Research on Robot-Based Gesture Interactive Decoration Design

Liu Yang, Department of Art and Design, Zibo Vocational Institute, China*

ABSTRACT

Recently, automata has become a research hot spot in various countries for a wide range of applications in areas such as security, entertainment, and industry. The authors designed a mobile robot that is based on gesture control with the focus on gesture interaction control that includes gesture definition, hand gesture acquisition, hand gesture decoration, hand gesture pre-processing, and hand pointer recognition. This paper uses a combination of qualitative analysis and quantitative research to study the mapping characteristics of gesture interactive decoration and its internal connection with user interaction behaviors by using a combination of theoretical analysis and situational experiments, which not only helps to understand the human-computer interaction decoration complexity but has important positive significance in theoretical discussion, research methods, and experimental methods. The experiments show that the average recognition rate of the five gestures obtained in this paper reaches 92.5%. The robot can accurately recognize gestures and make corresponding actions.

KEYWORDS

Contour Extraction, Gesture Interactive Decoration Technology, Gesture Recognition, Interactive Mapping, Mobile Robot

1. INTRODUCTION

Robots and humans can interact in a variety of ways, including vision, hearing, touch, and smell. Among them, humans obtain about 80% to 90% of information through vision. Therefore, although human-computer interactive decoration technology has now undergone five development stages of keyboard, mouse, touch, multimedia and virtual reality, there are also many interactive methods suitable for robotics, such as face recognition, expression recognition, voice recognition, Human motion recognition, gesture recognition, and gaze tracking, etc., but vision-based human-computer interaction is still the mainstream interaction method (Camille, 2020). Among them, the three human-machine interaction methods of human faces, human movements, and gestures have been initially applied in the virtual environment, and have a very broad application prospect. For this reason, the gesture interactive decoration design of robots has become increasingly important.

Most of the earliest gesture interactive decoration techniques used a special electromechanical or magnetic induction device to directly measure the posture of the hand and the angle of the joints. Aiguo developed a glove-type sensor system (SayreGlove), which is the predecessor of data gloves (Song, 2015). Ding has obtained the patent of “data glove”. Through “data glove”, the computer can obtain rich information such as the position of the hand and the extension of the finger (Qi, 2017). Ross Mead has also conducted research on gesture recognition, and can recognize 46 gesture symbols...
through similar devices (Ross, 2016). Judith Tiferes used a CyberGlove model data glove with 18 sensors in a domestic sign language recognition system, and for the first time in the world, realized a continuous Chinese sign language recognition system capable of recognizing more than 5000 words (Judith, 2018). Although this gesture interaction method can accurately and real-timely display 3D models of human hands, it is necessary to wear heavy data gloves when performing gesture interaction, which is not convenient for natural movement, and the equipment cost is high, and it is not easy to popularize. Therefore, the way of acquiring low-precision human movements through simple external devices is gradually emerging in the field of gesture interaction. Most of the earliest gesture interaction technologies used a dedicated electromechanical or magnetic induction device to directly measure the posture of the hand and the angle of the joints (Yiannis, 2016). Such as the glove sensor system (SayreGlove) developed by ToraDefanti and Daniel Sandin in 1977, which is the predecessor of data gloves. In 1983, Grimes first patented the “data glove”. Through the “data glove”, the computer can obtain rich information such as the position of the hand and the extension of the finger. In 1991, Fujitsu also carried out research work on gesture recognition, which can recognize 46 gesture symbols through similar devices. In 2000, Professor Gao Wen (now a professor at Peking University) of the Institute of Computing Technology of the Chinese Academy of Sciences and Professor Wu Jiangqin (now a professor at Zhejiang University) of Harbin Institute of Technology used CyberGlove model data gloves with 18 sensors in the Chinese Sign Language Recognition System. For the first time in the world, a continuous Chinese sign language recognition system capable of identifying more than 5,000 words has been implemented. Although this gesture interaction method can accurately and real-timely display 3D models of human hands when performing gesture interaction, which is not convenient for natural movement, and the equipment cost is high, and it is not easy to popularize. Therefore, the way of acquiring low-precision human movements through simple external devices is gradually emerging in the field of gesture interaction.

At present, the commonly used gesture recognition methods mainly include neural network-based recognition methods, hidden Markov model-based recognition methods, and geometric feature-based recognition methods. The neural network-based gesture recognition method has the advantages of anti-interference, self-organization, self-learning, and anti-noise capabilities. However, the number of samples to be collected during training is large, and the time series processing ability is not strong (Robin, 2019 and Junya, 2017). While gesture recognition approach based on Hidden Markov Model has the capability to describe gesture signals in detail, its topology is general and computationally intensive. An identification method with geometric features is based on hand region and the relationship of edge geometric features for gesture evaluation. This approach does not require temporary segmentation of the gestures, has small computational effort, and is applicable to gesture recognition in time-to-market applications.

