Virtual Piano

Proposal By:
Lisa Liu
Sheldon Trotman

November 5, 2013

I. Abstract:

Who says you need a piano or keyboard to play piano? For our final project, we plan to play and display music based on hand gestures that imitate piano playing. The project can be divided into three parts: video capture and processing, sound processing, and display.

Video capturing and processing is a large aspect of our project. We will use frame-by-frame comparison to determine if there is a significant enough difference in finger position. Individual fingers or sets of fingers will be tagged with colors to aid tracking and detection. This detection can be done in real-time to imitate the instantaneous reaction of a key press on a piano. Due to time constraints, it may be difficult to track 10 individual fingers, so primarily we aim to track 2 (perhaps to play Chopsticks), but ideally we would like to track all fingers.

At the most basic level, the Virtual Piano will play a pure tone based on what notes it sees. However, in real life, instruments do not sound like pure tones. Piano notes will be stored in memory and can be recalled and played when needed. In response to information about the speed of the hand, the volume of the note will increase or decrease. If time allows, we may adjust the attack and decay of the note as well.

Lastly, we would like to display the last few notes played on a staff on the computer screen. The note name should also accompany the notes. Given enough time, we can also display the length of each note.

II. Overview:

The Virtual Piano will need to detect gestures and output sounds and pictures based on those gestures. Figure 1 shows the basic subsystems we would like to design.

To detect a gesture, we will analyze a frame, store it in metadata storage, and compare it to the next frame in order to determine the change in position of a finger. A spatial analyzer will determine if a change in position represents a pressed key.

Once we determine that a key is pressed, we must display the notes and output a sound.

