

3D Sensor-based Library Navigation System

Abdel-Mehsen Ahmad*, Zouhair Bazzal, Roba Al Majzoub, Ola Charanek

Department of Computer and Communication Engineering, Lebanese International University, Lebanon

ARTICLE INFO

Article history:

Received: 15 May, 2017

Accepted: 15 June, 2017

Online: 30 June, 2017

Keywords:

Gesture-Based Control

Kinect

Processing Software

ABSTRACT

The discussed system in this paper uses the Kinect's 3D input (acquired through IR projector and Camera) to detect gestures and then based on these gestures perform certain tasks, and display certain results on an output screen. The programming language used here is the Processing open source software and a database was implemented to manage the data required within the system. The designed system was created for a library but the system's design is broad enough to allow its utilization in many more domains. Implementation, system screenshots, simulation and results are discussed along with testing the system in different condition of light and distance. Some simple additions have been added to this version of the program to facilitate the interaction between the user and the system further.

1. Introduction

The advancement of control systems that has taken large leaps in the past decade has raised human-machine interfaces to a whole new level where they became more and more remote from the user's body like voice control, eye control and body and hand gesture control which have changed the concept of human-machine interface from the typical and complex concept to become more natural and human-like, also known as Natural User Interfaces or NUI. Most of the new systems are trying to enhance the user interface further more so that a larger number of people can be able to use the new technology with less effort and time consumption. These types of interfaces allow users to bypass the limits of age, education and sometimes physical disabilities to have a better control over technological systems which have become an undisputable need for us on daily basis.

The system to be discussed implements gesture recognition to control the system where users interface with it remotely using simple gestures to navigate the system. With all the possibilities available for gesture detection, the most reliable method is 3D imaging, where the gesture is no longer dependent on the 2D properties of the hands, like color, shape etc. to determine and recognize the gestures of the user, which is very expensive in terms of processing and may lead to delays within the system when the user is expecting real time feedback.

Although the system was designed for a library however its possible implementations can go way beyond the walls of libraries

*Corresponding Author: Abdel-Mehsen Ahmad, Department of Computer and Communication Engineering, Lebanese International University, Lebanon
Email: abdelmehsen.ahmad@liu.edu.lb

into museums, hospitals, malls and many more applications, it can also aid elderly or people with special needs control many things from electrical appliances to electrical doors, applications etc... It will follow the hand gestures of the user to allow him to explore different books with different genres and authors available within the library, where the books will be classified into topics, and within each topic related books available will be displayed. Further information about each book (including author's name, ISBN, a short summary about the contents of the book and its location within the library) can be displayed to the user. This paper demonstrates the design and implementation of a 3D gesture-based control system that was previously discussed in 2016 2nd International Conference on Open Source Software Computing (OSSCOM) [1].

2. Similar Systems

2.1. Maintaining the Integrity of the Specifications (Heading 2)

Similar control systems were created using different techniques. A "Wireless Gesture Controlled Robotic Arm with Vision" was presented in [2] using accelerometers that allowed the user to control the robotic arm wirelessly, by using 3-axis DOF's (Degree of Freedom) accelerometers controlled via RF signals. The robotic arm is mounted on a movable platform to signify the hand and leg of the user, where the robotic arm correspond to the user's arm and the platform corresponds to the user's leg. Accelerometers are fixed on the user's hand and leg to capture their gestures and movements allowing the robotic arm and platform to move accordingly.

Another gesture controlled user interface called “Open Gesture” was proposed in [3]. A web cam is used with an open source program to implement the system in smart TVs for almost all users. Although the system can be used by most people, still it faces some drawbacks; the main tool for capturing gestures is a web cam, meaning that to identify gestures, image processing is required, and that is computationally expensive and will not allow the system to receive the same quality of information obtained by 3D scanning. Another drawback is the fact that the web cam is light-dependent and will not work in all lighting conditions, where dim lit and unlit places create a challenge for the system to identify the gestures performed. The system also assumes that all users are of a certain educational background, and that is not true, so the system is limited to these people only. Even though the system we propose in this paper is for a library, however the gesture control can be performed over anything and almost everywhere we want with some small changes.

