Controlling mechanical arms using 3D coordinates gotten from the AR technology

Yuxiang Ren[1], and *Cheng* Cai[1,*]

[1]School of Electronic Information Engineering, Shanghai Dianji University, Shanghai, China

Keywords: Mechanical arms, Arduino, AR application, 3D coordinates, the angle information.

Abstract. How to control the movement of the mechanical arm is a challenging topic recently. It requires much time and energy to program every joint. This paper proposes a new method based on 3D coordinates of the human arm joints, which helps the mechanical arm to imitate the movement of human arms and only needs to calculate angles of human arms' joints according to the 3D coordinates. Here in this process the key is to get the information of 3D coordinates of human arms. This paper applies an AR (Augmented Reality) application of body detection with LiDAR on iPad Pro 2020 to get 3D coordinates. Then the calculated angle information is transferred from the host computer to the Arduino board controlling the movement of the mechanical arm. Eventually the same movements will be made by the mechanical arm and the human arm.

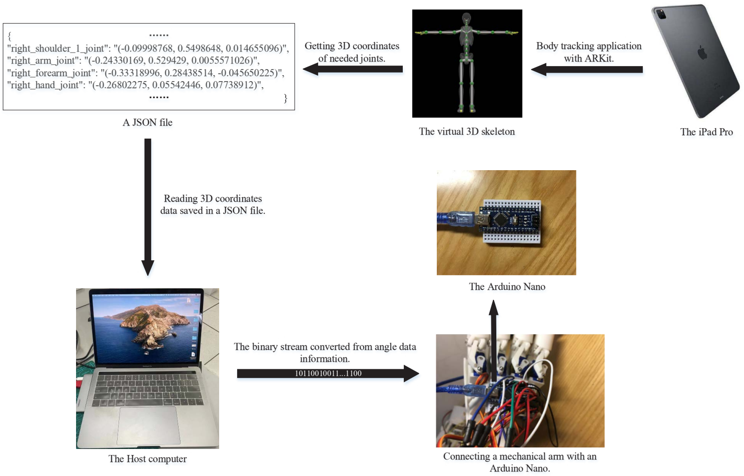
1 Introduction

Traditionally, the method is to put a pair of data gloves on human arms, and use the analogue signal generated on the glove to drive the mechanical arm to follow the movement of human arms. But it is not convenient for humans to wear a pair of data gloves as there are lots of wires and potentiometers on the data glove. This paper aims to search for another effective method based on computer vision to replace data gloves. Therefore, it is of great significance to realize the system proposed above.

In recent years, AR applications have had rapid development, providing an excellent way to get accurate 3D coordinates of human arms' joints. ARKit is a suite of tools for building AR applications on iOS, first released in 2017[1]. In 2019 Apple Inc. has revealed ARKit3 understanding the position of people real-timely in the scene by using computer vision[1]. One ARKit3 technology of Apple Inc.—body tracking—is used to get 3D coordinates on iPad Pro 2020. Then according to these 3D coordinates, the angle data information of human arms' joints can be calculated precisely. The architecture is completed by the hardware component, which consists of a RGB camera on the iPad Pro to look at the scene[5].

[*] Corresponding author: caic@sdju.edu.cn

Arduino Nano driving the movement of the mechanical arm in this paper is a small, complete, and breadboard-friendly board based on the ATmega328P released in 2008[2]. It can be programmed using the Arduino IDE, which is common to all Arduino boards, running both online and offline, and includes a serial monitor allowing simple textual data to be sent to and from the Arduino board[3]. The angle information is converted into a binary data stream and transmitted from the host computer to the Arduino board. The whole operating steps are shown in Fig. 1.



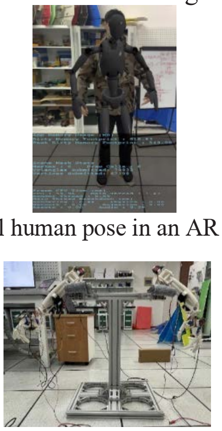
```
{                                   ......
"right_shoulder_1_joint": "(-0.09998768, 0.5498648, 0.014655096)",
"right_arm_joint": "(-0.24330169, 0.529429, 0.0055571026)",
"right_forearm_joint": "(-0.33318996, 0.28438514, -0.045650225)",
"right_hand_joint": "(-0.26802275, 0.05542446, 0.07738912)",
                                    ......                        }
```

A JSON file
Getting 3D coordinates of needed joints.
The virtual 3D skeleton
Body tracking application with ARKit.
The iPad Pro
Reading 3D coordinates data saved in a JSON file.
The Arduino Nano
The Host computer
The binary stream converted from angle data information.
10110010011...1100
Connecting a mechanical arm with an Arduino Nano.

