Interactive Projection Using 3D gesture Recognition

Parth Panse, Chinmay Paranjape, Atharva Pore,

Prof. Parth Sagar

Abstract—Gesture Recognition is the analysis of human ges- tures and interpreting the meaning behind them. While primarily modeled for eHCI, they also lend a helping hand in terms of iHCI. Gesture Recognition can be performed in various ways and by using various devices. The devices are broadly categorized as Input devices, which generally require user to hold or operate using a physical device, or they may be touchless devices which support natural gestures made by human motions. The proposed system provides a natural and intuitive way to interact with a projection using the principle of 3D Gesture Recognition. The system uses Skeleton Tracking in accordance with depth sensing, and the gestures are recognized in terms of the relative movements of the skeletal joints with respect to each other. Based on the gestures performed by the user, the system will perform the required action accordingly.

Key words —Artificial Intelligence, Computer Vision, Gesture Recognition, Kinect, Skeletal Tracking, Depth Sensing

1 INTRODUCTION

Artificial Intelligence is the ability to make machines think for themselves and make decisions accordingly without any explicit input from humans.It still remains one of the most expansive subjects in the field of Computer Science. The main advances in the past sixty years have been mainly in algorithms of advanced searches, machine learning, statistical analysis, heuristics, and some others. Recent decade, however, has experienced a surge in AI due to various technological advancements.

The domain of artificial intelligence is very diverse in every aspect. While broadly speaking, some common areas of research in AI are Expert Systems, Neural Networks, Fuzzy Logic, Natural Language Processing, Robotics, Speech recognition and synthesis, and Computer Vision.

The following paper mainly focuses on Computer Vision and its aspects related to Gesture Recognition. Computer Vision aims to bestow machines with high-level understanding of the digital images or videos. It extracts the images from the environment, analyzes the images, and processes them in order to produce information and make decisions based on them.

The paper is organized as follows. Section I introduces the concept of artificial intelligence along with gesture recognition. Section II provides literature survey for understanding the different systems. Section III provides and overview of Kinect. Section IV provides an explanation of the system architecture for the above mentioned kinect . Section V explains the

them, and analyses them in order to produce information or make any decisions.

The paper is organized as follows. Section I introduces the concept of 3D gesture recognition. Section II provides litera- ture survey for understanding the different systems. Section III describes a general system architecture of 3D systems. Section IV summarizes the brief working of discussed systems. Section V explains the algorithm for gesture recognition. Section VI briefly details some applications for 3D gesture recognition. Finally, Section VII concludes the paper and Section VIII provides acknowledgement.

2 LITREATURE SURVEY

Gestures are extensively used as a medium of non-verbal communication.Sign language can be considered as a language which uses some gestures that involve hand orientation, posture, shapes, as well as body movements and facial expressions which are used to convey user's thoughts, as opposed to acoustic sound patterns. Among various forms of gestures, hand gestures are most widely used as a simplest medium of sign language based communication framework. Hand gestures also serve as a powerful tool to interact with the computers, as opposed to unintuitive interfaces.Moreover, Recognizing gestures as an input allows computers to be more accessible for the physically-impaired and makes interaction more natural.

Various techniques have been developed to extract information from the performed gestures. Broadly speaking, the gesture recognition is done using either by controllers such as accelerometers, wired gloves or gyroscopes; or some hands- free methods such as Stereo cameras, or 3D depth cameras.

Some types of depth sensing devices are :

1) Stereo cameras

2) Time - Of - Flight

3) Coded Aperture

4) Kinect

5) Leap Motion Controller

3. KINECT

A. Overview

People prefer a more user-friendly and intuitive interaction in controlling a machine or a device. Gestures create more attractive form of interaction than that of traditional devices like keyboard, mouse, and joysticks. Human body interaction, along with voice interaction, create a Natural User Interface(NUI), which can be called as an evolved form of Graphical User Interface(GUI).

Kinect is an RGB-D Sensor that provides color and depth images along with their respective information. It was designed by Microsoft that initially served as an input device for X-Box Console. By using 3-D Gesture Recognition, it enables interactions without the need for any physical controllers[1]. Recently, the computer vision society has discovered the depth sensing technology of Kinect[2] and observed could be extended far beyond gaming and at a much lower cost than traditional Stereo cameras[3] or Time of Flight Sensors[4].

