Conducting a Robotic Musician

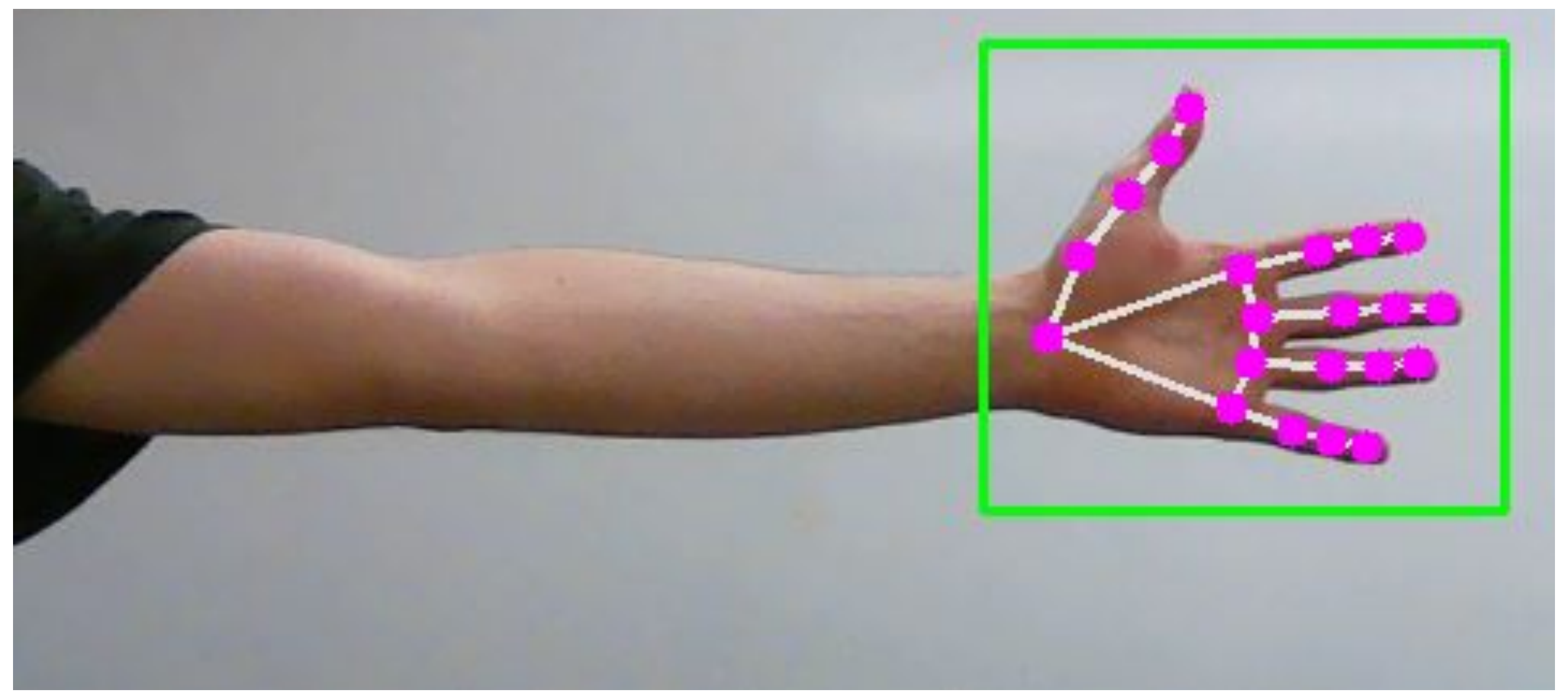
Will Sullivan, David Allen, William Barr

