Gesture Controlled AI-Robot Using Kinect

B Jayasurya, Jino Justin, C Kharat Pooja,
Mrutyunjay A Hasaraddi and T Kavitha

January 8, 2021

Gesture Controlled AI-Robot Using Kinect

Jayasurya B, Jino Justin,Kharat Pooja C,Mrutyunjay A Hasaraddi,Dr. T Kavitha

Dept. of Electronics and Communication Engineering

AMC Engineering College
Bengaluru,India

***Abstract*— Gesture recognition for human computer interaction is an area of active research in artificial intelligence and computer vision. To estimate gesture recognition performance in a real-life environment, we collect the gesture data that takes into account cluttered backgrounds, various poses and movement of robots, and then we evaluate the performance of robot. This involves skeleton tracking, where the skeleton data is produced from the depth images obtained via Microsoft Kinect sensor. The human gestures in 3 D space is captured by the kinect processed and replicated by the robot. An Arduino controller is used to govern the motion of robot which takes the input as joint angles from the kinect sensor and feeds it back to robot circuit and thus controls the robot actions. The primary goal of gesture recognition research is to create a system, which can identify specific human gestures and use them for device control. The gesture control robot will save huge cost of labour in future. The basic advantage of this robot is that it will be cost effective and no remote control is required.**

Keywords— Gesture, kinect, robotic movements.

I. Introduction

Gestures are visible body actions used for non-verbal communications to convey important messages. Gesture recognition is a technology for the extraction of human's meaningful expressions with mathematical interpretations through computing devices.

Gestures are usually divided into two types: static and dynamic gestures. A static gesture is a steady posture, represented by a single image which is captured. A dynamic gesture is a moving gesture, represented by a sequence of various images which are captured continuously. Algorithms on static gesture recognition have developed rapidly in recent years, such as gesture recognition based on artificial neural network and computer vision. However, simple static gesture cannot meet the requirements of the industry application and methods on dynamic gesture recognition have become a focus in the area of research instead. Owing to the diversity and complexity of gestures and flexible positions and shapes of hands, it makes gesture recognition a multi-discipline and challenging project.

Human-Computer-Interaction technology has become very important over the last years. The desire to provide a natural interaction between humans and machines has become a focus on gesture recognition. With the development of computer vision technology and depth cameras, the ways people interact with the electrical devices are being improved. In this project, Microsoft Kinect sensor is applied in a remote robot control system to recognize different gestures and generate visual Human-Robot interaction interface without calculating complex inverse kinematics to make the robot follow the posture of human. This kind of system aims to enrich the interactive way between human and robots which help non-expert users to control the robot freely, making human-robot interaction much easier..

II. Related works

Gesture Recognition or Sign Language Recognition (SLR) has been studied for decades since human computer interaction stepped into people's lives. Computational interpretation and translation systems can facilitate daily communication for speech and hearing challenged people. Understanding the structure of sign languages is the starting point to further solve the problems in developing hand motion tracking method. Literature on SLR, especially about 2D video-based recognition, is reviewed in Section 2.2.1

Studies aiming at tracking information of hands, including shape, position, and motion, can be traced back to the 1970's. Optical, magnetic, or acoustic sensing devices were attached to hands to report their positions. Later on, a glove-based system was described and implemented, which became a common and matured approach in the field of hand tracking. Throughout the past 20 years, a large variety of glove devices as input media for HCI have been built, of which some have remained in research labs and others have reached the marketplace. Literature about recent glove-based hand tracking research is reviewed in Section 2.2.2.

A more recent technology for gesture recognition is to incorporate the information of object distances, or depths, normally captured by a 3D camera, or a set of cameras that produce 3D images. It is also a contact-less user interface, in contrast to glove-based devices or handheld remote sensors. Different algorithms are proposed to separate objects from the background and to extract features of the objects, such as the body skeleton or hands, in order to track their motion. Literature related to depth-based recognition is reviewed in Section 2.2.3.

