

Viano

K. Kennedy, EE, C. Cunniff, CSE, A. Wildman, EE, and C. Chipimo, CSE

Abstract—The Viano, or virtual piano, allows music enthusiasts to play and record via GarageBand on the go without the hassle of having to carry a full-size midi keyboard around. The Viano is a portable, two-octave, life-size projected keyboard that gives the user the most optimal playing ability and alleviates the struggles of trying to play/record on too small of keys, which is unlike other portable alternatives.

I. INTRODUCTION

GARAGEBAND is a free application for Macs, iPads, and iPhones but there are no products on the market that are small, portable piano keyboards that seamlessly interface with GarageBand while still giving the user the most life-like experience. In just its fiscal year for 2014, Apple sold 160.1 million iPhones, 67.9 million iPads, and 18.5 million Macs[1], that is approximately 247 million new devices that have the ability to use GarageBand. The Viano solves the problem of being able to use GarageBand on the go while still having the most life-like experience one can have when not using an actual full-size midi keyboard.

There are several mini-midi keyboards on the market but these are all not very portable and do not Bluetooth connect to GarageBand. One example on the size of these keyboards is 25" x 9" x 3.5"[2]. An alternative option is using a Mac, iPhone, or iPad as the piano keyboard. The problem with using a Mac is that the user is using a QWERTY keyboard to play the GarageBand piano and this is not ideal when trying to provide the user with the most life-like experience. When using an iPhone or iPad, the user is able to directly play from the GarageBand piano keyboard but the keys on the iPad and iPhone are not close to the size of the keys on a real piano. The width of the GarageBand white keys on the iPad in the 2-octave mode are only 0.5 inches wide, while keys on a real grand piano are about double that size at 7/8 inches wide. This dramatic change in size leads to the "fat finger" problem. A study in 2012 found that 40% of mobile ad clicks were accidental or even fraudulent[3]. So although the user is able to directly play on the GarageBand piano keyboard via iPad or iPhone, there is a higher chance of pressing the wrong key leading to frustrated consumers and less music being composed. The Viano allows the consumer to directly

compose on a piano keyboard with keys that are closer to the actual width of a grand piano key, it is portable-can fit in a small bag, and it can sync seamlessly with GarageBand over a Bluetooth connection. This combination gives way to happier users who will be able to learn how to play, record, and compose music on the go.

The Viano has that capability to change the music industry by breaking the boundaries of where one can play and record, and it opens new doors for those who want to learn how to play the piano; whether it be from home, school, or even on a picnic table in the middle of Central Park.

A major component to the Viano is that it is portable and in Table 1, the specifications for the Viano are listed.

TABLE I
SPECIFICATIONS

Specification	Value
Weight	<1 kg
Height	<38 cm
Length	<15 cm
Width	<10cm
Battery Life	>2 hours*

*May change due to power IC restrictions

II. DESIGN

A. Overview

The Viano uses an image of a 2-octave, life-size piano keyboard that allows consumers to play and record through GarageBand anywhere they please. The technology used for this device is a pico projector, Raspberry Pi, Bluetooth, Raspberry Pi camera, and an IR laser. This technology will permit the Viano to be portable, life-like, and have the ability to communicate with GarageBand. There is a QWERTY laser-projected keyboard on the market that uses similar technology, which gives proof of concept[4].

When considering how to create an image of a piano onto any flat, opaque surface, many technologies were deliberated. In most of the available QWERTY virtual keyboards, the image is achieved through the use of a Diffractive Optical Element (DOE), optical lenses, and a laser [5]. The DOE, in this case, is the image of the keyboard etched onto a thin phase substrate. The optical lenses are methodically placed in front of the DOE to expand the image to the desired size. Finally, the laser is positioned behind the DOE and lenses and the resulting image reflects the laser onto a surface.