Another approach was presented in [4], which uses the WiiMote from Nintendo to control smart homes (appliances, lights, windows...), where the 3D accelerometer will be used to detect the gestures and a classification framework composed of machine-learning algorithms like Neural Networks and Support Vector Machine (SVM) will be used for gesture recognition classification and training. Although this system has a high level of accuracy for gesture recognition due to the use of the classification framework, a setback lies in the use of the WiiMote, where it restricts the gesture recognition for the user to only the time when the controller is held by hand, which after long periods may result in inconveniences for the users especially the elderly.

Yet another gesture controlled system was presented by [5], where Kinect’s depth imaging was combined with traditional Camshift tracking algorithm along with HMM (Hidden Markov Models) for dynamic gesture classification to control the motion of a bi-handed Cyton 14D-2G. A Client/Server structured robot tele-operation application system is developed, which provides a favorable function of remotely controlling a dual-arm robot by gestural commands.

3. Proposed System

3.1. Design Specifications

From the many choices of NUI we chose to utilize hand gestures because they are familiar and universal to all users where even small children learn to use gestures before they even learn to talk, and so this supports the fact that this system can be compatible with all backgrounds and ages of users.

Gesture recognition was accomplished with the help of the Kinect, which is a line of motion sensing input devices that includes a 2D camera along with a 3D scanner system called Light Coding which employs a variant of image-based 3D reconstruction that requires the work of an IR projector and IR camera.

Some of the great advantages of using the Kinect sensor are the fact that its price is affordable and that it can catch great depth images in different lighting conditions like dim lit or unlit places as well as places with good lighting. This is a property not

available in any of the other 2D image capturing sensors. Another important factors is the fact that Kinect uses 3D imaging instead of 2D imaging, which is independent of the light (as previously mentioned) and thus does not require image processing and resolves the issue of computationally expensive algorithm we would have been forced to use, making the system more robust and increasing gesture detection therefore increasing the speed and dependability of the system’s response to the gestures.

The Kinect will be connected to a computer with a Processing program designed to detect gestures and respond accordingly to them. Since we only needed the 3D imaging of the Kinect, 2D images were not used and thus the real time streaming of the surrounding environment was done in grey scale images of the Kinect sensor.

The system design targets the majority of people, meaning that users’ age, physical abilities and educational backgrounds will vary, so we had to take into consideration the controlling gestures since they had to be performable and known to all users with ease and without resorting to difficult physical actions that on the long term may strain the users. The system was only programmed to respond to two types of gestures: hand wave and hand click, where the wave gesture was used to initiate the program and then when inside the program it was used to go back to the previous screen/menu while the click gesture was used as a selection gesture.

The programming language used was Processing, which is a free open source programming language and runs on Mac, Windows, and GNU/Linux platforms. Processing has many libraries related to Kinect, and because it was easy to learn and because it is an object oriented programming language, it allowed us to have more flexibility when dealing with Kinect input and manipulating information and output.

Data were stored in a database that can be updated according to the needs of the system where the program fetches required information from the database depending on the type of gesture detected. If the gesture is a click for example, then the system will fetch all the information related to the object clicked on by the user. And so for that purpose MySQL database was used to store the information about the books and the other information to be displayed within the program.

The following figure (Fig.1) represents the interaction between hardware components of the system designed:

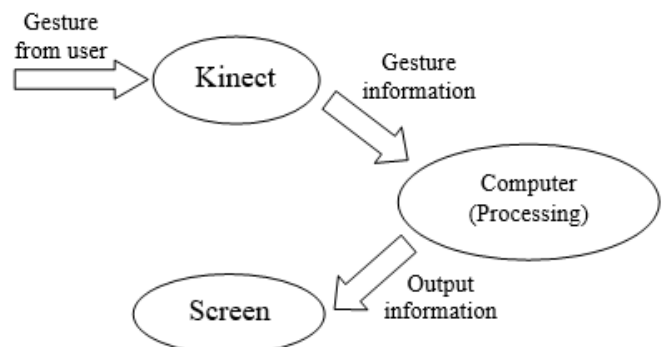


Figure 1 Hardware components of the system

4. Implementation

The system developed is an event-response application, where the user's gestures are the events that change the state of the system (the displayed information and the navigation through the system) to trigger the response to these events allowing him/her to control the system.