Some researchers were done with Kinect since its first launch in 2010. Among them there are several systems focusing on hand gesture recognition, most of which were making use of depth- based techniques. Literature about most related work on using Kinect is reviewed in Section 2.2.4.

A. *Sign Language Recognition*

Starner et al. presented two vision-based SLR systems using hidden Markov models (HMM): one used a second-person view with a desk mounted camera and the other was the first-person view with a camera mounted on a hat worn by the user. HMM was used for training and continuous motion tracking. Both systems used a skin color matching algorithm for hand tracking. Once a pixel of skin color was found. They checked the eight nearest pixels to search for similar color areas. The facial area was discounted based on the assumption that its position is almost fixed while hands are always moving. They were not able to separate two hands when they overlap each other due to the 2D video limitation. Therefore, they simply assigned the whole area to each hand whenever occlusion happened. Both systems were trained to recognize American Sign Language (ASL) sentences randomly chosen from the form of "personal pronoun, verb, noun, adjective, (the same) personal pronoun" for a 40-word lexicon; four hundred sentences were used for training and one hundred sentences were used for testing. In comparison, the second-person view system had a word accuracy of 92% while the first-person view system had a word accuracy of 98%. The high accuracy indicated that HMM is good for the purpose of continuous motion tracking. However, neither of these two systems has provided a feedback view for the signer himself. In addition, there is no good solution to compensate for head and hand rotations, especially for the first-person view system, in which the rotation of the head may significantly affect the quality of recognition. As lexicon grows large, it requires defining and extracting additional features to maintain the accuracy. Furthermore, no finger-spelling recognition was attempted in this project.

B. Glove-Based Gesture Recognition

Kenn *et al.* came up with a way to integrate glove-based devices into multiple applications with the help of a context framework. The textile glove with integrated electronic device was cool in appearance, and was able to perform in at least three applications as demonstrated: to move/zoom/select parts of a map, to navigate to a remote control in presentation, and to direct a toy robot to move left/right, forward/backward. Gestures were chosen to be simple but sufficient in interacting with applications. One problem was that this device can only detect gestures in the X and Y axes, without the ability to detected motion in the Z axis such as the so-called "yaw". In addition to achieve wearability, light weight, and cool appearance, the accuracy of recognition was sacrificed.

The glove systems described so far mainly work with information. However, real life applications involve gestures in the 3D space. More recently, the trend of glove-based systems is pointing to 3D recognition and the reduction of price without sacrificing accuracy.

Kim *et al.* developed a 3D hand motion tracking and gesture recognition system using a data glove called KHU-l. The data glove interacted with a PC via a Bluetooth device. It successfully performed hand motion tracking such as fist clenching, hand stretching and bending. A rule-based algorithm was used in simple HGR at two different angular positions: horizontal and vertical. Three gestures (scissor, rock, and paper) were tested with an accuracy of 100% for 50 trials each. The recognition in 3D and the wireless transmission were good improvements, but they incurred time delay. However, the tested gestures were too simple and too few to prove the robustness of this system.

C. Gesture Recognition using Depth data

Van den Bergh *et al.* used a ToF camera instead of stereoscopic cameras to enhance recognition. A ToF camera with a low resolution (176 x 144 pixels) was used to get depth image for segmentation, and it was paired with an RGB camera with a high resolution (640 x 480 pixels) for hand detection. The ToF camera and RGB camera were calibrated at the beginning, and a threshold distance was defined to discard background images according to the depth data. The remaining pixels were passed through a skin color detection to get hand data. The skin color used for detection was decided by a pre-trained, adaptive skin color model, which was updated with color information taken from the face. Three situations· were evaluated in hand detection: the hand was next to the face, the hand overlapped with the face, and a second person was behind the tester. Depth-based detection achieved more than 98% accuracy in all three situations, while the accuracy of color-based detection decreased dramatically from 92% of the first situation to 19.8% of the third situation. Average Neighborhood Margin Maximization transformation was used to build the classifier for gesture recognition, where the Haarlet coefficients were calculated to match hand gestures stored in a database. The RGB-based recognition showed an accuracy of 99.54%, and the depth-based recognition showed an accuracy of 99.07%, while the combination of the two methods showed 99.54%. This suggests that depth-based recognition may be good enough to form the basis of a recognition system especially in hand segmentation.

