HoloPyramid

Author: Breyden Wood, Grace An, Jullia Tran: Electrical and Computer Engineering, Carnegie Mellon University

Abstract**—**Holograms and holographic illusions seem the realm of science fiction, but the technology is here, as easily encountered as a phone app and "do-it-yourself" hologram plastic pyramid. Currently, holographic illusions are currently largely used only for entertainment, gimmicks, and at a relatively small scale with premade renders. Our goal is to leverage this existing technology to create a highly useful and immersive presentation tool: a holographic pyramid that displays a scaled-up 3D illusion of a smaller object from a local studio. The presenter would be able to easily move the object and display moving objects with no overhead of preparation or rendering time. Thus, our product would be invaluable for archeologists, researchers, and scientists, who may need to show an artifact or small piece of equipment, small animals, or robots to a large audience.

Index Terms—Holographic Illusion, Presentational Tool, FPGA, Real-Time Image Processing

I. Introduction

The use case of our project is to provide an immersive and interactive presentation tool that displays small archaeological objects or small moving things in 3D with enlargement for easier viewing by a group of people. The goal of the project is to display an enlarged, 3D holographic illusion of a moving object in real time and offer ways to interact with this object for the presenter.

Because of this, our requirements will be focused on several criteria including usability, enlargement, timing, video frame quality, and illusion. We want to give the user an integrated, compact design that is easy to use. To better display the object to a larger audience, our project should enlarge a 3'' object at least four times over its real-world size. Because this is an interactive presentation tool, our design must maintain a stable real-time capture and output of all video feeds of a uniformly lit object for ease of showcasing actuated objects. To maintain the details in the object, our project needs to generate a crisp (see 2.E) video stream to the display. Lastly, this display needs to recreate a 3D projection of the local object under office lighting using a reflective, transparent pyramid to generate the Pepper's Ghost effect illusion.

Currently there are existing designs in the market that use pyramid displays with a smart-phone screen. However, these are generally designed for entertainment purposes and are very small (Fig. 1) -- typically using pre-baked video feeds. These videos would need to be pre-rendered and pre-recorded to be streamed through the smartphone to create the holographic effect. Because of these existing limitations, we want to expand this design from just an entertainment tool to a full-fledged presentation tool by providing a real-time video feed at a larger scale while still maintaining the holographic illusion.

II. Design Requirements

A. Object Size

Our holographic display must be able to replicate and enlarge objects up to 3" on the diagonal, without losing any edges to clipping. We will test our system with objects at or under 3" that are in a variety of differently shaped objects (small toys, credit cards, a hamster, jewelry, etc.)

B. Enlargement

Our holographic display must enlarge any 3" object by a factor of at least four times. Thus, a 3" object should be enlarged to a virtual size of at least 12". To test this, we will place a variety of static objects of known sizes less than 3" and project it to the hologram, then measure the apparent size using a ruler held inside the middle of the hologram. Because the virtual object appears to be suspended in the pyramid, our virtual-image measurements must be taken planar with the virtual object to be accurate. We will compute the enlargement by dividing the virtual size as measured by the ruler by the actual size of the object in the studio and making sure that this factor is greater than or equal to four.

C. Latency and Real-time Performance

The holographic pyramid must display objects in real-time for there to be no perceivable delay between interaction with the object and its projection. Thus, the sum of all latencies in the system (end-to-end latency) be less than 250ms, which is roughly the limit of what humans have as reaction time [2]. Total system latency is defined as the delay between a change or interaction with the object and a change in the virtual

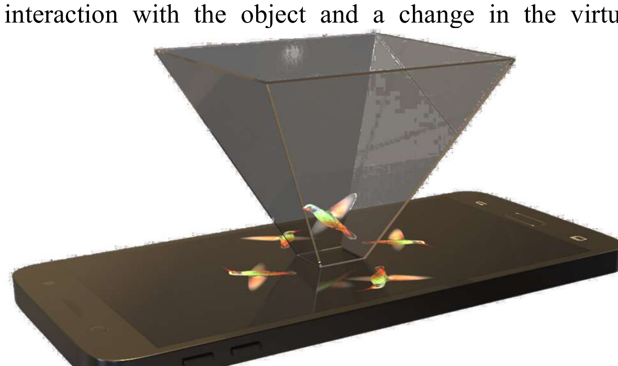

Fig. 1. Example of holographic illusion with a phone and pre-baked images [1]

projection of said object. This is tested by the following: Flash a light in the studio and measure the time it takes for the flash to appear in the hologram using a high-speed camera. The iPhone XS slo-mo camera (owned by one of our team members), or any other smartphones with slo-mo cameras, records at 240fps, allowing us to take latency measurements with a resolution of ~4.2ms. Thus, at 240fps, our total end-to-end latency must be less than 60 frames, which can be counted from the video.

D. Frame Rate Stability

Our holographic display must smoothly display the video. In video feeds, stutters and hitches that contribute to loss of smoothness occur when the time it takes for the video data to be processed and output exceeds the frame-time of the display. This is detrimental to the user experience, so we require that all processing must be done within one frame. The standard for display frame rates is 60Hz, thus all our processing is bounded to 1/60th of a second. Since all this processing occurs on the FPGA, we require that the FPGA processing for every frame be under this time. This can be done by counting the number of cycles taken for processing and ensuring it always remains under 1/60th of a second during normal operation. We plan to use the 7-segment displays to display the maximum number of cycles taken for all frames in operation and periodically checking it remains under the cycle count corresponding to 1/60th of a second over a period of several hours displaying a variety of images.