To allow the system to understand gestures we needed a gesture recognition algorithm to start with. So we decided to use a pre-made gesture recognition program that was available to save time and effort and to focus on the main idea of the project.

Although two approaches were available for us to control the system, one was 2 dimensional (2D) and the other was 3 dimensional (3D) depending on the algorithm used for the detection of gestures, however the 2D approach proved to have poor reliability due to the delays it showed within the system and the program's inability to follow the users hand and fingertips proficiently, thus it was halted and the 3D approach was adopted due to its good response to the users hand movements and gestures adding to the reliability of the system.

4.1. 2D Approach

The Finger Tracker library was our first alternate for gesture detection where it detects the tips of the fingers of the user. It does that through slicing the space detected by the Kinect into layers of 2D surfaces and then determining the fingertips through scanning each 2D surface and finding "inflections in the curvature of the contour" then drawing red ellipses in their positions, as seen in Figure 2.

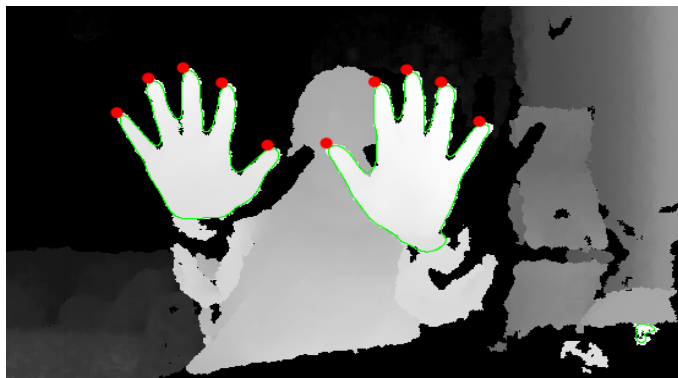


Figure 2 Fingers tracked using FingeTracker library

Finger's gestures and movements were used as the events that control the system. So we first eliminated the depth image (the grey-scaled 3D-image captured by the Kinect seen in Figure 2) and only drew the contour of the hand along with the fingertips (fingertips displayed in red and contour in green in Figure 2) with a rate approximate to 30 fps (frames per second), which is the frame rate of the Kinect, on a black background, and like that we were able to observe the real-time motion of our fingertips on screen. Next, fingertips previously drawn as 2d images were replaced by special 2D graphics known as "Sprites" as displayed in Figure 3. Sprites are 2D images that have special properties with boundaries and libraries that allow us to detect the collision between one sprite and the other, set their properties, motion, speeds and so much more. The library used here is called Sprites library, and allows us to control all the previously mentioned properties of sprites.

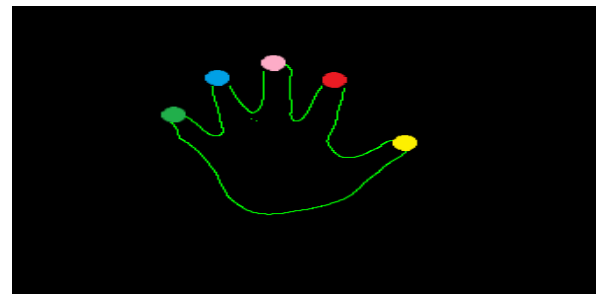


Figure 3 Sprites inserted on the fingertips

Images of books were inserted as sprites as well (see Fig.4), so to be able to detect collision between them and the fingertip sprites, an algorithm was created so that certain actions can be executed upon detection.