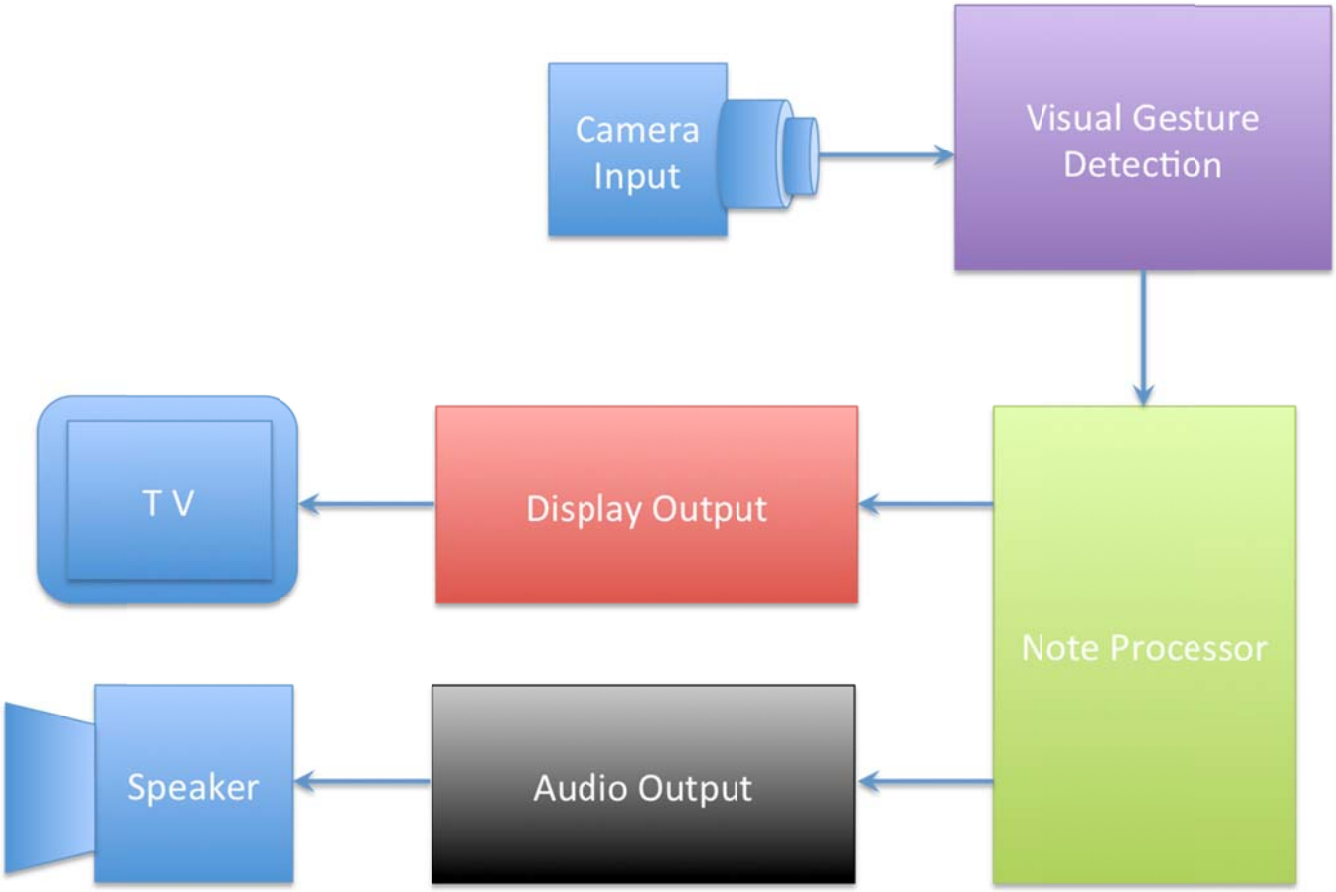

Figure 1. **General Block Diagram:** Graphical explanation of project design. We take in a camera input and after visual and logic processing we output display and audio data.

III. Sub-Modules:

Shown below is Figure 2, which illustrates the low-level implementation strategy used in the Virtual Piano.