While the DOE is simple in theory, its construction is more complicated in practice. A quick Google search will yield thousands of vendors specializing in the design and fabrication

K. Kennedy from Manhattan Beach, Ca (e-mail: kellyk@umass.edu).

A. Wildman from Douglas, Ma (e-mail: awildman@umass.edu).

C. Cunniff from Walpole, Ma (e-mail: cjcunniff@umass.edu).

C. Chipimo from Port of Spain, Trinidad and Tobago (e-mail: pchipimo@umass.edu).

of DOEs. Unfortunately, a quick call to any one of these vendors describing the scope and application of the Viano, will yield a “rough estimate” of anywhere from \$2,000-\$12,000 for the custom DOE displaying a piano keyboard. With a project budget of \$500 maximum, creating a Diffractive Optical Element was unachievable.

Further research led to the examination of a Computer Generated Hologram (CGH) approach. This is a MATLAB computer simulation technique through which a Fourier hologram of the desired image is generated by using the Fraunhofer diffraction formula and printed onto a film [6]. The image is then projected similarly to the DOE—with optical lenses and a laser. The simulation algorithm is composed of four main steps: the construction of the transfer function of the reference and object planes, the calculation of the far field amplitude, hologram construction and spatial filtering, and reconstruction of the hologram [6]. In the first step, the piano image is loaded and converted into a matrix. The far field calculation is achieved through a Discrete Fourier Transform (DFT) algorithm, and the resulting intensity, or square of the absolute values of the amplitudes, is imaged. Due to the Fourier transform property: $G^*(x,y) = G(-x,-y)$, a non-ideal twin image appears during reconstruction, which will be further discussed in the paragraph below. The matrix is then passed through a spatial filter to reduce the intensity of the peak at center image, and finally the matrix is Fourier transformed to reveal a fully reconstructed image.

Given the ample research available on this subject and succinct dissection of the process, creating a CGH was manageable, but would take a great deal of calculation, trial, and error. The most frightening of which being deciding on what size image to load and create a CGH for, that will be small enough to print onto a thin film substrate, such as glass, and keep the design of the Viano portable and pocket-sized. While UMass possesses many printers that would be up for the task, many questions were difficult to find answers to, such as which printer is best suited for this task? How much will printing cost? If the first trial is unsuccessful, is a second or third attempt still affordable? How long will the printing process take? What thin film substrates are available? Where can they be found?

Even if an ideal CGH was fabricated, due to the Fourier process described in the above paragraph, the result of optical reconstruction displays an undesired mirrored image and a distractingly bright beam will appear in the center of the image, as demonstrated in Figure 1. For the purposes of the Viano, having a second, mirrored piano keyboard would not only be distracting, but extremely confusing, and would lead

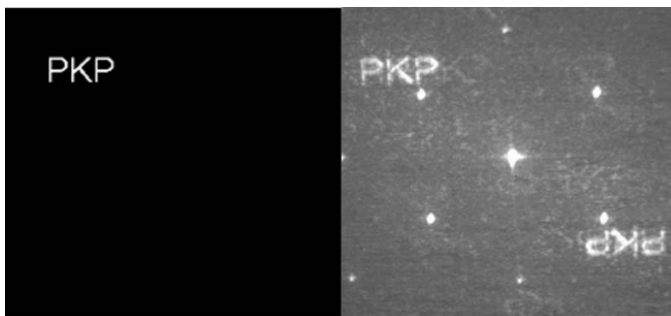


Figure 1. (Left) Uploaded Image (Right) Reconstructed Image via computer generated Fourier hologram [6]

the user to question which sets of keys to press. For this reason, and for the time required to answer the myriad of questions discussed, creating a Computer Generated Hologram was inefficient.

The third and final method considered was a simple projector. Pico projectors are available on the market, ranging from \$75-\$1,000, and their compact size is ideal for the Viano project, which was intended to remain portable and ideally pocket-sized. With further research, the SVP PP003 Portable Pocket Projector was selected to project the virtual piano. Discovered on Amazon for \$74.99, this pico projector is only 4.2” deep, 2.2” wide, less than 1” tall, and weighs less than 4 ounces, but can project an image anywhere between 6-54” diagonally. It has a low luminous flux of 6 lumens, but as the price of the pico projector rises as the luminous flux increases, the SVP PP003 meets the requirements of the Viano. The integration of the SVP PP003 will be further discussed in section B, Block 1: Projecting the Virtual Piano Image.