E. Frame Quality

Because of our project's design goal as a presentation tool, it is critical that details are preserved on the holographic illusion. Thus, each video frame must be photographically crisp. This can be determined quantitatively through an MTF test [14], which measures how fine-grained details can be displayed in an image. A photographically crisp video frame is one in which separation and contrast between fine details is maximized as much as possible or limited solely by the resolution of capture. To evaluate our MTF score, we will capture identical images of our projected objects using a high-end digital camera owned by one of our team members and adjust the resolution to the same as that of our holographic display. Because the high-end camera's MTF score is definitionally resolution-limited in this comparison, our hologram must match its score to pass this test.

F. Illusion

Our final requirement is the strength of the holographic illusion. To achieve a strong holographic illusion, the object needs to appear suspended in the display. To do this, we need to sufficiently remove the background behind the object in the studio while avoiding incidental object removal. We will measure this and define our criteria for success by capturing the display output and testing it in Lightroom to determine that we have removed >95% of the background and <5% of the object.

III. Architecture Overview

We want to begin this section by introducing the user experience of our project. The user will be able to place an object in the studio and see a holographic enlargement of the object being displayed on the pyramid, appearing to be floating in the center. The user can either interact with the object inside the studio or the object can actuate and these interactions will be captured inside the live studio through cameras, where the video stream would be processed inside an FPGA and outputted to the display in real-time. This flow of data corresponds directly to our system design, as shown in Fig. 2 (Simplified version of the system diagram for the purpose of showing data flow).

The cameras are positioned in a small live studio, where the object will also sit. This live studio consists of four cameras positioned at the center of four walls surrounding the object. This is so that the full size of the object can be captured in each frame of the video. The walls of the studio are designed in one solid color, chosen specifically for the purpose of background removal and minimizing glare. This design choice is made based on our frame quality and illusion requirements (See 2.E and 2.F). The size of the studio is calculated based on our object size requirement (2.A). The top part of the studio will be open, allowing the presenter to interact directly with the object with a tweezer, upholding our usability requirement. A more detailed description of the studio design can be found in section 5.F.

The pipeline diagram (Fig. 2) shows the data flow of the camera inputs, flowing into the FPGA. From here, the video stream will be decoded and processed using three filters: chroma-key, contrast and brightness, and sharpness. These filters are designed so that the hologram effect would be enhanced, following our illusion requirement. Through the removal of background using our chroma-key filter, the object will appear floating, increasing the effect of the illusion. Similarly, the contrast and brightness filter will also help to fulfill our illusion requirement through making the object more distinguishable from the background, enhancing the focal point of the presentation. The sharpness filter will help to maintain the details of the object when it is displayed at large scale. After being processed by these filters, the data will flow through an image combiner, where the processed data will be assembled into one frame before going to the VGA protocol controller that outputs to the display.

We introduce a simple 3D display based on the principle of Pepper's Ghost with ray optics, named after the creator John Pepper [3]. This is an old technique that causes the object to appear floating in air. This optical illusion involves a large plane of glass or other reflective surface, placed at an angle to project a person or an object from a room or a screen. This is the main principle on which the pyramid in our design functions. We expand on this principle: instead of just having one pane of reflective surface, we have four. This is so that the viewer does not only see one aspect but can see four sides of the object when they walk around. Hence, the illusion of a 3D object can be observed, giving an effect of a live hologram.

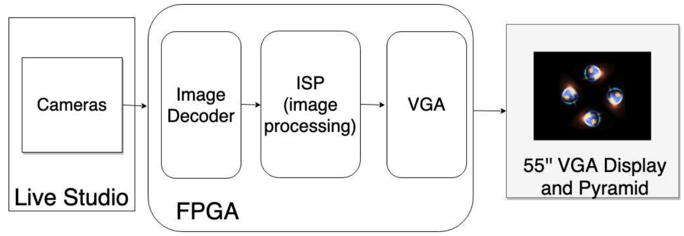

Fig. 2. Simplified system diagram showing data flow

IV. Design Trade Studies

In this section, we discuss the design trade-offs and design choices behind the components of our project.

A. Choices of FPGA

The FPGA is used for most of the computation in our system: receiving video frames from the OV7670 cameras, processing the video frames, and outputting each resultant combined video frame to the monitor over VGA. Thus, our FPGA also needs to be able to support the four input decoders, image processing unit, frame buffers, and VGA controller. As a result, we need a board that maximizes the number of logic elements, amount of memory, and number of I/O pins. We identified two choices of FPGAs in Table I. The DE0-Nano has a greater number of GPIO pins, but fewer logic elements and less memory, hence, we are selecting the DE2-115. While the DE2-115 does not readily has as many I/O pins as that of the DE0-Nano, it can be extended using a daughter card as needed. Memory-wise, our FPGA needs to at least store each camera's video frame (four 640x480 frames) as well as the output video frame (720p display). At the OV7670's RGB565 color scale, this comes out to 5.22MB. An estimate of how much memory is required is shown below.

Minimum of memory used by to store camera frames:

$$240 * 240 * 12 * 4 = 0.3456 \text{ MB} \quad (1)$$

Preferred estimate of memory used to store camera frames:

$$480 * 640 * 16 \text{ (bits per pixel)} * 4 = 2.4576 \text{ MB} \quad (2)$$

Memory used by display frame buffer:

$$720 * 1280 * 24 \text{ (bits per pixel)} = 2.7648 \text{ MB} \quad (3)$$

Furthermore, this board needs to have enough I/O pins to interact with the cameras in the studio. Our FPGA needs to interface with four OV7670 cameras, which each have 18 pins that need to be connected to the FPGA. Some of these pins can be tied together and do not necessarily connect to the FPGA's GPIO pins, such as power and ground lines. However, as a conservative estimate, our FPGA needs to have 72 pins to connect to the cameras. The DE2-115 does not have this many GPIO pins out of the box, however, it has an HSMC expansion header which we are using to add the necessary GPIO pins with an easily available daughter card. With this card, the Altera DE2-115 meets both the memory and GPIO pinout requirements.