This paper introduces the implementation method of the main controller STM32 to acquire the data of MPU9250 sensor and ultrasonic sensor, the transmission of status information and the motion control in the software design of the host computer. Then it briefly introduces the reading of status information and the control commands in the design of the host computer software send. This paper proposes a saliency calculation based on multi-scale global region contrast, then introduces a proposed saliency calculation based on multi-feature and multi-scale global region contrast, and finally introduces a method of incorporating high-level prior knowledge. Experiments show that the proposed method can effectively reduce. The adverse effects of lighting changes, uneven lighting and the presence of shadows in home environments on gesture detection. The experiments show that the average recognition rate of gesture interaction based on 5 robots in this paper reaches 92.5%, and corresponding work can be done on gestures.
2. PROPOSED METHOD

2.1 Acquisition and Recognition of Robot Gesture Decoration

Gesture recognition method based on skin color, the recognition effect is better when the background color and hand color are significantly different, but if there is a large area in the background with skin area or skin-like area other than the hand (such as Case), it will seriously affect the effect of gesture recognition. In this paper, gesture recognition is performed using a depth image obtained based on Kinect 2.0 (Lu, 2016). This method can solve the above problems well. When Kinect 2.0 acquires a depth image, because the distance between the target and the sensor is different, the depth value is different, so the situation where the hand is in front of the face can be identified according to different depth value ranges (Jean, 2017). The gesture acquisition and recognition process based on the depth image needs to first define the gesture; then acquire the gesture image based on the depth image and the bone data node; finally, recognize the gesture, as shown in Figure 1:

Figure 1. Acquisition and recognition of gesture decoration

(1) Gesture definition

Before performing gesture recognition, it is necessary to define the correspondence between gestures and motion modes to ensure the interactive control between the upper computer and the lower computer. The mobile robot system designed and developed in this paper recognizes a total of
six gestures, and the corresponding relationship between specific gesture instructions and movement modes is shown in Table 1:

Table 1. Correspondence between gesture instructions and movement modes

| Serial number | Way of exercise     | Instruction array name                      |
|---------------|---------------------|---------------------------------------------|
| 0             | Rotate in place     | MotionSpinaroundHandGesture                |
| 1             | Go ahead            | MotionGoForwardHandGesture                 |
| 2             | Back                | MotionGoBackHandGesture                    |
| 3             | Turn left           | MotionTurnLeftHandGesture                  |
| 4             | Turn right          | MotionTurnRightHandGesture                 |
| 5             | Stop                | MotionStopHandGesture                      |

(2) Gesture acquisition

Gesture image acquisition is divided into two steps: first acquire the depth image, then acquire the human bone data stream and acquire the gesture image according to the bone nodes.

1) Get depth image

The depth image was acquired using Kinect 2.0, and the acquired image size was 512 * 424. The pixel value of the image represents the distance. The smaller the depth value is, the closer the distance sensor is; the larger the depth value is, the farther the distance sensor is (Seokwon, 2013).

2) Gesture image extraction

After identifying the commander and obtaining the commander’s bone information, quickly lock the commander’s right-hand nodes (right wrist node, right fingertip node, right-hand node, right-hand thumb node, etc.) and calculate the commander based on the node position Relationship to the position of the robot, and use dynamic boxes to circle the position of the hand in the image.

(3) Gesture preprocessing

After obtaining the gesture image, gesture preprocessing is required. The gesture preprocessing part of this paper mainly includes: double threshold processing, image rotation correction and scale normalization. As shown in Figure 2 below.

1) Double threshold processing

In this paper, the hand is separated from the background through a double threshold (a dynamic threshold $T_1$ and a fixed threshold $T_0$) to obtain gestures. The maximum value of the depth value at the three nodes of the hand, thumb and fingertip in the bone data stream is used as the dynamic threshold $T_1$ of the hand segmentation. Threshold processing using the OpenCV library function threshold (Wu, 2016 and Xu, 2017).
2) Image rotation

Due to the variability of the angle of the master’s gesture recognition, in order to ensure the accuracy of gesture recognition, this article first rotates the image after the background difference before the gesture recognition, so that the fingertips of the hand image to be recognized are always on, palm and wrist Down. The following figure illustrates the principle of image angle rotation as an example. In a computer, the image origin is the upper left corner, and in mathematics, the image origin is the image center. Therefore, image coordinate conversion is required before image rotation (Yang, 2016). Assume that in the mathematical coordinate system, the rotation angle is $\theta$, and the point after the point $(X_1, Y_1)$ transformation is rotated by $\theta$ through the rotation matrix is $(X_2, Y_2)$, as shown in Figure 3 below.