In Figure 2, the block diagram for the Viano is provided. Each block is described in great detail in Section II. Design B through E.

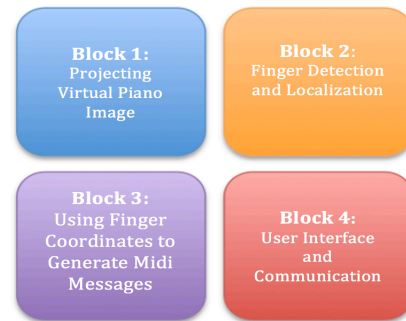


Figure 2: Overview of Blocks for Viano

B. Block 1: Projecting the Virtual Piano Image

The entirety of this technical block is realized through the SVP PP003 Portable Pocket Projector. An open source image of a two-octave piano keyboard is saved in JPEG format on an external memory card and is now placed in the SVP PP003 port. After powering on the pico projector, the memory card option should be selected from the main menu using the directional arrows and OK button on the projector, and the piano image should be selected from the memory card option. Once the piano image has been selected, the pico projector will project the virtual piano for a continuous 90 minutes without simultaneous charging, leaving ample time to compose a new musical piece, practice an old piece, or merely play for entertainment purposes.

Two complications that have arisen with the implementation of the pico projector include resizing the image through optical lenses and the luminous flux of the projected image. In order to project an undistorted, clear piano keyboard, the pico projector must be placed relatively high in the vertical plane (perpendicular to the horizontal surface being played on). In order to keep the Viano compact and portable, the pico projector must rest much closer to the horizontal plane. Therefore, optical lenses are introduced to

the system in order to project the same quality image from a much closer distance. The unfortunate predicted tradeoff is the loss of clarity and intensity of the redirected image. Placing a lens in front of the pico projector causes diffraction, resulting in a softer image. The low lumen count of the SVP PP003 does not allow for much diffraction before the image becomes granular in appearance, and undesirable to work with. Learning the perfect balance of lens magnification and luminous flux loss is imperative to constructing an optimal virtual piano image.

C. Block 2: Finger Detection and Localization

The goals of this block are to detect when a user touches the playing surface of the Viano with one or more fingers, and return a list of unique 2D coordinates describing where each point of contact was made. The goals of this block are accomplished using the setup shown in Figure 3.

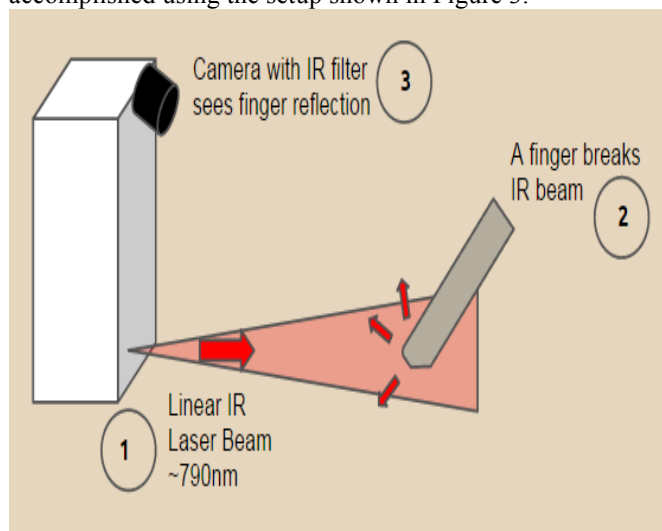


Figure 3

First, an IR linear laser is used to illuminate the user's fingertips, which break the planar laser beam upon making contact with the playing surface. Next, an IR sensitive camera (fitted with an IR band pass filter to match the laser) is used to observe the IR laser beam's reflection off each finger. The reflected beams create a region of high pixel intensity in a particular region of the camera's two-dimensional CMOS array, and this causes a blob to appear in the image for each finger's reflection. Lastly, an embedded microprocessor is used to process the camera's images in real-time and determine the centroid of each blob, and this uniquely identifies where each point of contact was made within the camera image.