B. Hardware

The hardware of kinect basically comprises of a normal RGB camera, an infrared depth sensing camera, an infrared projector, a microphone array, and finally a motor for tilting operations. With the aforementioned hardware, kinect is able to process RGB images, Depth signals, and audio signals simultaneously.

- RGB Camera: As the name mentions, the RGB camera provides three color components of the video. The RGB camera has a resolution of 640X480 pixels, and a refresh rate of 30Hz. It is also possible to record a scene at a higher resolution of 1280X1024 pixels, although the downside being that frame rate is reduced to 10 frames/sec.[5]

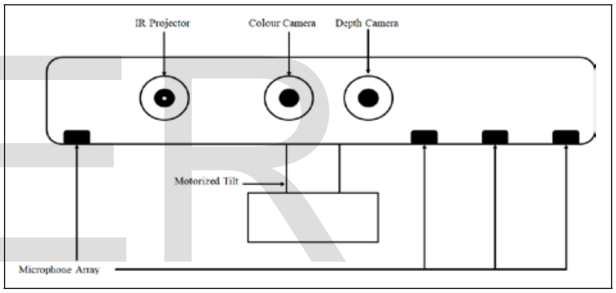


Fig. 1: Kinect

- 3-D Depth Sensor: The depth sensor cumulatively consists of an Infrared Depth Sensing Camera and an Infrared Laser Projector. Together, the projector and the camera creates a depth map, which gives the distance between the object of interest and the camera. The sensor has a range limit of 0.8m to 3.5m. The Infrared laser projector projects an Infrared speckles dot pattern into the 3-D scene while the Infrared camera catches the reflected Infrared speckles. The speckles are invisible to the RGB camera(as well as naked eye) but can be they can viewed by IR camera. Since each location of the projected speckle is unique, mapping between observed speckles in the image with the calibrated projector pattern can be easily done. The depth of a point can be calculated by the relative trigonometry between the object and the camera.[7]

- The Motorized Tilt: The Motorized Tilt is used for physical orientation of the sensor. The sensor can be tilted up to 27°either upwards or downwards.

C. Software

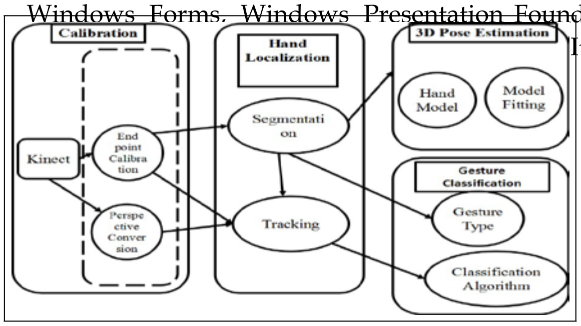
Kinect provides a development library wherein algorithmic components are included as well. OpenNI and Microsoft Kinect SDK are the most predominantly used libraries for development.

- Microsoft Windows SDK v1.8 The Microsoft SDK enables developers to develop applications that support voice and gesture recognition using kinect. The downside being that, this SDK works only on Windows platform, contrary to OpenNI. However, Windows SDK supports 20 skeletal joints, and as specified before, Windows SDK does not require any specific pose for calibration. But, this makes it more prone to false positives.
  Microsoft SDK is able to track a user's upper body even if lower body is not visible. This particularly proves useful when analyzing the human postures with a sitting position.Microsoft SDK recognizes simple gestures, such as grip and push action, whereas OpenNI focus more on hand detection and hand-skeletal tracking.

- Microsoft Visual Studio(IDE):
  Microsoft Visual Studio is and IDE that is used to develop computer programs, as well as web sites, web apps, web services and mobile apps. Visual Studio uses Mi- crosoft software development platforms such as Windows API,

Windows Forms, Windows Presentation Foundation, It can

D.



