**3D PROJECTED IMAGERY FREESPACE CONTROL UNIT (PIFCU): DESIGN, IMPLEMENTATION, AND ANALYSIS**

Holly T. Ferguson, Stephen W. Turner  
The University of Michigan-Flint  
303 E. Kearsley Street Flint MI, 48502 United States  
hfergus2@nd.edu, swturner@umflint.edu

**ABSTRACT**
A desirable mechanism to control a computer would be to use a multifaceted, three-dimensional interactive display system that provides varying levels of control of an API via a visually bounded freespace containing projected imagery for interactions. This work presents such a system and its proof-of-concept, which includes two components: the 3D Projected Imagery Freespace Control Unit (PIFCU)—the built hardware used to create 3D imagery and function as the system’s control unit, and the Gesture-Controlled 3D Interface Freespace (GCIF)—the term used for this “sensorized” volume of space in which the 3D images and gestures will reside. This system is intended to ultimately remove the need for keyboards and similar interface hardware.

The hardware of this device consists of an array of linearly adjoining slices of concave mirrored surfaces with openings at the top and bottom. This combination advances related work such that the imagery can be perceived as three-dimensional images in free-space, and the computer is controlled using hand gestures recognized by sensors. The benefits to other professions include providing new methods of construction, navigation, gaming, as well as presenting a solution for certain physical limitations that traditional computing experiences currently exert on users (i.e. rigid hand angles for typing).

**KEY WORDS**
Applications, Process Control, 3D Projected Imagery Freespace Control Unit, Control Systems, Components, Freespace, Gesture-Controlled 3D Interface Freespace.

1. Introduction
The methods for which projected imagery can be used to simulate 3D holographic volumes and solve existing spatial problems, especially within building systems, is the foundation of interest surrounding this project. Methodologies that are tangibly human-interactive are already popular areas of research for the field of computing. The techniques that allow human interaction and control of computers can now be combined with electronics used in built architectural spaces. Where the limits of these interests cross is the starting point for the work presented below. This is a proof of concept project that illustrates the feasibility of combining 3D projected imagery along with gesture recognition. This project examines the use of gestural recognition capabilities in combination with projected display systems.

The boundaries of interactive and visible imagery are explored to test capabilities that would be useful in transitioning the functionalities of current and future computer systems into a different type of control environment. Instead of moving towards the substantial, yet more common, research that works with gesture-recognition tailored for image manipulation or resizing, this project uses it in conjunction with holographic-simulating mediums for controlling OS GUI/APIs. The goal was to produce certain types of computer controls without the need for physically touching any piece of hardware; in a full model, this would remove the need for keyboards, computer mice, other interface hardware, or some combination thereof. The work to achieve this goal includes two major components: the 3D Projected Imagery Freespace Control Unit (PIFCU)—the built hardware used to create the 3D images (that emulate the creation of holograms) and function as the system’s control unit, and the Gesture-Controlled 3D Interface Freespace (GCIF)—the term given to the “sensorized” volume of space in which the 3D images and hand gestures will reside. To our knowledge, there are similar types of research that work with the singular various pieces required here or certain similar groupings, but none that offer a solution identical to the combination of elements in this project and/or to the same end. For the base experimentation, the GCIF uses a single depth sensor to detect linear distance to a given object. The system then creates the associated image projection on the PIFCU display. Currently, the experimental functionality implemented for this work includes a basic selection of the alphabet for typing with the control unit. Again, this selection helps to provide a proof-of-concept for this system, as well as inspiration for additional controls, educational uses, and accessibility ideas.

The remainder of the paper is organized as follows: Section 2 presents related work. Section 3 describes the design and structure of the PIFCU. Section 4 presents an analysis of the PIFCU and a discussion of its current state. Section 5 presents conclusions and future work.

2. Related Work
2.1 Displacement/Gesture Sensors
There is a variety of research on technology used to detect and track physical events, and there are a range of physical measuring principles available [1]: these should be examined before further addressing this current project. Sensor types of interest to this work and its overall goals
are capacitive sensors, laser sensors, and miniature confocal sensors, to name a few. Displacement/proximity sensors are the main focus of this work due to their ability to easily sense changes as events. While this is currently possible and efficient, there is still a continuous effort to increase sensor efficiency and reduce sensor size. This project is particularly interested in the fact that non-contact sensors also offer the added capability of freeing the user from direct physical contact with the hardware [2].