D. Most related work using Kinect

After the launch of Microsoft Kinect in November 2010, several exciting recognition systems based on this device were developed in less than 18 months. The resolution of its HGB camera and depth camera are both 640 x 480 pixels, which are fairly sufficient under many situations.

Yang *et al.* proposed a gesture recognition system using depth information provided by Kinect, and implemented in a media player application. It was able to recognize eight gestures to control the media player, with a maximum confusion rate of 8.0%. The algorithm for hand tracking is to first find the hand waving motion based on the assumption that a user tends to start an interaction session with such a motion. A continuously adaptive mean shift algorithm was applied to track the hand by using the depth probability and updating the depth histogram at each frame. The hand trajectory was examined by a 3D feature vector, and the gesture was recognized by an HMM. This system demonstrates the applicability of using Kinect for gesture recognition in a contact-less UI.

III. Design and Methodology

Gesture recognition refers to the positioning of certain characteristic motions of the human body, target detection, extraction, recognition and tracking, and obtaining the state of human motion such as position, velocity and motion trajectory, thereby completing processing and analysis, and obtaining a certain amount of Valuable practical parameters to achieve automatic processing and analysis of human motion to complete the task of motion tracking, and apply to other aspects. This chapter consists of overall block diagram of our project work and the design flowchart. This system uses the Kinect visual image sensor to identify human bone data and complete the recognition of the operator's movements. Through the filtering process of real-time data by the host computer platform with computer software as the core, the algorithm is programmed to realize the conversion from data to control signals. The system transmits the signal to the lower computer platform with microcontroller as the core through the transmission mode of the wireless communication, thereby completing the control of the robot. The block diagram includes process like Input Gestures, Kinect, Skeleton Tracking, wireless communication module, hardware processing module and finally Robot control module. The respective flowchart of both transmitter and receiver side is also included, along with its stepwise explanation. The different methods and technologies used in our project of controlling a robot wirelessly through gesture recognition is also mentioned and explained in the following section.

A. *Block diagram*

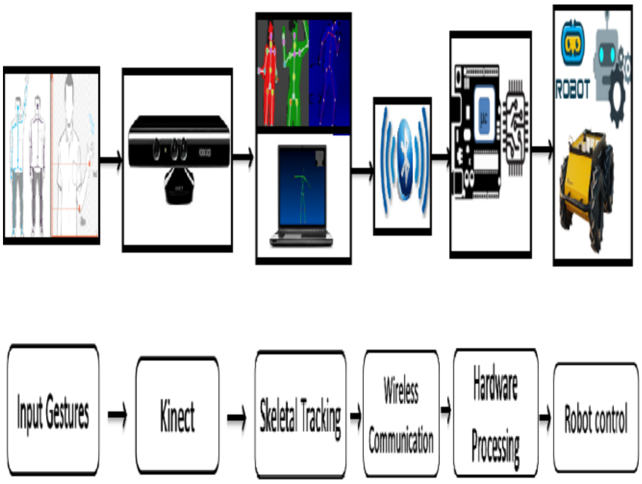

Fig. 3.2 describes the parts of the project in the form of block diagram. The overall block diagram of the control system consists of four main parts: A visual image processing sensor (input gestures & Kinect), a skeletal tracker, a hardware processor, and a robotic control platform (executive component). The sensor used here is Kinect visual image processing sensor, the data processor/skeleton tracking is controlled by computer platform, the hardware processor is completed by Microcontroller and related electronic modules, and the robotic control platform completes the actually expected action. The computer platform receives the real-time joint state and information returned from the Kinect visual image sensor in real time, and transmits it to the computer in the form of data stream. The data is processed in real time by the programmed algorithm, and then sent to the lower computer microcontroller in real time by wireless communication also can display the real-time joint angle in the software operation interface if needed.