The IR laser is designed to operate at ~790nm, as is the IR band pass filter installed on the camera. The optical filter is imperative to blocking out most of the visible light and IR from external sources in the environment, as the chosen camera (a Raspberry Pi NoIR Camera module) is sensitive to both the visible and IR components of the electromagnetic spectrum. The embedded microprocessor chosen is a Raspberry Pi with an ARM11 processor and 512 MB of RAM, running at 700 MHz, with a Videocore 4 GPU. These hardware components all work together to enable touch

detection and finger localization, when combined with image processing techniques performed in software.

The software components consists of several freely available utilities and libraries, all running on the *Raspbian* Linux distribution. The *raspicam* utility is used to capture the images from the Raspberry Pi camera, using a dedicated Camera Serial Interface (CSI) bus that connects directly to the Raspberry Pi's GPU hardware, and removes a huge burden from the CPU. This leads to significantly better performance than using a USB camera.

The OpenCV and cvBlobs libraries are used to perform real-time image processing on the Raspberry Pi. The OpenCV library is used to load the RGB image returned by the camera, convert it to grayscale, and apply a preset threshold to generate a binary image resulting in blobs that arise from the IR linear laser beam's reflection off each finger. The process of generating the binary image is shown in Figure 4. With the binary image, the cvBlobs library is used to analyze each blob, defined as a region of connected components, and to record several useful properties about each blob, such as its area and centroid. The blobs are then filtered by area to ignore blobs that are too large or small to correspond to a finger, and the centroids of the resulting blobs are taken as the finger-coordinates in the camera image frame.

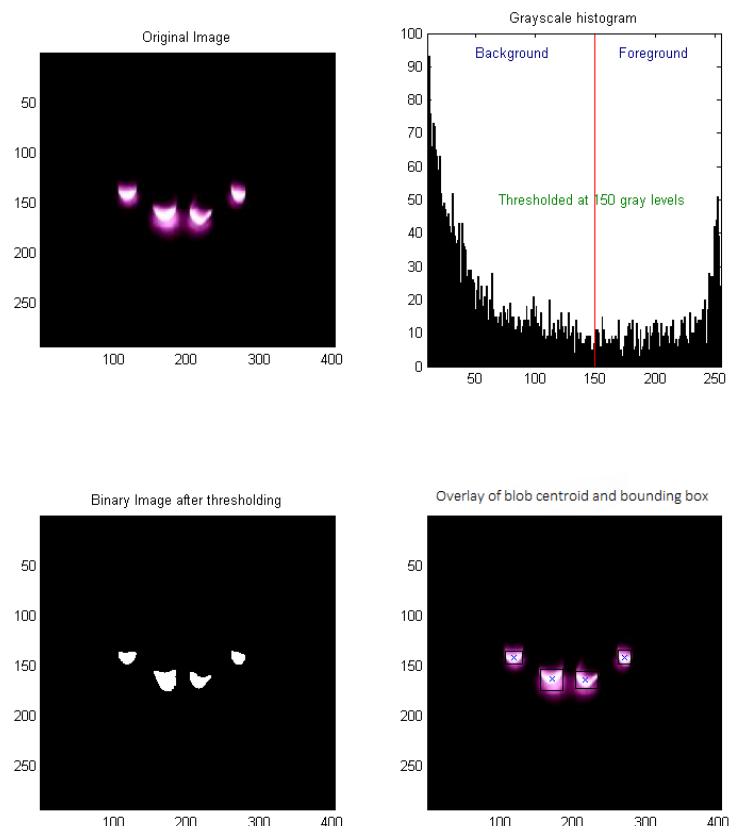


Figure 4

Both the OpenCV and cvBlobs libraries are made available to software developers in Java, python, and C++. For the best user experience, C++ was used to implement this block because it provides better performance, less overhead, and improved efficiency over both Java and python. This ultimately results in a lower latency between the user pressing a key and hearing a note.