Knowing that the height of the picture is $H$ and the width is $W$, the conversion relationship between the image icon $(X_0, Y_0)$ and mathematical coordinates $(X_1, Y_1)$ is written as a matrix expression as shown in formula (1):

$$
\begin{bmatrix}
X_1 & Y_1 & 1 \\
\end{bmatrix} = 
\begin{bmatrix}
1 & 0 & 0 \\
0 & -1 & 0 \\
-0.5W & -0.5H & 1 \\
\end{bmatrix}
\begin{bmatrix}
X_0 & Y_0 & 1 \\
\end{bmatrix} 
$$

(1)

After the image rotation is performed through a mathematical formula, the image coordinates need to be converted into the image coordinates of the new bitmap. It is assumed that the coordinates
of the image are the height \( H \) and the width \( W \) after the rotation. The inverse matrix conversion formula is (Lei, 2018 and Kang, 2015):

\[
\begin{bmatrix}
X_0 & Y_0 & 1
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
0 & -1 & 0 \\
0.5W' & 0.5H' & 1
\end{bmatrix}
\begin{bmatrix}
X_1 & Y_1 & 1
\end{bmatrix}
\tag{2}
\]

As shown in Figure 3, suppose that a point \((X_1, Y_1)\) on the mathematical coordinate system is obtained by rotating \((X_2, Y_2)\), the rotation angle is, the line between the point \((X_1, Y_1)\) and the origin and the horizontal angle is Then there are:

\[
X_1 = R \cos a
\tag{3}
\]

\[
Y_1 = R \sin a
\tag{4}
\]

\[
X_2 = R \cos(\alpha - \theta)
\tag{5}
\]

\[
Y_2 = R \sin(\alpha - \theta)
\tag{6}
\]
The sine and cosine theorems are shown in formulas (7) and (8):

\[
\sin(a - \theta) = \sin a \cos \theta - \cos a \sin \theta \quad (7)
\]

\[
\cos(a - \theta) = \cos a \cos \theta + \sin a \sin \theta \quad (8)
\]

From the formulas (3) to (4) and the sine theorem (5) and cosine theorem (6), we know:

\[X_2 = X_1 \cos \theta + Y_1 \sin \theta\]  
(9)

\[Y_2 = Y_1 \cos \theta + X_1 \sin \theta\]  
(10)

Write the formulas (9) and (10) as matrix expressions:

\[
\begin{bmatrix}
X_2 \\
Y_2 \\
1
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & -\sin \theta & 0 \\
\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X_1 \\
Y_1 \\
1
\end{bmatrix} \quad (11)
\]

The inverse operation is:

\[
\begin{bmatrix}
X_1 \\
Y_1 \\
1
\end{bmatrix} =
\begin{bmatrix}
\cos \theta & \sin \theta & 0 \\
-\sin \theta & \cos \theta & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
X_2 \\
Y_2 \\
1
\end{bmatrix} \quad (12)
\]

Certificate completed.

The angle is calculated according to the Anglecal function, and the image is rotated using the ImgRotate (SrcImg, angletemp, dstImg) function according to the principle of angle rotation, where SrcImg is the original image, dstImg is the target image, and angletemp is the rotation angle. The right finger tip node and right wrist node are calculated (Sophie, 2018 and Jin, 2018).

3) Scale normalization

Normalization of scale to guarantee the complete acquisition of hand information, we use dynamic box acquisition of hand images in this paper to obtain different sizes of the acquired images. A resize function in the OpenCV library is used in this paper to perform scale normalization using either region interpolation to ensure one hand image size of 100*100 (Zhao, 2015).

(4) Gesture recognition
The gesture recognition process is based on the shape characteristics of the hand to recognize based on the number of fingertips, which mainly includes extracting hand contours and center of gravity extraction and gesture recognition based on the number of recognized fingertips. It also gives the presence of multiple people Gesture Priority Recognition Method (Liang, 2015).

We use the findContours function for performing contour search in the OpenCV library to perform this paper. Because the hand image obtained was obtained based on the hand skeleton skeleton node information with the image size being exactly the entire amount of the hand, therefore the hand gesture contours are usually the largest for all silhouettes. Therefore, in this paper, the maximum area contour is labeled, and the hand contour is drawn in the hand image after threshold segmentation using the drawContours function, and the center of gravity of the hand is marked with the red origin on the gesture image with the hand contour drawn (Julio, 2019). The method of obtaining the center of gravity is: for the binarized picture, the OpenCV library function moments is used to calculate the center of gravity. The specific implementation method is as follows: where skinArea is the binarized picture pixel matrix.

### 2.2 Gesture Interaction Decoration Design

It is used to describe the characteristics of user behavior, interface performance, interaction tasks and approach to interaction. In general, there are three types of interaction patterns: designer patterns, system patterns, and consumer patterns.

1. **Equipment**

   The device is the most important part of IN-RT system, and also the most used part of users. It operates and controls the whole INRT system through the interface provided by the device. According to the function, it can be divided into sensor module, image processing module, communication module and motion control module.

2. **Sensor module**

   The sensor module is used to control the external input equipment, including video / audio acquisition and playback, infrared / ultrasonic ranging, GPS positioning system. The sensor module interface consists of 10 interfaces, i.e. iasrimagecapture, iasrimagecapturetofile, iasrimagecapturetofile, iasrimagecapturetofile, iasrimagecapturetofile, iasrimagecapturetofile, iasrimagecapturetofile, iasrimagecapturetofile, iasrimagecapturetofile, iasrimagecapturetofile. This system mainly uses two interfaces, i.e. iasrimagecapture and iasrimagecapturetofile, to collect and store video (Barka, 2015).

