A Human Computer Interface System Using Kinect

Sandineni Karthik[1], Abhilash Vanam[2]

B. Tech, Mahatma Gandhi Institute of Technology, Gandipet Main Rd, Kokapet, Hyderabad, Telangana

Abstract- Developing natural and intuitive interaction techniques is an important goal in human computer interface (HCI). Typically, humans interact with computers using mouse and keyboard. Outside of computer interaction, we are used to interacting with the world using our hands, body, and voice. Interfaces based on interaction with hands are a natural and intuitive way to interact with computers. Such an interface could be used for robot and human collaboration, virtual reality, scientific visualization, geographic information systems (GIS), or games. With the widespread use of the Microsoft Kinect, increasingly more people are creating interfaces based on full body and voice recognition. Finger and hand gesture recognition with the Kinect is still an open problem due to its low resolution (640 × 480), especially considering how hands occupy a much smaller portion of the full body image. Traditional vision-based hand gesture recognition methods are far from satisfactory due to the limitations of the optical sensors used and dependency on lighting conditions and backgrounds. Data gloves can be used for precise accuracy, but require the user to wear a special glove; this may hinder the naturalness of the hand gesture. The Microsoft Kinect is a commodity hardware device that can be used for designing natural gesture based interfaces.

Keyword; *Human Computer Interface, Microsoft Kinect, hand gesture recognition.*

I. INTRODUCTION

The interface using the hand gesture is popular field in Human Computer Interaction(HCI). Recently, a new possibility is provided to HCI field with the development of the sensors and technology [1, 2, 3]. This development has made possible robust recognition which is like to identify fingers and hand gesture recognition in bad condition such as dark light and rough background. At the same time, the interface for natural interaction was required from many users. However, it doesn't reach to this requirement of users. The term natural interprets often to mean the mimicry of the real world. It is the design philosophy and source for that metric standard and the repetitive process manufactures the product. In this research, we discuss the touch and gestural interaction as one mode making the organization of the natural user interface possible. However, as to we, moreover NUI believes. Actually, we can think new that he/she spread so that he/she could indicate the mouse and keyboard, voice command, and in-air gesture and we are able to make a kind of interface for the cell phone and so forth after the design guide. They don't prescribe that in them oneself or the input and print technique guarantee them to provide the opportunity for making the natural user interface from us. The natural user interface is in UI. It optimizes the route the optimizes in the expert that it knows as the experience and that it is made so that we can use in this technology it has an effect on the potentiality of the new technology to the reflector human ability in which the method we betters than. It is applied to the given context and technical goal of the invention. It satisfies our necessity. The Fig 1 depicts the HCI (Human computer interface). The range of the imagery in which that hears that quickly is awakened. Therefore the term natural is strong. First, and, the important situation for understanding uses that so that we can describe the in fact outside attribute in the product itself [4,5].

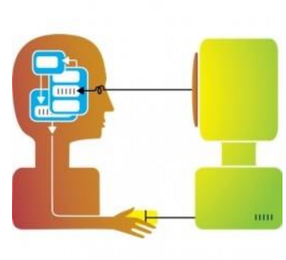

Fig. 1 Human Computer Interaction

Recent advances in 3D depth cameras such as Microsoft Kinect sensors (www.xbox.com/en-US/kinect) have created many opportunities for multimedia computing. The Kinect sensor lets the computer directly sense the third dimension (depth) of the players and the environment. It also understands when users talk, knows who they are when they walk up to it, and can interpret their movements and translate them into a format that developers can use to build new experiences. The Kinect uses an infrared (IR) laser to spray out pseudo-random beam pattern. An IR camera captures an image of the dots that are reflected off objects (as in the picture below) and the electronics inside the Kinect figures out how much the dot pattern has been distorted. The distortion is a measure of distance from the camera. The Kinect is shown in the below Fig 2.

