PerfectPitch

Grace Cheung Karl Rieb

Massachusetts Institute of Technology
Department of Electrical Engineering

November 3, 2007

Abstract

Fourier transforms and frequency domain analysis are being used to generate music scores from an audio sample. A Frequency Signature Algorithm has been developed that takes the result of applying a fast fourier transform (FFT) to the audio input and compares the frequencies with different frequency signatures to separate the audio by instrument. Frequency domain analysis is performed on the separated input to find peaks corresponding to the pitch in the audio sample. The frequency peaks are matched to their corresponding music notes, which are used to generate the music score.

Contents

List of Figures

1 Overview

The goal of *PerfectPitch* is to allow users the ability to create sheet music for their favorite songs. The system takes in the headphone output of any music player device and converts it into analog line input that is fed into an audio codec. The audio codec amplifies the signal and converts it to digital. Once the audio sample is in digital format, it is stored in memory and used by the fast fourier transform (FFT) module for analysis in the frequency domain. Peaks in the signal are then detected and compared with a pre-determined set of music instrument frequency signatures stored in an internal look-up table. The comparison allows the system to determine which instruments are playing in the audio sample. Once the instruments are determined, a filtered set of peaks are sent to a note look-up table which matches the peak with its corresponding musical note.

The notes and instruments are sent to a timing module that keeps a tempo for the song and calculates the duration of the notes in the audio sample. After the note, instrument, and duration are determined, this information is sent to the video modules.

The video modules display staves and notes as they are processed by the system. The notes are displayed in different colors depending on which instrument has produced the sound. The shape of the note is determined by a look-up table that contains a set of sprites associated with different note durations.

2 Module Specification

The *PerfectPitch* system is comprised of three main components: audio input, signal processing, and video display. See **Figure 1**.

2.1 Audio Input (Grace Cheung)

The audio input module takes the analog output from a headphone jack and converts it to a digital audio signal that's used by the rest of the system. See **Figure 2**.

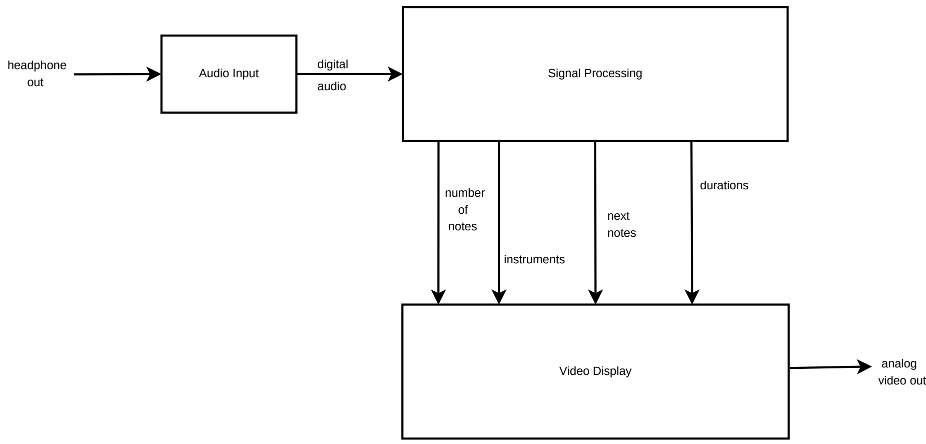

Figure 1: Top Level View of System Modules

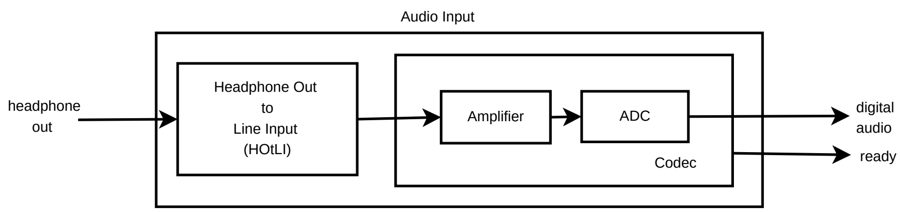

Figure 2: Audio Input Modules

2.1.1 Headphone Out to Line In (HOtLI)

The analog signal from a headphone output on most music playing devices is not compatible with the signal requires for the line in input. The HOtLI module transforms the signal appropriately before it is passed to the audio codec. The transformation is done completely outside of the digital system.

