Project Proposal: Virtual Conducting

Andy Lai Lin and Brandon Yoshimoto

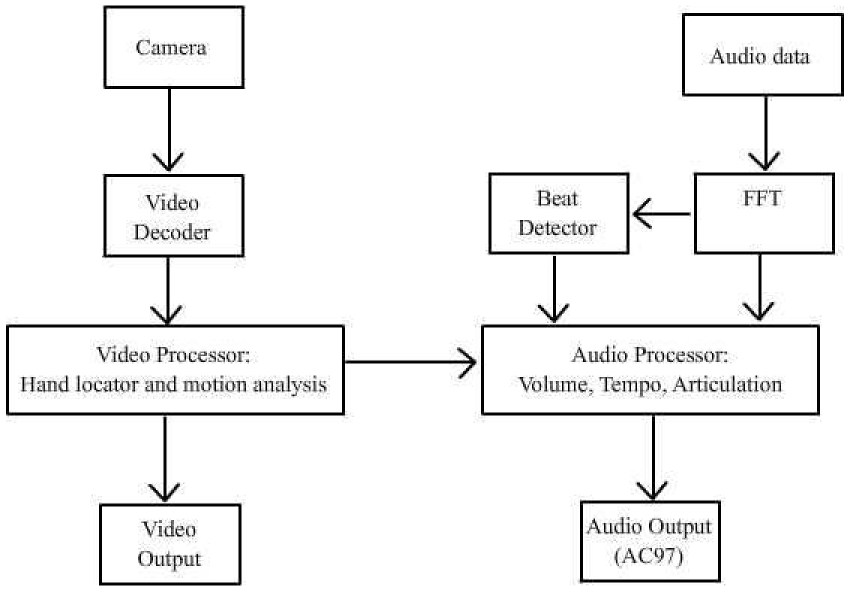
Introduction:

The purpose of this project is to design and implement an interactive music player which allows the user to control the sound of a composition through hand movements. The idea is to emulate the experience of a conductor directing the flow of a musical performance.

The design will use a camera to detect hand movement of the user and adjust musical qualities such as tempo, dynamics, and articulation in response to these motions. The camera will detect the user's hands on the screen. One hand will control the treble, while the other will control the bass of the piece. Volume of each part will be controlled by the size of the corresponding hand's movement.

The frequency of the hand motion will control the tempo of the playback. Similarly, the acceleration of the hand movement will dictate the articulation of the music: more acceleration corresponds to more pronounced articulation such as staccato while less acceleration corresponds to smoother articulation such as legato. Additionally, the design will include a screen which will display a visualization of the hand movements, following the path of the conductor's hands. The screen will also display useful information such as the current tempo, volume, and articulation to provide feedback for the user.

Implementation:


Camera
Audio data
Video Decoder
Beat Detector
FFT
Video Processor: Hand locator and motion analysis
Audio Processor: Volume, Tempo, Articulation
Video Output
Audio Output (AC97)

Camera:

The user will stand in front of the camera wearing gloves of a distinct color from the rest of the background. For the design, the user must stand such that each hand is in a different half of the screen. The camera will output frame data capturing the hand movements for processing in the Video Decoder.

Video Decoder Module:

This module will decode the stream of data from the camera into pixel information.

Video Processing Module:

Hand locator:

This part will determine the position of the two hands on screen. Firstly, it will store the information from the Video Decoder in a ZBT. The rate at which it stores new frames will depend on the time required for processing information on the position of the gloves.

Next, the module will determine the position of the gloves based on the information stored in the ZBT. Finding the position will require detecting where the data matches the expected color of the gloves. Having obtained two areas of the desired color, the coordinates assigned to each hand will be determined by taking the center of mass in both the x and y directions for each hand.

Once these coordinates are obtained, they will be stored in BRAM1 as binary coordinates. Successive frames will fill BRAM1 with these coordinates over time. The coordinates will continue to load into BRAM1 until a "beat marker" is detected. Beat markers correspond to the momentary pause of the user's right hand when making a sudden change of direction. In effect, it marks when the user ends a beat stroke. In our design, a beat marker occurs when the position of a coordinate of the right hand stays stationary within a certain bound for longer than x samples.

Once the module detects a beat marker, it will store the address and coordinates of the right and left hands in BRAM2, then clear BRAM1 and wait for a signal to start receiving new coordinates. For indicating the start of a new beat, the module will check for sufficient acceleration in the right hand's position values. This will signal when to start inputting new coordinates into BRAM1.