Fig.2 Kinect Sensor

II. LITERATURE SURVEY

Park, Hojoon, François Bérard[6] proposed a Barehanded technique that means that no device and no wires are attached to the user Fig 3, who controls the computer directly with the movements of his/her hand. In the last ten years, there has been a lot of research on vision based hand gesture recognition and finger tracking. Interestingly there are many different approaches to this problem with no single dominating method. The basic techniques include color segmentation, infrared segmentation, blob-models, and contours. Typical sample applications are bare-hand game control, and bare-hand television control. Most authors use some kind of restriction, to simplify the computer vision process:

- Non real-time calculations
- Colored gloves
- Expensive hardware requirements (e.g. 3D-camera or infrared camera)
- Restrictive background conditions
- Explicit setup stage before starting the tracking
- Restrictions on the maximum speed of hand movements

Changing the bright condition and background clutter additionally, the most of systems has the problem. Any of presented work doesn't provide the strong tracer technique for the fast hand movement. Moreover, the most of systems demands any kind of the device-step before the interaction can leave.

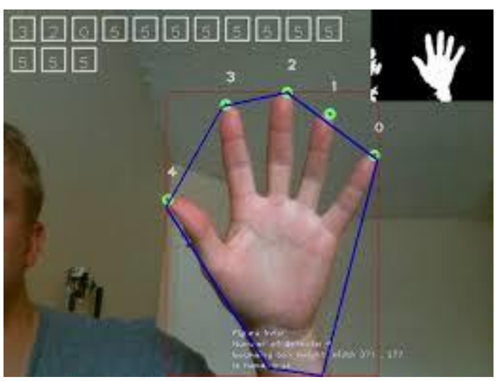

Fig. 3 Bare Hand Tracking

Victor Adrian Prisacariu, Ian Reid [7] proposed challenge of hand tracking mainly comes from the fact that the hand is a highly articulated object and cannot be treated as a rigid object, and the speed of gesture can be rather fast. Typical algorithms for hand tracking are: optical flow, Camshift, Kalman filtering, particle filtering, condensation algorithm etc. All these algorithms are based on certain assumptions and the integration of different algorithms can often generate better results.

Optical flow methods are pixel-based approaches to object motion estimation where a direct relationship is assumed between object motion and intensity changes within an image sequence [6]. Among the existing methods for optical flow estimation, gradient based techniques are characterized by the assumption of brightness constancy and spatial-temporal smoothness [7]. The major advantage of using optical flow in tracking is that it does not require any prior knowledge of the object appearance. However, this advantage may be compromised by its high computational complexity. For real-time hand gesture recognition applications, the Lucas–Kanade method based on block motion model is often used, which does not contain the iterative procedure [8]. Optical flow based methods perform well when objects move not very fast and there are only a few moving objects. For applications demanding high accuracy, such as robot navigation, optical flow itself may not be sufficient due to the factor of drift, which is caused by error accumulating. In this situation, exploiting the color information in optical flow tracking may result in better results.

III. DESIGN AND METHODOLOGY

Vision Based

Hand gesture recognition involves three important phases: detection and segmentation, tracking, and classification. The detection and segmentation part is to detect hands and segment the corresponding image regions from the background. Tracking exploits the spatial and temporal information of successive image frames to estimate the trajectories of hand motion. Classification takes hand trajectories as input to identify the specific types of gesture. The framework of a typical hand gesture recognition system is shown in Fig. 4.

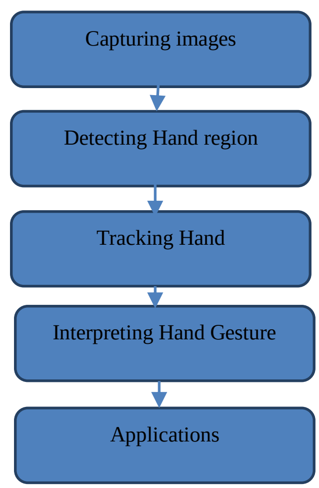

Fig.4. Framework of a typical hand gesture recognition system.

