ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Enhanced Application of Natural User Interface Using Kinect

C. Dinesh Kumar, D. Lohitha

M.Tech Student, Dept. of Embedded System Technology, SENSE, VIT University, Vellore, Tamil Nadu, India

M.Tech Student, Dept. of Embedded System Technology, SENSE, VIT University, Vellore, Tamil Nadu, India

ABSTRACT: As of now we are in such a era when we are almost surrounded by the technology in one way or other, in many ways technology had provided us ease and now we are in such a condition that we are evolving more and more in the field of technology. The communication with technology is a very important aspect of evolution. Communication with technology means communication with machines, computers, robots and many more. Since computer is invented The medium of communication with computer have improved a lot with compare to earlier if we start seeing the history we will come to know how we started communication with computers using punch cards and now we are using many kind of interface to communicate with computer. This paper presents the Enhanced Application of Natural user interface in which we have used the Natural User Interface as a input to computer, here our objective is to use the body of user as an input device for computer. For such kind of interfacing we are using KINECT Camera introduced by Microsoft which provide a ease to use the body of user to get an input as they have included feature of depth sensing so that using such device we can come to know the position of any point in 3D space and we can get coordinate of any desired point in 3D.

In This paper we have used skeleton detection feature of such camera which is provided by Microsoft as in the form of Software Development Kit(SDK). The feature of skeleton detection give us room to deal with coordinates of different joints in body all together. we can use the joint's coordinate in real time and can process with them, so we have collected data of the hand and then processed them in such a way that we can change the slides of PPT just by moving our hand in a particular direction. This paper opens the room for many other application which could be made in really efficient and time constrained using this feature of natural user interface.

KEYWORDS: SDK; PPT; 3D; Robots; Microsoft

I. Introduction

Now a days when computer are becoming more and more equipped day by day equipped in sense we are providing more and more feature in single computer itself which no one would ever imagined and still we are on. As more and more features are getting included we are also getting more and more benefits form our computers. Anyways whatever improvement takes place in technology how much computer get equipped after all its human wellbeing which we are seeking . So to facilitate our communication with computer we have introduced many kind of interfaces and many kind of devices. We are using mouse ,key board, console , printer ,and a lot more we have invented many interface to communicate with computer now time has came when we are going to use our body itself as a tool to feed input in computer in technology term we name it NATURAL USER INTERFACE (NUI). While dealing with such kind of interface we use user's body as input device itself there could be many ways to make such thing done. Peoples use many wearable devices for user body and yes they getting popular too as we can see google glass Samsung 's watch wearable Bluetooth module and what not.

Now we are also using NUI with a difference we are not going to use any device to wear on the body while we will use natural body itself. We will use a camera by which we will record the video and in real time we will process such video to detect any human came in visual frame of the camera if so then we will process each frame in real time itself and detect the joints a body should have our camera has classified 20 different joints in a normal human body so we will detect the joints in the body and we will get data from the frame like position of joint or coordinates in 3D as we

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

have depth feature included in our camera by getting the position of joints in 3D space we can easily make an virtual skeleton on our video screen. That feature is actually called skeleton tracking. Now we are using coordinates of hand to process.

The camera which we have used is kinect for windows this camera gives not only RGB frames of video but also it gives depth frame to process it gives a depth matrix of each frame in which the value of each pixel is function of the distance of such object from camera.

II. RELATED WORK

Natural user interface is indeed a technique to use natural objects to do interfacing .when we talks in terms of kinect we can say that body is using as natural component. So to use body as a component or as a input device we must have a set of different different symbols to provide input to computer as we desired so to produce different different kind of input we have selected some posture of body which would be using as input. So the main process it that to decode the posture of body correctly if the body is doing a certain posture then certain thing should happen if body is doing another thing another thing should happen as we have feature to get coordinate of some selected joints in the body in frame we just have to make some mathematical analysis to detect the posture. once if a particular posture is detected then we have to perform the task which we want to perform on occurrence of such posture earlier it was done using some wearable peripherals like glows for palm or many others that was also a natural user interface but in that user was bounded to peripherals the user have to wear some peripheral but now as we come to such a position in the evolution of natural user interface that we really don’t need any kind of peripheral for this kind of application but still wearable peripherals have also their significance in other particular application like Bluetooth speaker that can not be replaced because of the application in which we are using it.