3. **Image processing module**

   All this module used to handle image files, with image loading/saving, image color space translation, image encoding, other related graphic algorithms, soccer recognition and positioning. Which mainly includes iRimage, iRimageProcessColor, iRimageProcessGray and iArfootball interfaces. Input the gesture detection algorithm, gesture feature evaluation and histogram recognition algorithm proposed in this paper into this module for image application (John, 2015).

4. **Communication module**

   The communication module is used to operate network data, including transmission of network data, IP multicast of data, broadcast of network data, network monitoring, and operation monitoring of network cards.
(5) Motion control module

The motion control module is used to control the motion of the robot. The IASRMotion interface is provided in the IN-RT development system to control the motion. The IASRMotion interface includes two modes of open-loop motion and DSP closed-loop. The interface has corresponding functions for setting / obtaining speed, acceleration, displacement, etc., which can easily perform simple motion control on the system. If you want to perform complex motion control, you need to implement specific functions for motion control yourself.

2.3 System Functional Requirements and Electromechanical System Design

(1) System function requirements

In order to better study the gesture interaction control method, this paper designs and develops a mobile robot based on gesture interaction, which aims to help users (including the elderly and most disabled people) improve the quality of life (Chipofya, 2015). Therefore, the design of the entire robot system is customized around the needs of the user. The specific functional requirements of a mobile robot based on gesture control are as follows:

1) It needs to be applied in indoor environment;
2) Can move in all directions;
3) Highly needed to meet daily use needs;
4) During exercise, you can avoid obstacles autonomously, the detection range of obstacles is 3-300cm
5) Ability to detect its own motion status, including acceleration, angle and orientation information
6) Able to interact with people through gestures and recognize 6 gestures;
7) Secondary development and function expansion are possible.

(2) Structural design of the robot system

In accordance to the functional requirements of users, we have designed the overall structure of the robot in this paper. There are five parts in this paper: the complete robot: the chassis, the stand, the lower device system, for the upper device system and the power circuit. That the chassis is responsible for supporting the whole system; the scaffolding is responsible for supporting the camera part of the robot and insuring the whole altitude; the lower computer system includes the lower computer main controller, sensors, motors and motor drives, of which the sensor part includes two parts, the MPU9250 sensor module and the ultrasonic sensor module. The function is Avoid obstacles autonomously, detect its own motion status, receive commands from the host computer and complete corresponding actions according to the commands; the host computer system includes a camera module and an industrial computer module, which are mainly used to read the status of the host computer, complete the collection of video signals and gesture recognition and It processes and sends the recognition and processing results to form commands to the lower computer; the power circuit is responsible for supplying power to the upper computer system and the lower computer system at the same time. The upper computer system and the lower computer system communicate through the RS232 serial port (Wang, 2015 and Dadochkin, 2016).

The robot chassis is the hardware building platform for the entire robot system. According to different mobile structures, robots can be divided into leg-type crawler and wheel-type structures. The choice of structure depends on different application requirements and application scenarios. The robot designed and developed in this article is mainly used for home use. The application scenario is indoor. When choosing the chassis, it should try to meet the advantages of simple control, good stability, low noise and cheap price (Wang, 2015).
(3) Robot system hardware design

1) Camera

The camera is equivalent to the “eye” of the entire robot, and can acquire image information in the environment, including gray information, distance information, and color information of the target object or human body. When designing a robot in this article, we must consider the portability, scalability, and ease of maintenance of the robot, so we need to use as simple a structure as possible to obtain as much data as possible to ensure the practicability, stability, and redevelopment of the robot (Zhu, 2015). Based on the above factors, this design chose the second-generation Kinect for Windows (referred to as Kinect 2.0 for short) in the field of human-computer interaction for robot design and development.

2) IPC

Because Kinect 2.0 is selected as the camera module of the robot system in this article, the configuration requirements of Kinect 2.0 must be considered when selecting the industrial computer module. Kinect 2.0 development requires a 64-bit (x64), physical dual-core, 3.1GHz clock speed or faster processor. It also needs a dedicated USB3.0 bus interface, 2GB of RAM, and a graphics card that supports DirectX11. Windows 8 and above operating systems (Gianni, 2015).

Based on the above factors, this article chose an industrial computer module with i7 dual-core, 2GB RAM, four USB3.0 bus interfaces, and support DirectX11 graphics card. At the same time, the industrial computer module was installed with Windows 8.1 operating system, Visual Studio 2012, OpenCV3.0 library and Kinect2.0 SDK development kit are used for subsequent programming design.

3) Main controller

The main controller module is based on the ARM Cortex-M3 series highest configuration chip STM32F103ZET6 launched by ST company. It is packaged in a 144-pin LQFP package, including 512KB FLASH, 64KB SRAM, 3 12-bit ADCs, 4 general-purpose timers 2 PWM timers, etc. The STM32 microcontroller is designed and developed by C language under the Keil4 software environment, and the code is downloaded using J-LINK.