Fig. 1. Main operating steps.

2. The Apple ARKit technology

ARKit3 released in 2019 has a body tracking function and makes great contributions to the increment of accuracy of 3D coordinates because of LiDAR. Concretely, it can track a person in the physical environment and visualize their motion in 3D by applying the same body movements to a virtual character as shown in Fig. 2.



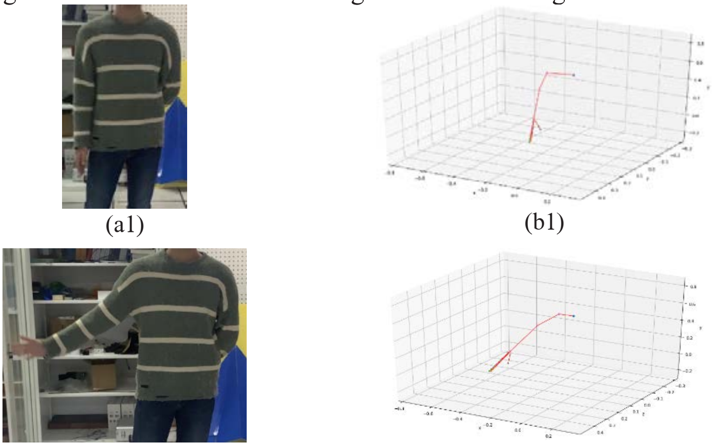
(a) The initial human pose in an AR application.

(b) The initial placement of the mechanical arm.

Fig. 2. The virtual scene and the real scene.

As can be seen from Fig. 2, the body tracking application with ARKit processes a video stream whose FPS is 60, thus demonstrating extremely good real-time performance.

Therefore, the 3D coordinates of human arms' joints can be obtained real-timely in each frame. Then these 3D coordinates can be saved as a file on iPad Pro 2020 for use in the next stage. Fig. 3 below has visualized an image of the tracked right arm's skeleton.



(a1) (b1)

(a2) (b2)

(a1), (a2) are the AR application's screenshot capturing the pose of a real person's right arm.
(b1), (b2) are the imitation of the corresponding real person's arm pose.

Fig. 3. Simulating the right arm's pose of a real person in a virtual world.

The accuracy of the obtained coordinates can be checked as shown in Fig. 3. The code of the application and the position of the camera on iPad can be adjusted in time according to simulation results.

3. Mechanical arms and an algorithm for calculating angles

3.1 Mechanical arms

Besides an Arduino Nano board is installed on the back of the mechanical hand as shown in Fig. 1. There is one servo which is used to control the mechanical elbow. There are two servos used to control the wrist and five servos controlling fingers respectively. Rotation ranges of all relevant servos are 0 to 180 degrees. The servo controlling the mechanical elbow' movement is mainly used in this paper.

3.2 The algorithm

In Section 2, the precise 3D coordinates have been achieved by applying an AR application with the body tracking function, saved as a file in iPad. During the movement of the real person's elbow, its angle must be able to be calculated by using the following formula.

$$\cos\theta = \frac{a \cdot b}{|a||b|} \tag{1}$$

The above two formulas are the classic formulas in mathematics. Assume that the elbow joint's coordinate to the shoulder joint's coordinate is $\boldsymbol{a}$ whose coordinate is (x_1, y_1, z_1) and that the elbow joint's coordinate to the wrist's coordinate is $\boldsymbol{b}$ whose coordinate is (x_2, y_2, z_2). Then three formulas can be obtained as follows.

$$a \cdot b = x_1 x_2 + y_1 y_2 + z_1 z_2 \tag{2}$$

$$|a| = (x_1^2 + y_1^2 + z_1^2)^{\frac{1}{2}} \tag{3}$$

$$|b| = (x_2^2 + y_2^2 + z_2^2)^{\frac{1}{2}} \tag{4}$$

Substituting formulas (2), (3), and (4) for formula (1), then the formula (5) can be obtained.

$$\cos\theta = \frac{x_1 x_2 + y_1 y_2 + z_1 z_2}{(x_1^2 + y_1^2 + z_1^2)^{\frac{1}{2}} (x_2^2 + y_2^2 + z_2^2)^{\frac{1}{2}}} \tag{5}$$

In the above formula (5), the variable θ stands for the angle of bend elbows while moving.

4. Communicating between the mechanical arm and the Arduino Nano board

The next key step is to transfer correctly the calculated angle data to the Arduino Nano. Because of the serial communication technology of Arduino Nano boards, these angle data generated by each frame need to be converted into one byte on the host computer before they can be received by Arduino Nano boards. The whole communication process between them is presented in Fig. 4.