Importantly, the Altera DE2-115 also has general PLLs. The PLLs are especially important for the VGA output and for the cameras, as 720p output requires a pixel clock that is higher than the 50MHz internal clock and the OV7670 requires a pixel clock lower than the 50MHz internal clock. Using them, we can generate separate clock signals for both interfaces. The DE2-115 also has a VGA port and DAC that are capable of 720p video output to the display. We have chosen VGA for our output as the protocol is easy to implement in hardware, can scale to different resolutions, and is supported by our FPGA without the need for an additional daughter board. Furthermore, this protocol is easy to adapt to other protocols (such as HDMI) allowing us to use our display.

TABLE I. FPGA Comparison

FPGA	DE2-115	DE0-Nano
Embedded RAM	3,888 Kbits	594 Kbits
SRAM/SDRAM	2MB SRAM 128 MB (4x32MB) SDRAM	32MB SDRAM
Number of logic elements	114,480	22,320
Number of GPIO pins	~40 + 80 (on HSMC expansion card)	153

B. Choices of Camera

Our design requires cameras that have at least 240x240 resolution and RGB565 color-scale for the following reasons: 240x240 resolution is needed because four video frames are combined and projected onto a 720p display, as shown in section 5.B. As shown in Fig. 3, about 1/9 of the cropped 720p display displays the processed output from each camera, so we need at least 240x240 resolution from each of the cameras. Any extra resolution available would also be preferable because it allows us to potentially scale up the resolution, zoom, pan, or use other image-enhancing techniques with the cameras. Finally, at least an RGB565 color scale is needed to provide sufficient color detail for the human eye.

There are several camera groups that are available commercially that operate using NTSC, USB, or VGA protocols. Through comparing these protocols, VGA provides ease of implementation and less overhead needed for the decoding unit when implemented on an FPGA board. Three VGA cameras that meet our budget and achieve our resolution and color-scale requirement: OV7670, OV7725, and OV5642. These are all viable options, but we chose the OV7670 because of its inexpensive cost and better documentation and reference support. The OV7670 supports 640x480 resolution and RGB565 color scale [11].

C. Material of Pyramid

We have four main requirements for the composition of the pyramid panels. This material must be:

1. Fully transparent and reflective - Pyramid panels must be able to reflect the display and easily transmit light to create the holographic illusion.
2. Stiff - Pyramid panels must be able to maintain shape precisely to avoid distortion in reflection.
3. Thin - Pyramid panels must be thin to avoid doubled reflections that distort the holographic illusion.
4. Durable - Our pyramid panel needs to be both shatter resistant and strong so that thin, large panels are not easily damaged during movement.

To create the Pepper's Ghost illusion, we need to pick a material that is transparent. We also need to keep in mind the usability requirement, which would involve picking a material that is not too heavy for ease of the user's setting up. Because of this, we have two choices for the material: glass and plexiglass. Both glass and plexiglass have the property of being transparent and have properties of reflection and refraction of light that is needed to create the holographic illusion. To fulfill both our usability and maintaining an illusion requirement, we will be looking at the reflective index, light transmission rate and specific gravity. We will also consider the factor of ease of construction because this panel will be constructed in-house, which can be measured through tensile modulus of elasticity.

A refractive index that is closer to 1.00 - that of air - would result in a more transparent material and less optical distortion [4] of the background, which is less preferable for the purpose of this project as we want to minimize distortion in the illusion to uphold our illusion requirement. As presented in Table II, plexiglass has a lower refractive index than that of glass, which would help to create a stronger illusion. Both materials have the required light transmission percentage needed for the Pepper's Ghost effect. Through comparing the specific gravity between glass and plexiglass, we can compare how heavy the two materials are when deciding on which one would better fulfill the usability requirement. With the property of being much lighter than glass as shown with smaller specific gravity, plexiglass is also lighter than glass which will make it easier to be mounted on top of the TV. Because of this, we will be using plexiglass for the pyramid's panel.

Another point we considered is also the thickness of the plexiglass needs to be kept at minimum. This is because we want to avoid reflections from the back surface of the material, which would create a "double reflections" effect, where the backside reflection has a slight offset from the front reflection. This can be avoided using thinner sheets so that the two beams would overlap, which further narrows our choice of plexiglass to be around ⅛'' thick, as this is the thickness that is commercially available.

A lower tensile modulus of elasticity means that the material is more prone to deformation when stress is applied, or less stiff and more flexible. With lower tensile modulus elasticity, plexiglass is more flexible and less stiff and so it is easier to be cut during construction of the pyramid.

TABLE II. GLASS AND PLEXIGLASS COMPARISON

Material	Glass	Plexiglass [6]
Refractive index	1.52 [7]	1.49
Specific gravity	2.4 [8]	1.19
Tensile Modulus of Elasticity	50 - 90 GPa (7.25e6 - 13.05e6 psi) [9]	450,000 psi
Light transmission	90% [10]	92%

D. *Image Processing Alternatives*

Because it is much easier to write software compared to hardware descriptions, a justification should be provided for the use of hardware. Alternative devices using software such as a Raspberry Pi or a CPU are unable to meet the timing requirements and the ease of use that our project requires.

The Raspberry Pi struggles to maintain a high FPS when handling multiple video streams with extensive image processing, and its image processing fails to scale for increased resolution and processing requirements. The Raspberry Pi also cannot meet real-time timing constraints as its performance dips when the OS handles background tasks.