A. *Tracking Joints of the user:*

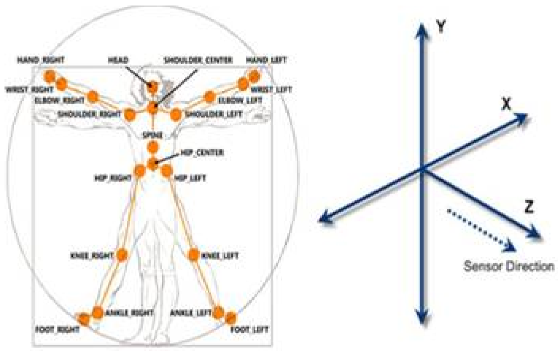


Fig 1: Tracking Joints with 3D view

B. *Microsoft Kinect Camera:*

It's an exceedingly creative mix of cameras, microphones and programming (software) that transforms your body into the computer game controller. The name Kinect is motivated by the words "kinetic," which intends to be in movement, and "connect," which implies it "unites you to the partners and incitement you desire". It's not simply the recreations that make them move, either: Kinect transforms your Xbox 360 into a voice-actuated console with video catching and facial recognition, pertinent for everything from selecting a Television event to making advanced craftsmanship/digital artwork.

Copyright to IJIRCCE

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

Fig 2: Microsoft Kinect Camera

C. *The Kinect Sensor and its Architecture:*

The creative innovation of Kinect is a combination of hardware and simulation to that hardware done inside Kinect, it has a Kinect port from which the gadget can draw power, yet the Kinect sensor accompanies a force supply at no extra power. For a computer game to utilize the hardware equipment's elements, it should likewise utilize the restrictive layer of Kinect programming that empowers body and voice acknowledgment from the Kinect sensor.

There's a trio of equipment advancements cooperating inside of the Kinect sensor:

i. **Color VGA video camera -** This camcorder guides in facial acknowledgment and other location highlights by distinguishing three shading segments: red, green and blue. Microsoft calls this a "RGB camera" insinuating to the shading(color) segments it identifies.

ii. **Depth sensor -** An infrared light projector and a monochrome CMOS (Complimentary Metal-Oxide Semiconductor) sensor cooperate to "see" the room in 3-D paying little respect to the lighting conditions.

iii. **Multi-array microphone -** This is a variety of four receivers (mic) that can detach the players' voices from the noise in the room. This permits the player to be a couple feet far from the receiver (mic) and still utilize voice controls.

A further take the specialized determinations for Kinect uncover that both the depth and video sensor cameras are of 640 x 480-pixel resolution and keep running at 30 FPS (frames per second). The specifications additionally propose that you ought to permit around 6 feet (1.8 meters) of play space in the middle of you and the Kinect sensor, however this could change contingent upon where you put the sensor.

III. DESCRIPTION

A. *Hardware Interface:*

The Kinect sensor interface with the computer through a standard USB 2.0 port, but in addition power supply is required as the USB port cannot give the sensor's power directly.

B. *Software Requirements:*

As indicated by Microsoft, the PC that is to be utilized with the Kinect sensor must have the accompanying least abilities: (a) 32-bit (x86) or 64-bit (x64) processors,(b) Dualcore, 2.66-GHz or speedier processor,(c) USB 2.0 port committed to the Kinect, and (d) 2 GB of RAM. To get to Kinect's capacities, the accompanying software programming is additionally required to be introduced on the engineer's PC: Microsoft Visual Studio Express or other Visual Studio version. The improvement programming languages that can be used are C++, C# (C Sharp), and Visual Basic.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