One of the more popular tracking sensor systems is the Microsoft Kinect, originally intended for use with the Xbox 360 gaming system. This device uses a combination of depth sensors and RGB cameras to input real-time data used to generate information about the 3D, unconnected space [3]. In the interest of the work presented here, it is important to note that “unconnected” is referring to physically detached from the technology itself. Because of its portability, available libraries, price, and its capability to be used with a variety of programming languages, the Kinect sensor is a useful starting point for projects requiring the type of input data that is used by the PIFCU. This type of programming capability will eventually allow for efficient conversion to parallel computing, since the algorithms work strongly with pixel analysis and changes [3][4].

Research in the field of robotics is also of interest to this project not because of the robotic machinery itself, but because of tracking techniques that are used to manipulate robots. In one study, a laser tracking system (LTS) produced the exact orientation and coordinate position of robotic end effectors within a built, physical space [5]. This study found that measurements could be accurately taken using a laser beam and two rotary axes. Diffraction patterns are used to calculate orientation, and a tracking unit follows the movements; this offers an efficient solution to capture motion dynamically. While still a possibility, this approach would limit the hardware-free environment that Kinect would potentially allow because tracking devices would still need to be attached to the user.

Another study found advantages with detecting events in sensor networks using proximity queries [6]. While this could prove an interesting approach in the future, these systems are more often found when tracking vehicles or when using surveillance as a type of security, as opposed to developing natural user interfaces. As far as the programming of the control unit is concerned, the minimum capability needed to perform the data collection for this project would be that generated from a simple binary sensor or series of binary sensors. Useful data could be gathered via mechanisms such as air servo displacement sensors [7] or even by using fiber optic displacement sensors (FODS) with concave mirrors [8]. Even for these sensor types, it was found that the linear range of the sensor strongly depended on the parameters of the mirror(s). Yet another study used an inductive probe and measured oscillation frequencies to achieve a digital value for the displacement of an object [9]. This approach lends itself well to complex measuring systems. However, it would be insufficient when turning to work with 3D fields in space that should be achievable for the interface design presented in this paper. Alternatively, a binary sensor scheme could be employed well for those situations where 1D and 2D arrays are useful [10]. These would, however, pose the same problem as other examples when transitioning to 3D fields of space; this is an issue the Kinect sensor system (and similar inventions) have already solved. For these reasons, the Kinect sensor was chosen to be used in this proof-of-concept; it provides the functionality required at a fraction of the costs compared with other available sensor types.

### 2.2 Current Holographic/Simulated Interfaces

The other category of research that plays an important role in the development of the PIFCU is the work in the area of holographic computing/interfaces. Manipulating an interface in a space that is free from the physical computer hardware is used for a variety of applications, e.g., gaming. However, it only partially meets users’ needs when considering other functionalities discussed in this project such as implementing a hardware screen as a 3D image in free space. Holograms can be described as “three-dimensional representations of physical objects” [11]. Much of the research surrounding holography creates holographic images that are either surrounded by their functional hardware or limited by other hardware. Holograms are useful for optical elements, certain display types, biomedical imaging, laser applications, physics research, and computer data and storage [11]. Of more interest to this project solution is the analysis of these approaches and the overall effects of their implementation. While there are arguments that these accessory hardware devices remove the system’s definition as being a “true” hologram, there is still merit in investigating successful research that produces a similar type of spatial field/quality required for our project. The 3D projected imagery (3D virtual objects) produced from the PIFCU are free from the limitations these other holographic systems imply, now allowing freedom of physical space for user interactions.