A CPU can handle image processing of large, high-resolution images, but not in real-time. CPUs are designed to handle multiple tasks at once and cannot be relied on to provide processed images at 60FPS and with stable timing. Additionally, a full PC cannot be easily integrated into this setup and relying on the user to configure the hologram creates the potential for more user error in a complex setup.

In contrast, an FPGA can support real-time image retrieval from multiple cameras with image processing, all while maintaining stable and predictable frame timing. An FPGA can also be easily integrated into our presentation tool; our FPGA design also serves as a proof of concept for an ASIC to be created for a real-time holographic illusion display. Thus, our choice of hardware is most advantageous over possible alternatives.

V. SYSTEM DESCRIPTION

A. *Monitor*

For the Pepper's Ghost effect to work, an image source must first be displayed and then reflected off the standing pyramid. The effect works by taking a square panel and dividing it into 9 sub-squares, four of which have the side images of the object projected. Each sub-image is reflected off one of the pyramid sides to form one side of the virtual image. To ensure the projected image is as large as possible, we have chosen a 55"

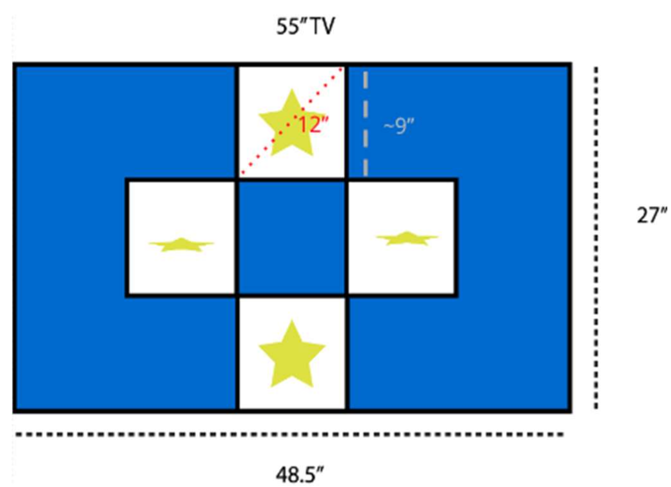

Fig. 3. Diagram showing the layout of images on the monitor

flat screen HDTV (16:9) as the image source to be placed underneath the standing pyramid. This sets each sub image to ~9” square with a diagonal size of ~13”, meeting our design requirement of enlargement. Furthermore, the HDTV has a flat panel with accurate colors, allowing us to display the image on the pyramid without color or image distortion. Our FPGA outputs video data over the VGA protocol, which unfortunately does not match the input of our TV as it only accepts HDMI. To solve this, we are using a simple VGA-to-HDMI converter which allows us to keep both the benefits of VGA output as well as the benefits of the HDTV.

$$\text{Image side length} = (3” * 4) / \sqrt{2} = \sim 9” \quad (4)$$

B. Pyramid Design

The pyramid is where the holographic illusion appears. To create an illusion of a hologram that appears floating in the middle of the pyramid, the tilt angle on each pyramid panel needs to be at a 45˚ angle to the line of vision. This angle is to ensure that the light will travel from the monitor, reflecting about 10% and transmitting the remaining 90% of the incident light [11]. In Fig. 5, the rays from the brightly lit monitor reflects from the pyramid panel and travels towards the viewer on the left. To a viewer who instinctively sees light as traveling in straight lines in open air, the object appears to be three-dimensional and occupying the middle of the pyramid.

To ensure the full illusion of the object is shown, each panel of the pyramid must be large enough to hold an entire image that is coming from the monitor. Because of this, the height of the pyramid needs to be at least the side length of each of the image’s panels, as mentioned in section 5.1 to be 9’’ (which is derived from our requirements). With this requirement, we use trigonometry to compute the rest of the dimensions as labeled A, B, and C in this picture. The most important aspect of this design is the inner angels of the isosceles trapezoid, which are

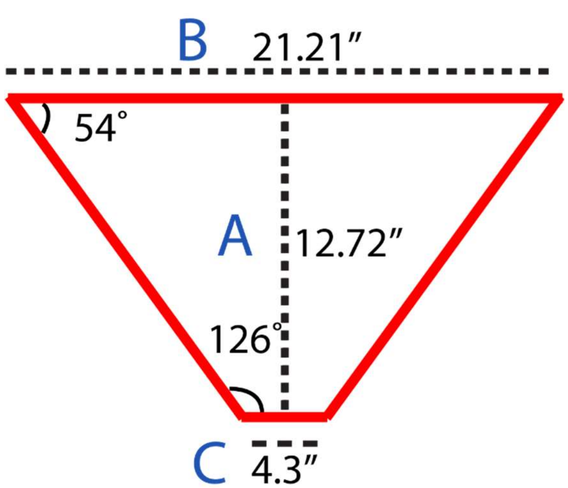

Fig. 4. Diagram showing the angles necessary for one panel of the pyramid

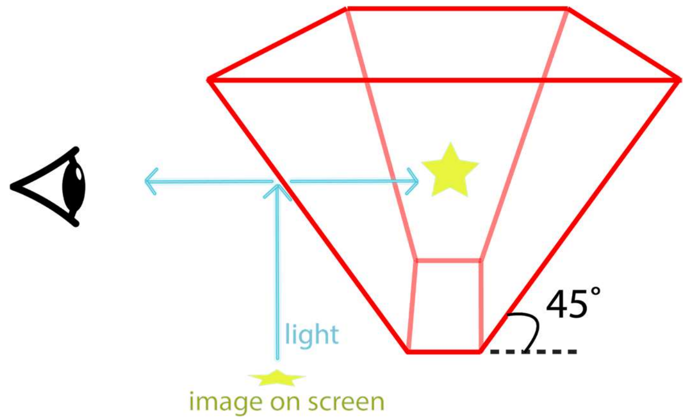

Fig. 5. Diagram showing how Peppers ghost illusion works in the pyramid

computed to be 54˚ and 126˚, so that when all four panels come together, we will have the 45˚ tilt.