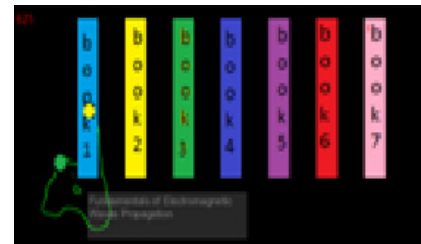


Figure 4 Book information displayed upon the detection of collision between a fingertip sprite and a book sprite

Two events were defined to control the system and change its states, a pointing gesture and a pinching gesture. A loop was used to iterate over the seven pictures given, then a condition checks for any collision between any book and a predefined fingertip (detected when the user is pointing at the book) which is shown using the yellow sprite in Figure 2. If the condition is satisfied, the name of the book that the finger sprite collided with will be printed on screen (see Figure3). Then the loop iterates over the books again to check for collisions. To select the required book two fingertips were used, such that if a pinching gesture was performed and the distance between the boundaries of the finger sprites was smaller than a certain threshold and the collision between these sprites and the book sprite was detected information about the selected book will be displayed, thus the system would transit from the initial state where the books are displayed to either state depending on the detected event; a pointing gesture results in displaying the name and a pinching gesture results in displaying the information. The following state chart displays how the system transits between states based on the events created by the user. (Figure 5)

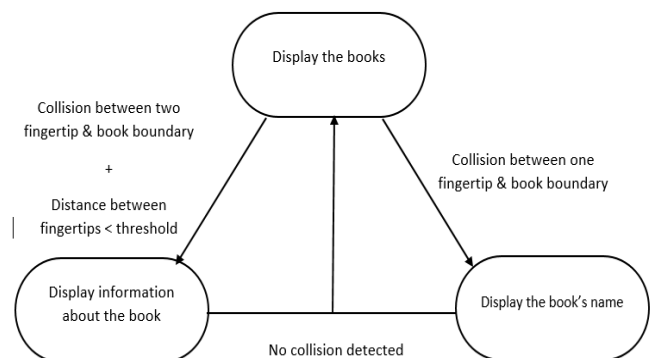


Figure 5 State chart that shows the transition of the system between states based on events created by user

Figure 6 displays how the software components in this approach work with each other:

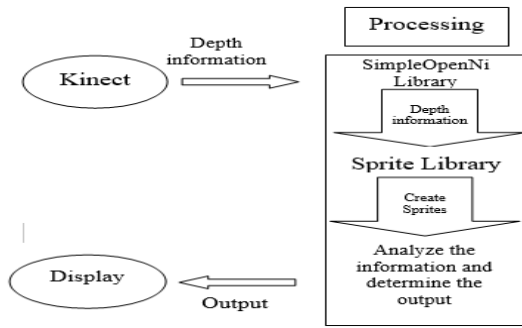


Figure 3 Software components of 2D approach

4.2. 3D Approach

The other available option for gesture implementation and detection was within the SimpleOpenNi library which is the “Hands” sample code that allows us to detect three hand gestures: Wave, Click, and Hand Raise which will be used as attributes for events to change the state of the system (and therefore change the displayed output). Upon detecting any of the three gestures, the detected hand is given an ID number between 1 and 7 and a unique colour is assigned to it where a circle of that colour is drawn in the upper middle part of the user’s palm and displayed on the screen, as can be seen in Figure 7.

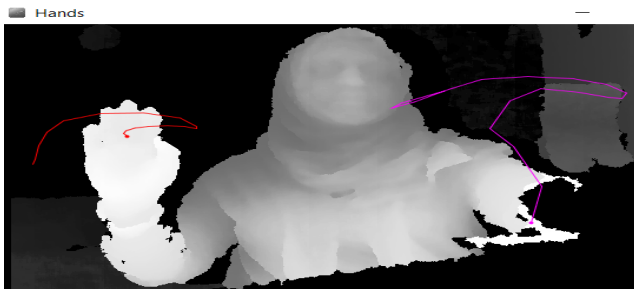


Figure 4 How hand code displays detected hands after gestures

Each gesture is assigned an integer number from 1 to 3, where Wave = 0, Click = 1, and Hand_Raise = 2. Because our system only needed the presence of two gestures (one for selection and another to go back to the previous menu) the click gesture was used for selection and the wave gesture was used for going back. Notice that the system focuses on detection of hands only, thus the user’s profile, age, physical abilities and educational background are of no importance, as long as the user can use his/her hands effectively in front of the Kinect.