Researchers in [12] developed what could be classified as a 3D holographic video system. Their system reduced power usage compared to previous models by using 800,000 mirrors and 2D projectors to produce a holographic film. There were challenges, if not limitations, with creating larger arrays of mirrors that continually need to be smaller in size. Issues with these types of systems caused the next generation to be defined by applying color to the images. Eye position sensory systems and compact electronic display systems are both advancements of holographic display units that work sufficiently, but still need types of hardware filters, glasses, etc., to produce the 3D spatial effects; this is some of the restrictive hardware this current project eliminates. The authors in [13] found that using arrays of half mirrors for projection-type imaging allowed conversion and/or simultaneous displays in both 2D and 3D layers of images. These prototypes belong to the category of 3-dimensional displays, even though there was not an
industry-leading standard at the time for the implementation of these types/classes of displays. Creating the hardware to house the holographic control device and associated sensors needs to be something that is large enough to allow the chosen controls to be displayed. In a similar way to the projection-type integral imaging [13], an array of concave mirrors is the chosen approach for this proposed PIFCU.

A number of recent Microsoft inventions [14][15][16] in the field of holography are some of the main motivators for our project, since gesture-based user interfaces and video processing are incorporated into many display systems they develop. One approach uses a 3D stereoscopic display that presents the illusion of a hologram. This mechanism detects the user’s eye positions and prepares the computer screen accordingly; the screen then produces 3D images [14] that do not require mediums like 3D glasses, but this approach is slightly different than what is needed in our project. A project of particular interest is the Microsoft Vermeer system [15]. The virtual image produced through the aperture of this device is, at the correct range of angles through 360*, as vivid as the real object itself and can be manipulated to a certain extent by analyzing the aperture size and the size of the parabolic diameter of the concave mirror. Our solution advances the structure used for Vermeer and takes the new construct further by omitting the size restrictions that the full use of the 360 degree Vermeer image implies, allowing for additional uses and sizing arrangements.

2.3 Combining Sensors and Holograms

Where and how these two research areas overlap is the foundation for creating the PIFCU. This category of emerging display systems almost by definition has a co-existence or identity in the areas of natural user interfaces (NUIs). Some consider a NUI to be that which reproduces the “real world”, that is, a “source for the metrics enabling an iterative process to create a product” [17]. The gathering of such metrics is particularly important when gesture recognition is introduced into the PIFCU display.

One example project demonstrating the overlap between these two research areas involved the ability to draw 3D surfaces in the free space next to the actual computer mouse and monitor. These organic formations are created via tracking sensors that are placed onto the hand and head of the user, translating his/her movements into the rendered virtual space on the connected computing software [18]. A different project created an in-air typing ability using robotic eye sensors attached to a Smartphone screen interface. The user is able to type in the space above the phone/keypad that is provided on its screen via a camera that tracks the position of a finger and vibrates slightly when a key is selected [19]. These examples are integral background work, but they still use spatial interactions that are neither metered by an overlaying visual aid nor place the user completely in an occupyable yet usable control space.

Other researchers [20] worked to create a “touchable hologram,” which provides the sensation of touch by using pressure created by ultrasonic waves. Wii systems were used to track the user’s hand. This system still requires hardware to surround the hand before it will work properly, but its potential uses could range from holographic/virtual light switches in hospitals to touchable, holographic books.

Another research study conducted at Sony worked with a 360* interactive holographic display [19], which does not require assisting glasses to be worn; It was created to work as a 3D hologram wrapped in a cylindrical shape. Microsoft has also developed certain display systems using gesture recognition and holographic technology. In particular, the Microsoft Surface [21] uses one of several integrated PC systems to operate a touch-screen surface, often in the formation of an actual tabletop. This is unique in its advancement of a class of interfaces, but for the purposes of our project it does not allow interactions to be free from direct contact with the physical hardware. Vermeer is described as providing an interactive 360* viewable 3D display that also provides viewpoint-corrected, stereoscopic 3D graphics to simultaneous users 360° around the display, without need for special eyewear or other user instrumentation [15]. This technology is another successful approach that can be advanced into an interactive hologram that is not hindered from its required mechanical parts. However, it suffers from limitations similar to that of the traditional mirascope’s optical illusion, in terms of its range of useful viewing angles, optical calculations of images, and so on.

