach + Changes



Complete Solution

(Click on the black squares below to view our video demos!)



Quantitative Testing Plan

Environment

Quiet room without metronome/percussive noises
Noisy room with percussive noises

Backend Tests
Latency & Accuracy

Backend functions are tested 2x separately for accuracy with the following tempos: 60, 90, 120, 150, 180, 210, 240 BPM

Integration Tests
Page Flipping & Cursor

Use integrated system with eye tracking and head tracking and measure latency and page flipping rates.

Quantitative Tests (1)

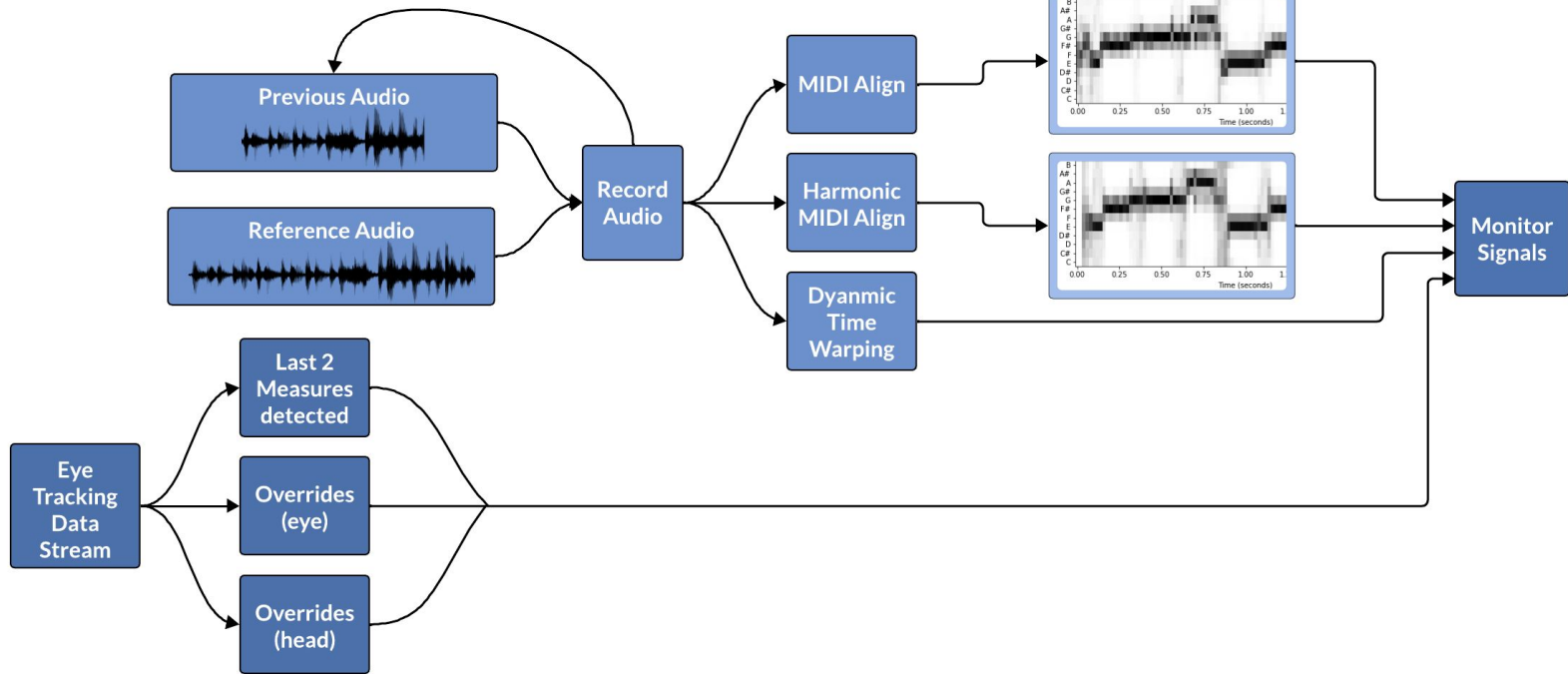
Metric	Test	Target	Actual	Met
Backend (Eye) Latency	Measure how often the front end reads from the backend in order to update information	< 500 ms	< 500 ms, variable	✓
Page Flipping Accuracy	Play through music and determine if page flips occur within 2 bars of the end of the page	> 95%	90%	✗
Visual Model Improvement	Measure how much page flipping accuracy improves with eye tracking and head tracking	> 5%	10%	✓
Eye Tracking Accuracy	Measure variation in data points	< 4 cm	< 1 cm	✓
Eye Tracking Precision	Measure variation in data points	< 1.5 cm	< 1 cm	✓

Quantitative Tests (2)

Metric	Test	Target	Actual	Met
Backend (Audio) Latency	Play portions of two octave scale and ensure the audio aligns	< 500 ms	158 ms	✓
Audio Alignment	Measure if cursor is within 1 bar of current position in music	1 Bar	> 1 Bar	✗
Audio Robustness	Track alignment for intermediate missed notes played during piece given correct beginning and end notes.	10%	up to 50%	✓
Eye Tracking Robustness	Eyes moving off screen, eyes looking at least few measures randomly, erratic eye movements, etc	no false positives	no false positives	✓

Tradeoffs

- Google Coral Dev Board vs Laptop
 - OS compatibility issues with Python libraries
 - Google Board Edge TPU no longer needed
- Speed vs Accuracy
 - Speed → real time application
 - Accuracy → design requirement
 - Robustness of DTW comes at the cost of speed



Speed vs Accuracy

Algorithm	Latency	Accuracy
MIDI Align	158 ms	Not robust with incorrect notes, supports missed notes
MIDI Harmonic Align	763 ms	Filters percussive noise, not robust with incorrect notes, supports missed notes
Dynamic Time Warping	~ 500-1000 ms	Very robust, needs longer note sequences

Technical Challenges

Audio

- Make MIDI Align algorithm more robust

Eye Tracking

- Dynamically reducing confidence of glances

Integration

- Integrating multiple confidence levels from both systems
- Bayesian integration

Project Management