2.1.2 Amplifier

Amplifies the analog signal before it is converted to digital.

2.1.3 Analog-to-Digital Converter (ADC)

Takes an amplified analog signal and converts it to digital. This allows the remainder of the system to analyze the signal and generate the sheet music.

2.2 Signal Processing (Karl Rieb)

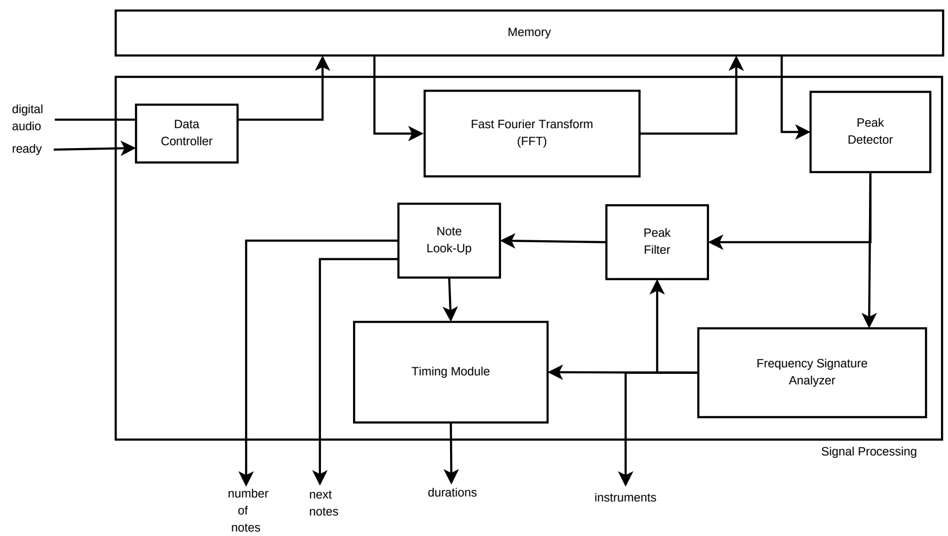

Figure 3: Signal Processing Modules

The signal processing module takes the digital audio signal from the audio input module and applies a fourier transform to analyze the signal in the frequency domain. A peak detecting module and frequency signature module determin the note frequencies and instruments in the audio sample. A look-up table is used to determine the note and the information is sent to the video display module to be displayed. See **Figure 3**.

2.2.1 Data Controller

The data controller module takes in the digital audio sample from the audio input module and a ready signal. When the ready signal is active, the digital audio sample is stored in memory for later use by the system.

2.2.2 Fast Fourier Transform (FFT)

The FFT module reads audio samples from memory and applies a fast fourier transform to them and saves the result (the signal in the frequency domain) into memory.

2.2.3 Peak Detector

The peak detector module reads the audio signal in the frequency domain from memory and attempts to locate all the frequency peaks above a certain peak detecting threshold. The threshold is adjustable and should be tailored appropriately for different types of music. Note that this module does not pick only one peak, but mutliple peaks.

2.2.4 Frequency Signature Analyzer

The frequency signature analyzer module takes peaks detected by the peak detector and compares them to a set of pre-defined instrument frequency signatures. Each instrument has different harmonics, and this can be used to determine what instruments are playing in the audio sample. The module outputs the instruments played.

2.2.5 Peak Filter

The peak filter module takes the peaks found by the peak detector, and filters them with a different threshold: filter threshold. The peak filter attempts to determine which notes were played in the audio sample by only outputing the peaks with the largest magnitudes. Depending on the type of music being sampled, the threshold may need to be adjusted appropriately. The peak filter also takes the instruments that are being played as input and matches them to the notes it finds.

2.2.6 Note Look-Up

The note look-up module is a table of music notes and their associated frequencies. The module takes the notes played and matches them to the correct music note.

2.2.7 Timing

The timing module takes in the instruments and the notes they play to determine the duration of the notes played. If different instruments play the same note, the module takes this into consideration and assigns appropriate durations for each instrument.