3. PIFCU Overview

3.1 PIFCU Overview

The interactions made between the user and computer interface screens in a traditional Windows computing environment can be altered to allow operations outside of their hardware, in a sense. We posed the following question: can this be done in conjunction with Kinect motion sensing abilities, such as using other displacement sensors or imaging means to produce a visual interface system free of traditional laptop screens and computer mice? Existing applications that have used displacement sensing technologies (to observe interrupts/events that represent motion being sensed) work within a 2D or 3D array in space. This led to an array of freespace itself that becomes the interface system for the 3D projected imagery, which also controls the system.

The goal of this project was to produce hardware and programming for a type of 3D, freespace interface applied for some of the basic commands used within a Windows OS API. It draws from the ideas of other interactive and/or viewable 3D display systems, but it expands on existing ideas and research and includes some new design specifications. The extensions made are combined to achieve a control unit that uses projected holographic-like images within a 3D freespace; this creates an interface that is usable outside of physical contact with the computer hardware.

The display unit consists of an array of concave mirror slices; 3D imagery is calibrated in combination with the
light projections. For prototyping purposes, it was constructed to work with any simple laptop screen space. The control unit used the chosen sensor system to work primarily with gesture recognition and depth recognition. This part was combined with the display hardware so as to create the GCIF, and data is collected via human interactions with the freespace. The data is translated into various API commands and the user’s simulated holographic screen is visually updated respective to each selection.

3.2 Structure of the PIFCU Concave Mirror Sections

Many types of existing holographic imagery function on input/output information as described above, but these previous works require various hardware assistance, such as surrounding boxes, smoke, etc. This means that they include hardware that will physically interfere with the free space in which the projected imagery will reside-this same space needed to properly operate such a system. Algorithms for gesture recognition are readily available for use with proximity sensors, so the components of this project requiring these are used. There are physical advantages and many limitations when working towards a type of 3D projected interface that uses an unhindered “freespace.” However, overcoming these limitations allows the PIFCU to be used as a physically screen-free interface system. In this situation, we felt that the use of virtual images produced by a set of double-curved concave mirrors are the effective option to allow freespace for system control.

We observed by the first author that when a user is viewing a virtual image created by two concave mirrors, they are doing so only from a single line of sight. There is no need for a single user to have a 360° view, apart from personal preference. For example, only viewing a laptop computer from a single line of sight is not viewed as a problem, it is simply all that is required for general use. This is not to say that there are not great advantages to 360° display systems. However, separating this desire from the current circular device may also provide alternate advantages for a new projection display system [Fig. 1].

If the same aerial and sectional views of a typical Mirascope are considered, then also consider the portions of the Mirascope that are unnecessary for viewing the image(s) [Fig. 1.a], [Fig. 1.b] from the direction depicted by the arrows [Fig. 1.c]. The interesting option comes from the section of this device that is left if the two shaded sections [Fig. 1.a], [Fig. 1.b] are literally removed. When gone, the remaining slice [Fig. 1.e] will still produce a usable virtual image for this new system along the two-directional axis indicated by [Fig. 1.c]. This is the only section that is absolutely necessary due to the fact that the optical ray tracing still exists within the remaining slice; in other words, the image still projects perfectly along the same axis at which the user will view the image(s) [Fig. 1.d]. At this point, there is a usable visual creation, made from hardware that also has a linear edge or quality to it, and this provides further options for usage.

These slices can now be arranged in a variety of arrays, creating holographic-like images that can be as small or as large as is feasible, but which will exist within the space from which the sensor(s) can retrieve and translate information [Fig. 2].

Figure 3 shows a single technical drawing example of this type of system given a set of mirrors that are curved to match a different parabolic equation than the mirrors used for our model. These virtual images could be generated at a different height according to the optical calculations. The parabolic curves of the slices of mirror used can be altered to potentially place the virtual image at some distance above the concave mirrors. In theory, one example of where the virtual image could potentially exist is being roughly demonstrated [Fig. 3]. A prototype of this example could not be made at this time due to cost considerations and its custom nature. However, the idea is presented as one possible path of many for future endeavors. The selection of parabolic curves and the specific arcs chosen are abstracted in the example presented, but they are one combination that illustrates this possibility.