A. Detection and Segmentation this part belongs to the pre-processing stage, which locates the task-relevant regions of the images and eliminates the background. It is both important and challenging for the reason that it generates the source data for the subsequent tracking and classification stages and the background might be very noisy in real-world situations. A large number of methods have been proposed based on various types of visual features or their combinations. For dynamic hand gesture recognition, the motion of hands instead of the pose of hands is the major concern and the computational complexity should be kept as low as possible. Consequently, low-level features and simple methods such as shape and background subtraction are often preferred.

1) Shape

The geometric features of hands can be also used fordetection in different ways. The contour of hands can beobtained by applying edge detection operators, such asCapturing Images. Canny, Sobel, Prewitt, Roberts and Laplace operator.Compared with skin color detection, the contour feature hasthe superiority of being independent of illumination andskin color variation. However, its effectiveness can behindered by occlusions and viewpoint change. Furthermore,edge detection may often result in a large number of edgesthat belong to irrelevant background objects. Therefore, theintegration of shape, texture and color features may producebetter performance [9].

2) Background Subtraction

Background subtraction is a widely used for detecting moving objects. The ultimate goal is to “subtract” the background pixels in a scene leaving only the foreground objects of interest. If one has a model of how the background pixels behave the “subtraction” process is very simple. Of course, in real world scenes, the situation will be more challenging. Background subtraction usually consists of three attributes besides the basic structure of the background model: background initialization (construction of the initial background model), background maintenance (updating the background model to account for temporal changes in subsequent frames), and foreground/background pixel classification. The process is shown in Figure 5.

Depth Image

The Kinect uses an infrared (IR) laser to spray out pseudo-random beam pattern. An IR camera captures an image of the dots that are reflected off objects (as in the picture below) and the electronics inside the Kinect figures out how much the dot pattern has been distorted. The distortion is a measure of distance from the camera. This approach is commonly called Structured Light. All this happens at 30 frames per second.

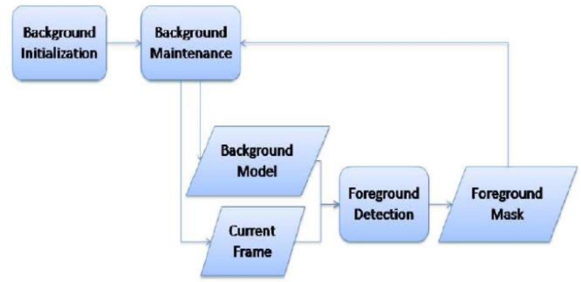

Fig. 5 Background Subtraction

Proposed System

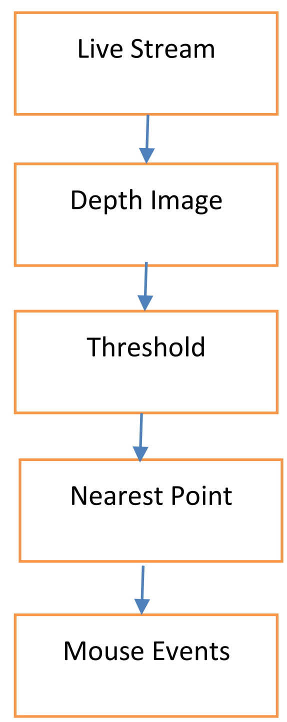

Fig. 6 Data flow diagram

Threshold

A particular threshold value is set so that the scene can be divided into two clusters. The first cluster in the area where we are tracking the movement and the second cluster is hollow space where the Kinect doesn’t track any of the movements. This clustering is done using the K-means algorithm where the k is 2 in this case. The clusters are set with a specific boundary values which can be altered for the user convenience.

Nearest Point tracking

After the clusters are the made, we are going to find the nearest point in the tracking cluster with the help of KNN algorithm .Here we are going to assign a huge value of a pixel to a variable and then update its value if any other pixel is closer than that pixel thus, finding the nearest pixel to the Kinect.

