International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 14 (2015) pp 33987-33989

Control And Programming Of A Robotic Arm Using Gestures And Skeleton Tracking

Vasanth.S

Assistant Professor Department of Mechatronics
SRM University, Kattankulathur, India
vasanth.s@ktr.srmuniv.ac.in

Gayathiri.D

Assistant Professor Department of Mechatronics
SRM University, Kattankulathur, India
gayathiri.d@ktr.srmuniv.ac.in

VinothKumar.P

Assistant Professor Department of Mechatronics
SRM University, Kattankulathur, India
Vinothkumar.P@ktr.srmuniv.ac.in

Abstract
The advent of the Microsoft Kinect sensor and the release of the NITE middleware by PrimeSense revolutionized the field of Computer Vision. The depth camera along with NITE offered various advanced features at a very low cost that were not once available to a civilian. Skeleton Tracking is one such feature that made tracking the joints of a human user, very easy and hence provided new possibilities in Human Computer Interaction. The Skeleton Tracking framework allows the computer to understand a person's body position in 3D and to have a quite accurate idea of where the person's joints stand in space at every point in time. Using this information it is possible to calculate the angle of these joints at every point in time. This paper explores the possibility of designing a robotic arm which will be controlled and can also be programmed using the skeleton tracking data obtained by using the NITE middleware along with the SimpleOpenNI library. The arm built will have 3 degrees of freedom with a gripper to pick objects. An Arduino board is used to control the actuators. Processing is used for programming and will act as an interface between the Arduino and the Kinect. Once the processing sketch is run, it will display the depth image obtained from the Kinect. Once the user is identified and his joints are tracked, various methods are used to calculate the angles of these joints tracked. These angles are then optimized and sent to the Arduino using serial communication. The elbow and the shoulder angles are directly calculated from the kinect data. As its not possible to calculate the shoulders swivel using a single kinect we use a separate gesture to control the swivel. These gestures are derived from the hand tracking feature available in SimpleOpenNI library. Moving a tracked hand in a circular motion will control the swivel action of the arm. Swiping the hand in a particular direction will close or open the gripper accordingly. All these data sent to the Arduino is stored in an Array List. In other words, from the start of the program all the movement the user imparts to the robotic arm is stored. On a mouse click, the skeleton tracking is stopped and theses stored data are continuously sent to the Arduino. This will ensure that the arm repetitively performs the motion that the user programmed it to do my moving his hand around in real 3D space. This method will provide a very easy and highly interactive way of programming a robotic arm. Instead of typing in a code or by moving the arm step by step and recording its position, the proposed method will be a lot easier and requires very minimal programming knowledge to use.

Keywords-Robotic Arm; Skeleton tracking; Programming; Kinect

I. INTRODUCTION

Robotic arms are mechanically controlled devices designed to replicate the movement of a human arm. The devices are used for lifting heavy objects and carrying out tasks that requireextreme concentration and expert accuracy. Various types of robotic arms are available. Each type has its own specifications. For example, the number of joints in the arm varies. The direction in which the joints rotate varies also. Some of the most common types of robotic arms include vertically articulated, selective compliant assembly, polar, cartesian, cylindrical and parallel. Robotic arms are used in various industries. The devices are useful especially in transporting large objects from one place to another, for example on building sites. The machines also are used in situations where toxic fumes could be dangerous to humans, such as the application of paints or varnishes. Robotic arms often are used in factories to assemble dangerous equipment. Robotic arms are used in the medical profession to carry out certain precise operations. The arms also are useful in the collection of specimens in contaminated areas. NASA has been known to use robotic arms to hold astronauts in place and to assemble space equipment. Robotic arms have become useful in many applications. In the future, the devices could be used to replace lost limbs. The Defense Advanced Research Projects Agency has been working on a replica human arm that reacts to messages from the central nervous system. The robotic arm is an invention that could potentially change lives.

The Xbox Kinect:

The way we chose to control our robotic arm is by tracking the motion of the real human arm. And we are implementing this feature using the Microsoft Kinect sensor. Kinect is a motion sensing input device developed by Microsoft for the Xbox 360 video game console and Windows PCs. It is basically a horizontal bar connected to a small base with a motorized pivot and is designed to be positioned lengthwise above or below the video display. It is the cheapest depth camera available. The device features an "RGB camera, depth sensor and multi-array microphone running proprietary software", which provide full-body 3D motion capture, facial recognition and voice recognition capabilities. The one important feature which we are using extensively in our project is the Skeleton Tracking. Prime Sense, the real manufacturer of the kinect, have released a package called NITE which will allow us to do the tracking using kinect. NITE is not an open source software and hence its codes are not publicly available. Therefore, NITE is installed like any other software and can be used by importing the SimpleOpenNI library in our program. Using the depth image acquired and the tracking we calculate the angle of each and every joint and map the data to a limited
Value. This value which is proportional to the angle of the joint is sent to the Arduino using serial communication. According to these values a PWM signal is sent to the servo from the Arduino output pin which will result in movement of the arm proportional to the movement of the users arm.

II. TRADITIONAL METHODS OF CONTROL

Limited Sequence Robots:

The limited sequence robots represent the lowest level of control. Limited sequence robots do not use servo-control to indicate the relative positions of the joints, instead they are controlled by using limit switches and/or mechanical stops. Using this method of control the joints in the robot can be moved to either extreme of a set of limits. There is generally no feedback used by the robot to indicate whether the position of the joint has been achieved or not. Applications of these types of robots generally involve simple operations such as pick and place.

Playback Robots:

These robots use a much more complex control system where in a set of positions or motions are taught to the robot. These positions or motions are recorded into the memory and then repeated by the robot. The method of teaching the robot can be achieved by the use of a teach pendant or by programming the positions directly into the memory of the robot. These robots do have feedback. The playback robots can be classified as point to point robots and continuous path robots. The point to point playback robots are programmed by storing various points in the robots memory. The robot then traces the path taken to get from one point to the next.
The continuous path robots are capable of performing motion cycles in which the path followed by the robot is controlled. The robot is made to trace a series of very closely spaced points which describe the desired path. The individual points are described by the control unit rather than by the user. The user only defines the start and the end point of the path, and the control unit calculates the sequence of the individual points that allow the robot to follow the desired path. The continuous path control is required for some industrial applications such as spray painting and arc welding.

Intelligent Robots:

Intelligent robots constitute a growing class of industrial robots that possesses the capability not only to play back a programmed motion but to also interact with its environment. Invariably, the controller consists of a digital computer or similar device. Intelligent robots can alter their programmed cycle in response to conditions that occur in the workplace. They can make logical decisions based on sensor data received from the operation. The robots in this class have the capacity to communicate during the work cycle with humans or computer based systems. The kinds of applications that are performed by intelligent robots rely on the use of high-level language to accomplish the complex and sophisticated activities that can be accomplished by these robots.

III. PROPOSED METHOD OF CONTROL AND PROGRAMMING

The general block diagram of the process happening is shown in Fig.2.

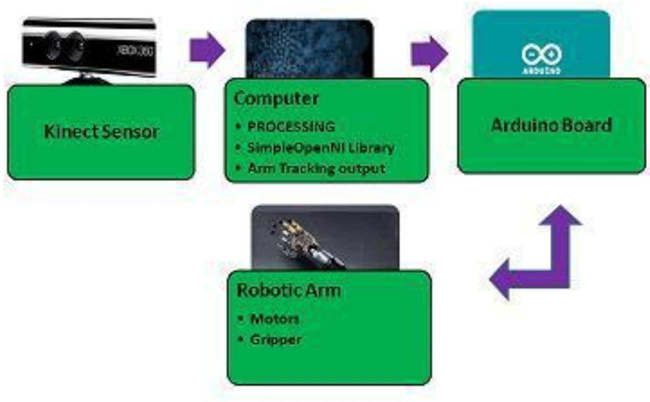

Fig.2. Block Diagram

Initially the Kinect sensor constantly acquires depth images and sends it to the computer. These images are acquired and processed by the processing program. The program initially searches for a user in the image in the starting pose. The pose is called PSI, where the user stand in a hands-up position. Once a user is identified, several NITE call back functions are called, which ensure the skeleton is tracked. To provide a visual feedback to the user, limbs on the right arm are drawn using a separate function. Using the three dimensional coordinates of the elbow joint, the shoulder joint and their adjacent joints, the shoulder and elbow angles can be calculated. To ensure a smooth motion the angle are linearly interpolated with the values of the same at a previous at a previous instant. Using a specified serial protocol the interpolated angles are sent to the arduino. The arduino then accordingly actuates the servo motors in the robotic arm. These angle values are also stored in an array list (PVector Type) along with the shoulder swivel motor angle. The shoulder swivel motor is controlled by a separate gesture as the angle between the shoulder and the body of the human cannot be found using a single Xbox kinect. Hand tracking is implemented for this purpose. Moving a tracked hand in a circular motion controls the swivel of the shoulder. This gesture can be seen in Fig.3. A visual feedback of a circle on screen is provided, that indicates whether the hand is being moved in a clockwise or an anticlockwise motion. The gripper will be controlled by a swiping gesture. Swiping your hand to the left will actuate the gripper to open and swiping your hand to the right will actuate the gripper to close. The gesture can be seen in Fig.4. The gripper angle values are stored as a separate integer variable

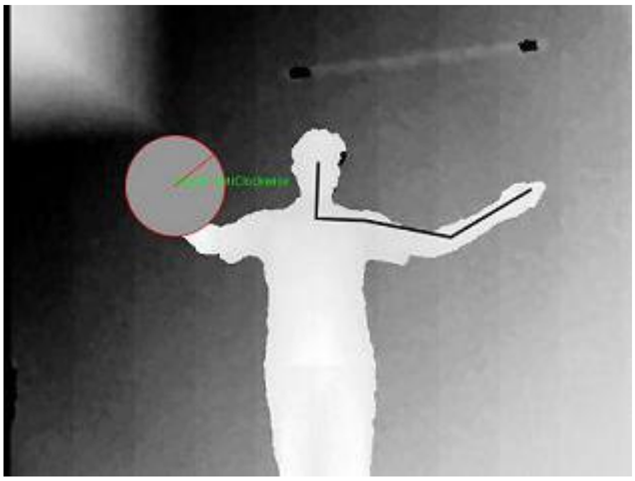

Fig.3. Gesture for the swivel motion

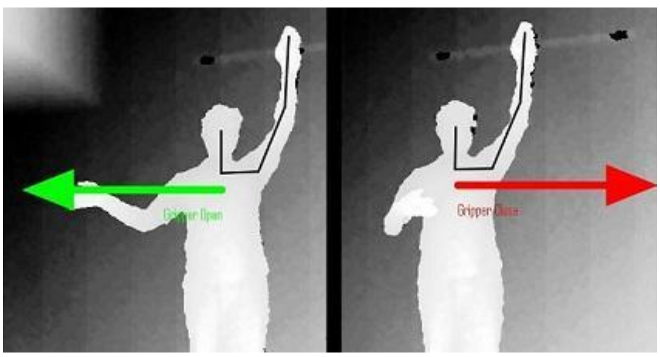

Fig. 4.Gripper Open/Close Gesture

All the four angles are continuously stored till a mouse click event occurs. Once that event occurs these values are lopped and published to the arduino.

IV. RESULT AND CONCLUSION

The robotic arm was successfully controlled using skeleton tracking. The elbow and shoulder rotations were controlled using the human elbow and shoulder rotations. The shoulder swivel and the gripper actuation was accomplished using the gestures mentioned above. The system was able to continuously store these movements and when initiated it was able to repetitively perform the programmed action very precisely. We found the programming method very easy and interactive.

Hence, the robotic arm will be repetitively tracing the motion performed by the user from the start of the tracking till the mouse click event occurs. This is how we program a robotic arm by actually moving our arm in the three dimensional space.

V. REFERENCES

[1] Greg Borenstein, Making Things See: 3D vision with Kinect, Processing, Arduino, and MakerBot.

[2] C. Keskin, O. Aran, and L. Akarun, “Real time gestural interface for generic applications, ” in European Signal Processing Conference, EUSIPCO’05, 2005.

[3] Mikell P. Groover, Mitchell Weiss, Roger N. Nagel, Nicholas G. Odrey, Industiral Robotics.

[4] C. M. Bishop, Pattern Recognition and Machine Learning. Springer, 2006.

[5] Richard Szeliski, Computer Vision:Algorithms and Applications.

[6] Enrique Ramos, Melgar Ciriaco, Castro Díez, Przemek Jaworski, Arduino and Kinect Projects Design, Build, Blow Their Minds

[7] http://openkinect.org/wiki/Main_Page.