3.3 Physical Structure Surrounding the PIFCU

The PIFCU system is physically illustrated [Fig. 4]. All sections are connected in some fashion to the associated computer in order to display the transferred data. The overall space in use is comprised of the sensor system observing the PIFCU, the projection system projecting visual data into the PIFCU, the PIFCU mirror array itself, and the computer used to run the main program and to capture the effects visually on the screen or API. The PIFCU is set within a table opening at the traditional 36°
Figure 3. Potential Image Improvement with Better Mirrors. The sensor is placed directly above the control unit at a height of 4 feet above the array of mirrors in order to emulate the effect of it being part of the ceiling construction, for example. The projector used to cast the bitmap images into the PIFCU is placed directly below the array of mirrors, projecting directly upwards into the device (however this is only one inexpensive solution to create the real object within the mirrors). The sensor and projector are connected directly to the computer running the associated API program, and the mirror array is connected conceptually by the user interacting with simulated holograms projected by the system within the free space (or within the GCIF) between the sensor and the image projection/creation.

As completed in this demonstration and arrangement example, the sections of the project that are adjacent to the array of concave mirrors are those two paths that transfer signals between the computer program and the imagery. The first signal path feeds data between the sensor and the running application. In this usage, the sensor used is the depth sensor, which monitors for certain changes in its view. There are also ergonometic considerations that affected the sensor readings and decisions. That is to say, we attempted to place the sensor such that it did not define the sensorized space – i.e. it simulated a completely free space occupied only by the simulated holograms or imagery. This construction was also coded such that the sensor would not be triggered until a very specific depth was reached. The depth is measured so that a user’s hands might hover within the 2” height of space directly above the 3D images and not trigger the sensor until one finger or another is lowered to reach the event-triggering distance from the sensor. The hands hovering above the 3D images, coupled with the sensor activities, are used to simulate the well-known and comfortable interactions one already has with a traditional computer keyboard. The framed view of the sensor is treated as a 2D plane for the sake of monitoring for the changes needed in this project. The PIFCU hardware is centered horizontally and perpendicular to the view of the two dimensional image and the cross-sections are taken at the center point of each of the characters in the array of mirrors. This means there are any number of points the sensor is monitoring at once, or measuring at every frame as indicated by the rate of the video stream provided. An example of the frame divisions are shown via the intersections of the dashed lines [Fig. 5].

The points that are monitored in this application are determined by dividing the height of each bitmap by two and then multiplying the width of each bitmap by 4/10, 5/10, or 6/10 for each frame. These mark the three cross-sections needed for our proof of concept demonstration, given the proportions of the given bitmap image; these translate into the same positions of the respective three simulated holograms used for the final construction. To gain better control over the 24+ frames per second, a timer is applied to the program to correspond to the time a human hand takes to press a normal button on a keyboard. This timer allows the user to use the partial keyboard of 3D projected images without causing the program to send more API commands than expected to the program interface.
mirrors at the point beneath the projected imagery where the real objects need to exist to create the virtual image above the mirrors. This is the spatial volume that will trigger the sensor to create a certain command if the user's hand gestures within the volume of sensor-monitored space.

The third major section of the project is that of the PIFCU itself, or the array of concave mirror slices set into the work table. The user in the final project interacts with two sets of three images, each set of which can be swapped in and out as the user selects the "#" virtual image key. However, these commands can control anything once detected and can be in any number, as the coding allows for additional applications. Selection of the projected virtual image control keys creates a "touchability" of simulated holograms/images existing in the freespace (or GCIF). This "touchability" refers to the instance where the user’s hand "touches" (interferes with) these image(s). The two sets of virtual keys used for the example built hardware are the beginning set {A, B, and #} and the alternate set {1, 2, and #}. The concave mirror slices are arranged linearly over a set of bottom openings, through which the images are projected. These are the same projected bitmaps as described in the previous paragraph. The framework for this mirror array is constructed at a table height, as stated earlier. The images can then be operable for a user standing before the apparatus or seated in front of it. The real objects used in the bottom opening of the mirror array are planes angled at 45 degrees relative to the plane of the work surface. These are backed with opaque vellum for the sake of this construction, but these planes may be made out of a range of materials with some variation allowed in terms of opacity, in this case 50%. This sets up the mirror slices such that the projections coming from below are cast onto the opaque surfaces, reversed within the slices, and re-created as the simulated holograms above the construction. This does, however, require the cast real image bitmap to be backwards relative to the desired resultant user view. At this point, the reversed images can appear at an orientation to read the words, pictures, or individual characters.