- Gestures are given as input commands by the user. The body movements of the user are captured using the Kinect sensor, which consists of a depth sensor and RGB camera, obtains the skeletal data from the user. The information is acquired in the form of positions of the various joints using the 3 dimensional Cartesian co-ordinate spaces.

- The captured image data is sent serially through the USB port to a computer machine where the captured skeleton is displayed and the co-ordinates of the joints are processed. Processing of the gestures requires a program compatible with the Software Development Kit (SDK) of the Kinect. Thus the skeleton tracking is done. Using mathematical calculation, it calculates the angle deviation between the different joints based on the movements.

- After the recognition of the gestures is completed, the data is processed in real time by the programmed algorithm and the corresponding angle deviation data measured by the Kinect sensor is transmitted from the computer to the robot through any wireless transmitter. In this project we are using Bluetooth for this operation. The use of the wireless communication module will increase mobility.

- The transmitted data is received in the buffer of the Bluetooth module connected to the robot and given to microcontroller for further processing.

- The received information of the gesture via Bluetooth is processed in the microcontroller for various tasks to perform in real time. We are using Arduino board for performing this operation which will convert the received angle deviations values and generates the equivalent PWM. Thus the robotic control platform completes the actually expected action.

B. *Methodology*

Our system is divided into two sections: Transmitter and Receiver section. Fig 1(a) is a flowchart for Transmitter section Gesture recognition) and Fig 1(b) Receiver section (Robot control). After the start, the initialization object operation is needed to determine the tracking target, open the sensor, and start collecting the depth data and joint data using Kinect's image acquisition function. After filtering, display current image and data information in the software operation interface.

1) *Transmitter section*

a) **Data Acquisition:** In the data acquisition process, the gestures are captured in the form of video. The user has to stand in front of the device that is used to capture the input. Here Microsoft Kinect is used as an input capturing device. The Kinect can capture the RGB data as well as depth data. Using the Kinect sensor's visual perception technology, in principle, it automatically tracks the human joints within the target range before output, and displays a real-time dynamic map showing the position of the human joint. The coordinates acquired by Kinect's depth data acquisition are depth image coordinates, but in order to make the human motion vector calculation in the actual realistic three-dimensional coordinates, it is necessary to pre-process the data, complete the coordinate conversion, and thus calculate the subsequent joint angle

b) **Skeleton *Tracking*:** The significant innovation in Kinect is its skeleton tracking ability. The sensor can elegantly track the skeleton of the person standing in front of it. Locate and trace person's major joints and provide helpful content of its locations which can be processed by computer. This is done by the sensor in following manner:

i) Depth sensor: The depth sensor is a combination of IR projector (laser) and IR camera. Depth values are coded in gray values the closer the object the darker will be its depth image pixels. The objects distance from the sensor is flawlessly coded into gray values depending on the time taken by the IR lights to return back to the sensor.

ii) Joint Tracking: The joints are classified it to various categories like (head, hand, neck, shoulder etc.). The sensor in all track 20 major joints of human body. These joints are effectively located by the sensor and tracked continuously. Based on the coordinates of the joints a vector is drawn between two joints which represent the bone of a skeleton. The combination of the joints and vectors is represented as the skeleton of the user. The joints are interconnected to obtain the person's skeleton. This result is optimized by tracking the joints at a rate of 5ms per frame. The sensor also provides the 3-D co-ordinates of the tracked joints.