The raw data obtained from kinect cannot be directly applied to computer vision algorithms. It is necessary to first acquire the RGB and the Depth data from the two respective cameras.In order to accurately combine RGB image with its corresponding depth data, it is necessary to align the RGB camera output with depth camera output. Some further processes conduct depth filtering and recalibration as well. and depth data filtering.[8]

- Recalibration:
  Kinect is calibrated during manufacturing itself. The cam- era parameters are stored in the devices memory, which can be used to fuse the RGB and depth information.This calibration information is adequate for casual usage, such as object tracking. However, it is not that much accurate for constructing an entire 3-D map of the environment. Moreover, the manufacturers calibration does not correct the depth distortion, and is thus incapable of recovering the missing depth.

  In this method, 3-D coordinates of the feature points on the calibration card are obtained from the RGB cameras coordinate system. The spatial Feature Point mapping helps the points to get their true depth values based on the RGB camera's coordinate system. Meanwhile, the depth camera measures 3-D coordinates of those feature points in the Infrared camera's coordinate system.

- Depth Data Filtering:
  Another preprocessing step is depth data filtering, which can be used for depth image de-

noising or missing depth (hole) recovering. A naive approach considers the depth data as a monochromatic image and thus applies existing image filters on it.

4. SYSTEM ARCHITECTURE

Preprocessing is the most initial step that performed in any 3D sensor. After preprocessing the data acquisition occurs. In this preprocessing step there are several sub steps, those must be performed by the user before initializing the system in full. Those steps are briefly narrated below:

Fig. 2: System Architecture of Kinect

In the above diagram there are four blocks named as Calibration, Hand Localization, 3D poses Estimation and Gesture Classification.

- Calibration: The calibration step involves two sub steps named as end point calibration and perspective conversion. For those sub steps an input has been taken from the user via 3D sensor.
  1) End point calibration: The end point calibration step takes an input from the user along with four points of projection. The reasonable grounds behind those four points is to refer those points for the frame of reference[40].
  2) Perspective Conversion: The sensor has few cameras (such as IR cameras, color cameras). Though those cameras are at different positions on the sensor. So, because of their different positions they have different co-ordinates. Fortunately, the resemblance between those cameras is fixed [39].
- Hand Localization: In hand localization there are two important steps that user has to carry out. Those are Segmentation and Tracking. Through these steps the hand modelling and model fitting is done which are the sub steps of 3D pose estimation.
  1) Segmentation: The segmentation process is carried out to initiate the Human-Computer Interaction. In this step the users hand will get detected by the aforementioned cameras. This segmentation is done to highlight the hand from the background region. To obtain this result system have to eliminate the background region.
  2) Tracking: In tracking the output from segmentation will be taken as the input and the hand will get tracked via proposed system. In this step the calibrated surface is also considered because, the operation will be done in that calibrated surface only.
- Gesture Classification: In the gesture classification, the gesture get classified through classification algorithms and the different types of gestures which are stored in a database which are recognized by the system. To get recognized by the system the gesture classification step has to take an input from the Segmentation and tracking.

5. ALGORITHM

In this section, analysis of computational efficiency of the tree based fastNN algorithm is done while trying to offer additional perception on the future performance of the approach on unknown gesture datasets.

As a first step, the implementation for fast K Nearest Neigh- bor (fastKNN), which is a direct generalization of fastNN [46] is used .Some of the key points are :

- It is supervised learning method.
- Data is represented in a vector space.
- The target function may be either discrete value or real value.
- KNN classifiers are slow when organizing test tuples.
- By simple presorting and arranging the stored tuples into search tree, the number of comparisons can be reduced .

1) Nearest neighbor search : : Given the training set of vectors,

$$U = u^i \varepsilon R^d,\ i = 1, ..., M$$

and a query vector,

$$qsR^d$$

the problem of Nearest Neighbor consists of locating the training vector u which shows the minimum distance,

$$d(q, u^i) : \hat{u} = arg - min - d(q, u^i)$$

In this work, we used Euclidean distance between two vectors q,u:

$$d(q, u) = |q - u|^2$$

2) Exact Nearest neighbor algorithm :: The obvious brute force algorithm (Full Search) is to compute all distances in a linear way, showing computational complexity which is linear to the number of training examples, M, and the data dimensionality, d, i.e. O(M,d). Partial Distance Search (PDS) improves the Full Search algorithm through early termination of local distance computation when it exceeds the running minimum distance, but its cost remains nearly linear.