The code used to program this system began with the basic Kinect SDK for the depth sensor, which had to be programmed to create the API necessary to demonstrate the functionality of the project in its current state. This was a choice of time availability, however the PIFCU can be created with alternative sensors and programming. The incorporated elements of the final application were taken from an earlier version of the interfacing mock-up and included the text edit box with word wrap and all the necessary character functionality, the window holding the current bitmap used for projecting into the mirror array, the buttons within the same program that can simultaneously be used to control the bitmaps viewed, and the adjusted video feed from the sensor data. After these pieces were added, the sensor data itself needed to be handled and manipulated.

The code responsible for operating the PIFCU begins, in part, by initializing the 3D image projections, the sensor dialogue, the current state of the programming interface, the message queue used for handling the stream of commands, and the mode bits. One of the goals was to be able to switch between types of commands to allow for the case in which multiple control types were desired or the size of the array was limited. This was achieved by using mode variables/bits that are set at the start of the program. The bit(s) used here are set via a Boolean variable since there are only two modes utilized to demonstrate the control options. False is used for letter characters, and true for number variables; this is easy to adjust into other numbers of controls, but a true/false state for the "mode" was the approach for this version.

There are three sections of code to be described [Fig. 6]. First, the simulated holograms are initialized through the projection of the bitmap images. The particular bitmap used depends on the current mode that is read by the program and displayed by the bitmap window. This data is then sent through the projector until the mode and bitmap are both changed. This occurs when the user later selects the control switch indicated by the user interacting with the "#" image key. The second section of code is the sensor dialogue. After the sensor sends data back to the API, the depths at certain points in the area of view are monitored for each of the given modes. At this point two directions can be taken. For each frame of depths, the data is sent to the API to display on the view regardless of a gesture detection. Also at this point, the depths at the cross-sections are analyzed and processed to determine if there is a gesture event that should be triggered. If there is no gesture event to be processed, then the program returns to continue monitoring for future events; however, if there is a gesture detected, then the program determines the type of gesture. That is to say, the program will have marked at which location the gesture(s) occurred and maps that to a command sequence. If the command input is to switch modes via the "#" image as used in this demonstration, then the selection is translated into the API command needed (in the case of the "#" selection the mode bit is switched and new 3D images are loaded into the PIFCU), and the new message is sent to join the API message queue. The program then updates all of the associated character keys by checking the current mode settings, mapping the locations of the keys the new mode will display and the simulated holograms will update accordingly. The third code section, which performs queue processing, is always running simultaneously with the other two sections. All of the command messages are sent to pass accordingly.

![Figure 6. System Flow](image-url)
through this third section and it is here where the screen view is updated if needed after the sequential commands are executed. The program continues to retrieve the next command to execute, executes the command, and updates the screen view for the user. Depending on the specific command, the program will update the view, print the desired character to the text box, or switch modes involving a window and bitmap change. At this point, if the program is not terminated, it continues to monitor for changes and execute commands as directed.

In terms of the hardware used, one area that was limited in the final project was that of the components structuring the control unit. As previously stated, it used three sets of facing, double-curved, concave mirrors, a single depth sensor, and a projector. This type of hardware is expensive, so these decisions were based on the limitations of the sizes and quantities (mainly those of the mirror types needed) that could be used for the built project demonstration. Regardless, there were enough resources to build the project, demonstrate, and validate it as a proof-of-concept. The final built project was enough to establish an example of an array of three adjacent mirror slices and how this can create a series of projection controlling mechanisms. A sectional array of mirrored slices yielding an array of virtual images is the extent of the built hardware at this time. Given enough time and funding, the slices could continue in any number, direction, overlap, scale, distance, proximity, be curved based on any parabolic equations, and/or the projected imagery could be at varying distances from the invention [Fig. 3]. The functionality of the PIFCU can be added as a layer to many existing applications to improve use and educational exploration. Further work would allow the simulated holograms in the volume(s) of space they occupy to be static, operable to be switched by the user, operable via voice commands, combined with voice commands, be self-updated, changed in real time, or via video feed, and so on. Additionally, more sensors and ones of different types can be engaged to create the same effects, as the application desires.