Then the system was programmed to respond to a certain gesture in a certain position where the position is the other attribute of the events that allow us to change the state of the system and therefore navigate forth and back through them, so that if the user wanted to select a certain topic or book all he had to do was to click on the image of that topic or book and after getting the required information, a simple wave of the hand allows the user to go back to the topics menu or to the collection of

displayed books, allowing him to navigate back and forth within the system.

Meaning that the events here have two attributes; one is the gesture type and the other is the position where the gesture is performed, so upon the completion of the event when satisfying the conditions, the system transits to another state where the information matching the completed event are displayed.

A small description of the system along with some screenshots will be displayed for better understanding of how the system works.

The user faces a Kinect sensor and a Screen that will display to him the program he will interact with. The screen displays the user’s depth image, this shows that the program still isn’t functional till the user waves his/her hand, then a “START” button will be displayed in the middle of the screen (as seen in Figure 8) announcing that the program is ready and waiting for the click of the user.



Figure 5 After a wave of the hand the start button is displayed

Next the user will “click” within the boundaries of the start button opening the main page of the system where empty shelves and the topics of the available books are displayed on the screen. Figure 9 displays the state chart of the system, where each event allows the user to go back or forth within the system, changing the state and therefore the displayed output of the system.

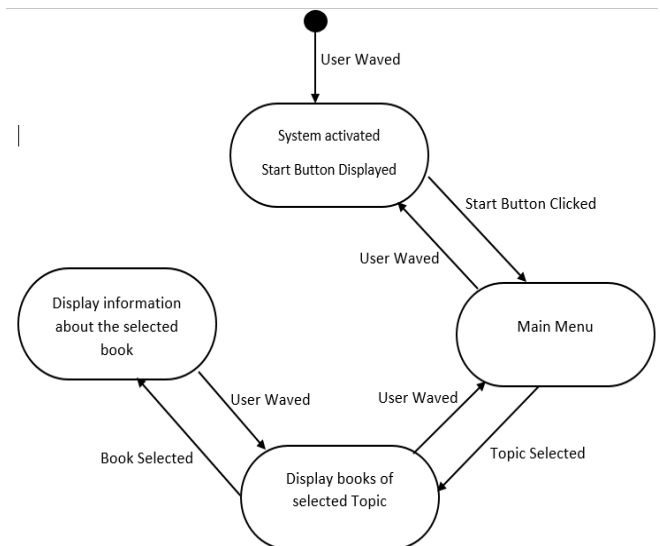


Figure 6 State chart displaying the overall states of the system and how they changed based on the user’s events (click and wave)

As the user clicks the start button, a new screen will appear with the topics of the available books displayed on the left side of the screen, and empty shelves in the middle of the screen, as seen in Figure 10. Here the condition was to perform the click gesture within the boundary range of the start button.

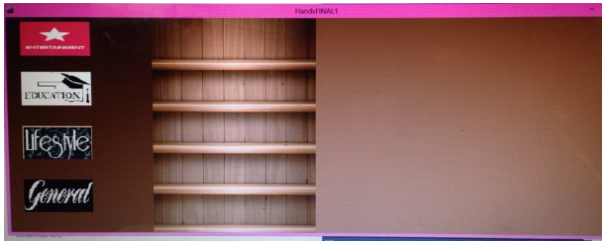


Figure 7 Main screen displayed after clicking the start button

Next the user can select any of the displayed topics he/she is interested in through a click gesture on the required topic.

As a response to this gesture pictures of the available books concerning this topic will be displayed on the empty shelves as seen in Fig. 11.