Motion analysis:

Motion analysis will use the beat marker information to calculate frequency, acceleration, and amplitude of the motion over time. The frequency of movement is calculated by examining the address of a particular beat marker stored in BRAM2. If the address is high, the frequency of the beat stroke is low. If the address is low, the frequency is high.

The amplitude of the motion will be calculated as the length of the segment between two successive beat marker coordinates in BRAM2. Since there are two hands - one for treble, the other for base - amplitude information will be calculated for each hand separately.

Acceleration will be calculated as the second difference between the samples at the start of a new beat in the right hand. If the second difference over a certain number of samples is high, the acceleration output will be high. This scaling will be used in the audio processing module for articulation modulation.

In summary, two motion qualities – frequency and acceleration – will be determined by the right hand only. This means the user controls tempo and articulation of the music with just the right hand. Amplitude information, on the other hand, will output two values, one for each hand, so they can be used to change the volume of the bass and treble separately.

Visualization Generation:

This module creates a visualization on the monitor tracing the motion of the batons. Possible effects include a fading trail behind the current position or a display of the frequency distribution from the audio FFT output.

FFT Module:

This module will be built from the FFT module provided from Xilinx. It will take in the audio signal and output its Fast Fourier Transform. Using filters, this module will divide the audio signal into treble and the bass.

Beat Detector Module:

This module will detect the beat frequency (tempo) in an audio signal. It will detect large pulses in the low frequencies and output a beat indicator, marking the beginning of each beat. The beat information is important for adjusting the tempo and articulation of the audio samples in the Audio Processing Module.

Audio Processing Module:

This module will control its submodules, which will consist of the Volume Modulator, Tempo Modulator, and Articulation Modulator. The Audio Processing Module as a whole will take parameters for volume, tempo, and articulation as inputs and change the original audio signal into a modified audio signal. The audio processing module will also use the beat indicators from the Beat Detector Module to feed in audio samples one beat at a time. The resulting modified audio signal will then be fed into the AC97.

Volume Modulator:

This module takes an audio signal and coefficient as inputs and outputs a signal with scaled amplitude, given the coefficient. The computation required will consist of multiplication.

Tempo Modulator:

This module takes an audio signal, original tempo, and desired tempo (beat frequency) as the input, and outputs a slower or faster signal based on the desired tempo input. The tempo modulation will be performed using the exclusion or interpolation of data points. If the desired tempo is greater than the original tempo, the module will exclude a certain number of points to achieve the desired tempo. On the other hand, if the desired tempo is less than the original tempo, the module will interpolate points between known audio samples to achieve the slower tempo.

Articulation Modulator:

This module takes an audio signal and acceleration data and outputs a modified signal with different articulation. The faster the acceleration, the more pronounced the articulation; for example, a large acceleration corresponds to *staccato*, while a smaller acceleration corresponds to *legato*. The articulation modulation will be performed using multiplication with a set of coefficients within one beat. The more smooth the articulation, the smoother the coefficient function. For example, if the articulation is smooth, the signal is multiplied with a coefficient function that is almost constant. On the other hand, if the articulation is pronounced, the audio signal is multiplied with a coefficient function that rises initially, then drops quickly.

Testing Strategy:

For the video, we need to test if values are being decoded correctly in the Video Decoder module. This can be done by holding patches of different colors up to the camera then checking if the frame's information is correct. Similarly, we can test the calculation of the center of mass for the different hands by holding patches of the desired color up to the camera and checking if the center is correctly calculated.

We can test for correct detection of beat strokes by moving the color patch back and forth, outputting the value of the current address and seeing if it resets to zero upon pausing movement. All this can be done on the Logic Analyzer. Even before this, however, it's necessary to test all these operations in the Modelsim simulator for ideal inputs before moving to the hardware.

For obtaining frequency and amplitude information, we can again use Modelsim and the Logic Analyzer for matching actual and desired outputs. We can produce known outputs using Matlab and compare the signals computed in Matlab to the actual signals. The beat detector can be tested by feeding in simple signals and viewing the resulting "beat indicators" under simulation and on the Logic Analyzer. The complexity of the test signals can be increased until we arrive at the final signal.

Division of Work:

Brandon will work primarily on the video decoder, video processing, and visualization modules. Andy will work on the FFT and audio processing modules, while also contributing to the video processing module.