Several techniques from past courses will be used to build this block. This includes capturing images on an embedded device, learned in Computer Systems Lab 2 (ECE 354), as well as performing image processing with software, learned in Computer Vision (CS370), and software development in C++, learned in Software Intensive Engineering (ECE 373). Building this block presents numerous learning goals. These include working on Linux and the Raspberry Pi, writing a multiple-class C++ solution that integrates several open-source packages and libraries, and learning how to use OpenCV and cvBlobs to process and render images. A simple experiment to test this block is to turn on the IR camera and laser, and to run the touch detection and finger localization code which can show a visual overlay of the finger coordinates, superimposed on the original captured image, when contact between the finger and the playing surface is detected. Preliminary tests found this overlay test to be very successful, as shown in Figure 5, and the Raspberry Pi was able to capture and process up to 20 images a second in real-time. By Nyquist theory, this allows for a theoretical 10 chords/second to be played on the Viano without the effects of aliasing.

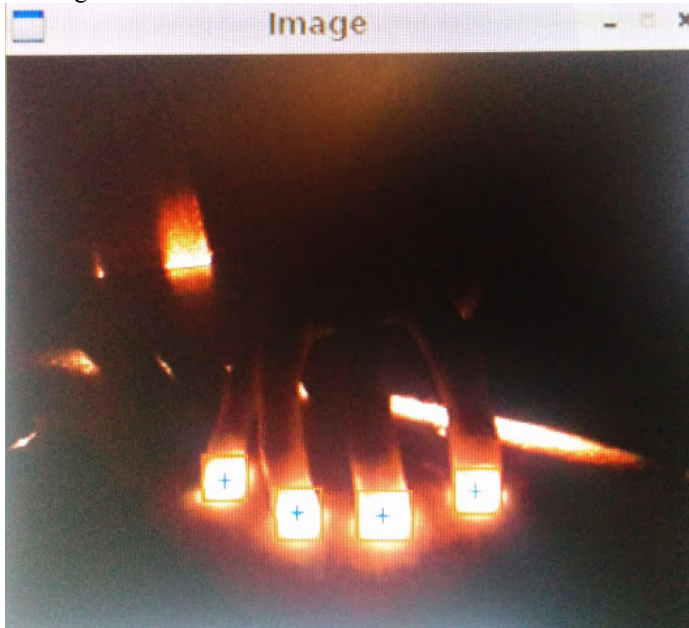


Figure 5

D. Block 3: Using Finger Coordinates to Generate Midi Messages

This block uses information from current and previous finger coordinates in the camera image to determine what MIDI messages will be sent to Garage Band, using coordinate conversion, a virtual keyboard, and a key event processor.

Coordinate conversion is used to map points $P'(x', y')$ from the camera frame of reference (where the projected keyboard is viewed as a trapezium) to points $P(x, y)$ in the world frame of reference (where the projected keyboard is viewed as a rectangle). This problem can be generalized to performing a mapping between two arbitrary convex quadrilaterals, but specifically it reduces to performing a perspective transformation between the perceived trapezium shape of the projected piano in the camera image and the known

rectangular shape of the piano keyboard in the world view. The advantage of performing the coordinate conversion is that the dimensions of the projected keyboard are rectangular and known in the world frame. This makes it simpler and more efficient to implement a virtual keyboard layout with rectangular key components in the world frame, rather than to having to define a keyboard layout with trapezium shaped keys in the camera image frame. The virtual keyboard can then be used to identify what particular key is pressed when supplied a coordinate in the world frame.