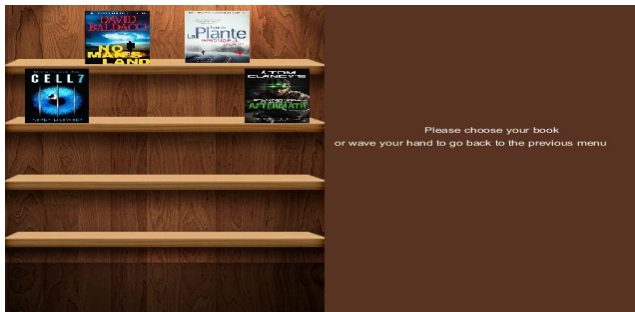


Figure 8 Books displayed on the shelves

With a simple click on the book a larger image of it is displayed along with information about it (author, ISBN, summary, position within the library...) on the right hand side of the screen as seen in Figure 12.



Figure 9 Information displayed about the selected book

Going back to the previous menu can be done with a simple wave of the hand. The same gesture can take us back one step wherever we are within the program, as seen in Figure 11.

Figure 13 displays the software components of the 3D approach of the system.

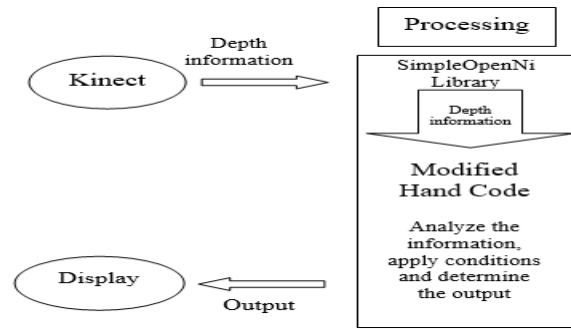


Figure 10 Software components of the 3D approach

4.3. Database Implementation

MySQL database was created to store the attributes and information and picture paths of the books to be displayed on screen, and information were retrieved from the database, using queries triggered by events created by the user.

For the system to retrieve the data a function was created which returned the desired information as a string array. The function then

- Performs a query: Selects all rows from the specified table
- Iterates over all of the selected rows
- Gets the "attribute" value of each entry
- Increments the loop to hop to the next position in the array thus retrieving the paths of pictures and books' information (summary, author...)

This implementation makes this control system quite broad where if we wanted to change the application field of the system from a library management system to another different domain, then the users must only supply the database with the new information to update the tables, so the system can then perform different actions (display images, control devices etc...) thus extending the range of the system's application field.

5. Simulations and Results

5.1. Test Cases and Acceptance Criteria

To consider the system as acceptable for interfacing it should satisfy the following criteria:

- Be able to detect gestures without too much effort from the user, and without any delays.
- Be well organized so that the user wouldn't be confused about the information or about how to select items.
- Use gestures that are easy and familiar to people so that all users would be able to perform them without effort.
- Limit the range of gesture performance such that it wouldn't be too wide, so that the hand movements wouldn't strain the user, and they must be swift and not involve too much movement.
- Have a high percentage of accuracy and fidelity.

The tests were performed on both approaches mentioned before, the 2D and 3D approaches.

5.2. Simulation Results

Although the 2D approach involved finger gestures that have become quite common among most people due to their use in touch interface (smartphones and tablets etc.), the continuous flickering of the fingertip sprites and the fact that the detected IDs of the finger sprites were inconsistent made collision detection much harder reducing the accuracy of the system significantly, leading to considering the approach as invalid.

The 3D approach underwent many tests to validate its efficiency, fidelity, real time response to users' gestures and the response of users towards the used gestures. The results proved the system to be well compatible with the criteria mentioned above. The range of gesture performance has also been detected to be good and reachable by the majority of the users who have performed this test which included users of different ages and different educational levels. The gestures used, namely the waving and clicking were very well detected and responded to, on condition that the user faces the Kinect head on without any tilting and without any obstacles and that the Kinect would remain stationary after calibration. The accuracy has increased significantly from the first prototype, and fidelity was very good, where users were able to go forth and back while navigating within the system.