C. Hardware -- OV7670

Our design uses four OV7670s to capture side-views of the object at even, 90-degree angles. An Arduino Uno connects to all four cameras to control their color-scale using the SCCB protocol, which is compatible with the I2C protocol. Information about this interaction can be found in the following section on the Arduino Uno (5.D). The cameras also communicate video frame data to the Altera DE2-115 board using eight data pins and handshake signal lines, including PLCK, HSYNC, HREF, and VSYNC. More information about these handshake signal lines can be found in Section 5.E.1.

D. Hardware -- Arduino Uno

An Arduino Uno is used to configure the OV7670 cameras. These cameras require setup and configuration with the I2C protocol to set the scale, color scale, and color matrix settings. The Arduino Uno connects to the camera’s SCCB signals: the SCCB_E signal (serial chip select output), the SIO_C (serial bus control signal), SIO_D (serial bus data signal) and SCCB_E (serial bus enable/disable signal), and PWDN signal (power-down signal).

E. Hardware -- Altera DE2-115

The Altera DE2-115 is the center of our project, taking in the signals from the OV7670 camera, performing processing and filtering on those images, and outputting those signals over VGA to the holographic display. It has 6 internal modules that

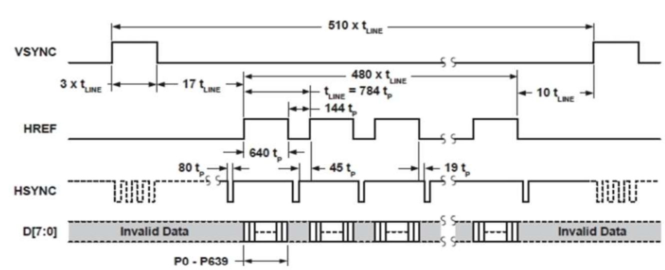

Fig. 6. Timing diagram of OV7670 Input

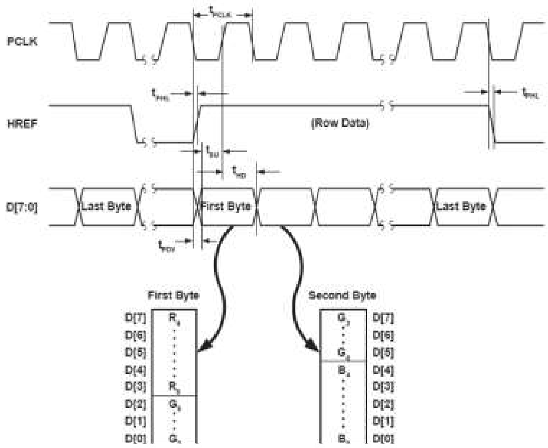

Fig. 7. Timing diagram of OV7670 5/6/5 pixel data transfer

work together to handle this processing and handshaking as described below.

1) Image Decoder

Each OV7670 is connected to an image decoder, which handles the eight data pins, as well as the handshakes signals such as PLCLK, HSYNC, HREF, and VSYNC. Hence, there four decoders work in parallel to decode the data. This module will oversee retrieving data from the camera module. As can be seen in Fig. 6, signals VSYNC and HSYNC provide us references about the location of a pixel data regarding a frame and the timing of its arrival. The image decoder will handle the different RGB formats used by the OV7670 since this camera supports RGB565, RGB555, and RGB444 with these 8 data pins. The image decoder then outputs serial streams of data corresponding to each camera to the first frame buffer, outputting enabling lines and completed lines as it goes to tell the FFB which cameras are having serial data written at any given time and when they are done.

2) First Frame Buffer (FFB)

The first frame buffer technically consists of four buffers, one for each image signal processor module. This is to enable data pipelining and simultaneous processing of all four camera frames. Each frame buffer takes a serial stream of data from the image decoder and stores this data into SRAM and SDRAM to build the buffer along with FIFO queues to output data to the next module, the ISP. The FFB module has an input data line from the image decoder and an output data line to the image signal processor. The FFB module also has enable and completion control lines indicating when data is ready to be output to the image signal processing modules. This data is needed for direct convolution involved in the sharpening filter.

3) Image Signal Processor (ISP)

The ISP is designed to enhance the video quality going out to the display. It takes in signals from the FFB for the serial data being read in as well as control signals telling the ISP when the FFB has enough data to begin processing. It will output four serial streams of data and enabling lines to the SFB to tell the SFB when it is processing data and the data to be stored into the buffer. To sufficiently strengthen the holographic illusion and preserve details, we have three image filters through which video frames are processed in order in the ISP module:

Chroma-key filter: Chroma keying is a technique to remove backgrounds from images by selectively removing all pixels that match the color of the background. Our background removal filter is designed to remove >95% of the background and <5% of the object. This will use a hard-coded background color value and threshold distance that determines how many colors close to the background color are filtered out. The background color will be the color used for the background of the live studio. This algorithm is shown below.

$$\text{if(distance(pixel, target)} < \text{threshold), pixel} = 0 \quad (5)$$

Contrast and brightness filters: This filter will operate on each pixel to increase the overall contrast and brightness of the image, to improve the visibility of the holographic illusion in office lighting. This algorithm is shown below, where “a” represents the contrast and “b,” the brightness.

$$f(x) = a(x-128) + b + 128 \quad (6)$$

Sharpness filter: The video frame is convolved with the 3x3 image filter shown in Fig. 8 to increase local contrast. This will be done through direct convolution instead of alternatives such as FFT-based convolution. While computationally more expensive, direct convolution uses significantly less memory than FFT-based convolution [13]. Because of the need to be processing four video frames from the four cameras simultaneously, memory bandwidth will be the limiting factor, rather than computational units such as adders and multipliers on the FPGA. Additionally, our image filter is relatively small, so the difference in computational expense between direct and FFT-based convolution is very low.