c) **Feature extraction:** Once we have the skeleton data of the user, we use basic trigonometric functions to calculate the angle between the vectors**.** The algorithm of the system is based on the feature recognition of the target operator's joints, and the depth coordinates are converted into three-dimensional real coordinates, thereby utilizing the nature of the space vector and the relative position between different joints to calculate the inter-bones. The angle, so that the computer completes the processing of the data returned by the Kinect sensor and sends out real-time control signals.

d) **Data Filtering:** Because Kinect has a fast refresh rate of 30 frames/s, it may cause the coordinates to change continuously during the same minute time period in the bone recognition process. Therefore, some means are needed to process the data collected in real time. This process is called "data filtering". In order to reduce the negative stability of its calculation, we adopt the average method which is easier to implement. We continuously use a fixed set of N real-time values as a set of data once, arranged in group, the group leader is fixed to N unchanged, the new data at the next moment is stored in the end of the group, and the data at the head of the group is removed, and so on, so that different values at different times are obtained, and the N data in the obtained group are taken average value. You can expect a more stable result.

$$y_k = \sum_{J=m}^{n} p_i y_{k+j}$$

$$k = m+1, m+2, \cdots, N-n$$

In the formula, pi is the weight coefficient, and $\sum_{i=m}^{n} p_i = 1$, the system goes to n = 10, that is, every 10 sets of data is processed as a group, m = 0, n = 9, the first 10 data from the above formula cannot participate in the calculation, because it can measure 30 times per second, there will be 30 sets of data within 1 second, that is $y_1 - y_{20}$ are valid. Every 10 numbers take the arithmetic mean to get 2 data, and finally 2 data values are returned. By averaging the joints corresponding to the degrees of freedom, actual three-dimensional coordinate values of the groups returned within each second can be obtained.

e) **Angular Feature Calculation:** After the filtered stable value is obtained, the angular feature calculation can be performed. For example, if a system operates with only 4 bones of one of the identified bones. First, the vector between the four bones is obtained from the actual three-dimensional coordinates, and a total of three sets of vector values are obtained, $\vec{a}, \vec{b}, \vec{c}$. According to the correspondence, the shoulder angle is, $\theta = \langle \vec{a}, \vec{b} \rangle$. The elbow angle is, $\beta = \langle \vec{b}, \vec{c} \rangle$ using these actual vectors to map the angular features.

$$\text{By} \quad \theta = \frac{[\vec{a} \cdot \vec{b}]}{|\vec{a}| \cdot |\vec{b}|}$$

$$\text{Got} \quad \theta_{i-j} = \frac{x_{i_1} x_{j_1} + x_{i_2} x_{j_2} + x_{i_3} x_{j_3}}{\sqrt{x_{i_1}^2 + x_{i_2}^2 + x_{i_3}^2} + \sqrt{x_{j_1}^2 + x_{j_2}^2 + x_{j_3}^2}}$$

$$\text{Calculated} \quad \begin{cases} x_i = x_{i_1} x + x_{i_2} y + x_{i_3} z \\ x_j = x_{j_1} x + x_{j_2} y + x_{j_3} z \end{cases}$$

Therefore, the angle is the angle characteristic value of the current two bones. Based on this, it can be extended to 20 bone recognitions of the human body. The angle between any two bones joint can be extracted. This identification method has a series of benefits, which not only is only operability and also eliminates interference from disturbances such as lighting, background and operator position. In summary, in order to obtain accurate and stable target return value, a large amount of real-time data must be filtered to improve system performance and reduce fault tolerance.

f) **Wireless Serial Transmitter:** Once we have the angle deviation values, we transfer the angle deviation through wireless communication module to receiver side. The wireless communication module allows us to increase the mobility of the robots by eliminating the connection of the robot to the computer. On the receiver side, we use the data uploaded through the wireless communication module to control the robot actions.