Besides PDS, a lot of fast Nearest Neighbor (fastNN) methods have been proposed in the literature, achieving nearly logarithmic time on average. Such algorithms are typically based on partition trees, arranging training vectors in a tree data structure in some appropriate way (initialization step). Initialization typically starts from the root of the tree and distributes training vectors across its b children based on b 1 hyperplanes; it then continues recursively for all children nodes. During the searching step, fastNN algorithms navigate the tree recursively until the first leaf node is reached; at this point, a first solution is acquired (Depth Only Stage DOS). At that point, backtracking to previous tree nodes is performed, examining the rest of the training vectors. Computational efficiency comes from pruning many training vectors out of the search based on effective lower bounds.

6. BRIEF WORKING

Gesture recognition is an advanced way of interacting with machines. To recoup 3D information of a scene and to inspect depth map and skeletal joints, the Microsoft Kinect is used, and actions being performed by the person get identified by the Kinect.

Using the infrared projector along with depth sensing camera, Kinect can recognize upto 6 users in its field of view. Out of these six users, two users can be fully identified and tracked in detail, and rest four user can be only tracked. The infrared emitter of a Kinect sensor projects a pattern of infrared light, called as infrared(IR) speckles. These speckles of infrared light are used to calculate the depth of the people in the field of view, which allow the recognition of different people and their respective body parts.

- Object Detection:
  Object detection and tracking are hot topics in RGB-based image and video analysis applications. A widely used approach is background subtraction, when the cam- era is fixed. In this particular approach, the background remains stationary over time, and hence, the foreground can be extracted easily by subtracting the input image from the background model. However, in reality, de- tecting objects in images or videos using background subtraction techniques is not that easy due to the high possibilities of probable configurations of scenarios, such as changes of illumination conditions and subtle move- ments or noise in the background. For such cases, the background is not static at the signal level.

- Hand Detection:
  Similar to the object detection introduced before, hand detection can be carried out either on depth images only or by fusing the RGB and depth information. The former aims to obtain a fast algorithm, whereas the latter targets an accurate system. Hand detection can accomplished by the k-means clustering algorithm with a predefined threshold.[9]

- Skeletal Tracking:

Windows SDK provides Skeletal Tracking which allows Kinect to recognize people and follow their actions. This allows the application to locate the joints of the tracked users in 3-dimensional space and track their movements. Skeletal Tracking is optimized to recognize the users.

standing, sitting, or facing the Kinect. Sideways poses prove to be challenging regarding the part where the user is not directly visible to the sensor.

If more than one Kinect is used to illuminate the target area, there maybe a reduction in the accuracy and precision of skeletal tracking due to interference with the infrared light sources. To reduce the possibility of interference, it is recommended that no more than one Kinect sensor (or infrared light source) points to a field of view where skeletal tracking is being done.[12]

The entire setup of the system consists of the following components:

- A Laptop/PC workstation.
- Microsoft Kinect Sensor.
- Power Adapter for Kinect.
- A Projector supporting VGA/HDMI connection.

Initially, after turning on the workstation, required drivers need to be installed to start interacting with the kinect. After installing the Windows SDK drivers, the kinect can be connected to the system. After powering the device, if drivers are installed successfully, should show Kinect as removable media in the notifications tray.

The blinking LED light on kinect assures that the device is getting the data optimally. Now, after a successful setup, the application of the designed system is started. The system detects and classifies the user in front of the sensor. After detecting the user, his joints are mapped and compared with the available data in the sensor. The Skeletal Tracking allows the system to get a skeleton stream of the user in Kinect's field of view. In the system being developed, the joints below the lower body are discarded as the gestures to be classified are concerned only with upper body(hands to be specific).

Now, the gestures are classified based on the relative motion of the joints in accordance with the relative motion of every other joint. For example, When a user's hand crosses the joints of the elbow and spine sequentially, then a swiping action is said to have been performed. This action is then mapped to a keyboard input and given to the system in the form of that corresponding key.

A functionality of mouse cursor control using hand tracking has also been added using similar terms. This maps the cursor to user's palm and can be moved at will.