The key event processor is used to keep track of whether a key has just been pressed, if it is being held, and when it is released. From a cold start, the block will keep track of what keys have been already been pressed using a Boolean array. Using simple logic, if a key was not pressed before but it is currently, then that would register as a key press; if the key was pressed previously, and still is, then that would be a key hold; if the key was pressed previously but is not now, then that would be a key release. All the events will be both recorded in batch, by adding them to a queue of events, and will subsequently be processed in batch, by removing each and processing each event from the queue. At the processing stage, the necessary MIDI data to send to garage band will be determined and available to send to via Bluetooth to Garage Band.

Techniques from courses the team will use to build this block include performing image transformations using matrices and homogeneous coordinates, learned in Computer Vision (CS 370), and software development in C++, learned in Software Intensive Engineering (ECE 373). The team will need to know how to obtain and use the perspective transformation and homogeneous coordinates to perform the coordinate conversion.

To test the coordinate conversion, the team simulated the perspective transformation using MATLAB, to show that it was possible to map coordinates between two arbitrary quadrilaterals. The team tested this graphically and numerically by inputting a series of coordinates into the first quadrilateral and outputting the corresponding coordinates in the second quadrilateral. Sample points included the four vertices, and several arbitrary points inside and outside the quadrilateral, as well as along the edges. If the vertices and arbitrary points correspond visually and numerically in both quadrilaterals, then the coordinate conversion can be deemed successful.

To test the virtual keyboard, the team plans to supply it a coordinate for a known key, and request the corresponding note. If the note returned matches the note of the known key, for all known keys, and coordinates that fall outside the defined virtual keyboard range are rejected, then the virtual keyboard implementation can be deemed successful.

To test the key event processor, the team would then have the system log each key event in real-time, and identify that the correct action event was recorded for the correct key, for all keys.

E. Block 4: User Interface and Communication

This blocks purpose is to allow the user to communicate with the Raspberry Pi, allowing them to control certain settings of the system from their laptop, along with interfacing

music notes sent from the Raspberry Pi with third party music software on the users laptop. There are three main components of this block. The first is creating software interfaces to send data from the Raspberry Pi via Bluetooth. The second piece is to initiate a Bluetooth communication between the Raspberry Pi and the users laptop while also maintaining a standard protocol for data communication between the two devices. The final piece involves creating a user application to interface the Raspberry Pi with audio editing software applications, allowing the user to record and play sound from the instrument emulated by the Raspberry Pi device.

For CDR, the first piece of this block will include a software interface created in C/C++ on the Raspberry Pi, allowing for Bluetooth connectivity with a connected user laptop, the technology used for this connection is explained in more detail in the next paragraph. Additionally, this software will provide interfaces, which will abstract the communication standard for sending standardized data from the Raspberry Pi, so it can then be interpreted correctly by a connected user laptop.

The Bluetooth interface between the Raspberry Pi and the Laptop will be created using Bluetooth Low Energy, also known as Bluetooth Smart. Bluetooth Smart is a power friendly version of Bluetooth wireless technology, consuming only a fraction of the power as its predecessors. Bluetooth Smart only “supports very short data packets (8 octet minimum up to 27 octets maximum) that are transferred at 1 Mbps” [7]. Our application will be sending minimal data over this connection; in fact only 3 bytes are needed to transmit a single instrument note, encoded within the physical layer of the Bluetooth Smart architecture. Figure 4 notes the data rate needed for a range of typical song tempos, measured in BPM (beats per minute). Even at the higher end of the spectrum at 200 BPM, only an 80 bps transfer rate is needed, falling well with in the 1 Mbps threshold of Bluetooth Smart. Currently, this component of the *Communication Block* has been successfully tested using an older, more power hungry Bluetooth v2.0. However, Bluetooth Smart is the more practical solution, which will be used in the implementation for CDR.

