Phantom Sight Reader

Dilini W, Lance Collins, Jing Han

Abstract

"Phantom Sight Reader" is the future of music playing. It captures an image of sheet music from an external camera and produces an audio output matching the sheet music. It is comprised of three components: image capture, note recognition, and audio production. Image capture involves interfacing a camera with the FPGA and capturing a still image to memory. Note recognition involves identification of whole notes, half notes and quarter notes and the position on the treble clef in one octave. Audio generation involves generating tones similar to those that would be played on a real instrument. The final goal of the project is to be able to recognize notes from regular sheet music and produce tones as if it were played on a piano.

Work distribution

Lance will be in charge of the Player module and the Audio Processor. Dilini will be in charge of the Note Decoder module and Master FSM. Jing will be in charge of the Converter Box, the ZBT Memory module, and the Video Display module.

Y, Cr, Cb module

Input: camera image (bit stream)
Output: extracted Y, Cr, Cb values

Converts bitstream image from camera to Y, Cr, Cb values from an look up table (LUT). We will be using a pre-written module for this.

RGB module:

Input: Y, Cr, Cb values
Output: RGB values

Converts Y, Cr, Cb values to RGB values from an LUT.

ZBT Memory module:

Input: HSL values, enable
Output: HSL values

The ZBT memory will store the HSL values of the image pixels before passing a complete image in HSL to the note detector module for recognition. The ZBT memory has 512K x 36, more than enough to store a full page of pixels, which would be obtained by a camera with a 480 x 320 resolution.

Video Display module:

Input: recognized notes from the note detector (16-bit bit stream)
Output: video display of notes

The video display will function as both a testing method and a user interface. Initially, the simple digital number display on the labkit can be used to test whether the notes are being correctly identified. The final goal is to have a user interface that is both easy to use and visually pleasing. Capabilities will include: a GUI for adjusting the volume, selecting which instrument to play, and music notes that display on a staff as the music is being played.

Memory Array module:

Input: 16 bit stream from note detector module
Output: 16 bit stream to the player module

The memory array will serve as a storage for the notes before they are played. The size of the memory array will be 16 x 128.

Note Decoder module:

Input: HSL values from the ZBT memory module
Output: [15:0] bit stream into the Memory Array, [15:0] bit stream to the digital display module

The Note Decoder module undertakes the task of identifying the individual notes in the music score sheet and determining the duration of each note. The input into this module is the HSL values from the ZBT memory module.

The Note Decoder module comprises of three functions: The staff recognition, the frequency identifier and the beat identifier. The staff recognition function will determine where the staffs in the music score sheet are located. This is done by scanning the image from top to bottom and determining where black and white pixels are located. Once the top and bottom of each staff is determined the individual lines are differentiated and given individual "y" values: y1, y2, y3, y4, y5 for each line in a staff. This method will place a certain restriction on where notes can be located on a particular staff. We assume notes can only be between the top and bottom staff lines. This constraint will make it is easier to detect one staff from the other because there will be a considerable amount of white pixels between them.

The frequency identifier's task is to determine the individual notes. By scanning row wise between each "y" value, each note will be identified by determining whether the majority of the black pixels are between two consecutive "y" values or between two consecutive spaces. We will assume at the beginning that there are only individual notes present at one given time (no chords).

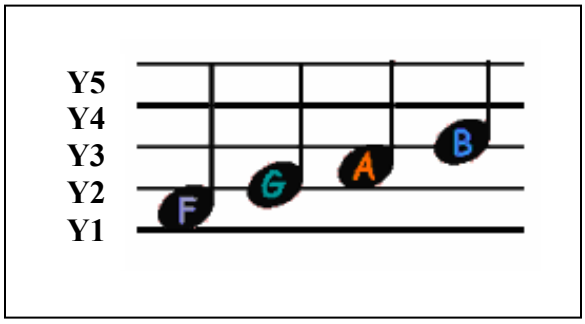

Figure 1: Notes on a staff

As shown in the figure 1 above, the note F has a greater amount of black pixels in between the Y1 and Y2 values. Likewise, the note B has a greater amount of black pixels between Y2 and Y4 lines instead of Y2 and Y3 values.