4) Sensor module

The working environment of the robot system designed and developed in this article is indoor, so the GPS module cannot be used to collect the bearing signal. In order to obtain the information of the robot’s movement direction and attitude, this article uses the MPU9250 sensor with 9-axis motion tracking function. The MPU9250 is a QFN-packaged composite chip consisting of a three-axis accelerometer, a three-axis gyroscope, and a three-axis electronic compass. The three-axis magnetic field meter consists of a three-axis monolithic Hall sensor, which has the characteristics of large range, low power consumption, and high accuracy. The maximum measurement range is ± 4800 μT (Jean, 2017). Accelerometers and gyroscopes have the same direction and polarity, while electronic compasses have different directions and polarity.

5) Power supply design
The robot system has many components and different subsystems have different voltage standards. Therefore, a dedicated power circuit needs to be designed to ensure the power supply and operation of the entire system. The entire robot system requires three voltage specifications of 12V, 5V, and 3.3V to maintain the normal operation of the system (Wang, 2019 and Lu, 2016). In order to avoid excessive external power, this article will use a 12V lithium battery as the only power supply directly to each module of the robot system.

(2) Software design of robot system

The system software design of the robot mainly includes two parts: software design of the lower computer and software design of the upper computer.

1) Sensor data acquisition

The MPU9250 sensor includes three parts: acceleration, gyroscope and electronic compass. When reading the sensor data, the corresponding three-axis acceleration, three-axis angular velocity, and three-axis magnetic field data are read by sending the addresses corresponding to the acceleration, gyroscope, and electronic compass. The sensor address in MPU9250 is composed of 7 address bits (A7-A1) and 1 read / write bit (W / R). In the read and write bits, 0 represents write and 1 represents read, as shown in Figure 4 below.

![Figure 4. Sensor address](image)

2) Motion control implementation

The system uses 3 motors to control the robot movement. A single L298N driver chip can control up to two motors at the same time, so two L298N driver chips are needed to control the robot system in this paper. The output pins of L298N are composed of OUT1, OUT2, OUT3, and OUT4, which are respectively connected to two motors. The input pins are composed of In1, In2, In3, In4, PWMA, and PWMB. In1 and In2 are used to control the Running state, In3 and In4 are used to control the running state of the second motor, and PWMA and PWMB are used to control the rotation speed of the first and second motors, respectively. The logic functions of In1 and In2 in L298N (the logic functions of In3 and In4 are the same) are shown in Table 2:

The speed of the motor is adjusted by PWM. The principle is: assuming that the on-time of the switch in a cycle is t and the cycle is T, the relationship between the average voltage U across the motor and the power supply voltage Vcc is shown in formula (13).

\[ U = V_{cc} \times \frac{t}{T} = V_{cc} \times a \]  

(13)
Among them, \( a = \frac{\omega}{T} \) is called the duty cycle. Because the speed of the motor is proportional to the voltage at both ends, and the voltage at both ends of the motor is proportional to the duty cycle of the control waveform, the speed of the motor is proportional to the duty cycle of the control waveform.

### 3. EXPERIMENTS

#### 3.1 Test Subject

1. **Robot physical map**

   After completing the separate tests of the upper and lower computers of the system, the overall function test of the entire robot system is performed. The physical appearance of the robot designed in this article is shown in Figure 5.

   As shown in Figure 5 above, wheeled mobile robots generally have two types: universal wheels and omnidirectional wheels. Universal wheels are generally used with multiple fixed casters. Fixed casters are responsible for providing power, and universal wheels are responsible for providing direction. The restriction cannot be turned in place. Compared with the universal wheel, the omnidirectional wheel has multiple side pulleys. When moving in the direction of the wheel axis, rolling friction can replace sliding friction, reduce the friction between the wheel and the ground, and there is no dead zone limit. In robotic systems.

2. **Experimental hardware composition**

   The hardware components of this interactive system include: image acquisition equipment (Kinect equipment), interface between computer and Kinect equipment, computer, display and robot. As shown in Figure 6 below.

3. **Composition of experimental software**

   The software flowchart of the robot is relatively simple. It mainly receives the recognition result of the recognition module through the wireless receiving module and converts the recognition result into corresponding control commands and signals. Finally, the robot’s professional chip control system is used to guide the robot’s movement, as shown in Figure 7.

#### 3.2 Experimental Methods

Each test subject will be given several warm-up tasks to familiarize them with the entire test process. During the exercise, the tester will explain to the test subject the entire experimental process and what the test subject needs to complete. Test subjects then practice until they are fully familiar with the position of the entire test process. Testers will always be present. The experiment used a repeat

| In1 | In2 | Running state |
|-----|-----|---------------|
| 1   | 0   | Forward       |
| 0   | 1   | Reverse       |
| 1   | 1   | Brake         |
| 0   | 0   | Stop          |

Table 2. L298N logic function table
Figure 5. Robot physical map

Figure 6. Structure of system
test within the target. Each test subject needs to complete 128 small tasks (32 circles x 2 repeats x 2 positions).