The figures previously displayed were screenshots captured during the simulation of both approaches, where each of them displays exactly how the system responds to the actions of the user.

Statistics were made based on new review from test users of the system, where two surveys was conducted concerning the concept of the system as a gesture based control system, whether users had previously used similar system and if they would buy such a system themselves (see Figure 14). The survey was done on social media and on field with test users.

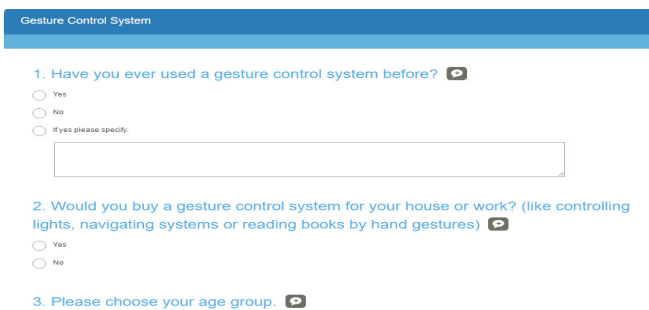


Figure 11 Online survey

The results of the survey were as follows: 64% of the users stated that this was their first time using a gesture controlled system while 36% stated that they have a similar previous experience (Xbox, telephone), 72% of them stated that they would buy such a system to use in their daily life while 28% declared that they would not. (Figure 15)

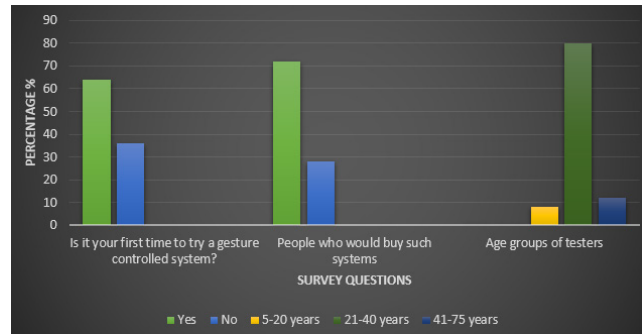


Figure 12 Statistics of Surveys

Although the feedback of users varied about buying such a system or not, but all agreed that the system was easy to use and learn, and that the gestures were fairly simple to perform. While the wave gesture was the easiest to perform with a 100% on its performance and real time response, the click gesture was more challenging where the response to the gesture was at 50% before making the Kinect stationary, but rose to 90% after fixing it to a specific location, making the gesture detection easier and the system response to users better, where the optimal distance for the user to be at is between 2 and 2,5 meters away from the Kinect with the Kinect stationed at 1.5 to 1.65 meters high.

The system was also tested for other gestures that are not defined within the program, but there was no reaction from the system whatsoever, so the system only responds to predefined gestures (the ones mentioned before).

The size of the population was increased from the previous paper, where about 20 people previously tested the system, and now the test subjects were doubled so the responses were more various and the feedback was more accurate.

The ground truth for the gestures used for us as programmers was obtained through tests, and then when the system is presented to the users, a tutorial is given to make sure the gestures and how the system works are both clear to the users, and after the tutorial, users adapted immediately to the system and the feedback was very satisfactory. These are some of the users' feedbacks on the system: "Gestures are easy to learn and perform", "The wave gesture is like saying hello and goodbye", some noted that the tutorial is very important for controlling the system properly, and so within each display window instructions were included to help first time users understand what to do while navigating the system. (See Figure 16)



Figure 13 Directions displayed to help users

A better step would be to include a video tutorial about the gestures to make sure that even people who can't read can still understand and use the system.

To test the performance of the system under different conditions we decided to perform the test with a single user but on different distances, and in different lighting conditions.

The evaluation of the response was classified as Excellent (100%), Very Good (85%), Good (70%) and Bad (0), and the distance started from 1.5 m from the Kinect to make sure that the user's hand wouldn't extend to its blind spot, where the range of detection starts from 1 m and extends to almost 6m away from the Kinect.