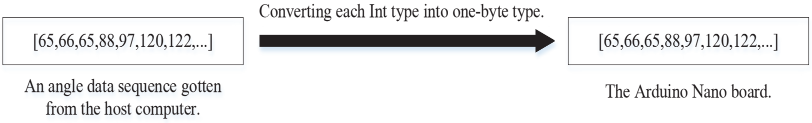

Converting each Int type into one-byte type.
[65,66,65,88,97,120,122,...]
[65,66,65,88,97,120,122,...]
An angle data sequence gotten from the host computer.
The Arduino Nano board.

Fig. 4. The whole communication process.

Virtually the host computer transfers ASCII values to the serial port and then the Arduino Nano board reads data from the serial port one by one, converting ASCII values into integer data automatically.

5. Experimental results and conclusions

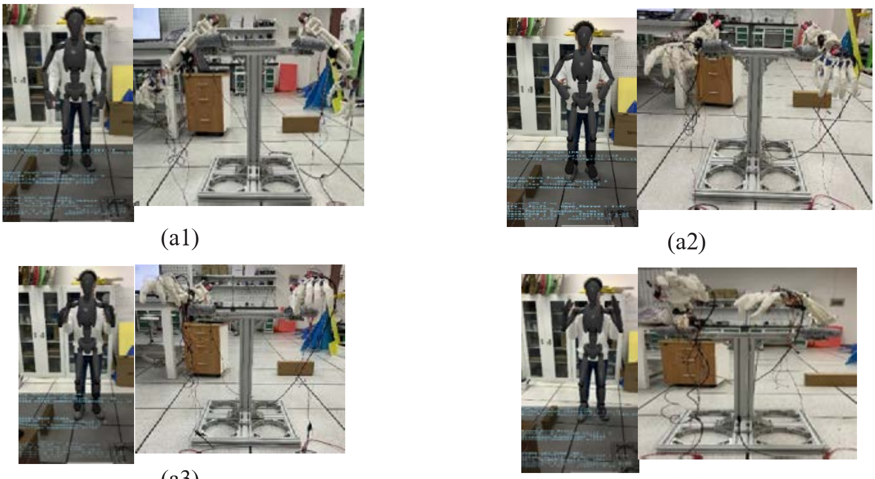
5.1 Experimental results

The result of the mechanical arm imitating the human arm posture is shown in Fig. 5.

As can be seen from Fig. 5, the mechanical arms can imitate human arms' posture accurately with a relatively high degree of completion.

5.2 Conlusion

In this paper, the AR technology has been used to get 3D coordinates of human arms' joints with high precision. Although a relatively satisfactory result has been achieved eventually, yet there is still much room for improvement in future work. For example, how to control each finger on the mechanical arm to imitate human fingers' poses and movement. This is also what our next research focuses on.



(a1)

(a2)

(a3)

(a4)

(a1), (a2), (a3), (a4) are poses of the mechanical arm imitating human arms.

Fig. 5. Imitative results.

Supported by Natural Science Foundation of China (Grant No. 6207024186)

References

1. Lang, B. (2019, June 03). Apple Announces ARKit 3 with Body Tracking & Human Occlusion. Retrieved December 15, 2020, from https://www.roadtovr.com/ios-arkit-3-body-tracking-motion-capture-human-occlusion/.
2. Arduino Nano. (2020, December 02). Retrieved December 15, 2020, from https://en.wiki-pedia.org/wiki/Arduino_Nano.
3. Aqeel, Adnan, Introduction to Arduino Nano, M. The Engineering Projects, (2018).
4. Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang and A. L. Yuille, CRAVES: Controlling Robotic Arm With a Vision-Based Economic System, C. Long Beach, (2019) 4209-4218.
5. Della Santina C, Arapi V, Averta G, Learning From Humans How to Grasp: A Data-Driven Architecture for Autonomous Grasping With Anthropomorphic Soft Hands, J. IEEE Robotics and Automation Letters, (2019) 1533-1540.
6. C. Erdogan, A. Schröder and O. Brock, Coordination of Intrinsic and Extrinsic Degrees of Freedom in Soft Robotic Grasping, C. IEEE International Conference on Robotics and Automation (ICRA), (2018) 4251-4256.
7. Omran Mohamed, Lassner Christoph, Pons-Moll Gerard, Neural Body Fitting: Unifying Deep Learning and Model-Based Human Pose and Shape Estimation, C. International Conference on 3D Vision (3DV), (2018) 484-494.
8. W. Zhang and X. Qi, Design and Implementation of Comic and Animation APP Based on AR Technology, C. IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), (2017) 244-251.