4. Analysis of the PIFCU System

4.1 Results and Benefits of the Current Setup

The current state of the system [Fig. 4] offers the most clear depth change for the sensor to read relative to other attempts made as far as arrangement of hardware is concerned. When pointed directly downward onto the PIFCU, the position gives the sensor enough layers of data to be able to distinguish the range of depths that will trigger an event in the program. The sensor can get these layers from the table level and different components of the control unit, assuming it is at a distance larger than the 1 meter minimum the Kinect sensor requires for accuracy. This also allows the field of view of the sensor to be simplified, which increases the effectiveness/accuracy of the set event range. At this point, the ability to hover with one’s hands over the simulated holograms is possible with fewer errors in depth detection events than the other states allowed. It also creates an environment where the sensor events are not triggered unintentionally due to the existing desire to hover one’s hands over a keyboard. The relations of the components as seen in the final project also allow the field of view of the sensor to be more stable in terms of data retrieved because of the fixed background. In this case, the fixed background means the elements under the field of projected imagery, including the table it is built into that has a fixed location/height. Thus, the only movement affecting the sensor data is that made by the hands of the user, giving the program as little cause for false event-sensing as possible.

Since one of the main goals was to offer an unbounded freespace from which to utilize simulated holographic controls that are not confined to a small box for their existence, the final state of the project was selected because it solved this problem. For many applications, this system can be constructed into built/human-occupyable spaces. This allows ample space around the images, and if the sensor part of the system were installed into ceiling constructions, for example, then there would be even less negative interference. The other components could be concealed or built into the space of a traditional work table, as in this implementation. This arrangement is also very effective as the existing application-replacing a keyboard, or those similar.

The source of light in this system creates a situation in which the device can be seen and used in both light and dark rooms. The placements allow for this function when the PIFCU is placed between the user and source of the light projections. In this state, the user is viewing the projected images created from the light source after they are passed through the concave mirrors. This light source is projected toward the GCIF and reflected upward from the control unit into the user’s view. The cast light is coming from the space created within the mirrored interior of the control unit, thus, this space is lit up regardless of the lighting conditions outside of this construction. This is why the simulated holograms can still be seen and used if the room in which the system is installed is either light or darkened. Other tested arrangements of the hardware components would not necessarily maintain this feature, which in effect is actually incredibly useful, depending upon the context.

4.2 Discussion of Alternate Configurations

Other hardware configurations [Fig. 7] were attempted prior to achieving the current state. One alternative had configured the components by keeping them all at the same horizontal level of the table space, or as close as possible to this. The projections in this instance could still be successful and would still bounce the light into the control unit to allow for use in the light and dark. Depending on how the bitmaps are reversed, the image source may need to be tilted or key-stoned to be captured correctly by the mirrors. While this does not hold back the image quality or
usage, it does take up far more horizontal area (table space), which is problematic with limited square footage. Sensor placement in this example is more problematic. The sensor can be programmed to trigger events based on different ranges of distances, since the imagery would appear at increasing depths. The problem arises with the space to the left of the image accounting for the background of the sensor view. Since the space is not bounded by a statically placed plane, there were far more instances of visual noise that confused the sensor and made it very difficult to determine depths consistently. Furthermore, the depths of this background may not be within the range that this particular sensor can manage, making the depths of user interactions null or inaccurate relative to the sensor. Thus, the system will not work as intended in some configurations. Even if the background is within range to be interpreted correctly, there is also too great of a possibility that people will come into the space and/or lighting will change within that background space, decreasing the effectiveness of the sensor data collected or ruining it altogether.