IV. RESULTS AND DISCUSSION

Processing is a flexible software sketchbook and a language for learning how to code within the context of the visual arts. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. There are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning and prototyping.

- Free to download and open source
- Interactive programs with 2D, 3D or PDF output
- OpenGL integration for accelerated 2D and 3D
- For GNU/Linux, Mac OS X, Windows, Android, and ARM

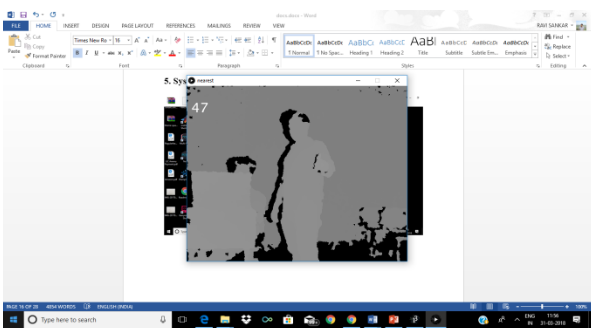

Fig 7. User is outside the threshold

The sensor detects the object (finger) only when it is with in the threshold. It shows the detected part in pink color leaving everything in the grey scale. When there is no point in threshold then the mouse pointer resides at coordinates (0,0). In the above Fig. 7 the user is not in the frame so the mouse pointer is in the top left corner (0,0) of the screen.

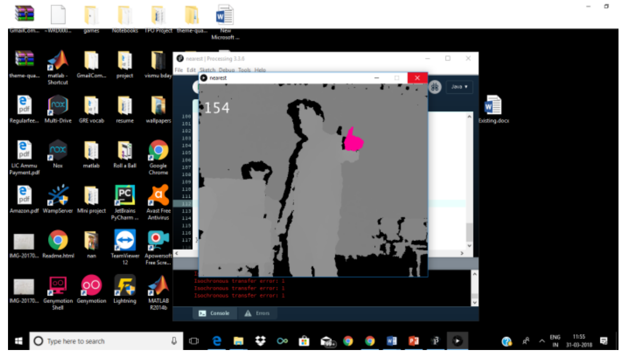

Fig 8 the user is within the threshold

When the user is in the threshold it takes the nearest point the cluster thus giving the mouse control to that point. The user can perform all the mouse events with in the threshold itself. A counter is taken to see how many seconds the cursor remains at a particular position. This counter is incremented if the movement in the mouse pointer is not more than 50 pixels. If the movement in the cursor is more than 50 pixels then the counter is again set to 0 and if the count reaches 200 then the respected mouse event happens. In the above Fig 8 the user is controlling the pointer by the hand movement. If the value 154 in the figure reaches 200 then the mouse event occurs.

Depth Image from Kinect

The program starts with the Kinect using an infrared (IR) laser to spray out pseudo-random beam pattern. An IR camera captures an image of the dots that are reflected off objects (as in the picture below) and the electronics inside the Kinect figures out how much the dot pattern has been distorted. The distortion is a measure of distance from the camera. The Fig 9 describes the depth image obtained from Kinect. This approach is commonly called Structured Light. All this happens at 30 frames per second.

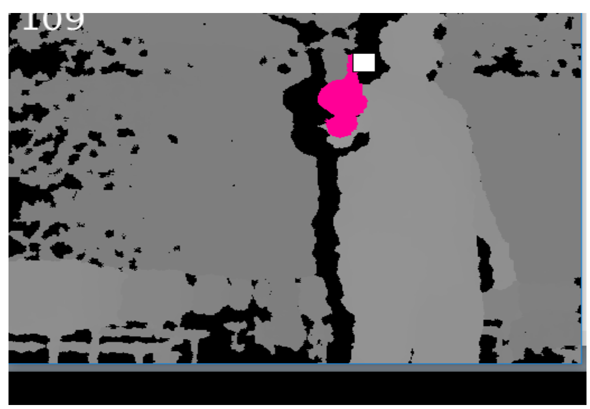

Fig 9. Recognized hand

Point tracking in Threshold

After the clusters are the made, we are going to find the nearest point in the tracking cluster with the help of KNN algorithm .The Fig.10 depicts the point tracking. Here we are going to assign a huge value of a pixel to a variable and then update its value if any other pixel is closer than that pixel thus, finding the nearest pixel to the Kinect.