2) Receiver section

a) **Wireless Serial Receiver:** The wireless receiver will fetch the data from the wireless transmitter module. The receiver is linked to the robot's control system.

b) **Processing of the data received using microcontroller:** The programming of robot flows the similar way as that of Kinect sensor. Fig. 4 shows the flow chart of the same. The robot when initialized, the Bluetooth module scans for the devices for connection from the remote computer machine. Once the connection between the two is established, the gesture information is received to the microcontroller. The microcontroller processes the received data and scans for the relevant operation to be performed. If the received information is for motion of the robot, the corresponding PWM signals are generated.

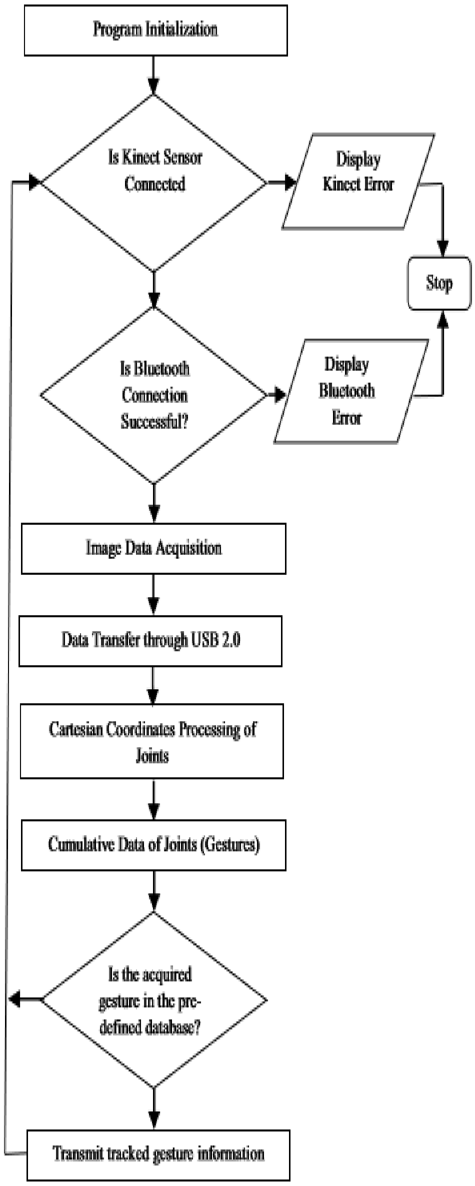

Fig 3.3(c) Flow chart for data processing in receiver side.

c) **Generation of PWM signals:** The PWM signals are Pulse Width Modulated Signals. They are analog signals generated from a digital source. If the PWM signals generated belong to the motor control, the received angles are first quantized and then the axel rotation of the motors takes place and the robot follows the commands of the user. The magnitude of the voltage determines the speed of the motor, and positive and negative represent the steering of the motor. When the voltage difference is 0, the motor stop rotating. The PWM square wave is the control signal of the robot, and the transient characteristics of the duty cycle are used to change the rotational characteristics of the robot. There's a provision for obstacle detection too using the ultrasonic sensor. It protects the robot from getting damaged by continuously monitoring its environment and maintaining safe distance.

d) **Control operations of the robot:** Robot can be controlled by simple gestures. The movements are processed by the computer in real-time and appropriate gesture is identified from the predefined feed.