4) Second Frame Buffer (SFB)

The second frame buffer sits after the ISP and serves to buffer the modified frames from each camera for later combining and output. It takes in four serial streams of data from the ISP corresponding to pixel data from each of the four cameras as well as enabling lines telling it which cameras are actively being filtered. It stores this data into SRAM and SDRAM to build the buffer along with FIFO queues to output data to the next module, the image combiner. To interface with that module, it gives the combiner access to the queue along with enabling lines telling it which camera buffers are being read.

5) Image Combiner

The role of the image combiner is to rotate images by appropriate multiples of 90 degrees. This will be done through selecting pixel data according to their relative position of the final frame from the RAM frame buffer to output a serial stream of bits. This data is being read and stored selectively through a FIFO queue protocol and handshake signals with combinational reads and synchronous writes. The image combiner will map

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Fig. 8. 3x3 Sharpness Convolution Filter

consecutive bits of the final frame into pixel locations in the four camera frames, which are implicitly rotated in this process.

6) VGA Protocol Controller Module

The VGA Protocol oversees taking the pixel data and output from the Image Combiner and output this pixel data along with the appropriate VGA timing protocol (HREF, VSYNC, HSYNC). This data will be outputted directly into the VGA DAC, where the signals can be displayed on to the TV. Timing will be synchronized between the Image Combiner and the VGA Protocol Controller so that the output signal is delivered with the same rate that is needed for the display. It will take in a serial feed of data from the image combiner as well as timing signals to let the combiner know when it needs new data to output.

F. Live Studio

An 8" by 8" live studio is used to capture side-view videos of the object. The size of the studio comes from the cameras: The OV7670 provides a 25˚ vertical field of view from the horizontal, so the camera must be placed 4" away from the object to show the full size of the object. This live studio is made of cardboard and uses a blue backdrop. While green backdrops are often used for the purpose of background removal, using a blue backdrop reduces the color spill from the background onto the object. Less color spill improves detail preservation, which is critical to our presentation tool, fulfilling our usability requirement. LED lights are placed at the corners to ensure even lighting. The more uniformly lit the backdrop is, the better our chroma-keying filter can remove the background. Our live studio also includes accessories such as blue tweezers to enable seamless interaction with the object during presentations, without obscuring the object. Being the same color as the background, the blue tweezer will also be removed by the image processing unit, hence, there will not be obstruction of the object.

VI. Project Management

A. Schedule

Our schedule is constructed so that the most important components are worked on at first, so that any issues that arise have sufficient time to be fixed. These critical tasks include the following: integrating the FPGAs with the camera, output video frames from the FPGA through VGA output, and image decoding and image combining, which includes movement of data into and out of SDRAM on the FPGA. We ordered our critical hardware components first (FPGA, cameras, display, pyramid materials) so that we could begin working on the MVP before getting to the image filtration.

Following the completion of our MVP by the interim demo on April 12th, we will be focusing on improving our project and image quality, through writing and adapting the image filters as well as integrating and testing our FPGA.

B. Team Member Responsibilities

Breyden Wood

- Image rotation and combining
- PLL with VGA output
- Testing of image quality in Lightroom

Jullia Tran

- PLL with Camera interface with FPGA
- Image decoding and frame buffer
- Integration of FPGA

Grace An

- Building the pyramid and live studio
- Designing and building image filters on FPGA

C. Budget

As shown in Table III, our largest hardware components--the Altera DE2-115 and HDTV--are preexisting components. Added to the cost of the ~9$ cameras, the $55 HSMC expansion board, $20 VGA-to-HDMI adapter, and $120 for plexiglass and other building materials, our entire budget comes out to approximately $212.98. The remaining portion of our $600 budget may be used for any issues that arise, such as purchasing additional plexiglass if any gets damaged and needs to be replaced.

D. Risk Management

Our risk management plan is focused on providing fallback plans to compensate for physical constraints such as number of logical elements or limited RAM on the FPGA board that we are using. We are able to:

1. Reduce video frame quality: If we struggle with the timing requirement, memory constraints, and/or logic element number on the Altera DE2-115 board, we can reduce the resolution, frame rate, and color-scale [12]. We can reduce the 480x640 camera input to 240x240 camera input by sampling every other pixel. We can reduce the VGA output from 1280x720 to 640 x 480, and then use a VGA-to-HDMI adapter and an HDMI upscaler to maintain video frame quality at low resolutions. We can also reduce the speed of 60 frames per second down to 30 by reducing the clock speed.

2. Reduce pyramid size: The pyramid needs to be stiff (able to hold the pyramid shape at exact dimensions and angles), transparent (for high reflection quality), and thin (to avoid doubled illusions). If we have difficulty meeting all these constraints for a large pyramid, we can reduce the pyramid size (and the video frames used) to maintain the holographic illusion.

