

Human Puppetry Using Microsoft Kinect

Anand Padmanabhan, Geethu Joy, Dr. K. P.Soman

Abstract

The aim of this paper is to infuse the idea of building a novel interactive application that can control an external device using the skeletal movement of out joints employing Kinect. This paper articulates in detail about Skeletal Tracking and its share in controlling a robotic puppet that impersonates the user's actions that the user performs using his/her body movements. The basic ideology behind the implementation of the paper is Forward Kinematics. This paper ideally discusses about the utility of Kinect and conveys the applications a user can exploit using this inevitable tool.

1. Introduction

In the recent times the bridging gap between the humans and machines has shrunk considerably. During the evolutionary age of computers the only mode of interaction to the device was by employing keyboard and mouse. Later in the contemporary times, especially after the advent of Kinect, user is able to interact with the computer at will without the aid of any of the conventional input devices. Kinect has revolutionized the terminology of "interactive applications" and has taken that experience to a whole new level. Kinect's utility is exploited and applied in many disciplines of science and engineering. Kinect's role in robotic applications is enormous as it possesses a unique capability to comprehend 3D discernment of human's movements. Specifically for indoor direction-finding using the method of reverse engineering where its generally accepted functions are to assist in android navigation, surroundings mapping and entity manoeuvring. Another big application of this robotic interface using Kinect is Telerobotics. Telerobotics plays a huge role in space missions where there lies the need for an external robotic to probe the area without human aid. To neglect the inherent risk and cost involved in human space travel and space walks there is mounting interest in using robotic devices to carry out

most of the satellite and space station maintenance labour.

Speculating about Posture inference is a prime issue of concern in the field of computer vision. Amongst the most significant methods by which human beings communicate with each other in reality, postures are one of the main. The intricacies involved in constructing a computerized method that has the ability of receiving as well as classifying this type of data has been one of the main challenges for the developing engineers. Using the depth camera, Kinect bridges the gap existing in Human-machine interaction. In the era of human-robot interaction, a major requirement is that the robot should have the ability to steer freely and make use of its hands to manoeuvre matter. For the Humanoid robots to accomplish regular tasks the robots in the first place should understand and recognize the actions of people in its environment and reciprocate to those actions accordingly in a logical manner. The idea behind this paper is to make a quintessential interface for teleporting humanoid robot that is inexpensive, person-independent and easy to handle. The aim of bilateral telemanipulation is to permit a human user to manipulate and communicate with a remote environment by employing master and slave robotic devices. So it should be an interface where trouble shooting is made easy and user-training is made very minimal. This paper puts forth an implementation where in Skeletal tracking principles are involved to present a neoteric approach of blending Kinect and Arduino together to present a working of Human puppet which in turn can be extended to control a Humanoid robot for its control and functioning. The concept that is implemented in this paper is to use the Kinect for interactive applications in which the user can make a puppet to replicate the user's body moments with high precision and speed in real time. This idea of human puppetry is realized using Kinect as an interface where Arduino acts as the brain of the system.

In this paper a 2D planar robotic framework which has a structure similar to that of human skeletal anatomy is

used to replicate the user's exact body movements. The body movements of the user standing in front of the Kinect are captured and the depth information (post user calibration) that is obtained is processed using processing. Using this data, the angles made by the joints are calculated in real time. This angle information is then given to an Arduino board via serial communication. Here the Arduino that works as the brain of the arm in turn controls the servo motors that attain these angles in real time with high precision and speed. Every single servo motor being used in the framework can cover a span of 0° to 180° and it corresponds to the respective joint in the human skeletal structure and this establishes the concept of human puppetry.

2. Kinect

The Kinect is a Microsoft product. It is a stand-alone device that comes along with the XBOX. The Kinect is basically a depth camera. The conventional camera collects the light that bounces off the objects in front of them. Then that light is changed into an image that resembles what we view through our eyes. The Kinect on the other hand, trace or records the distance of the entities that are placed in front of it. It employs IR light to compose the image (depth image) that captures not what the entities look like but instead where they are in space.

2.1. Components in Kinect

Primarily there are two different ideologies of working with hardware in general and technology in particular, they are extremely important, those are nothing but the input and output. Input is generally the information that we derive from the system via an external source and output is considered to be the information that is let out from the system. The input devices of Kinect are sensors and they read the information in space regarding the physical entities that are placed in front of them. Similar to the inputs that are taken in from the system the outputs of the Kinect are nothing but actuators that permit it to write or act upon the physical space by altering in various ways.

The different entities on a Kinect sensor are as given as follows:

- IR Emitter
- Colour sensor
- IR Depth Sensor
- Tilt Motor
- Microphone Array

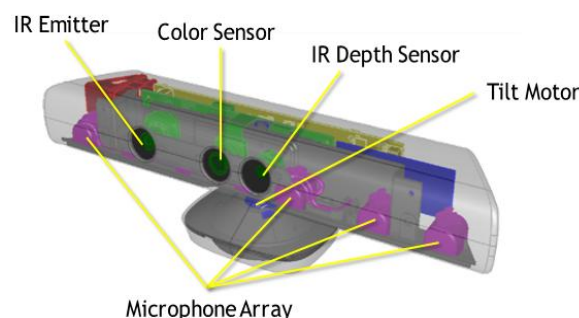


Figure 1. Microsoft Kinect [13]

2.2. Input components

There are four microphones (primarily on the corners) on the Kinect. It produces quadraphonic sound. Blended with highly advanced digital processing in the software side, there are four microphones with which one can do extraordinary things. The filtering of the background noise is done using the combination of the four audio inputs and this not only filters out the disturbances but also helps in detecting the relative position of any person speaking inside the room within the range of the microphone. From the front view we can see that there are three adjacent microphones on the right hand side and the fourth one is placed on the left hand side. Right next to it is the IR (Infra-Red) camera the most enigmatic package inside the Kinect is the triple-axis accelerometer present inside the device.

2.3. Output components

It functions in combination of the IR (Infra-Red) camera to obtain the precise position in space of everything inside the room it occupies. Another light-based output is the LED indicator. It cannot be easily accessed from the OpenNI frameworks. It would be the most efficient way to for the application to inform and alert the user that something is occurring without the use of a screen. For instance, In Matter Port, a 3-D capture tool, the user picks up the Kinect and roams around inside the room-away from the PC-to picture (photograph) objects. The computer provides and audible beep and lets us know once the analysis of a specific view is done. This beep can also be accompanied by the flicker from the LED's as an extra cue. Finally, the Kinect has an actuator that has the opposite functionality of a sensor and it is in the form of miniaturized driving gears that pitch the tilt of the camera up to a range of 30 degrees upwards and downwards.

3. Scripting the Kinect

Processing is primarily a Java application and therefore can be used to do everything that java can do. For example, Animation and drawing in 2D and 3D graphics, manipulating and altering images, reading and writing data, communicating using http and last but not least working with that data that we obtain from a Kinect sensor. But processing also has a scripting syntax and easier function calls made easier so that the novices who are new to programming find it easy and the professionals might find it efficient too. It is “self-contained” and inside that it executes and runs its own instance of Java Virtual Machine (JVM) on the computer. So this makes it extremely easy for set up. All one has to do is just download and launch. The best thing about processing is that it is compatible with multiple operations systems namely, Windows, Mac and Linux operating systems.

Processing is the best tool that an amateur can ever code on as it is the easiest of all programming tools. It is also the gateway to hacker down to say the least. Also it is extremely useful for the pro-programmers as it has its own charms: “the sketchbook” metaphor and also its own pluggable libraries is the best way to get into exploring something totally innovative.

3.1. Open NI framework

OpenNI is the abbreviation for Open Natural Interaction. Natural interaction basically points put to interacting with a device or technology without depending on input devices such as mouse or keyboard. The motive is to interact with technology in an analogous manner to how humans communicate with each other through gestures and speech. OpenNI is basically a multi-languge, cross-platform framework that defines APIs for developing applications that take exploit natural interaction. It segregates the API for the sensor from the API for the middleware that executes tasks such as tracking an entity in space. OpenNI is generally used with the Kinect, so that the need to be concerned with the API for the sensor is taken care of. Also all one has to do is to interact with the middleware. The diagram illustrates how OpenNI comprises different components and how it links itself to wrappers such as the Simple-OpenNI wrapper that is employed for Processing.

OpenNI comprises sensor modules that directly interact with external devices such as cameras or microphones.

The data that is obtained as output from the modules can then be interpreted by the middleware components.

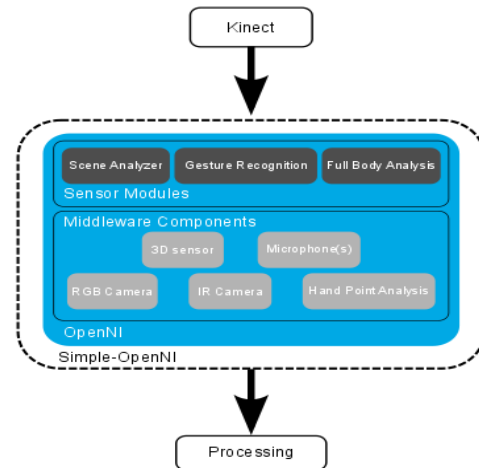


Figure 2. Different Components in Open NI

[14]

3.2. Middle ware components

The OpenNI middleware can be categorized into the following components.

- Full body analysis - tracks the joints of a skeleton such as elbow and knee.
- Hand point analysis - tracks the position of a hand.
- Gesture detection - identifies pre-defined gestures such as a hand wave.
- Scene analyzer - can identify items in a scene such as the floor and separate the figures in the foreground from the background.

3.3. Skeletal tracking

OpenNI has the skill to process the depth image for the user in order to detect and track people. As an alternative of having to loop through depth points to verify their position, the user can directly access the position of each body part of each user that OpenNI tracks. Once OpenNI has detected a user, it will inform us the location of each of the user’s visible joints like head, neck, elbows, torso, hands, shoulders, hips, feet and knees. OpenNI utilizes the phrase joint to point towards all the positions on a user’s body the library is competent of tracking, whether or not they are actual joints. Rather than attempting to compute the threshold of the depth image and then repeatedly go through every single point within it to locate the closest one and suppose that stand for the user’s outstretched arm, we

could have easily gotten the location of his arm from OpenNI and utilized that to update the location of our line. And, additionally, utilizing this joint information, we will now be capable of implementing much more complicated user interfaces that would have been formerly impractical. We can use interactions on hand gestures and on the whole poses of the body, we can follow body activities over time and contrast them, and we can determine distances among different sections or parts of the body.

3.4. Calibration

In order for OpenNI's algorithm to start tracking a person's joints, it requires the person to be standing in a recognized pose. In particular, the user has to stand with his/her feet together and their arms raised above their shoulders on the sides of their head. This pose is branded by various names. In the technical literature and in Prime Sense's individual documentation it is called the "Psi" pose. Other designers also refer to it as the "submissive" pose due to its close resemblance to the position that one would assume if someone pointed a gun at a person. OpenNI does have the potential to track users without the requirement for explicit calibration under certain special circumstances. OpenNI can record the calibration data from a single user and later it can use that to calibrate other users of identical body type. This technique is not as easy as it looks as and is a little intricate and won't work in all situations.

4. Kinematic modelling of planar arm

The essence behind this project is to learn about the concepts of kinematics. Robot kinematics concerns with application of geometry to the learning of the motion of multi-degree of freedom kinematic chains that shape the structure of robotic systems and also has significant use in computer animation. The importance of Kinematics in this project is that it is based on this concept that the association between each joint in the body and the angles of other joints that are in the same line are explained. The different postures made by human beings are decided by the several skeletal joints in the user's body that contribute towards it. This is where the concept of kinematics comes into play. The end position of the user's body is determined by the angles assumed in each joint. Starting with original position the user assumes the end position by rotating the specific joints responsible for this movement through a series of angles and finally presumes the position that is desired.

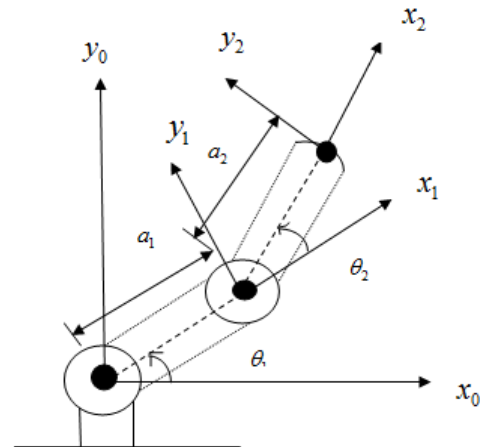


Figure 3: Two-link planar manipulator [15]

Consider the two-link planar arm of given figure 3. The joint axes z_0 and z_1 are normal to the page. We create the base frame $o_0x_0y_0z_0$ as shown. The origin is preferred at the point of intersection of the z_0 axis with the page and the direction of the x_0 axis is totally arbitrary. Once the base frame is established, the $o_1x_1y_1z_1$ frame is stationary as shown by the D-H convention, where the origin o_1 has been positioned at the intersection of z_1 and the page. The link parameters are shown in Table 1. The A-matrix is shown below

Table 1: Link parameter for 2-link planar manipulator

Link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1^*
2	a_2	0	0	θ_2^*

*variable

According to D-H notation we have the following matrices,

$$A_i = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.1)$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

The T-matrix is thus given by

$$T_2^0 = A_1A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_2 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_2 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.3)$$

It is visible that the first two entries of the last column of T_0 are the x and y components of the origin O_2 in the base frame; that is,

$$x = a_1c_2 + a_2c_{12}$$

$$y = a_1s_2 + a_2s_{12}$$

are the coordinates of the end-effector in the base frame. The rotational part of T_0 gives the orientation of the frame $o_2x_2y_2z_2$ relative to the base frames.

5. Human puppetry using Kinect

Technology is gaining good ground and has taken over, transformed and re-modelled the significance of conventional craftworks. A standing example for that is human puppetry using Kinect. This neoteric technology has revamped and renovated some orthodox practices into really interesting and mind boggling phenomenon. Kinect has given the user an approach to create an impersonating puppet that actually replicates the moves that the user enacts. It's a novel approach where the traditional art form is brought a new face. It has been transfused with the modern technology and has now made the old art form revive come back to life. It has made human puppetry contemporary and lucid.

The aim of this skeletal tracking application is to create a 2-dimensional humanoid robot and to make it replicate the actions that the user performs. Whatever action that is performed by the user is performed and impersonated by the Robotic puppet. The application embarks the future that's in store. The whole system basically works on the inputs (joints) that we get using Kinect (Skeletal tracking) and since each joint is considered to be a three dimensional vector, the angle between each joints is calculated(In this case the angles are :left elbow, left shoulder, right elbow, right

shoulder, left shoulder blade, right shoulder blade, right hip, left hip).These angles are in turn sent to the Arduino controller which acts as the brain of the human puppet and this controls the joints of the puppet. The data is sent via serial communication. So, the Arduino receives the angles and controls the different servo motors that are acting as the joints of the human puppet. Instead of building a whole puppet, a robotic arm alone is designed implemented as the similar logic goes for the rest of the joints too.

6. Output

The outputs obtained are shown in the following figures.

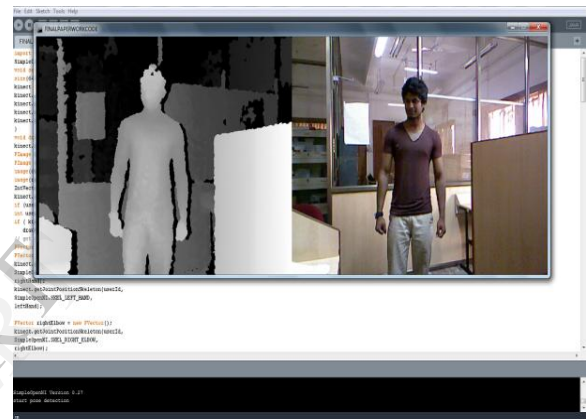


Figure 4: Before Calibration

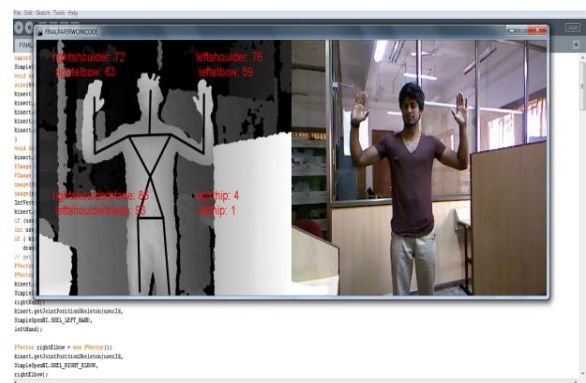


Figure 5: Calibration after PSI Pose

Angles calculated for every joint including both the left and right arm, neck, left and right shoulder blades and hip is shown in the figure below.



Figure 6. Angles Calculated



Figure 7. Corresponding RGB image of figure 6

7. Conclusion and future work

This paper demarcates the importance of Kinect and its share in constructing a user-interactive application that is proficient of controlling and manipulating an external device. The idea of building a robotic human puppet is thoroughly explained in this paper using the concepts of Forward Kinematics. The precision in the angle calculation stands as an inference to imply the successful interface between Arduino and Kinect. This whole idea can be extended and can be used for telerobotics in controlling a humanoid robot.

8. References

[1] Chanjira Sinthanayothin, Nonlapas Wongwaen, Wisarut Bholsithi, Skeleton Tracking using Kinect Sensor & Displaying in 3D Virtual Scene, *International Journal of*

Advancements in Computing Technology(IJACT), Volume4, Number11, June 2012

[2] Fan Wang, Cheng Tang, Yongsheng Ou and Yangsheng Xu, A Real-Time Human Imitation System, *Proceedings of the 10th World Congress on Intelligent Control and Automation*, July 6-8, 2012, Beijing, China

[3] Antonio Padilha Lanari Bo, Mitsuhiro Hayashibe, Philippe Poinet, Joint Angle Estimation in Rehabilitation with Inertial Sensors and its Integration with Kinect, *33rd Annual International Conference of the IEEE EMBS Boston*, Massachusetts USA, August 30- September 3, 2011

[4] Faisal Ahmed, Mohammad Adom Safiullah, Saddam Hossain Khan, Abdullah Moinuddin, Abu Mohammed Farhan, Assembly of Robotic Arm Based On Inverse Kinematics Using Stepper Motor, *Computer Modelling and Simulation (EMS)*, 2012 Sixth UKSim/AMSS European Symposium, 14-16 Nov. 2012, pp. 285 - 290

[5] Murilo M. Marinho, André A. Gerales, Antônio P. L. Bó, Geovany A. Borges, Manipulator control based on the dual quaternion framework for intuitive teleoperation using Kinect, *2012 Brazilian Robotics Symposium and Latin American Robotics Symposium*, 16-19 Oct. 2012, pp. 319 - 324

[6] Indrazno Siradjuddin, Laxmidhar Behera, T.M. McGinnity and Sonya Coleman, A position based visual tracking system for a 7 DOF robot manipulator using a Kinect camera, *WCCI 2012 IEEE World Congress on Computational Intelligence*, June, 10-15, 2012 - Brisbane, Australia

[7] Antonio Frisoli, Claudio Loconsole Daniele Leonardis, Filippo Bann`o, Michele Barsotti, Carmelo Chisari, and Massimo Bergamasco, A New Gaze-BCI-Driven Control of an Upper Limb Exoskeleton for Rehabilitation in Real-World Tasks, *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, vol. 42, No. 6, November 2012

[8] Daniela ChavezGuevara, Giuseppe Vietri, Mangai Prabakar and Jong-Hoon Kim, Robotic Exoskeleton System Controlled by Kinect and Haptic Sensors for Physical Therapy, *2013 29th Southern Biomedical Engineering Conference*

[9] Baocheng Wang, Chenguang Yang, Qing Xie, Human-machine Interfaces based on EMG and Kinect applied to Teleoperation of a Mobile Humanoid Robot, *Proceedings of the 10th World Congress on Intelligent Control and Automation*, July 6-8, 2012, Beijing, China

[10] Samiul Monir, Sabirat Rubya, Hasan Shahid Ferdous, Rotation and Scale Invariant Posture Recognition using Microsoft Kinect Skeletal Tracking Feature, *Intelligent Systems Design and Applications (ISDA)*, 2012 12th International Conference, 27-29 Nov. 2012

[11] Noriyuki Iwane, Arm Movement Recognition for Flag Signalling with Kinect Sensor, *Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2012 IEEE International Conference, 2-4 July 2012*

[12] Greg Borenstein, Making things see 3D Vision with Kinect, Processing, Arduino and Makerbot, Maker Media, Inc

[13] Google Images

[14] <http://learning.codasign.com> accessed on 6/7/2013

[15] <http://www.cs.duke.edu/brd/Teaching/Bio/asmb/current/Papers> accessed on 3/7/2013

IJERT