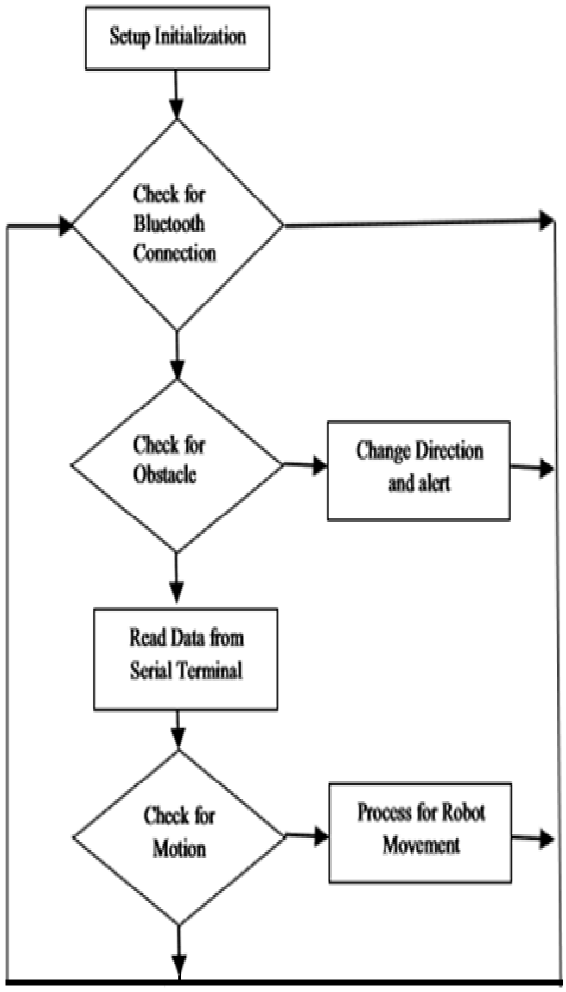

Fig 3.3(d) Flow chart for Robot control

e) **Motion Control**: In this project, we are designing a wheeled robot with four way motion. The robot is made self-equipped with collision detection. Furthermore we even make it more practical and portable by implementing it wirelessly using Bluetooth. However the use of 3Dimensional Gesture recognition used to operate the robot makes extremely easy to use. Any person can easily operate this robot on their own. The robot uses the microcontroller as the main controller, and completes the PWM square wave control of the robot motors. Before starting the start-up, the actual wireless connection is performed. Under different background environments and light intensities, identify the same actions of the operators and complete the corresponding requirements. The method to verify system stability and accuracy. The motors which are attached to the robot wheel control the further action of the robot. The primary work of this project is to create a user-friendly interface that recognizes different kind of gestures performed by the user. So, here are defined five types of gestures that should be associated with the robot movements. One of them is combined gesture; it means that it is combination of two different gestures associated with one robot movement. These kinds of gestures are as follows,

- Right Hand Forward, which is associated with movements of the robot for moving forward
- Left hand on the right side is associated with the movements of the robot for moving on the right
- Right hand on the left is associated with the movements of the robot for moving on the left
- Right hand backward or both hands backward are associated with the movements of the robots for moving its motors backward
- Both hands down means that robot does not move in any direction
- One of the gestures is combined gesture which means that user may associate two different gestures for determining one robot movement. This movement of the robot consists of these gestures: Right hand forward or both hands forward which is the same, combined with the left hand on the right, which gives the command to the robot for moving Right-Forward.

IV. Conclusion

Hereby the paper concludes that through gesture recognition, robots can be controlled easily and naturally. We design a control system for the given robot which allows the user to control the robot wirelessly using the actions or gestures. This project contributes greatly in the fields of kinect applications development, motion detection, human-computer interaction and gesture recognition. The experimental data shows that the Kinect- based motion recognition method can effectively complete the tracking of the expected motion and compute the robotic control, which has extremely high operability.

Through the tracking simulation of the movement of the human body by the motion recognition system, it wil replace the work of high risk, high difficulty and difficult working environment in the future.

References

[1] M. Cho and Y. Jeong, "**Human gesture recognition performance evaluation for service robots,"** 2017 19th International Conference on Advanced Communication Technology (ICACT), Bongpyeong, 2017, pp. 847-851, doi: 10.23919/ICACT.2017.7890213.

[2] Y. Liu, M. Dong, S. Bi, Dakui Gao, Y. Jing and L. Li, "**Gesture recognition based on Kinect**," 2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), Chengdu, 2016, pp. 343-347, doi: 10.1109/CYBER.2016.7574847.