3. Remove image filters: We can remove the image filters if we have insufficient logic elements or time to process the images in real-time. Instead of using chroma-keying for background removal, we can use a black background in the live studio, so that the background is automatically not projected onto the hologram. We can also remove the brightness and contrast filter by manually setting brightness and contrast in the cameras used. We can also remove the sharpening filter since it is not critical to the holographic illusion requirement. While it will worsen the video frame quality of the final video, this is not critical to the design.

References

[1] Artstation, https://www.artstation.com/artwork/lVwgAa

[2] Jain, A., Bansal, R., Kumar, A., & Singh, K. (2015, May). A comparative study of visual and auditory reaction times on the basis of gender and physical activity levels of medical first year students. Retrieved March 15, 2021, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4456887/

[3] Jeremy Brooker. 2007. The Polytechnic Ghost. Early Popular Visual Culture 5, 2 (2007), 189–206.

[4] Nick Schlüter, Franz Faul; Are optical distortions used as a cue for material properties of thick transparent objects?. *Journal of Vision* 2014;14(14):2. doi: https://doi.org/10.1167/14.14.2.

[5] Synowicki, Ron. "Transparent Substrates: How To Suppress Reflections from the Back Surface." *Https://Www.jawjapan.com/Document/Public/000/General_01_Transparent%20Substrates%20Suppressing%20Backside%20Reflections.Pdf*, Apr. 2010.

[6] Arkema Group. *Plexiglas Data Sheet*, Altuglas International, 2019, www.plexiglas.com/export/sites/plexiglas/.content/medias/downloads/sheet-docs/plexiglas-g.pdf.

[7] Apple Computer Corporation, *Index of Refraction*, 2000, hyperphysics.phy-astr.gsu.edu/hbase/Tables/indrf.html.

[8] Engineering ToolBox. *Solids and Metals - Specific Gravities*, 2003 https://www.engineeringtoolbox.com/specific-gravity-solids-metals-d_293.html

[9] Engineering ToolBox. *Solids and Metals - Specific Gravities*, 2003 https://www.engineeringtoolbox.com/young-modulus-d_417.html

[10] Li, Yan, and Shuxia Ren. "Plate Glass." *Plate Glass - an Overview | ScienceDirect Topics - Building Decorative Materials*, 2011, www.sciencedirect.com/topics/chemistry/plate-glass.

[11] Greenslade Jr, T. B. (2011). Pepper's Ghost. The Physics Teacher, 49(6), 338-339.

[12] Omnivision, "OV7670/OV7171 CMOS VGA (640x480) CameraChip Sensor with OmniPixel Technology", Aug. 2006

[13] Abtahi, Tahmid & Kulkarni, Amey & Mohsenin, Tinoosh. (2017). Accelerating convolutional neural network with FFT on tiny cores. 1-4. 10.1109/ISCAS.2017.8050588.

[14] Lens MTF test results. (n.d.). Retrieved March 15, 2021, from https://www.the-digital-picture.com/Help/MTF.aspx

TABLE III. GLASS AND PLEXIGLASS COMPARISON

Part name	Price	Quantity	Total price
Altera DE2-115 (owned by course staff)	$0.00	1	$0.00
HDTV (owned beforehand)	$0.00	1	$0.00
OV7670	$8.99	4	$17.98
HSMC expansion board (for DE2-115)	$55.00	1	$55.00
Estimated cost of pyramid and studio materials (plexiglass, LED strip lights, etc.)	$120.00	N/A	$120.00
Estimated cost of VGA to HDMI adapter	$20.00	1	$20.00
Total			$212.98