Another significant option for the arrangement of the hardware is best understood via an elevation view, showing how the components may be stacked on top of one another. This alternate state has the same sensor problems as described above for the sensor position. However, the sensor was not the focus for this attempt; the largest change is the position of the projector creating the image source. Instead of placing the projector below the control unit, it is placed either to the side or beneath the sensor, projecting the light from the right (or left) of the PIFCU array. This reduces the horizontal area for a single workspace, but only to twice the work area of a standard computer station today—not enough to be an efficient use of desk space. Also, while it is still possible to cast light from this direction, it is more difficult to use cast bitmaps from this angle relative to the location at which the real image needs to be produced for the respective simulated holograms. Thus, alternative states were explored.

Yet another alternative state was tried in order to solve the problems presented in the previous examples. To improve the problem of too much space used to set up the system, the projector was now used directly below the control unit. It solved this problem, as well as created a position of the hardware that allowed it to be constructed within a table workspace. With this problem solved, the sensor placement was now considered more fully. At first, the sensor was left to the right or left of the control unit as in the previous states, but now with the image light source from below. This, of course, maintained the same problems as before, so another change was necessary. For the sake of exploration, the sensor was moved to observe the control unit from behind, now facing the position of the user while the projector remained below. Now, the area the sensor is observing would include the user, which could eventually benefit the system if this same sensor was used to identify other face and hand gestures. However, the depth sensor was not proving to be accurate enough here because the depths were being compared and measured against the constant motion of the user on the other side of the control unit. Given the problems having the user in the view of the sensor was causing, another change was necessary to be explored to achieve the benefits of the system and overall goal of the thesis project.

Finally, the current state [Fig. 4] of the project was reached; that meaning the state with the sensor placed directly above, and the projector remaining below, the control unit/array of mirrors. This solved the immediate problems as described in this section, and offered the benefits for the overall system as delineated in the following section.

4.3 Future Improvements Currently Possible

There are a couple equally successful options for the placements of this hardware, assuming the expenses are not a limiting factor. There may be more ways than are noted in this section, but the few discussed here would offer particularly interesting or useful improvements to the system or its functionality.

One alternate way to achieve this type of system from the existing components would be to omit the projector and create the real objects in a way other than projecting images onto an opaque surface. For example, since the real objects are beneficial if able to change over time, whatever makes these real objects that turn into the simulated holograms need to be computerized and therefore be programmable. These could be created with a mini projector or better yet with a series of miniature LED screens all contained within the PIFCU. The screens would be fed from the running program or pre-programmed with the data necessary for the body of imagery/controls desired. This would completely omit the need for a projection device exterior to the PIFCU, and it would improve performance. Although this would have been an initial organizational scheme to produce, it would have also increased the cost of the system beyond feasibility at the time.

Another interesting path that could be explored if the funds were available would be that of varying numbers and types of sensors. While at least one sensor is necessary to create a functional and interactive system, additional sensors of the same or various types could be used for adding features. Each sensor could be responsible for a different set of controls - one for the control unit, one for other gesture recognition facing the user, and so on. Also, other types of sensors could be used instead to provide more control over
all of the programming. Furthermore, instead of using an external sensor, the whole system could be made more portable by using a single small sensor or array of small sensors (depth or otherwise) installed from inside the control unit or as part of the overall control unit and concealed within the protective barriers. With both of these changes, it would now be independent from building constructions or tables that the current configuration is using.

5. Conclusion

Although this project is successful thus far, it depends greatly on the quality of the equipment used, the construction of the hardware apparatus, the availability of the hardware, the structure of the surrounding spaces, the type and precision of the sensors, the type and precision of the mirrors, and the specific programming languages used. Future work will consider the highest quality equipment available, especially for the image creation, since that directly affects the quality of the imagery produced. The results of this project have the potential to improve any application that could benefit from the incorporation of a visual component; this simulated holographic element is currently found in very few places. This project was also constructed so it could alternatively be incorporated with larger systems, built or otherwise. In particular, the size and shape of the PIFCU and GCIF can change size or repetition to meet the needs of a user; this array of mirrors can be of any size, shape, parabolic equations, number, location, and/or command type desired. Future work may be inclusive of any additional features; these may more immediately include additional API/GUI functions, different gesture recognition, voice-recognition features, other placements, or various incorporations into existing applications and building systems. Whatever the chosen direction for future investigations, the current project is meant to be a demonstration and proof-of-concept for a functional system and ground work from which to compare and contrast future variations and additions.

References