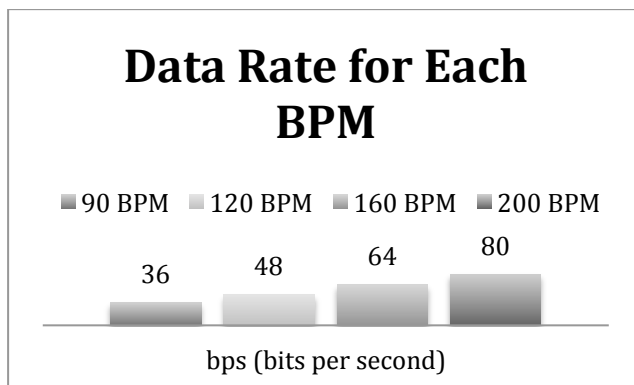


Figure 6: This chart shows the transfer rate in bits per second needed for each BPM song being played on the device. Anything over 200 BPM is less typical in modern music piece.

The third and final part of this block will reside on the user

laptop. A user application will be created which will provide an interface between our device, and a sound playing/recording application on a users laptop, such as GarageBand or Audacity. This will allow the user to listen to, and record the electronic notes being received in binary form from the Raspberry Pi device. Currently, this piece has been simulated by sending music notes from an Arduino and then interpreted by a custom-built C++/Objective-C command line application for OS X that resides on the connected laptop. This application successfully bridged the binary notes sent from the Arduino, to music playing/recording software, GarageBand residing on the laptop. For CRD, the application will provide a more user intuitive experience. It will include a Graphical user interface, which will likely be implemented using a user interface framework for C++ development, provided by Qt Project [8]. This application will also allow for the change of a select group of user settings. The possible settings currently include, the ability to change the type of instrument being played on the Raspberry pi, along with changing the velocity of the notes being interpreted.

III. PROJECT MANAGEMENT

In Figure 7, one can see the status of the Viano’s MDR goals. There were two main changes made from the original MDR plans, one being how the image was being projected and the other being the Device Driver for OS-X. Both of these changes have been described in full detail in the following sections; B, Block 1: Projecting the Virtual Piano Image and E, Block 4: User Interface and Communication.

MDR Deliverable	Description	Status
Virtual Piano Image	Computer Generated Hologram designed and in the process of fabrication	Changed design- using Pico Projection Alternative: Accomplished
Single Finger Localization	Track and display finger coordinates over image	Accomplished
Filtering Visible Light	Idealize the infrared light detection by filtering out visible light	Accomplished
Coordinate Conversion	Given $P(x',y')$ in image, convert to $P(x,y)$ in projection	Accomplished
Device Driver for OS-X	Allow for GarageBand integration with keyboard	Changed design- Application and no longer a driver: Accomplished
Bluetooth	Functioning connectivity	Accomplished

Figure 7

Figure 7 is the list of MDR deliverables and the status of where they stand. The team has successfully met all MDR deliverables and demonstrated the progress in the MDR presentation.

Team Viano is built on the foundation that we are only as good as our weakest part. Thus, there has been a tremendous amount of teamwork amongst the group. Although there are

individual blocks, there have been times where more than one of the members will be working on a part of the project to ensure that everything gets done in a timely fashion. Both Chitula and Christopher have excellent programming skills and have been taking on most of the software aspect of the project, while Kelly and Anna have been working on the Virtual piano image, the optical aspects of the image, the power system, and the IR beam/filtering out visible light.

In terms of communication, the team has been in constant contact via email and group texts. The team meets at least once a week with each other and every Friday with our Advisor, Professor Anderson.

Figure 8 is the Gantt Chart for the Viano. The first six parts listed in the chart were the deliverables promised for MDR while the proceeding parts are those that have been promised for CDR.