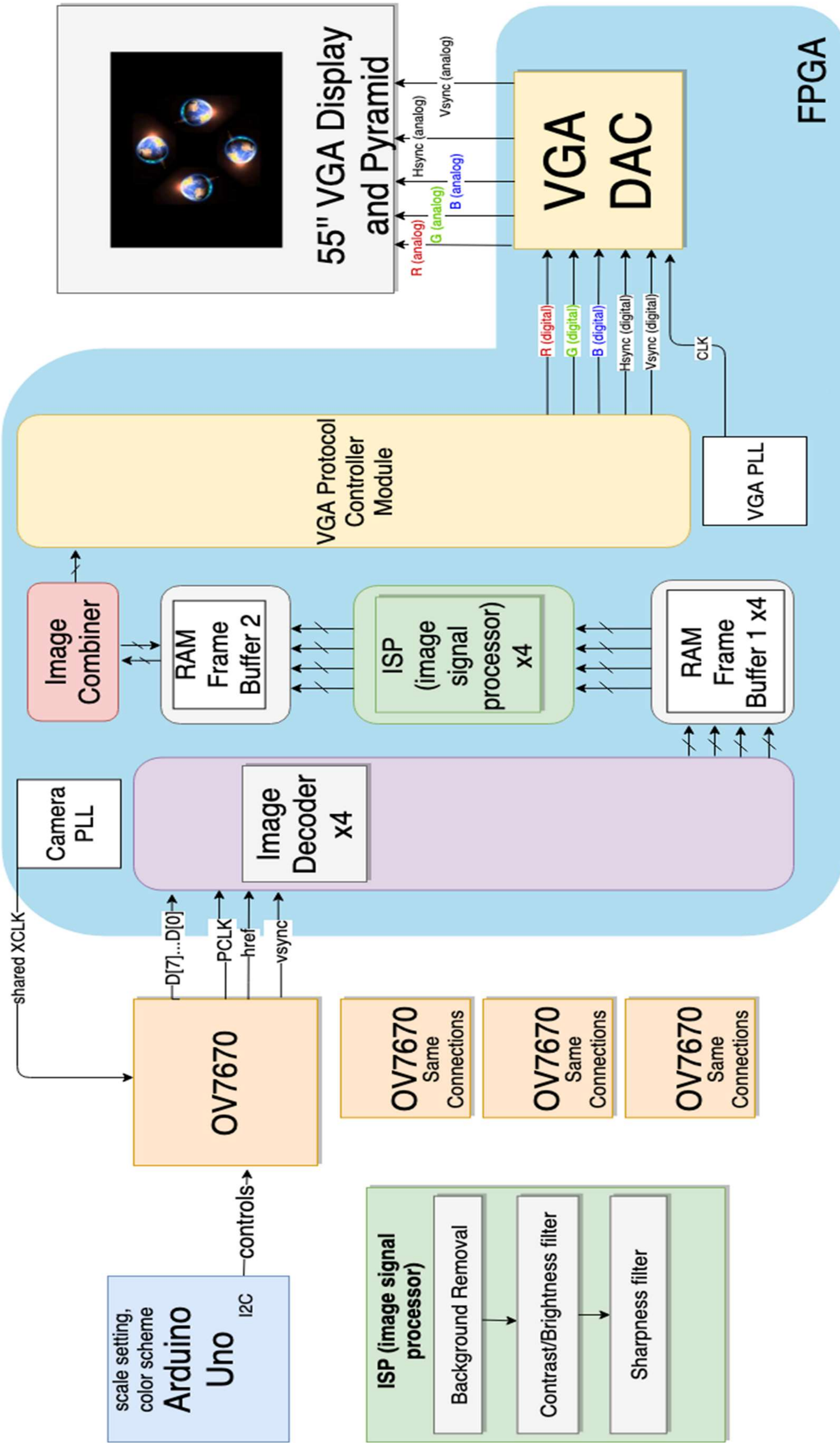

Fig. 9. Whole System Block Diagram

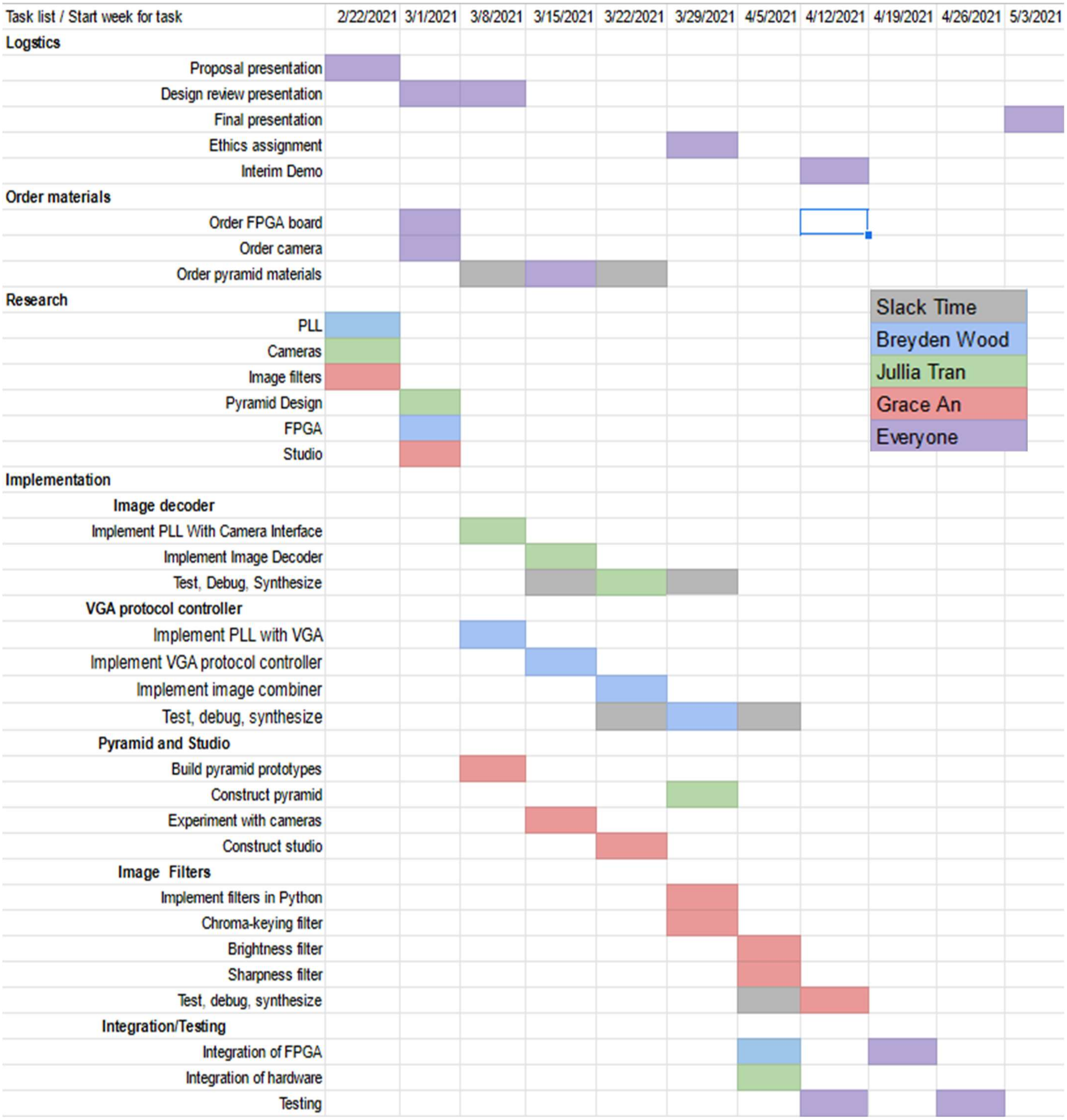

Fig. 10. Schedule