As for lighting conditions, the tester performed the gestures outdoors during daytime in an open space, where there was a lot of sunlight, then indoors but with sunlight as well, then the amount of sunlight was decreased and instead electric lighting was used mostly, then the lights were 85% dimmed and finally the lights were completely turned off and the sunlight was blocked to get the darkest possible environment.

Figure 17 displays the response of the system under the previously mentioned conditions. As we can see in the figure, the response of the system is optimal between 1.5 and 3 m and precisely at 2.5m which is the optimal distance for the user in all lighting conditions, however when the distance increases outdoors the response becomes very bad due to the large amounts of IR light present in sunlight which can affect the performance of the sensor and decrease its range of response. The performance gets better as the amount of sunlight decreases, however in the remaining lighting conditions it proved to have a good response till about 4m from the Kinect, then the response decreases and stops after 5m. But when all lights were blocked, the response was much better and the range increased to 6m which is the range of "vision" of the Kinect.

longer limit the educational background or age of users and it can be used by the majority of people.

The accuracy of the system was very good, however for better results new algorithms can be created specifically for the purpose of this system, which would enhance the accuracy and hence the overall system response.

6. Conclusion

The simulations of the system and the user feedback about the system have reflected a good response, as long as the Kinect remains stationary after calibration to ensure the best results. The simplicity of the system can allow it to go into various domains for control over software or hardware and so it can be implemented anywhere to control systems easier, however it can be improved through further studies and more specialized hand detection algorithms.

References

- [1] Ahmad A, Al Majzoub R, Charanek O, 2016, '3D Gesture-Based Control System using Processing Open Source Software', in A Ahmad, R Majzoub and O Charanek, , *Beirut 2016 2nd International Conference on Open Source Software Computing (OSSCOM), Lebanese University.*
- [2] Aggarwal L, Gaur V and Verma P "Design and Implementation of a Wireless Gesture Controlled Robotic Arm with Vision", *International Journal of Computer Applications*, 2013, Volume 79 - Number 13
- [3] Neßelrath R., Lu C., Schulz C. H., Frey J., and Alexandersson J., "A Gesture Based System for Context-Sensitive Interaction with Smart Homes", *Ambient Assisted Living*, Springer, 2011, pp 209-219.
- [4] Bhuiyan M. and Picking R, "A Gesture Controlled User Interface for Inclusive Design and Evaluative Study of Its Usability", *Journal of Software Engineering and Applications*, 2011, 4, 513-521.
- [5] Qian K , Niu J and Yang H, "Developing a Gesture Based Remote Human-Robot Interaction System Using Kinect", *International Journal of Smart Home*, 2013, Vol. 7, No. 4

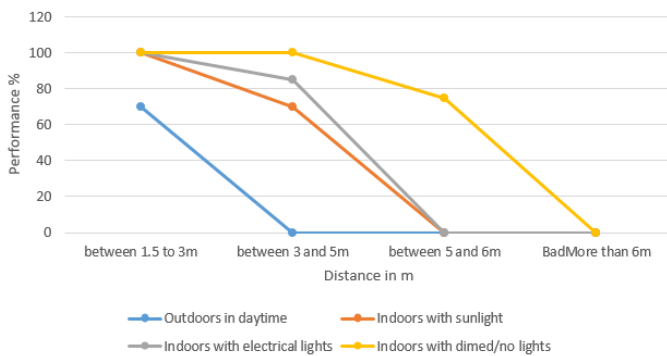


Figure 14 System response under different lighting conditions and distances

Another problem we faced in this implementation which was the calibration of the sensor distance from the user, where any change in the distance between the user and the sensor had greatly affected the system. This can be resolved by keeping the sensor stationary. Since the control system created was implemented in a library, then the educational background of the user is limited to a certain level, and age has its limits as well, however this type of control system is gesture dependent, meaning that if implemented in other domains to control things, like hardware, then it will no