When the mission begins, the large circle in the center will be lit, and the test subject will raise his right hand in a comfortable way and inform the tester that he is ready. Testers then used the OptiTrack motion capture system to record the user’s current posture. After recording, one of the small circles will be randomly lit to guide the user to move his hand to get the target. The illuminated small circles are used to clarify the direction and relative distance required for hand movement, and do not provide the test subject with any hints about the actual distance their hands moved. The experiment interface does not provide any visual feedback. When the test subject thinks that his hand is in a suitable position to obtain the target, he stops, then maintains the posture, and notifies the tester to record the posture. When the posture of the test subject is recorded, the large circle in the center will be lit, and the small circle that was originally lit will return to its original state, and then the above process is repeated until each small circle is lit twice. There are 4x8x2 = 64 small tasks in a large task. All testers need to test in two locations. The order of the test locations is random between the test subjects.

3.3 Experimental Collection

The proposed method is validated with the construction of a dynamic gesture dataset in this experiment employing Microsoft’s Kinect device, which is comprised of a depth camera and a VGA camera. With the depth camera, it is used to capture the depth information of the object. With its capture rate of 30 frames/second and resolution of 320x240, the depth measurement range is 1.2-3.5m, and to a depth resolution of about 5 mm; the VGA camera is used to capture RGB images, and its acquisition rate is also 30 frames / s, and the image resolution is 640 × 480. The data collection was performed under the three lighting conditions of morning, afternoon, and night. 10 gestures were collected, and a total of 10 people made gestures. Each gesture was performed 5 times for a total of 1500 gesture videos.
4. DISCUSSION

4.1 Experimental Analysis of Functional Test of Robot Gesture Decoration

The test process is as follows: find 5 people (A, B, C, D, E) respectively standing in the range of 80cm-150cm from the front of the robot and input 5 gesture commands (0, 1, 2, 3, 4) to the robot, and the same person makes the same gesture 10 times, and observes whether the robot's actions are consistent with the corresponding instructions, and counts the number of correct and incorrect instructions.

Table 3. Robot gesture command recognition test results

| Gesture | A  | B  | C  | D  | E  | Recognition rate |
|---------|----|----|----|----|----|------------------|
| 0       | 9  | 9  | 10 | 10 | 9  | 94%              |
| 1       | 9  | 9  | 8  | 10 | 9  | 90%              |
| 2       | 10 | 10 | 9  | 10 | 9  | 96%              |
| 3       | 9  | 9  | 9  | 9  | 9  | 90%              |
| 4       | 10 | 9  | 10 | 10 | 10 | 98%              |

Figure 8. Relation between gesture recognition rate and the number of lexicons based on each feature
As shown in Table 3 above, the average recognition rate of 5 gestures of 5 people is 92.5%. After testing, the robot can accurately recognize gestures and make corresponding actions.

This experiment uses a dictionary learning method to build a sparsely represented gesture dictionary from the training data set. Here, 200 pictures of each gesture are used to learn the dictionary, and 200 pictures are randomly selected for testing. Each gesture is learned separately during the dictionary learning A dictionary, so this article learns 5 dictionaries, then connects a total of 5 dictionaries learned by each gesture to a dictionary containing all categories to sparsely represent the test set, and then combines its corresponding markers for gesture recognition. The changes of the gesture recognition rate and the number of dictionary elements K are shown in Figure 8 below.

As shown in Figure 8 above, through experiments, it can be known that when the number of dictionary elements is less than 32, the recognition rate increases with the increase of the number of dictionary elements, and the increase is obvious, while the number of dictionary elements between 32 and 128 decreases the recognition rate. After it is larger than 128, it increases with the number of dictionary elements, but the overall trend is that it increases with the number of dictionary elements. Therefore, the number of dictionary elements used in subsequent gesture recognition in this paper is 200.

4.2 Comparative Experimental Analysis of Different Classification Algorithms

In order to verify the effectiveness of gesture recognition performance of sparse representation (Sparse Representation) compared to other classifiers, this paper uses extracted salient features to classify sparse representation classifiers without support of dictionary learning, support vector machine (SVM), artificial neural network (Artificial Neural Network), Bayesian Network and Decision Tree classifier for comparison. The artificial neural network uses two hidden layers. The number of nodes in the hidden layer is 40 and 20, respectively. In the Bayesian-based method, naive Bayes is used for recognition. In the decision tree-based approach, the C4.5 algorithm is used for classification. As shown in Table 4.

| Different classifiers | Gesture 1 | Gesture 2 | Gesture 3 | Gesture 4 | Average recognition rate |
|-----------------------|-----------|-----------|-----------|-----------|--------------------------|
| SR                    | 98%       | 96%       | 99%       | 93.5%     | 96.5%                    |
| SVM                   | 91%       | 86.5%     | 93.5%     | 89%       | 89.9%                    |
| ANN                   | 73.5%     | 75.5%     | 85%       | 87%       | 81.6%                    |
| BN                    | 83.5%     | 83%       | 92.5%     | 80.5%     | 85.4%                    |

The specific recognition results of the methods shown in Table 4 above, the results in the table show that the average recognition rate based on sparse representation is the highest, and the recognition rates between various gestures are not much different. The average recognition rate based on the support vector machine is the second, and the average recognition rate based on the decision tree method is the third, and the average recognition rate based on the artificial neural network is the lowest.