Yet another feature that has been added is voice recognition, which harnesses the basic microphone array of the Kinect. Simple commands like 'MOUSE ON', 'MOUSE OFF', 'HIDE WINDOW', 'SHOW WINDOW' are used to set alternative paths for same actions, and also make system more interactive.

7. PERFORMANCE EVALUATION

- Total Number of gestures performed

  T_g

- Gestures Detected Successfully

  tt_d

- Recognition Rate = RR

  RR = (T_g/$tt_{d)}$ $_{*\,100}$

The graph based on the perfoamce evaluation of the system of the system with the system which is partially similar to the

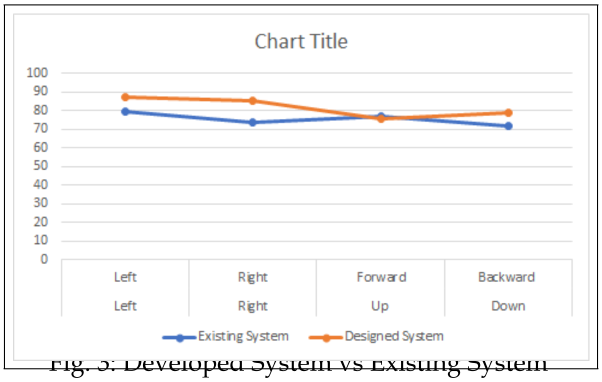


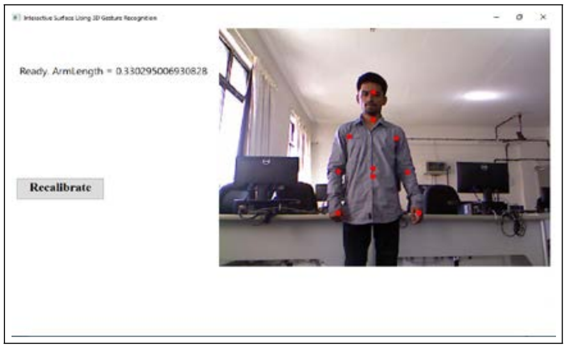
Fig. 3: Developed System vs Existing System

proposed system.System went through gesture testing 70 times which gives a recognition rate using aforementioned formula in percentage.

70 templates were divided in to 4 groups and the observation was in ascending order from template-1 to template-4 the per- formance of the system got increased gradually.The aggregate of 4 templates has been taken using aggregation formula as follows:

Aggregate = (T1+ T2 + T3 + T4) / 100

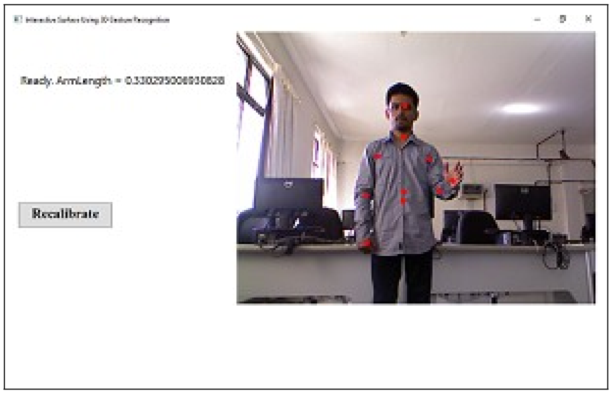
8. IMPLEMENTATION



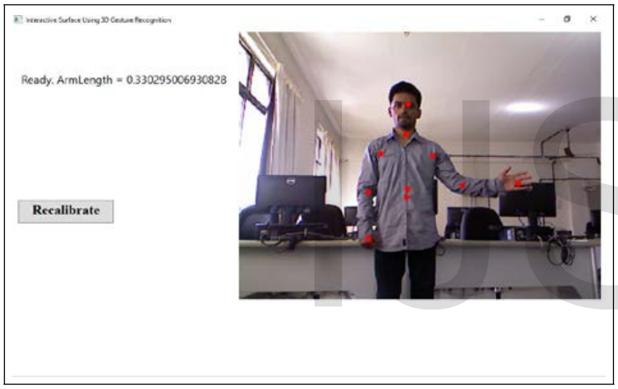
Ready. ArmLength = 0.330295006930828

Fig. 4: Skeleton Mapping

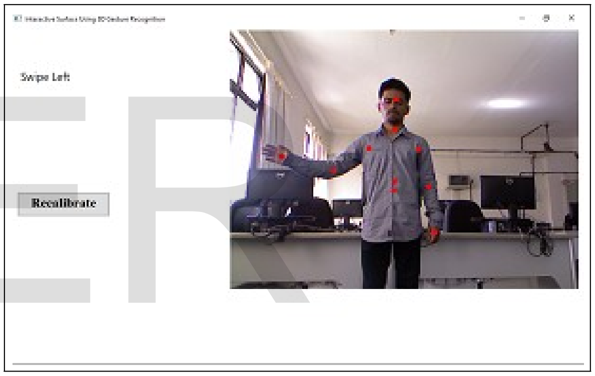
Initially the skeletal stream is obtained from the foreground, and respective joints are mapped.They are displayed by red circles.The joints which are mapped include: Head, Shoulders, Elbows, Neck, and Spine.



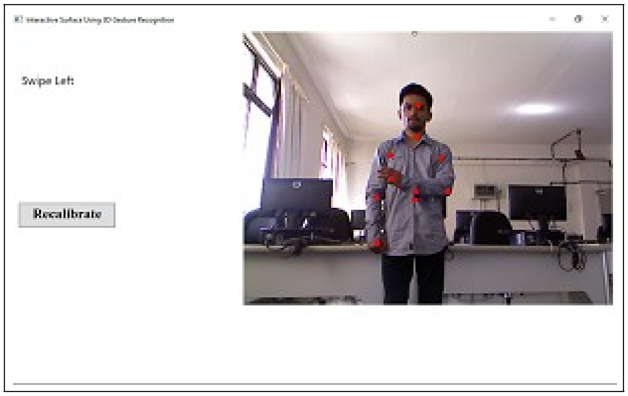
Ready. ArmLength = 0.330295006930828



Ready. ArmLength = 0.330295006930828



Swipe Left



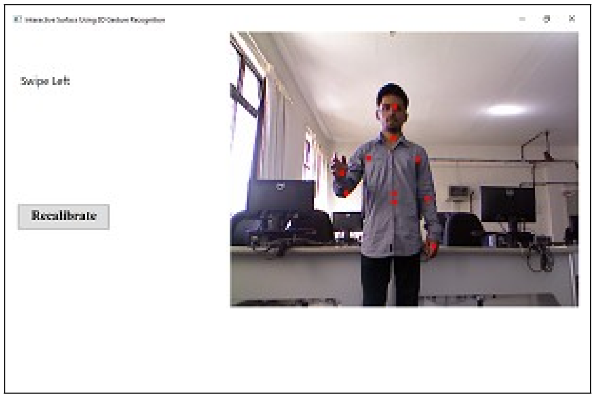
Swipe Left

Fig. 5: Swipe Left (Sequence 1)

Fig. 8: Swipe Right (Sequence 1)

Fig. 7: Swipe Left (Sequence 3)

Fig. 8: Swipe Right (Sequence 1)



Swipe Left

Fig. 9: Swipe Right (Sequence 2)

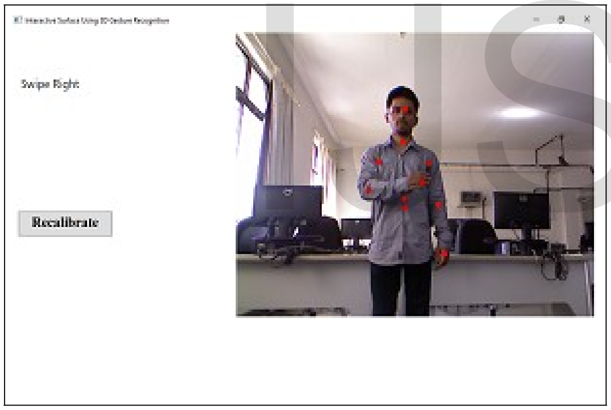


Fig. 10: Swipe Right (Sequence 3)

To perform transitions from left to right, or from right to left; First, the hand is taken to an end. Then when the hand crosses from elbow and spine sequentially, the swiping gesture is said to have been performed. The above images demonstrate the transition from one end.