2.3 Video Display (Grace Cheung)

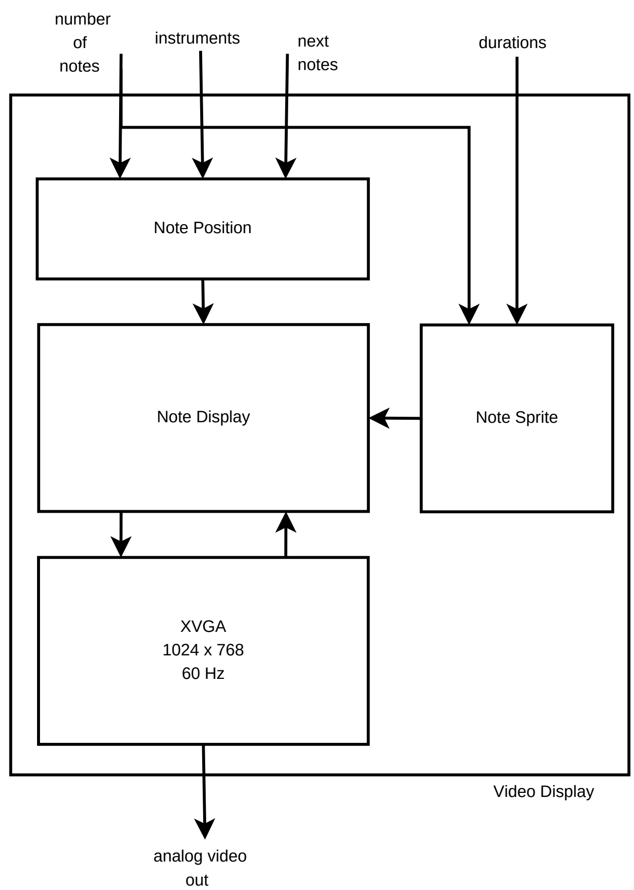

Figure 4: Video Display Modules

The video display module takes the digital signals on durations, notes, and instruments outputted by the signal processing module and converts this into sheet music on the display screen. The module is comprised of four submodules: note position, note sprite, note display, XVGA. See **Figure 4**.

2.3.1 Note Position

The note position module consists of an FSM that remembers where the last note was displayed, and determines where the next notes should be on the stave. This module takes in three inputs: next notes, instruments, and number of notes. It saves the last notes that were sent from the signal processing module, and determines where next notes should be in relation to these last notes. Number of notes specifies whether the next notes belong in a chord with the last notes, or just as single notes. It sends those coordinates to the display module, and also picks a unique pixel color for the determined musical instrument.

2.3.2 Note Sprite

The note sprite module stores all the sprite data for each type of note. Given a specified duration, this module produces a quarter note, half note, full note, etc. The number of notes input allows it to determine where to align the notes. This note information is then sent to the display module, which will draw the note appropriately.

2.3.3 Note Display

The note display module takes in the note coordinates and associated instrument color outputs of the note position module, as well as the note information from the note sprite module and displays the pixel outputs. Note display generates the appropriate notes at their respective coordinates with the correct pixel colors, depending on the type of instruments.

2.3.4 XVGA

The XVGA module takes in the clock input and outputs the hcount, vcount, hsync, and vsync signals to the note display module. The resolution and refresh rate are set at 1024x768 and 60Hz, respectively. The XVGA takes in the pixel output from the note display module, creates the image, and sends this to the monitor.

3 Testing

The system will be tested using audio samples with pre-existing music scores to compare the output of the system with the sheet music for the sample.

Since the thresholds used by the system are adjustable, multiple analyses on the same sample will be performed to find the best result. The result will be compared to the sheet music to determine accuracy given different musical compositions.

The minimal goal for *PerfectPitch* is to accurately analyze a monophonic audio sample with one instrument. The next steps are analyzing a monophonic audio sample with multiple instruments (tested using a synthesizer), a polyphonic audio sample with one instrument, and finally a polyphonic audio sample with multiple instruments.

4 Projected Costs

A summary of expenditures:

- Headphone out to line in (HOtLI) module may require the use of a transformer cable or built in-house using basic circuit components.
- Software synthesizer used for testing is free.
- All audio samples use for testing have already been purchased, however some sheet music will be required.
- Music player used to provide the audio analog samples has already been purchased, and thus incur no cost.

All other required materials are provided by the lab. The projected cost of the entire system is less than $100.