4.3 Impact of Different Kernel Functions and Different Features on Recognition

In order to effectively verify the effectiveness of the proposed kernel function, experiments are performed on different kernel functions respectively. Polynomial kernel (PK, kernel 1), inverse distance kernel (IDK, kernel 2), and inverse square distance kernel (ISDK), Kernel 3), Gaussian kernel (GK, kernel 4), in order to better compare the performance of different features on a specific
kernel function, this paper experiments on the performance of various features on 4 kinds of kernel functions, specifically. The results are shown in Figure 9 below.

Figure 9. Comparison of the recognition rate of various features in the inverse square distance kernel

As shown in Figure 9 above, from the recognition rate curves of various features on different kernel functions, it can be seen that the proposed salient feature has the highest recognition rate, which is superior to other commonly used gesture features, and the accelerated robust feature has the lowest recognition rate. Therefore, only the average recognition rate of salient features between different kernel functions is given here. The influence of outliers is large, and the performance of histogram cross-core and exponential histogram cross-core is not significant and is relatively stable. Therefore, the sparse representation based on histogram cross-kernel is better than the ordinary sparse representation, which indicates the direction for further research on kernel sparse representation gesture recognition.

4.3 Analysis of Hand Movement Amplitude

The shoulder view in the upward direction (Northeast, North, and Northwest) is significantly smaller than the shoulder view in the downward direction (Southwest, South, and Southeast). The interactive effect of the target distance * direction on the head angle shows that when the target distance increases, the head angle increases more in the east and southeast directions, and decreases less in the northeast and northwest directions. The interactive effect of the target distance * direction on the shoulder angle shows that when the target distance increases, the shoulder angle increases faster in the east
and southeast directions, and increases more slowly in the west and south directions. As shown in Figure 10 below.

Figure 10. Target position in test subject perspective

As shown in Figure 10 above. In order to analyze the relationship between the hand movement amplitude and the context of the interactive interface, for targets on the same straight line, here we calculate the linear relationship between the hand translation amplitude L, the head angle A and the shoulder angle B, and the target position. The directions East, Northeast, North, and Northwest are the positive directions of the corresponding data set. Analysis of linear regression results for each data set is shown. Among them, the blue line is the regression line of the translational amplitude l, the red line is the regression line of the head angle α, and the green line is the regression line of the shoulder angle. For the individual directions, in the experiments of D = 2.0m and D = 3.0m, there is a high linear correlation between the hand translation distance L, the head angle A and the shoulder angle B, and the position of the target in the angle of view of the test subject. Its linear correlation coefficient 2 R is higher than 0.95. However, the correlation between the target position and the translation range is higher than the correlation with the head and shoulder views.

4.4 Hand Movement in the Direction of Movement and Depth

For each target in different directions, when acquired, the deviation of the hand direction of motion of the test subject is shown with the value of the hand motion Dz when acquired for each one in depth direction in each target in the figure in the following. As shown in Figure 11 below.
As shown in Figure 11 above. When acquiring targets in the three directions of northwest, west, and southwest, the deviation of the hand movement direction is significantly higher than those in other directions. The target distance has no significant effect on the deviation of the moving direction (F (3, 1916) = 0.143, p = 0.934). For targets in different directions, the movement of the hand in the depth direction is significantly different (F (7, 1912) = 14.819, p = 0.000). In the northwest, west, southwest, and south directions, the absolute value of Dz is significantly larger than the values in other directions. When acquiring different targets, the movement value Dz of the hand in the depth direction is significantly different (F (3, 1916) = 29.181, p = 0.000), and the absolute value of Dz when acquiring target 4 is significantly greater than that of acquiring target 1, 2 and absolute value at target 3.

5. CONCLUSIONS

This paper designs and implements a mobile robot based on gesture control. It focuses on gesture recognition based on depth images and proposes a method based on double thresholds to extract gestures from the background of hand depth images. The robot can control the direction of travel through gestures, has the characteristics of self-movement detection, automatic obstacle avoidance, and can be developed twice. The process of gesture recognition is: first, quickly lock and intercept the hand depth image according to the position of the relevant nodes in the bone data stream; then, use the double threshold processing method to extract the binary image of the hand from the hand depth image; finally, according to the opponent The result of the binary image is used to extract
contour and center of gravity and identify the number of fingertips, and then recognize gestures to get corresponding instructions. This paper proposes a static gesture detection method that combines multi-feature and multi-scale global region contrast.

This paper presents a new multi-scale global area color contrast saliency calculation method for detecting regions of interest. Experimental results show that the proposed multi-scale global area contrast saliency calculation method is better than the other 11 significant calculation methods. Secondly, based on the above-mentioned significant calculation method, a multi-feature fusion-based static gesture detection algorithm based on multi-scale global region contrast is proposed. This algorithm incorporates information such as color and texture for multi-scale global region contrast calculation, and combines skin color and High-level prior knowledge such as objectivity is used for detection. By using accuracy, recall, and F-measure histograms to evaluate the performance of gesture detection, experiments show that this method performs better than gesture detection using other saliency methods.