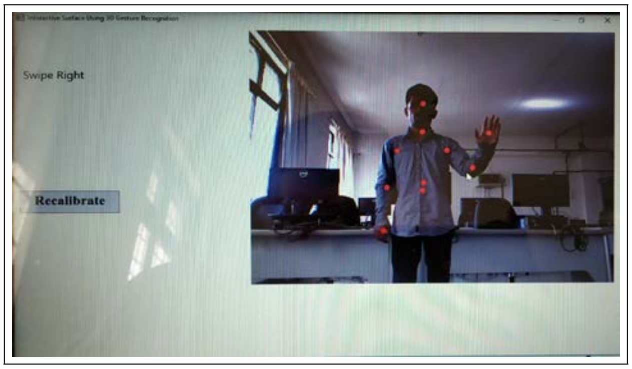


Fig. 11: Mouse Control

To initiate the mouse cursor movement, it can be performed by either two ways: One way is to Wave the Right hand in front of the sensor. An alternative way is to say ”Mouse On”.To de-activate mouse control, say ”Mouse Off”. Some other extra operations can also be performed like, ”Hide Window”, ”Show Window”. ”Close Window” by saying the respective commands.

9. APPLICATION

Gestures have always been a primitive form of communication. The recent advent of 3D gestures have improved this approach even beyond. Depth sensing is the founding pillar for 3D gestures. Some applications of such systems are explained below:

- Medical Applications:
  Gestures recognition can be used alongside robotics to detect and act over life threatening situations like strokes or heart attacks.

- Interactive Interfaces:
  Gestures recognition, combined with voice recognition, face recogniton, and object tracking can be used to

create a unique interface, called as Perception User Interface(PUI).

- Entertainment applications:
  3D Gestures are a fascinating way to immerse the gamers in a world like they have never experienced before.

- Automation systems:
  Gesture recognition can incorporate the routine life and greatly increase the usability of the daily devices like remote controls, car entertainment systems, etc.

- An easier life for the disabled:
  One of the biggest challenges faced by the society is creating an adaptive world for the disabled and handicapped.

While there is much room for improvement, gestures can make life a lot easier and comfortable for the unfortunate.

- Education:
  Students can perform practicals and as well as learn through 3D interaction. Moreover, hidden talents can be revealed as well.

10.CONCLUSION

The Expectation of the system to detect objects within kinect sensor FOV and take their depths from where they located.Using this methodology the system use to recognize the gesture provided by humans and respond accordingly.

11.ACKNOWLEDGEMNT

We are really glad to express our gratitude towards all who have contributed their views and opinions fro the comprehensive study of the research. We are really grateful to our guide and co-author Mr. Parth Sagar for his excellent guidance and support in the work. His suggestions and honest feedback which has helped us through out our research.

12.REFERENCES

[1] A review on gesture recognition using kinect, 2015 International Conference on Electrical Engineering and Informatics (ICEEI) , 2015.

[2] Kinect camera overview . http://www.xbox.com/en-US/kinect/default.htm

[3] Stereo camera overview. http://en.wikipedia.org/wiki/Stereo camera

[4] S. Gokturk, H. Yalcin, and C. Bamji, A time-of-flight depth sensor system description, issues and solutions, in Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops, 2004, pp. 3545.

[5] C. Zhang and Z. Zhang, Calibration between depth and color sensors for commodity depth cameras, in Proc. IEEE ICME, 2011, pp. 16.

[6] J. Smisek, M. Jancosek, and T. Pajdla, 3-D with Kinect, in Proc. IEEE ICCV Workshops, 2011, pp. 11541160.

[7] D. Herrera, J. Kannala, and J. Heikkila, Accurate and practical calibration of a depth and color camera pair, in Proc. CAIP, 2011, pp. 437445.

[8] M. Schmeing and X. Jiang, Color Segmentation Based Depth Image Filtering, in Proc. Int. Workshop Depth Image Anal., 2012.

[9] H. Liang, J. Yuan, and D. Thalmann, 3-D fingertip and palm tracking in depth image sequences, in Proc. ACM Int. Conf. Multimedia, 2012, pp. 785788.

[10]https://msdn.microsoft.com/enus/library/hh973074.aspx