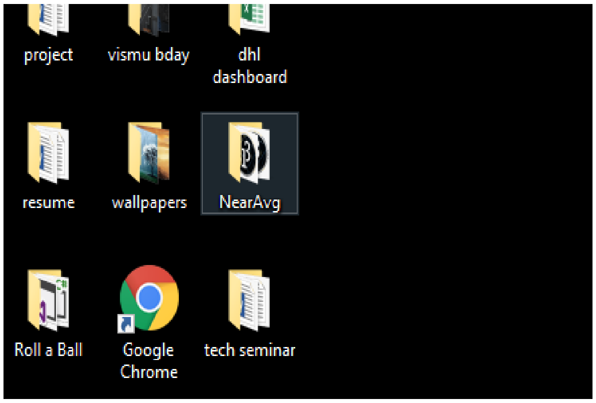

Fig 10. Computer interface

Double Clicking the desired folder

Finally, the mouse movement and its respective events are added. If the pointer resides on a particular option for 2secs then click event is performed at that particular site.

V. CONCLUSION

To conclude we use the "Hand Tracker" to get information about the position of the human hand in the 3D space, from optical data that they are provided by the Kinect RGB-D sensor. We use the hand's position aligned to the computer screen's coordinate system to move the cursor in the position the user points on the screen. Furthermore several gestures are recognized which activate specific computer mouse and keyboard commands when they performed by the user. The hand gestures are modelled using the Kinect. All the gestures are consisted of hand position. The positions are observed and the nearest point in the tracking cluster is found with the help of KNN algorithm .Here we are going to assign a huge value of a pixel to a variable and then update its value if any other pixel is closer than that pixel thus, finding the nearest pixel to the Kinect. Finally, the mouse movement and its respective events are added. If the pointer resides on a particular option for 2secs then click event is performed at that particular site. Therefore the u-user needs to hold his finger for a period of time for the event to get performed. The results of the experiments show the efficiency of the application especially if we consider that all the users had never use the application or similar applications before. Most of the users found it easy enough to operate the computer with this application, while they enjoyed the process.

VI. REFERENCES

[1]. Z. Ren, J. Meng and Z. Zhang. "Robust Hand Gesture Recognition with Kinect Sensor", MM'11, Proceedings of the 19th ACM international conference on Multimedia, 2011.
[2]. A. Mulder. Hand Gestures for HCI. Simon Fraser University: NSERC Hand Centered Studies of Human Movement Project, 2014.
[3]. Hewett & others. Curricula for Human-Computer Interaction. Carnegie Mellon University: The Association for Computing Machinery Special Inters Group on Computer Human Interaction, 2012.
[4]. J. Preece & others. Human Computer Interaction. Wokingham, England: Addison-Wesley Publishing Company, 2014
[5]. Ashwini M. Patil, Sneha U. Dudhane, Monika B. Gandhi, Nilesh J. Uke. Cursor Control System Using Hand Gesture Recognition. International Journal of Advanced Research in Computer and Communication Engineering. 2013.
[6]. Park, Hojoon. A Method for Controlling Mouse Movement using a Real-Time Camera. Brown University, Providence, RI, USA, Department of Computer Science. 2008.
[7]. Victor Adrian Prisacariu, Ian Reid. 3D hand tracking for human computer interaction. Image and Vision Computing. 2011.
[8]. Geovane Griesang, Rafael Peiter, Rolf Fredi Molz. Man-Computer Interaction System Using Kinect. International Conference on Industrial Engineering and Operations Management. 2013.
[9]. J. Preece & others. Human Computer Interaction. Wokingham, England: Addison-Wesley Publishing Company, 2014.