Introduction & Goals

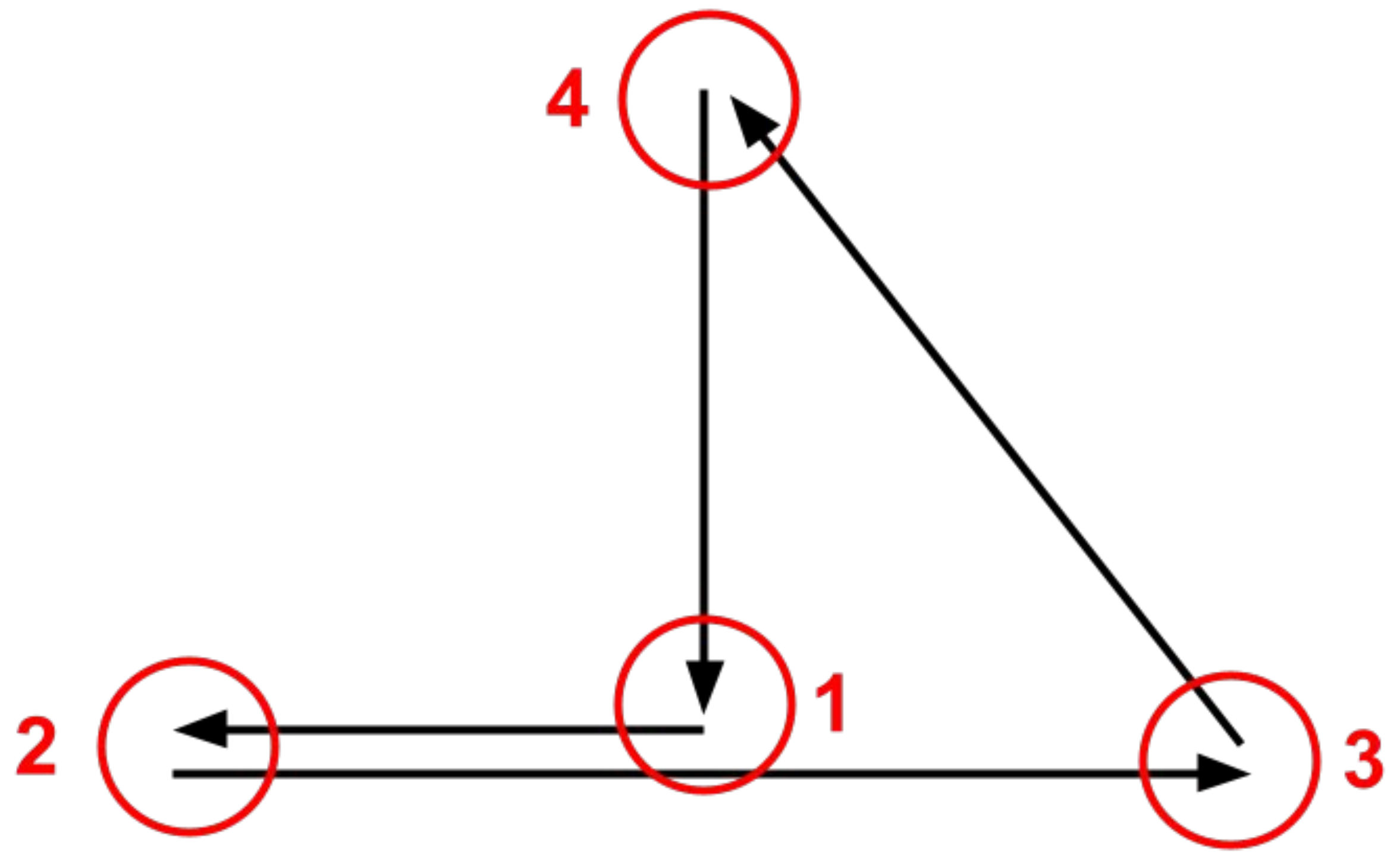
- The primary goal for our project is to enable a robot to track a conductor's motions and recognize hand gestures in real time, as well as synchronize the tempo of synthesized music with those gestures.