The beat identifier performs the task of differentiating between a whole note (semibreve), a half note (minim) and a quarter note (crotchet). This is performed by dividing each staff in to equal columns and evaluating the black pixel concentration in each column. Here we assume that the notes are evenly spaced. This can be achieved by printing the notes using a computer. This method is explained as follows:

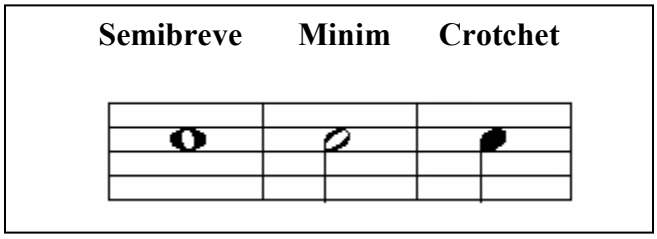

Figure 2: Beats of notes

As seen in figure 2, the crotchet has the greater amount of black pixels and the minim has the second highest black pixel count and the semibreve has the smallest amount of black pixels.

The output from the Note Decoder module is a stream of bits: 16 bits for each note. This information will be stored in a memory array to be used by the Player module and will also be used by the digital display module.

Master Finite State Machine (Master FSM):

Input/Output: enable signal from/to various modules

The Master FSM module undertakes the task of integrating all the modules in the system. This FSM comprises of three states: The IMAGE_CAPTURE state: DECODER state and the PLAY state. The integration of the modules is performed by the Master FSM by using enable signals. The Master FSM is in the IMAGE_CAPTURE state when the system is powered on. Once it receives a "0" for the enable_zbt it transitions from the IMAGE_CAPTURE state to the DECODER state by setting the enable_nd high. Once the enable_nd is low the Master FSM transitions to the third state: PLAY by setting the enable_play high. Once the enable_play is low the Master FSM transition back to the IMAGE_CAPTURE state.

Audio Processor:

Player

I. Parameters: None
II. Inputs: Note Durations (14 bit value)
III. Outputs: Note Enable Signals (7 bit value)

NoteBox

IV. Parameters: Note Frequency (21 bits)
V. Inputs: Note Enable Signal (1 bit)
VI. Outputs: Tone (18 bits)

Wave Generator

I. Parameters: Note Frequency (21 bits)
II. Inputs: Note Enable Signal (1 bit)
III. Outputs: Unmodulated Tone (18 bits)

Amplitude Modulator

I. Parameters: ADSR Envelope (20 bits)
II. Inputs: Note Enable Signal (1 bit)
III. Outputs: Tone (18 bits)

Audio generation in our project is divided into two main modules: the player and audio synthesizer. The player takes the information from the ZBT memory and sends the notes to the audio synthesizer at the correct timing intervals. For each beat, there is a 14-bit value in the ZBT (2 bits for each note of 7 notes). The first 2 bits correspond to A; the second 2 bits correspond to B, and so on. The 2 bits indicate whether the note was pressed during that beat the duration of the note as specified in the table below.

2-bit value	Duration
00	NOT PLAYED
01	Quarter Note (1 beat)
10	Half Note (2 beats)
11	Whole Note (4 beats)

The player generates a 7 bit signal which tells the synthesizer which notes should remain played during that beat. The audio synthesizer is composed of 7 note generators. The note generators each produce values for the note they represent and all these values are combined using an adder and sent to the audio output. Each note generator is comprised of a wave generator and an amplitude modulator. When a note is played, the wave generator begins generating the corresponding values for the wave based on the note's preprogrammed frequency. These values are scaled based on the amplitude modulators output. The amplitude modulator varies the amplitude according to the timing specifications for piano tones to ensure the note sounds like a real piano.

Testing and Debugging

The image capture will be tested by taking an image of a simple image (such as a black square on a white piece of paper) and displaying HSL values on the monitor. The Note Decoder module will be tested by observing the notes displayed on the digital number display of the labkit and comparing them with the notes on the scoresheet. The Audio Processor will be tested with user-generated tones that can be selected using switches or buttons.

Block Diagram: