

Controlling Servo Motor Angle by Exploiting Kinect SDK

Farzin Foroughi

College of Astronautics

Nanjing University of Aeronautics and Astronautics

Nanjing, China

Peng Zong

College of Astronautics

Nanjing University of Aeronautics and Astronautics

Nanjing, China

ABSTRACT

Nowadays controllers are one of main topic in engineering and researchers are to seek the produce simpler controller, so Body Gesture can be a good choice and it is one of popular way. This paper narrates the all proceeding of designing a system for control of a servo motor angle control using by body gesture of user. Hardware structure was wholly designed and implemented for this aim. Hardware structure was designed by an interface circuit that is based on microcontroller Atmega8 for analyze the data and generate command also circumambient architecture for control movement of different angle of the servo motor. For operate the servo motor were used Kinect. Kinect is used for receive information from user and provide communication between user and computer then send information of angle of servo motor to microcontroller. For suitable interaction for controlling servo motor by user, was created Graphic User Interface in visual C#. The main objective of this paper was to servo motor controlled by users without any background. No training is needed to user for controlling servo motor like traditional controller.

Keywords

Servo motor, Image Processing, Motion Capture, Human Computer Interface, Kinect.

1. INTRODUCTION

Remote controlling is a method for management of inaccessible system and there are many study and researches that have been done on this field. [1-4]. These researches have been conducted in specific field such as Medical, industry, art, military service and etc. the main implementation and usage of remote control are: work in hazardous environment and threat to human life. And also the place that need real-time controlling system [5-7]. Remote control has a prominent role studied by recent research in robotic science. Nowadays robots are being used in different fields especially in those missions that require high accuracy. Some of the implementation where robots can control by remote controller; are: exploring of unknown location or other planet, surgery, thwart the bomb and etc. [8]. Robots are being upgraded continuously because they have positive effect on development and human lifestyle [9]. Servo motor is a kind of technology that is being studies remote control and robotic science. As these kinds of motors are specially designed to be settled in specific angle or positions have an important role in robotic science.

Servo motors are self-contained device that use for those tools that needs high accuracy rotate. It is a small device with one output shaft. By sending special signal code to this shaft, servo motor will be set in specific angle. Till this signal code exists, servo motor will be kept in required angle. When signal code changes, servo motor will be turn to set in the new angle [10].

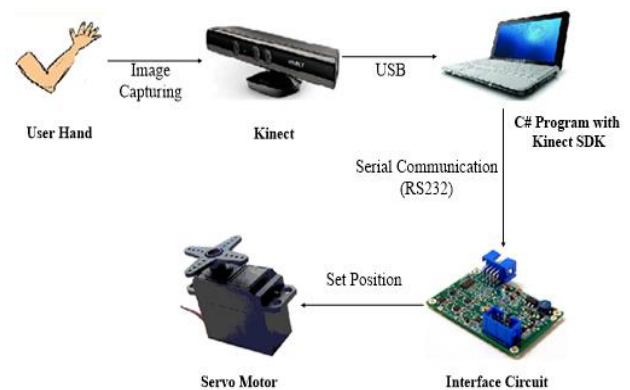


Fig 1: Scheme of whole process

This paper is based on control of servo motor via user's arm movement in front of Kinect and hardware controller. Hardware controller is used for receive data which Kinect obtained from user and transform these data in order to defined position of servo motor.

The movement that is done by the user is in 3D and Kinect catch the data then these data will be analyzed and then will send to the servo motor by signal code. This movement is obtained by Kinect in real-time [11]. This project has used elbow of user right hand movements which Kinect will use to control the spin of servo (Figure 1). Has written middleware in C# 2008 and it's based on the Microsoft Kinect SDK 1.8. Also hardware programming has written in Bascom. This study is aimed to servo motor controlled by users without any background. It means no training is needed for user to controlling of servo's angle like traditional controller. This study exhibit following steps:

- Getting information from user by Kinect.
- Analysis of the data by C# program.
- Sending analyzed information to interface circuit by serial communication.
- Commanding the servo motor to move to correct position.

2. BACKGROUND

Kinect is a composite device that is composed of IR projector, RGB camera and IR camera, which are together as a triangular process for measurement of depth in space [12]. Using depth camera and color camera can recognize details of image. RGB image, IR image and depth image can be rendered as a Kinect output [13]. Skeleton tracking is one of the most important ability of Kinect, for get more information beyond only the depth of each joint. Kinect uses depth stream to detect the skeleton tracking of human in front of camera.

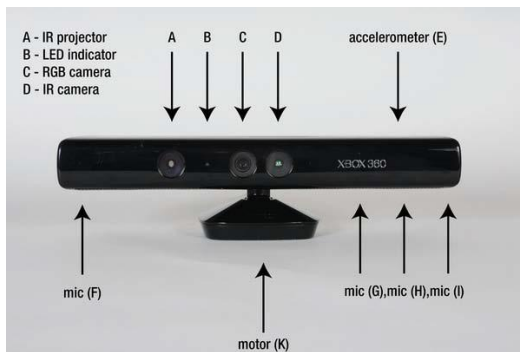


Fig 2: Kinect hardware at a glance

Due to this process Kinect can establish the positions of various skeleton joints on a human form [14]. A skeleton comprises 20 different positions, one for each joint of human body, as shown in Figure 3.

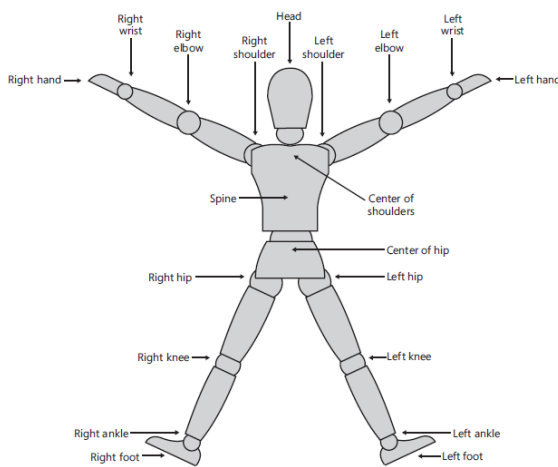


Fig 3: The 20 control points of a skeleton

Kinect was firstly designed for natural interaction in computer game but due its ability in the field of mapping and 3D modeling in space it became an attractive device for researches [15]. according to the prominence of Kinect and high usage of it in recent years and also to comfort the application development and raise research area by using Kinect, Microsoft released Kinect Software Development Kit (SDK) for Windows [16]. SDK allow developers to write their applications for Kinect in 3 different programming languages which are Microsoft Visual C++, C#.Net and Visual Basic. Microsoft Visual C#.Net which supports by SDK is widely used between programmers. C# is a multi-paradigm programming language which has strong environment for coding. Simplicity is one of the features of C# and also because of its advantage is considered by many programmers. There are many code examples in C# language to create solutions that support gesture by using the Kinect. In addition to C#, Microsoft Visual C++ and Microsoft Visual basic are also used by users which are supported by Kinect.

Tanaka et al. analyze the effect of motion detection software algorithms on the game control (Sony PlayStation Move, Nintendo Wii, and Microsoft Xbox 360 Kinect). Also they proposed an effective algorithm for motion detection by Kinect in game environment [17]. Raheja et al. introduced a method for fingertips detection and centers of palms detection distinctly for both hands using Kinect in 3D from the input image. In this method, the depth vector and centers of palms

were detected by using distance transformation on inverse image [18]. Vera et al. proposed Augmented Mirror method which is used two Kinect systems for motion capture, depth map and real images, and a gyroscope to detect head movements. This method implemented in simulation software for controlling an avatar. Although the propose method in this paper generated good performance, it needs many hardware, and also it is not suitable for real world application such as controlling robots in industries [19].

3. PROPOSED METHODE

To understand the whole process is provided a summery description of each part in this section. Figure 4 shows the block diagram of system which divided into several units. According to this figure, the User determines roles of angle of the servo motor. In fact, the initial and most important data produce by opening and closing elbow of user right hand. Opening and closing of this joint determines the angle of servo motor. After that, the Kinect sensor is used as the input and data transmitter device. It records the movement of the user right hand elbow in real time and transfers the data to the computer for processing. Then, Computer, by using the controller program which has written by C#, receives the information from Kinect, and processes and converts it into a skeleton image by Microsoft SDK toolkits. Moreover, this program calculates the angles of user right hand elbow by information of vector that it has received. After this step, Serial Interface receives the data from computer, and transmits it to the interface circuit. Next, Interface Circuit procreates PWM signals depending on the data receives to move a servo motor to an angle which choices by user. Data receives by microcontroller (Atmel At-mega 8) that is designed on interface circuit and after translating it generates signals. Lastly, Servo Motor moves to a specific angle by receiving PWM signal from microcontroller. This movement is exact that angle is selected by user. In fact it's final purpose. This paper explains the whole processes of proposed method, which are categorized in hardware and software parts, in the rest of this part.



Fig 4: Block diagram of whole process

3.1 Hardware

Interface circuit that is designed in this study is based on microcontroller. A microcontroller is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

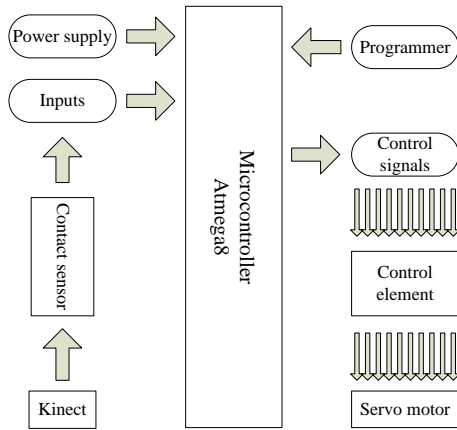


Fig 5: Block structure of the system



Fig 7: Final PCB

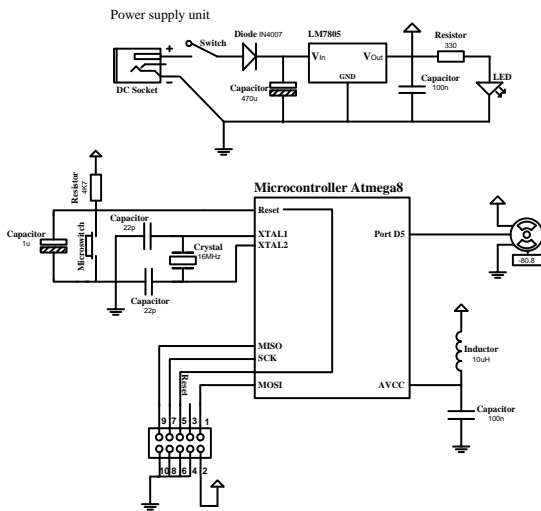


Fig 6: Schematic of hardware

Figure 5 shows the detailed hardware structure of the implemented distributed system. The device is a product of fully independent work. Production implied the implementation of the entire hardware circuit and construction of work algorithm. The structure consists of a microcontroller, a controlling element for the Servo motors.

Figure 6 shows schematic of hardware. In this figure, the servo motor allows for precise control of angular position, velocity and acceleration. It consists of a suitable motor coupled to a sensor for position feedback. It also requires a relatively sophisticated controller, often a dedicated module designed specifically for use with servo motors. Because of high accuracy Servo motors are used in applications such as robotics, CNC machinery or automated manufacturing.

Figure 7 illustrates the final Printed Circuit Board (PCB) of the schematic of Figure 5.

3.2 Software

3.2.1 Getting information from user by Kinect

Main act in first step is done by Kinect. User needs to stand in a fit distance in front of Kinect, and then it can recognize 20 joints of the person (Figure 8). Then gives output in coordinate of each joint is X, Y, Z axis. This project has done control servo motor angle using by user's right arm elbow, in another word when user moves right hand elbow to favorite position actually is making an angle and servo motor should turn to that angle. When user stands in front of Kinect, it can recognize 20 joint of user's body but only three positions is studied in this project. Right shoulder, Right elbow, Right wrist and right hand are these joints. We survey these three joints in order to control angle of servo motor by right arm. So we need information of our right arm and all information that Kinect can analyze and give as an output is these tree joint positions. So output of this step, X, Y, Z axis data of tree joint, is send to C#.

3.2.2 Angle computation by C# program

Prominent part is programming by C#. Microsoft company has written a library for those users want to write programs for Kinect in windows. This library can be attached in C#. In this part data gotten from user's movements by Kinect, transfers to computer and after that will be analyzed by C# program. As done Kinect have sent 3 joint position to the program that is written in c#. These positions are: Right shoulder, Right elbow, and Right wrist of right arm (Figure 9). With positions of these joints can create two vectors and calculate the angle between of these vectors.

As seen in Figure 9 X1, Y1, Z1 are coordinates of right shoulder, X2, Y2, Z2 are z. Angle of between two vectors can calculate by bellow equation:

$$\theta = \text{ArcCos} \frac{A \cdot B}{\|A\| \cdot \|B\|}, (1)$$

Figure 10 shows the pseudo code of calculating Equation 1.



Fig 8: Kinect recognize skeleton

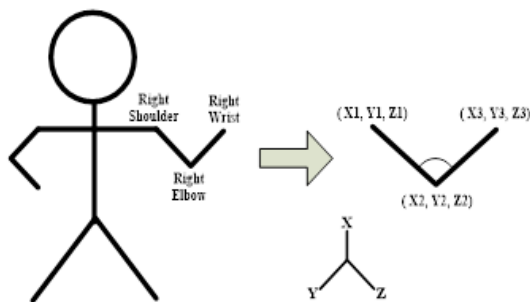


Fig 9: Constant scaling

Function Calculating-Angle

Initial inputs

Shoulder-Position-x, Shoulder-Position-y, Shoulder-Position-z,
Elbow-Position-x, Elbow-Position-y, Elbow-Position-z,
Wrist-Position-x, Wrist-Position-y, Wrist-Position-z

Consider outputs

Angle

Function body

Angle = 0;

VX is equal the subtract of Shoulder-Position-x and Elbow-Position-x;

VY is equal the subtract of Shoulder-Position-y and Elbow-Position-y;

VZ is equal the subtract of Shoulder-Position-z and Elbow-Position-z;

V is calculated by the second norm of "VX, VY, VZ";

UX is equal the subtract of Wrist-Position-x and Elbow-Position-x;

UY is equal the subtract of Wrist-Position-y and Elbow-Position-y;

UZ is equal the subtract of Wrist-Position-z and Elbow-Position-z;

U is calculated by the second norm of "UX, UY, UZ";

Numerator is calculated by the second norm of "VX * UX + VY * UY + VZ * UZ"

Mhshu=VX * UX + VY * UY + VZ * UZ

X = Mhshu / (V * U);

If the type of x is double and X is between or equal of 1 and -1 then

Angle-Rad is the arch sinus of X

Angle = Angle-Rad * (180.0 / π)

Else

Angle = 0

End if

End Function

Fig 10: Function of Calculating Angle

3.2.3 Sending analyzed information to by serial communication

The important part in third step is sending appropriate information from computer (C#) to interface circuit. Interface circuit is composed of microprocessor and peripheral circuits.

Correct transferring information evaluated in second step to microprocessor is important. This transferring is done by serial communication. In telecommunication and computer science, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

3.2.4 Commanding the servo motor to move to correct position:

In this part information is taken from computer is including of command of moving servo motor in appropriate position. Microprocessor's task is to translate data received from C# and send appropriate information to each motors to be in correct position.

3.2.5 Graphic User Interface (GUI):

The designed GUI for this project is shown in Figure 11, which is implemented by Microsoft C#.Net. As a whole, it is designed in five section; i.e. camera, skeleton, connect to micro, angle of joint, and camera position. Firstly, camera shows the normal video of a user who is stand in front of it. Next, skeleton section shows 20 joint points of the body. For these two, user only needs to stand in front of camera then Kinect SDK can recognize skeleton. Then, user can select the COM port and the baud speed for transferring data from computer to microcontroller. After that, angle of right elbow of user will be illustrated. If user move (open or close for this joint) the right elbow (Figure 9), angle of this joint will be calculate real-time. This calculation is shown in Figure 10. Finally, the angle of the Kinect can be changed. A lot of people are different heights so user may need to change the angle of Kinect to fit it in suitable position, so here user only with some click on the slider can change angle of Kinect and no need to trying to force the Kinect with hand.

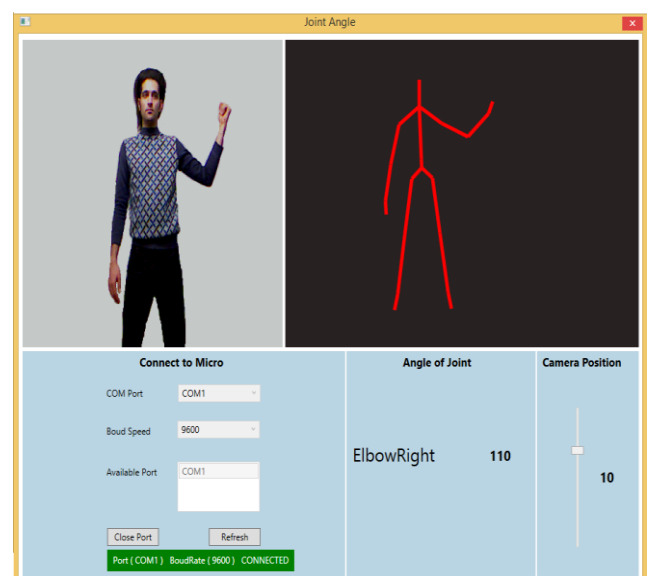


Fig 11: Graphic User Interface

4. EXPERIMENTAL RESULTS

As the propose method use in the physical environment, there is no standard metric for evaluating these type of methods [17-19]. As a result, we have designed some practical experience for evaluating our proposed method. This paper implemented some of related methods in C# environment, and compared the performance, response time, and error ratio of them with our proposed method. All of these results reported by averaging 50 times of repeating each experience, separately. The results will be illustrated in the rest of this part. Figure 12 demonstrates average of performance in our proposed method based on the number of action in each task.

In Figure 12, the vertical axis refers to performance and the horizontal axis refers to the number of actions in each single task. As this figure shows, the proposed method generated acceptable performance in most of the time. Figure 13 shows the average of response time in our proposed methods.

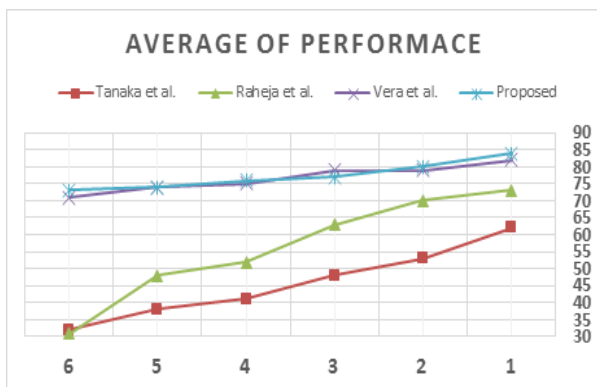


Fig 12: Average of performance

In Figure 13, the vertical axis refers to response time in second and the horizontal axis refers to the number of actions in each single task. As this figure shows, although the proposed method uses a Kinect for processing data, the response time and performance (see the Figure 12) of it are better in comparison with other methods. Figure 14 illustrates the error ration in our proposed method.

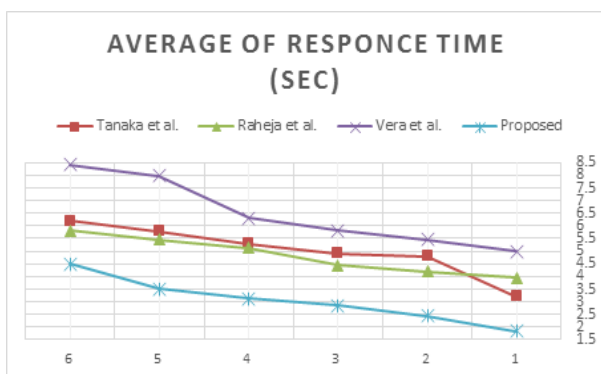


Fig 13: Average of response time

In Figure 14, the vertical axis refers to average of error number for sub-actions in 50 times independence running and the horizontal axis refers to the number of actions in each single task.

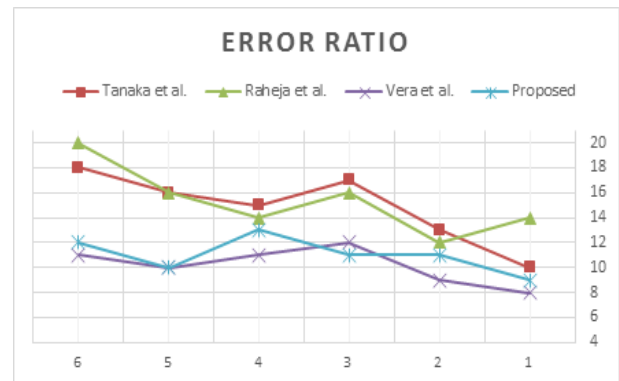


Fig 14: Error ratio

As this Figure 14 shows, although the proposed method cannot handle accurately all of tasks in comparison with some of previous method (Vera et al. method), it is acceptable because this paper uses only one Kinect for generating experimental results. Solving this problem can be as a potential for future works of this method.

5. CONCLUSION

This study introduced a controlled Servo motor angle by using Kinect SDK, remotely. This paper proposes an operating hardware, Graphic User Interface and algorithm. User can control servo motor angle by using right hand via Kinect. This system is based on microcontroller. Easy to develop and to be low cost are two main benefits of the proposed method. The Kinect, which is used as a sensor detects motion in this project according to its ability, is becoming popular device between researchers in this field. Controlling Body Gesture by using Kinect becomes popular way to communication and control many machines. By improving this project can control robot arm movement with proposing new framework based on Kinect SKD. In other words, user can stand in front of Kinect and start to move his/her arm, all of user's arm movement will be recognized by Kinect and then will be repeat (simulate) by robot arm. We have plane for design new hardware contains of robot arm.

6. ACKNOWLEDGMENTS

We would like to thank the Center for Artificial Intelligence and Robotics, Nanjing University of Aeronautics and Astronautics (NUAA) for providing the research facilities. This research work has been supported.

7. REFERENCES

- [1] T. B. Sheridan, (1993), Space teleoperation through time delay review and prognosis, IEEE Transaction on Robotics and Automation, October, vol. 9, pp. 592-606.
- [2] C. Sayers, (1999), Remote Control Robotics, New York: Springer Verlag, ISBN-10: 0387985972.
- [3] J. Velagic, M. Coralic, and M. Hebibovic, (2004). The Remote Control of Robot Manipulator for Precise Time-Limited Complex Path Tracking, Proceedings of the IEEE International Conference on Mechatronics and Robotics, Volume 2, 13-15 September, Aachen, Germany, pp. 841-846.
- [4] D. Lee, and M.W. Spong, Passive Bilateral Teleoperation with Constant Time Delay, (2006) IEEE Transactions on Robotics and Automation, April, vol. 22, no.2, pp. 269-281.

- [5] (2002), Bluetooth and Wireless LAN Applicability for Real-time Control, September, Lecture note in Tampere University of Technology, http://ae.tut.fi/research/AIN/Publications/automaatio03/paper_vladimir_lucanfinal.pdf.
- [6] Xin Liu , Yongtian Wang , Yue Liu , Dongdong Weng, and Xiaoming Hu, (2009), A Remote Control System Based on Real-Time Image Processing, 5th International Conference on Image and Graphics (ICIG '09), September 20-23, Xi'an, Shanxi, China, pp. 763 767.
- [7] Chui Yew Leong, and Abdul Rahman Ramli, (2011), Development of a real-time embedded remote triggering and monitoring system with SC12, Intelligent Systems and Robotics Laboratory (ISRL), Institute of Advanced Technology, March 5, Universiti Putra Malaysia. 43400 Serdang, Selangor.
- [8] S. E. Salcudean, S. Ku, and G. Bell, (1997), Performance Measurement in Scaled Teleoperation for Microsurgery, Proceedings of the 1st Joint Conference in Computer Vision, Virtual Reality and Robotics in Medicine and Medial Robotics and Computer-Assisted Surgery (CVRMed-MRCA 97), Grenoble, France, pp. 789-798.
- [9] Vedran Vajnberger, Tarik Terzimehi, Semir Silajdi and Nedim Osmi, (2011), Remote Control of Robot Arm with five DOF, Proceedings of the 34th International Convention MIPRO, 23-27 May, pp. 1707 1711.
- [10] J.F. Gieras, R.J. Wing, (2009), Permanent Magnet Motor Technology Design and Applications, August 26, Marcel Dekker Inc., New York, USA.
- [11] R.K. Megalingam, N. Saboo, (2013), Kinect Based Gesture Controlled Robotic Arm: A research work at HuT Labs, (2013), IEEE International Conference on Robotics and Automation, 20-22 December, Jaipur, India, pp. 294 299.
- [12] K. Khoshelham, and S. O. Elberink, (2012), Accuracy and resolution of Kinect depth data for indoor mapping applications, *Sensors*, vol. 12(2), pp. 1437-1454.
- [13] Jan Smisek, Michal Jancosek, and Tomas Pajdla, (2013). 3D with Kinect. *Consumer Depth Cameras for Computer Vision*, pp. 3-25.
- [14] Webb, J., Ashley, and J. Beginning, (2012), *Kinect Programming with the Microsoft Kinect. SDK*, a press Media LLC, New York, NY, USA, 2012; pp. 8586.
- [15] K. Khoshelham, (2011), Accuracy analysis of Kinect depth data, *Proceedings in ISPRS Workshop Laser Scanning*, vol. 38, pp. 133-138.
- [16] M. R. Islam, S. Rahaman, R. Hasan, R. R. Noel, A. Salekin, and H. S. Ferdous, (2013), A novel approach for constructing emulator for Microsoft Kinect X B O X 360 Sensor in the .NET platform, In proceedings of the 4th international conference on Intelligent Systems Modelling & Simulation, January 29-31, Bangkok, Thailand, pp. 1-6.
- [17] K. Tanaka, J. Parker, G. Baradoy, D. Sheehan, J. R. Holash, and L. Katz, (2012), A Comparison of Exergaming Interfaces for Use in ehabilitation Programs and Research, vol. 6, no. 9, ISSN 1923-2691.
- [18] J.L. Raheja, A. Chaudhary, and K. Singal, (2011), Tracking of Fingertips and Centre of Palm using Kinect , In proceedings of the 3rd IEEE International Conference on Computational Intelligence, Modelling and Simulation, Malaysia, 20-22 September, pp. 248-252.
- [19] Luca Vera, Jess Gimeno, Inmaculada Coma, and Marcos Fernandez, (2011), Augmented Mirror: Interactive Augmented Reality System Based on Kinect, 13th IFIP TC 13 International Conference, Lisbon, Portugal, 5-9 September, pp. 483-486.