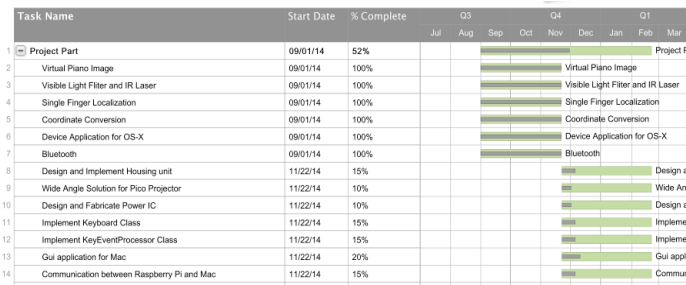


Figure 8: Gantt Chart with MDR and CDR deliverables

One of the tasks that remains incomplete is designing and fabricating an integrated power circuit. This IC will provide power to the pico projector and raspberry pi. The ideal design would include a rechargeable circuit with only one external wire that can be plugged into a wall outlet. This user-friendly design will require research to achieve, but will improve the overall design of the Viano.

Another task is designing and building the housing unit for the Viano. This will be done using a 3D printer, specifically the MakerBot Replicator 2, found in the MakersSpace in ELAB I. The software for this 3D printer has already been downloaded and the team is in the process of designing the housing unit. The purpose of the housing unit is to hold the pico projector, Raspberry Pi, Raspberry Pi camera, power IC, and IR laser beam all in one compact component that will be light weight and portable.

Finding a wide-angle solution for the image is another task that remains incomplete. Just by experiment, using a small wide-angle lens will allow for the pico projector to project the piano image from a lower height but there is still some distortion that needs to be fixed before the task will be completed.

The next phase of the Communication block involves interfacing a Bluetooth Smart module with our Raspberry Pi. This also involves creating the software interfaces necessary for sending and receiving data from the Raspberry Pi over the Bluetooth module.

The next state of the User Interface block involves porting the current command line application to a graphical user interface application. This will allow for ease of use for the user, this application will also see updated features from its

command line predecessor by CDR.

IV. CONCLUSION

There has been tremendous progress with the Viano; the projection of the piano image, finger localization, coordinate conversion, and Bluetooth communication with GarageBand have all successfully been completed. Great time management, teamwork, and early research were key factors for being able to accomplish these goals.

The next steps for the team are to continue expanding on our blocks and to have all blocks working with one another successfully by CDR. In Section III. Project Management, one can read in more detail on what parts of the project still need to be accomplished.

One of the major difficulties we expect to have is finding the best solution for the pico projector to project from a lower height, which would make the Viano much smaller and thus more portable. As of now, wide-angle lenses have been ordered and the team plans to have a solution for the height of the projection by CDR.

ACKNOWLEDGMENT

The authors would like to thank Professor Salthouse for his suggestions and forwarded academic article regarding Computer Generated Holograms.

REFERENCES

- [1] Investor Relations (2014). Apple Corporation [online]. Available: <http://investor.apple.com/>
- [2] iRig KEYS PRO (2014). IK Multimedia Production [online]. Available: <http://www.ikmultimedia.com/products/irigkeyspro/>
- [3] R. Kim, GigaOM (2012, Aug. 31). *Report: 40 percent of mobile ad clicks are fraud or accidents* [Blog]. Available: [#Professionalwebsitewherethedateofpublicationisnotimmediatelyapparent](http://guides.lib.monash.edu/content.php?pid=346637&sid=3402748)
- [4] Virtual Keyboard (2014). Brookstone [Online]. Available: <http://www.brookstone.com/laser-projection-virtual-keyboard>
- [5] J. Fuller. (2012). *How Virtual Laser Keyboards Work* [Online]. Available: <http://electronics.howstuffworks.com/gadgets/travel/virtual-laser-keyboards.htm>
- [6] A. Vijayakumar, B. J. Jakin and P. K. Palanisamy, "Computer generated Fourier holograms for undergraduate optic laboratory," *Physics Education*, vol. 28, no. 4, Dec. 2012.
- [7] (2014). *The Low Energy Technology Behind Bluetooth Smart* [Online]. Available: <http://www.bluetooth.com/Pages/low-energy-tech-info.aspx>
- [8] (2014). *Code Less. Create More* [Online]. Available: <http://www.qt.io/product/>