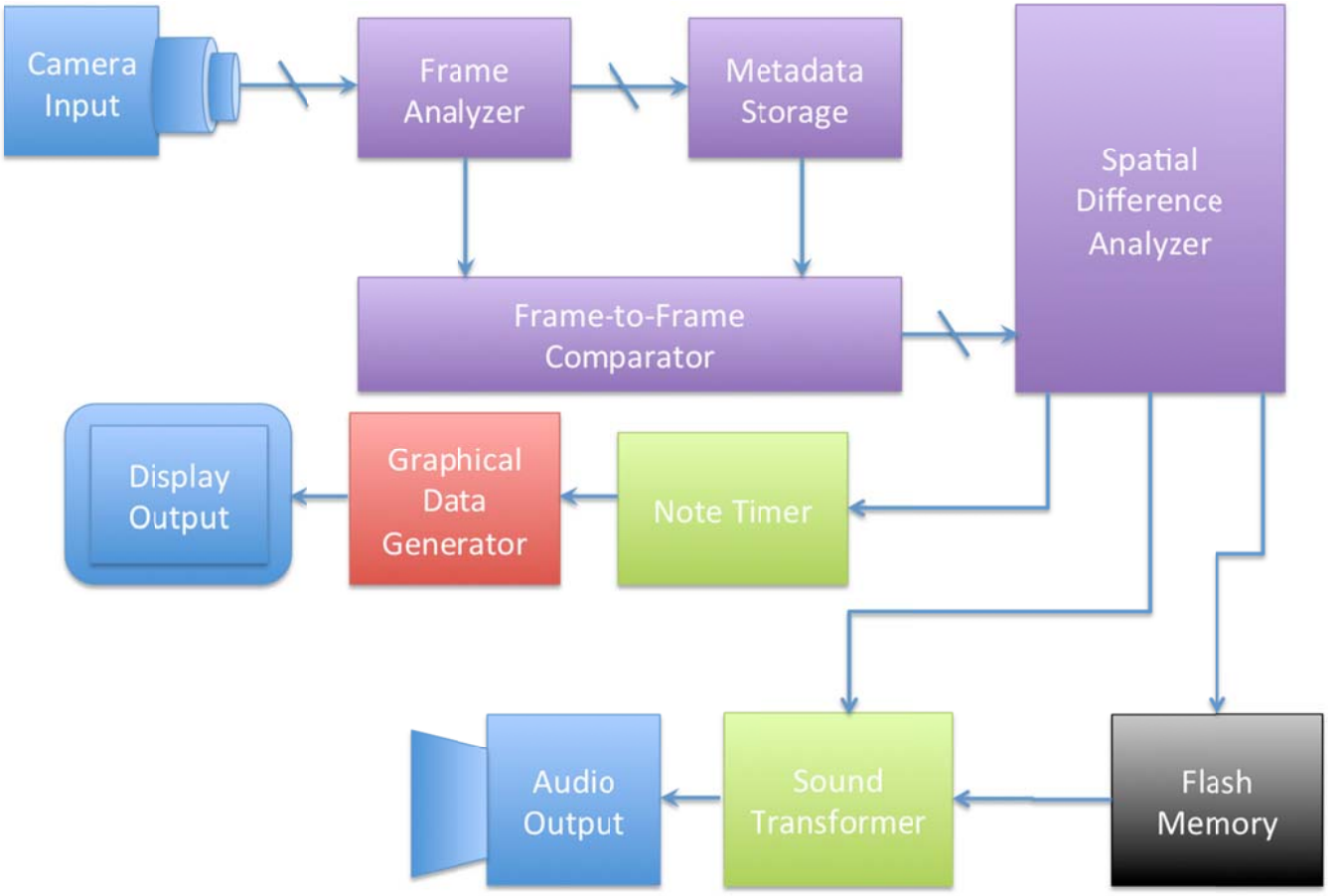

Figure 2. **Specific-View Block Diagram:** The purple blocks represent sub modules of the visual gesture system. The green modules represent the note processor. Black represents memory. Lastly, the red represents the graphical data generator.

Camera Input:

Raw data received from user interaction in the frame of the camera. This data is sent to the frame analyzer.

Frame Analyzer:

Here, detection of the specific colors used for the fingertips is done and passed on to be stored, as well as compared to previous data. The analyzer is one of the most complex portions of this project and will take the majority of the runtime.

Metadata Storage:

The data extracted from past frames are kept for future comparison. When a new frame arrives it is stored and the previous data is outputted to the Comparator.

Frame-to-Frame Comparator:

This module receives current data and previous data and does a direct comparison of finger placement difference from frame to frame. This is used in the next section information is used in the spatial difference analyzer. In future applications, velocity will come into play and will alter the sound.

Spatial Difference Analyzer:

Once the comparison of relative difference of the fingers is made, an overall spatial difference is necessary to see if the person has struck a key. The pressing of the key is handed off to the Note Processing module.

Note Timer:

This module times the length of each note played and sends the length and note to the display module.

Sound Transformer:

Receives a sound from flash memory. Based on information from the spatial difference analyzer, it will transform the sound to be louder, to have more attack at the beginning of the note, or other transformations. It will output the signal to the audio module.

Graphical Data Generator:

Given the x,y position of fingers, as well as where the note is located on the staff, serial information is developed and sent to the screen for display.

Display Output:

Receives information about what note was played and for how long. This note is then displayed onto staff bar and the associated key on a virtual piano keyboard is highlighted.

Flash Memory:

Receives current note to be played. Looks up address of this specific piano tone. Sends stored analog note to audio output.

Audio Output:

Receives analog signal, which is converted into audible sound.

IV. Project Plan

Week of 11/4

- Block diagram conference

- Start implement gesture detection
- Start figuring out how to store and recall audio

Week of 11/11
- Project design presentation
- Test and debug gesture detection. Be able to detect 2 fingers, hopefully more.
- Test and debug audio output. Minimum, be able to play unmodified tones.
- Start display module.
- Project checklist meeting

Week of 11/18
- Debugging of display module.
- Start implementing note-processing module.

Week of 11/25
- Thanksgiving
- Debug anything needed in note processing, display, gesture detection, or audio.
- Try to implement any stretch goals

Week of 12/2
- Debug and test whole design
- Prepare for project presentation

V. External Components

- Gloves w/ brightly colored fingertips
- NTSC Camera
- 6.111 Labkit

VI. Division of Labor

This project will be split as evenly as possible among the two of us. A specific breakdown of obligations is as follows:

Lisa:

- Graphical Data Generator
- Display Output
- Sound Transformer
- Note Timer
- Flash Memory
- Audio Output

Sheldon:

- Camera Input
- Frame Analyzer
- Metadata Storage
- Frame-to-Frame Comparator
- Spatial Difference Analyzer