Gesture Recognition

- Gesture recognition and finger position
- OpenCV video processing
  - We utilized OpenCV, an industry standard computer vision python library. In our project, it is used to show the video feed and to draw the visual elements of the GUI.
- Mediaipe hand tracking
  - We used MediaPipe's pre-trained neural networks to track the locations of the joints in the hand in real time.
  - This gave us real-time pixel location values of the user's finger, which let us calculate the velocity of their finger

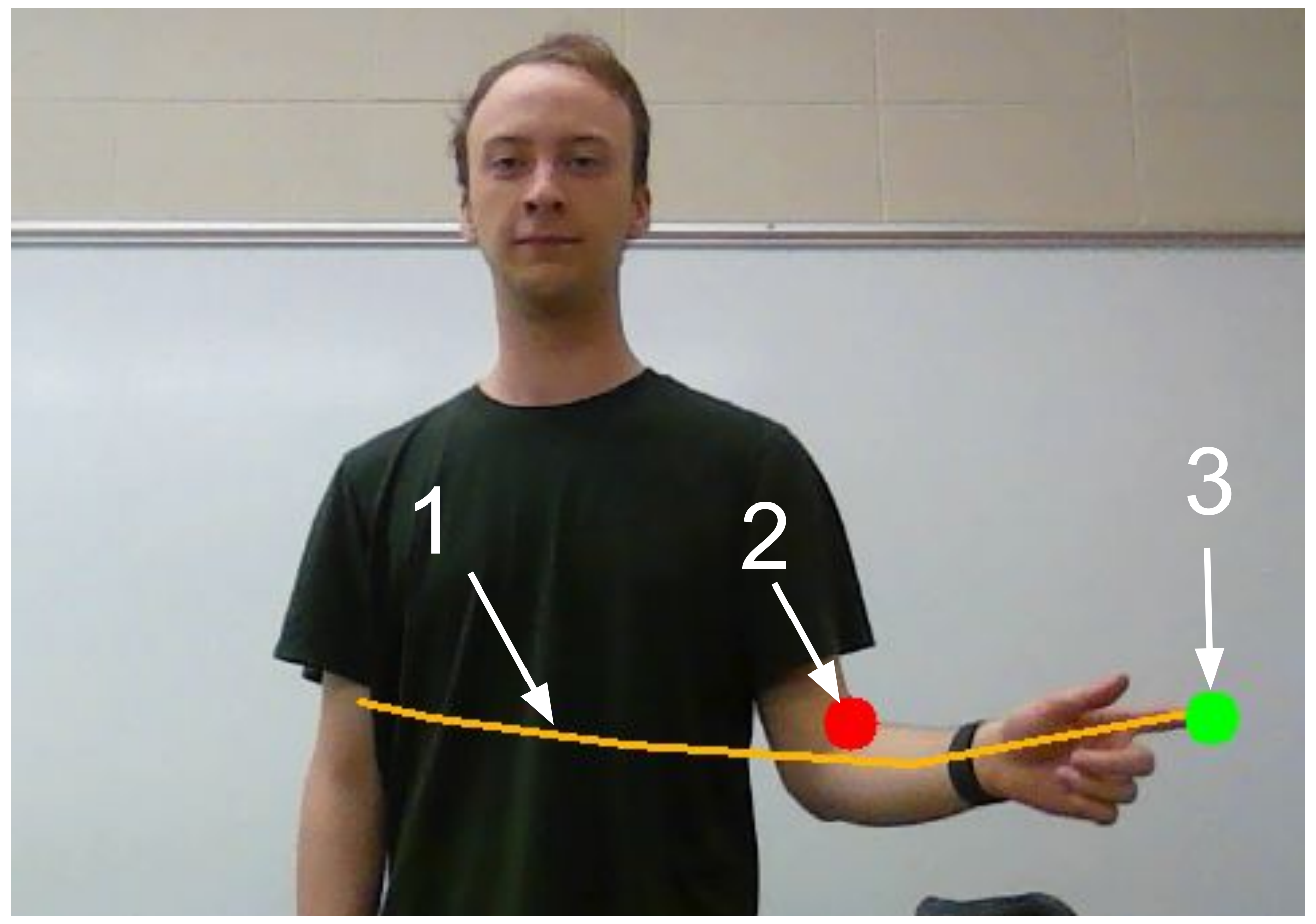


- The program waits for the user's finger to both exceed a predetermined velocity threshold and change directions in the same way specified by the pattern of 4/4 conducting



Error Correction

- Positional and velocity smoothing
  - The finger tracking was jittery, so we implemented a smoothing mechanism by calculating a rolling average of the last locations of the finger.
  - We used the same rolling average algorithm to smooth the velocity, and used that smoothed velocity to algorithmically detect beats.
- Debouncing
  - We added a 300 ms debounce period to prevent beats from immediately coming after one another