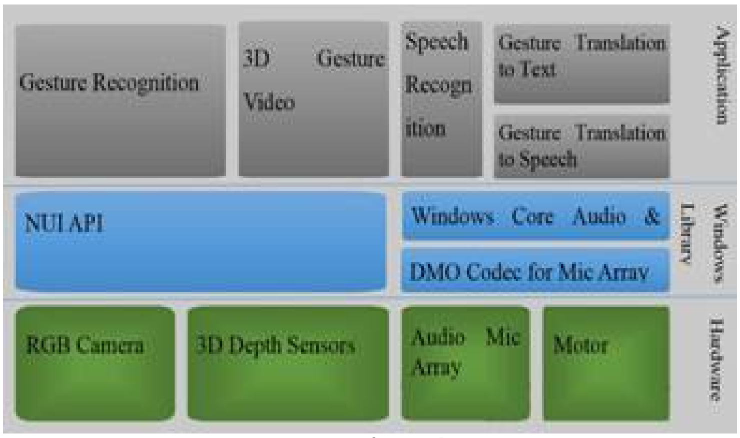


Fig 3: Software Flow

C. *Kinect for Windows SDK:*

Introducing Kinect for Windows SDK is important to add to any Kinect-empowered application. Figure 4 indicates how Kinect corresponds with an application. The SDK in conjunction with the Normal user(Client) Interface (NUI) library gives the devices and the Application Programming Interface (APIs) required, for example, abnormal state access to shading and adjusted depth pictures, the tilt engine, propelled sound abilities, and skeletal following yet requires Windows 7 (or newer version) and the .NET framework 4.0.

D. *Implementing a Gesture-Based HCI System Using Kinect:*

Three key stages are included in the improvement of the proposed HCI system. Firstly, the gadget must have the capacity to track the skeletal components of a user to identify the hand before any motions can be perceived and handled. Secondly, every motion must be appropriately perceived. Thirdly, gesture recognised must be translated to complete the activities identified with it. All in all, these three key stages compasses around skeleton tracking and signal acknowledgment capacities.

E. *Choice of Programming Language:*

Most Kinect-empowered applications are normally created with C#, C++, or Visual Basic. This interface is created utilizing Windows Presentation Foundation (WPF) in C# which gives a more comprehensible code and a more streamlined syntax structure.

IV. DEVELOPING THE APPLICATION: POWER POINT CONTROL

A. *Setup of the Kinect Sensor:*

The hardware and software are established together, the initial phase is to interface which includes the library files files of the Kinect referring Microsoft.Kinect.dll. The program written in C# includes the namespace directive of Microsoft. Kinect. The classes which is mentioned in the SDK acquires an occasion of the sensor based on the skeletal data. The occurance of the event of Kinect sensor acts like a "engine room." The Kinect has the tracking ability to identifies errors of the developer's program.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

B. *Obtaining the sensor data:*

The application should have the capacity to get the abilities like depth information, color data and the skeletal information, the data from these sensors and camera are fundamental for the fruitful execution of the application. The color stream of the Kinect given by the RGB video data, IR depth sensor will give the depth image of the user.

C. *Depth and camera image of sensor:*

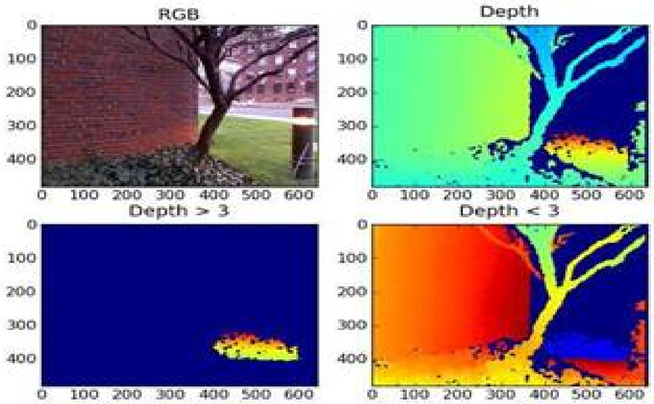


Fig 4: Depth Image Variations

D. *Parameters of the skeleton to be tracked:*

Parameter	Description	Default Value	Comments
Smoothing	Specifies the amount of smoothing	0.5	Higher values correspond to more smoothing and a value of 0 causes the raw data to be returned. Increasing smoothing tends to increase latency. Values must be in the range [0, 1.0]
Correction	Specifies the amount of correction	0.5	Lower values are slower to correct towards the raw data and appear smoother, while higher values correct toward the raw data more quickly. Values must be in the range [0, 1.0]
Prediction	Specifies the number of predicted frames	0.5	
Jitter Radius	Specifies the jitter-reduction radius, in meters	0.05	The default value of 0.05 represents 5cm. Any jitter beyond the radius is clamped to the radius
Maximum Deviation Radius	Specifies the maximum radius that filter positions can deviate from raw data, in meters	0.04	Filtered values that would exceed the radius from the raw data are clamped at this distance, in the direction of the filtered value

Table 1: Parameters for Skeletal Smoothness

E. *Tracking of skeletal joints and getting the skeletal data:*

When the data is obtained from the skeletal joints, that data stream is enabled which generates an event and provides data for the application. The frame of Skeleton is obtained by:

sensor.AllFramesReady+= newEventHandler(sensor_AllFramesReady) Eq(1)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

(a) (b)

Fig 5: (a) A Kinect Skeleton (b) A Kinect Skeleton Showing positions of the hand

The gesture presentation control uses joints data for (1) pointer(cursor) movement, (2) Mouse events are to be tracked by gestures, and (3) controlling presentation. Therefore, finding these joints plays a crucial role, and unless they are initiated, Kinect cannot begin to track them. The tracking of these joints is initiated by a code given below in figure 6.

```
Joint head = sd.Joints[JointType.Head];
Joint jointRight = sd.Joints[JointType.HandRight];
Joint jointLeft = sd.Joints[JointType.HandLeft];
Joint wristLeft = sd.Joints[JointType.WristLeft];
Joint shoulderLeft = sd.Joints[JointType.ShoulderLeft];
Joint wristRight = sd.Joints[JointType.WristRight];
Joint shoulderRight = sd.Joints[JointType.ShoulderRight];
Joint hipCenter = sd.Joints[JointType.HipCenter];
Joint elbowLeft = sd.Joints[JointType.ElbowLeft];
Joint elbowRight = sd.Joints[JointType.ElbowRight];
```

Fig 6: Definition of Joints to be Tracked

From Figure 7 it gives the C# code to find the movements of the pointer onto the screen to left and right. Left hand users also can move the pointer on screen with a ease. The height and width of the screen can be changed by moving the cursor to the borders of the screen.

```
if (sd.Joints[JointType.HandLeft].TrackingState == JointTrackingState.Tracked &&
    sd.Joints[JointType.HandRight].TrackingState == JointTrackingState.Tracked)
```

Fig 7: Tracking a User's Left and Right Hand

F. *Cursor Assignment to the Hand and Cursor Movement :*

Before assigning the cursor to the hand and moving the cursor, a Win32 function called SendInput (Figure 8) must be imported. This function can send Windows messages to the topmost window on the computer screen to control inputs (mouse, keyboard, hardware). It uses some basic Win32 structures defined as follows for the mouse input:

```
public static class NativeMethods
{
    public const int InputMouse = 0;

    public const int MouseEventMove = 0x01;
    public const int MouseEventLeftDown = 0x02;
    public const int MouseEventLeftUp = 0x04;
    public const int MouseEventRightDown = 0x08;
    public const int MouseEventRightUp = 0x10;
    public const int MouseEventAbsolute = 0x8000;
    public const int MouseEventWheel = 0x800;


    [DllImport("user32.dll", SetLastError = true)]
    private static extern uint SendInput(uint numInputs, Input[] inputs, int size);
```

Fig 8: Library for Mouse Events and Movement

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

V. SIMULATION RESULTS

A. *Calculation of the distance between two joints:*

The kinect depth sensor takes joint position in a 3D space coordinate system (X,Y,Z). First the distance between two joints is calculated in 2D using X and Y coordinates, and they are calculates as follows:

float dX = first.Position.X - second.Position.X   Eq(2)

float dY = first.Position.Y - second.Position.Y   Eq(3)

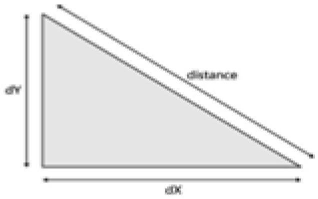
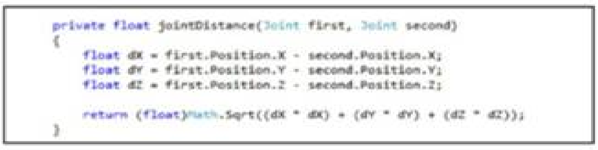


Fig 9: Triangles and Distances

Using Pythagorean Theorem, thus the distance could then be calculated as:

Distance = Math.Sqrt((dX * dX) + (dY * dY))   Eq(4)

The Math library in .NET provides a square root method called Sqrt that provides the distance value. The Pythagorean Theorem is also applicable in 3D view, and the only thing needed is to add the distance in the third dimension. This is used to create the method in Figure 10 that gives the distance between two joints.



```
private float jointDistance(Joint first, Joint second)
{
    float dX = first.Position.X - second.Position.X;
    float dY = first.Position.Y - second.Position.Y;
    float dZ = first.Position.Z - second.Position.Z;

    return (float)Math.Sqrt((dX * dX) + (dY * dY) + (dZ * dZ));
}
```

Fig 10: Method for Calculating the Distance between Two Joints

B. *Controlling PPT:*

The application that is implemented in this paper is controlling the power point presentation by hand gestures. The right and left push buttons of the keyboard permit the slide to move next and back. Using the positions of X, Y, Z which represents head, right hand and left hand and 2 predefines movements are used in the keyboard simulation. Once the left is moved 45cm away from the head then swipe left is perceived which moves the slide backwards and once the right hand is moved 45cm away from the head swipe right is perceived which moves the slide forward.

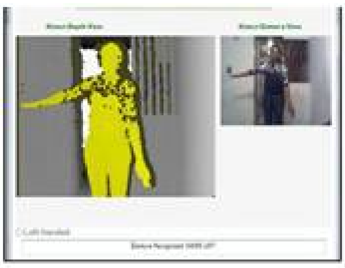


Fig 12: Controlling PPT using Right Movement

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 5, May 2017

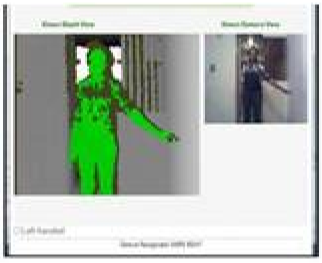


Fig 13: Controlling PPT using Left Movement

VI. CONCLUSION AND FUTURE WORK

Human Movements (Human movement analysis) are now input to the computer which analyze the motions of the body(hand). Hand movement recognition interface between the people and the machine, gesture recognition particularly hand motions are appropriate for a range of subjects, for example, solution reconnaissance, robot control, remotely coordinating, communication via gestures, facial motions recognition, diversions and animations.

REFERENCES

1. Kurtenbach, G. & E.A. Hulteen. "Gestures in Human-Computer Communication". 'The Art and Science of Interface Design, Laurel, B. (Ed.). Reading, Mass: Addison-Wesley Publishing Co., Wasley', pages 309-317, 1990.
2. Kawade Sonam P & V.S. Ubale. "Gesture Recognition" - A Review. 'In OSR Journal of Electronics and Communication Engineering (IOSR-JECE)', pages 19- 26.
3. Zimmerman, T., Lanier, J., Blanchard, C., Bryson, S. and Harvil, Y. "A Hand Gesture Interface Device". In Proceedings of 'CHI 87 and GI', pages 189-192, 1987.
4. Dipietro, L., Sabatini, A. M., & Dario, P. "Survey of glove-based systems and their applications". 'IEEE Transactions on systems, Man and Cybernetics', Part C: Applications and reviews, 38(4), pages 461-482, 2008.
5. Porta, M. "Vision-based user interfaces: Methods and applications". Elsevier, 'International Journal Human-Computer Studies', 2002(57), pages 27-73, 2002.
6. Rafiqul Zaman Khan & Noor Adnan Ibraheem. "Hand Gesture Recognition" – A Literature Review.'International Journal of Artificial Intelligence & Applications (IJAIA)', Vol.3, No.4, July 2012.

BIOGRAPHY

C. Dinesh kumar is a Student in Embedded System Technology (M.Tech) degree from VIT University, Vellore, TN, India. He received B.Tech degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University(JNTU), Anantapur, AP, India in 2014. He is Interested towards Communication, Embedded Systems, Web Designing etc.

D. Lohitha is a Student in Embedded System Technology (M.Tech) degree from VIT University, Vellore, TN, India. He received B.Tech degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University(JNTU), Anantapur, AP, India in 2015. She is Interested towards Communication, Embedded Systems, Microcontrollers etc.