We have designed a user-centered mapping technique in this paper and for the above-mentioned research finding, we have validated it. With these experimental studies, the intrinsic correlation between user priority, performance, etc. and gesture interaction mapping features were effectively discovered and tapped, which is important for understanding the behavioral characteristics of users in gesture interaction on the one hand and the intrinsic essences of HCI functions on the other, while the theoretical discussion and experimental techniques of related research are of great value for boosting the research conducted on HCI in China.
REFERENCES

Aiguo, S. (2015). Research on human-robot interaction telerobot with force telepresence. Science & Technology Review, 33(23), 100–109.

Borghesan, G., Scioni, E., Kheddar, A., & Bruyninckx, H. (2015). Introducing Geometric Constraint Expressions into Robot Constrained Motion Specification and Control. IEEE Robotics and Automation Letters, 1(2), 1140–1147. doi:10.1109/LRA.2015.2506119

Camille, G., & Henri, B. (2020). The Nature and Role of User Beliefs Regarding a Website’s Design Quality. Journal of Organizational and End User Computing, 32(1).

Chipofya, M., Lee, D. J., & Chong, K. T. (2015). Mobile Robot Motion Control Using Laguerre-Based Model Predictive Control. Applied Mechanics and Materials, 776, 403–410. doi:10.4028/www.scientific.net/AMM.776.403

Dadokhin, M. K. (2016). Coordinating the motion of a robot and the workpiece. Russian Engineering Research, 36(7), 587–590. doi:10.3103/S1068798X16070054

Fok, C.-L., Johnson, G., & Yamokoski, J. D. (2015). ControlIt! A Software Framework for Whole-Body Operational Space Control. International Journal of HR; Humanoid Robotics, 13(01), 29–43.

Guadarrama-Olvera, J. R., Dean-Leon, E., Bergner, F., & Pressure-Driven, B. C. U. R. S. (2019). Pressure-Driven Body Compliance Using Robot Skin. IEEE Robotics and Automation Letters, 4(4), 4418–4423. doi:10.1109/LRA.2019.2928214

Jean, M. M. G., Čáp, M., & Frazzoli, E. (2017). Locally-optimal multi-robot navigation under delaying disturbances using homotopy constraints. Autonomous Robots, 42(12), 1–13.

Jean, M. M. G., Čáp, M., & Frazzoli, E. (2017). Locally-optimal multi-robot navigation under delaying disturbances using homotopy constraints. Autonomous Robots, 42(12), 1–13.

Jijun, K., Peng, W., & Yunying, F. (2015). Research on Interaction between Urbanization and Economy Growth from the View of Economy Transformation. Journal of Chongqing University, 35(4), 295–298.

Jin, Y., & Lee, M. (2018). Enhancing Binocular Depth Estimation Based on Proactive Perception and Action Cyclic Learning for an Autonomous Developmental Robot. IEEE Transactions on Systems, Man, and Cybernetics. Systems, 49(1), 169–180. doi:10.1109/TSMC.2017.2779474

Jin-zhou, L., Ling-bin, Z., & Nan, Y. (2018). Research on industrial robot alignment technique with monocular vision. Optics & Precision Engineering, 26(3), 733–741. doi:10.3788/OPE.20182603.0733

John, C. (2015). Dallon, Lynnae C. Despain, Emily J. Evans. A continuous-time model of centrally coordinated motion with random switching. Journal of Mathematical Biology, 74(3), 1–27.

Junya, Y., Made Widhi Surya, A., & Takeshi, H. (2017). Cooperative Control of Human-robotic Network under Inter-robot Communication Delay: A Passivity Approach. Transactions of the Society of Instrument and Control Engineers, 53(12), 663–670. doi:10.9746/sicetr.53.663

Kantaros, Y., & Zavlanos, M. M. (2016). Global Planning for Multi-Robot Communication Networks in Complex Environments. IEEE Transactions on Robotics, 32(5), 1–17. doi:10.1109/TRO.2016.2593045

Landres, S. (2018). The First Non-Human Action Artist: Charlotte Moorman and Nam June Paik in Robot Opera. PAJ a Journal of Performance and Art, 40(1), 11–25. doi:10.1162/PAJI_a_00390

Lu, X., & Fei, J. (2016). Velocity Tracking Control of Wheeled Mobile Robots by Iterative Learning Control. International Journal of Advanced Robotic Systems, 13(3), 1. doi:10.5772/63813

Lu, X., & Fei, J. (2016). Velocity Tracking Control of Wheeled Mobile Robots by Iterative Learning Control. International Journal of Advanced Robotic Systems, 13(3), 1. doi:10.5772/63813

Mead, R., & Maja, J. (2016). Matarić. Autonomous human–robot proxemics: Socially aware navigation based on interaction potential. Autonomous Robots, 41(5), 1–13.
Liu Yang was born in Zibo, Shandong Province, China, in 1988. She holds a master's degree from Tianjin Academy of Fine Arts in China. She now works at Zibo Vocational College in Shandong Province. Her research interests include product styling and product interaction design.