[3] D. Maraj, A. Maraj and A. Hajzeraj, "**Application interface for gesture recognition with Kinect sensor**," 2016 IEEE International Conference on Knowledge Engineering and Applications (ICKEA), Singapore, 2016, pp. 98-102, doi: 10.1109/ICKEA.2016.7803000.

[4] E. Yavşan and A. Uçar, "**Teaching human gestures to humanoid robots by using Kinect sensor,"** 2015 23nd Signal Processing and Communications Applications Conference (SIU), Malatya, 2015, pp. 1208-1211, doi: 10.1109/SIU.2015.7130053.

[5] Y. GU, H. Do, Y. Ou and W. Sheng, "**Human gesture recognition through a Kinect sensor**," 2012 IEEE International Conference on Robotics and Biomimetic (ROBIO), Guangzhou, 2012, pp. 1379-1384, doi: 10.1109/ROBIO.2012.6491161.

[6] T. Starner, J. Weaver and A. Pentland, **"Real-time American sign language recognition using desk and wearable computer based video,"** in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pp. 1371-1375, Dec. 1998, doi: 10.1109/34.735811.

[7] H. Kenn, F. V. Megen and R. Sugar, **"A glove-based gesture interface for wearable computing applications,"** 4th International Forum on Applied Wearable Computing 2007, Tel Aviv, Israel, 2007, pp. 1-10.

[8] C. Yang, Yujeong Jang, J. Beh, D. Han and H. Ko, **"Gesture recognition using depth-based hand tracking for contactless controller application,"** 2012 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, 2012, pp. 297-298, doi: 10.1109/ICCE.2012.6161876.

[9] J. Kim, N. D. Thang and T. Kim, **"3-D hand motion tracking and gesture recognition using a data glove,"** 2009 IEEE International Symposium on Industrial Electronics, Seoul, 2009, pp. 1013-1018, doi: 10.1109/ISIE.2009.5221998.

[10] Guan-Feng He, Sun-Kyung Kang, Won-Chang Song and Sung-Tae Jung, **"Real-time gesture recognition using 3D depth camera,"** 2011 IEEE 2nd International Conference on Software Engineering and Service Science, Beijing, 2011, pp. 187-190, doi: 10.1109/ICSESS.2011.5982286.

[11] J. L. Hernandez-Rebollar, R. W. Lindeman and N. Kyriakopoulos, **"A multi-class pattern recognition system for practical finger spelling translation,"** Proceedings. Fourth IEEE International Conference on Multimodal Interfaces, Pittsburgh, PA, USA, 2002, pp. 185-190, doi: 10.1109/ICMI.2002.1166990.

[12] Han J., Jang W., Jung D., Lee E.C. (2017), **"Human Robot Interaction Method by Using Hand Gesture Recognition".** In: Park J., Chen SC., Raymond Choo KK. (eds) Advanced Multimedia and Ubiquitous Engineering. FutureTech 2017, MUE 2017. Lecture Notes in Electrical Engineering, vol 448. Springer, Singapore. https://doi.org/10.1007/978-981-10-5041-1_17

[13] Cekova, Katerina & Koceska, Natasa & Koceski, Saso. (2016**). "Gesture Control of a Mobile Robot using Kinect Sensor"**. International Conference On Applied Internet And Information Technologies. doi:251-258. 10.20544/AIIT2016.31.
[14] Ahmad, Abdel-Mehsen & Bazzal, Zouhair & Al Youssef, Hiba. (2017). **"Kinect-Based Moving Human Tracking System with Obstacle Avoidance"**. Advances in Science, Technology and Engineering Systems Journal. Doi:2. 191-197. 10.25046/aj020325.
[15] J. A. Tumialán Borja, E. Bernal Alzate and D. L. Mariño Lizarazo, **"Motion control of a mobile robot using kinect sensor,"** 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC), Cartagena, 2017, pp. 1-6, Doi: 10.1109/CCAC.2017.8276474.