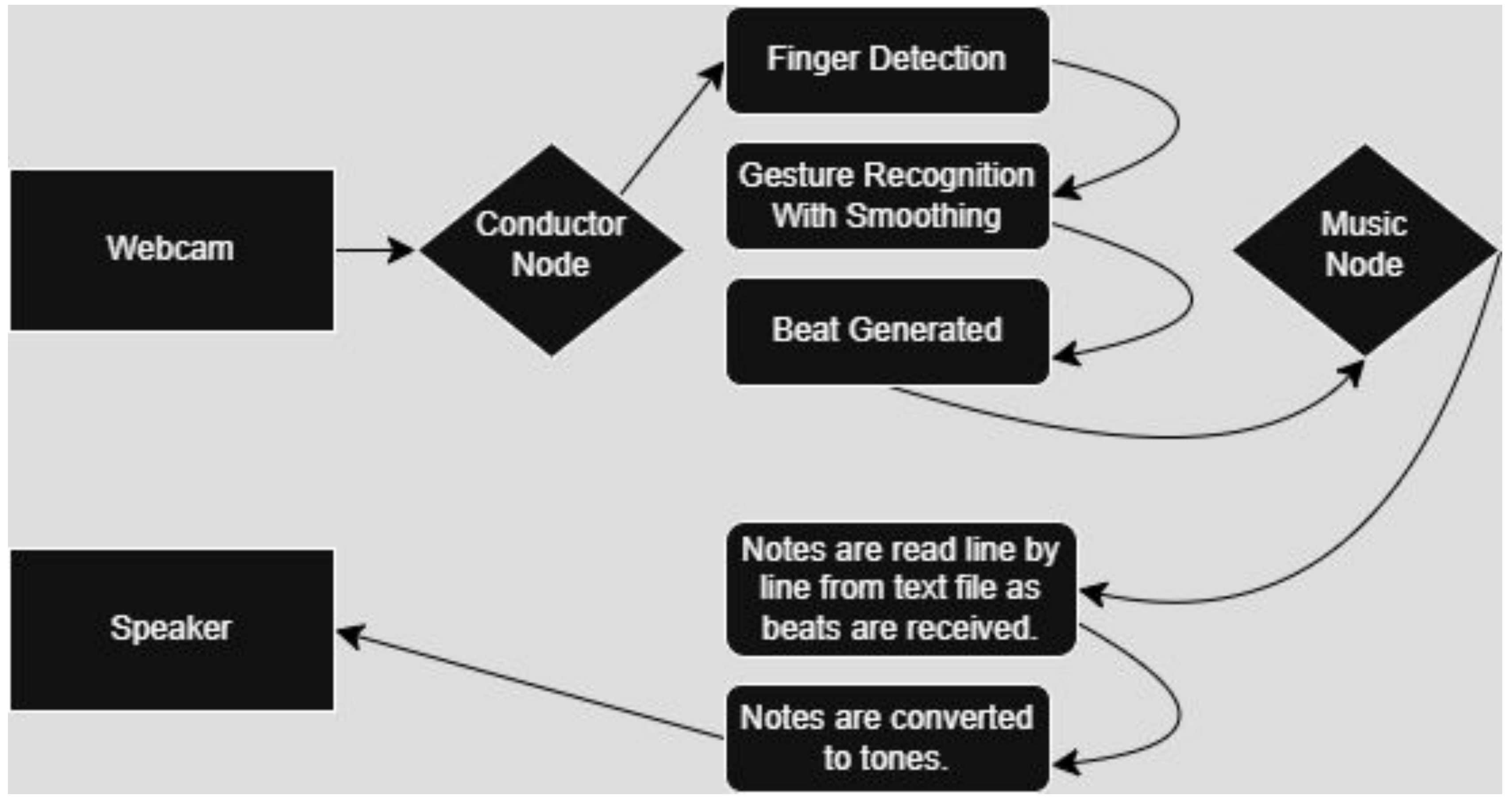
1. Index finger position history
2. Smoothed index finger position
3. Current index finger position

Usage

Upon Program Start:

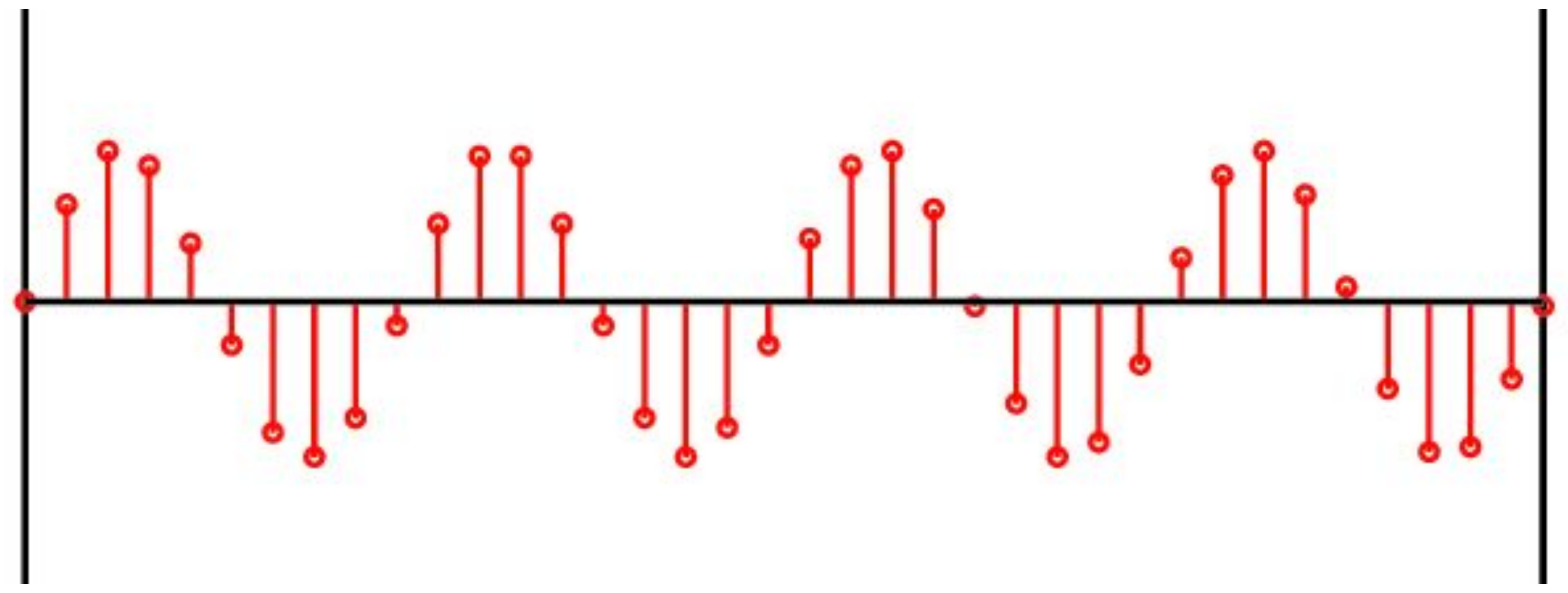
- The user positions himself within view of the camera display.
- Our program searches the video feed for the presence of hands and will then seek to identify the pointer finger of the right hand.
- The conductor(user) then proceeds to begin moving his hand in the style of a conductor giving tempo. The gesture consists of the movements down, left, right, up (as pictured to the left.).
- Our musician is trained to recognize these gestures and will generate music of the appropriate tempo.

Project Structure


Webcam
Conductor Node
Finger Detection
Gesture Recognition With Smoothing
Beat Generated
Music Node
Notes are read line by line from text file as beats are received.
Notes are converted to tones.
Speaker

Music Production

- We used pygame and the University of Princeton's audio playback library to manually synthesize sound samples from mathematical functions.
- Our code takes in a pitch and a duration and converts them to hertz and sample count respectively
  - This allows us to generate sine wave samples for the correct duration upon the user's command



Future Work

- Further implementation of expressive gesture recognition
- Further smoothing and increasing responsiveness of gesture recognition
- Deep learning can be used for gesture recognition instead of algorithmic computer vision