Research on Robot Control Technology Based on Kinect

Feipeng Jia *
Quanzhou University of Information Engineering, Fujian, China.
*Corresponding author e-mail: 33145440@qq.com

Abstract. With the development of robot control technology, the interaction between humans and robots has become more and more common. Using human movements and gestures to control the robot can replace complex and cumbersome program operations, easily and conveniently manipulate the robot, issue commands to the robot, and interact with the robot. Human vision and gesture recognition based on machine vision is an indispensable key technology for implementing a new generation of human-computer interaction systems. As a revolutionary product, Kinect somatosensory devices capture color images, depth images, and human bone images, and provide a new way of human-computer interaction. It captures and tracks human movements, gestures and sounds. In this paper, the Kinect somatosensory device is used for static gesture recognition and dynamic gesture recognition. Based on this, the robot action is used to realize the contactless interactive interaction of the robot through gesture control. This paper introduces a Kinect device-based robot control method, which is related to the field of robot control, and is applied to the two-wheeled self-balancing robot motion control and human-computer interaction system. This method utilizes Kinect for the motion control system of two-wheeled self-balancing robot. It is controlled with the DTW gesture recognition algorithm.

1. Introduction
Intelligent robots not only have entertainment functions, but also can do many jobs efficiently instead of people. With the development of robot technology, intelligent robots or corresponding intelligent products are increasingly entering homes, factories and enterprises, providing various entertainment, production and other services for human beings, making the interaction between humans and robots more and more universal.

Usually the control of intelligent robots needs to be done by computer, so the interaction between humans and robots is also transformed into the interaction between humans and computers. Human Computer Interaction (HCI) is a technical discipline that studies the communication and communication between people and computers through mutual understanding and, to the greatest extent, people's functions of information management, service and processing. After decades of development, the current human-computer interaction technology has adapted from the past interaction subject to the interaction object, and the current interaction object has continuously adapted to the new stage of the interaction subject. After the command line interface and graphical user interface, the current Natural User Interface (NU) reflects the user-centric concept of human-computer interaction. In terms of robot control, the previous interaction between humans and robots is mainly based on mouse, keyboard or touch screen. These methods require additional equipment for interaction, which is not suitable for
mobile robots and portable smart products. [1]. At the same time, these methods require some computer
input device operation methods, the interaction method is not direct enough, and it is not suitable for
interaction with the elderly and children who lack learning ability. Therefore, a natural human-computer
interaction method is needed instead of complicated and cumbersome. Program operation, in order to
easily and conveniently manipulate the robot, issue commands to the robot, and interact with the robot.
The human body movement is a natural, intuitive and easy-to-learn human-computer interaction means.
The human body directly acts as an input device for the computer, between the human and the machine.
Communication will no longer require redundant media, and users can simply define several appropriate
actions to control the surrounding machines. With the development of computer vision and image
processing technology, the way of machine vision-based gestures and computer interaction has become
the mainstream of new human-computer interaction [2] In November 2010, Microsoft's Kinect
somatosensory device has human tracking and pose estimation. Outstanding performance. As a model
of a new generation of natural human-computer interaction technology, Kinect enables users to interact
more naturally with computers through actions such as voice and gestures [3].

1.1. Introduction to Kinect
The Kineet for XBOX used in this article is the first somatosensory device in the Kinect series launched
by Microsoft. This device can get rid of the somatosensory handle used in the past, and the body
recognition and device control effect can be achieved through the human body, as shown in Figure 1 is
shown. Kinect includes an RGB camera head, a pair of 3D depth cameras, a set of microphone arrays
and a rotating motor. It has three functions on the Kinect body with instant dynamic capture, image
transmission, voice transmission and multi-person interaction. In the middle is the more common RGB
color camera, the left lens is the infrared emitter, and the right lens is the infrared CMOS camera. The
left and right lenses form a three-dimensional depth image sensor. The 3D depth image sensor is mainly
used to obtain the operator's motion, while the RGB color camera is used to identify the operator's
identity.[4] Located on the base of the Kinect is a motor that can only be rotated up and down. In the
lower part of Kinect, there is a -Y0 microphone system, which can be used for voice recognition. The
advantage of this system is that it can reduce noise by simultaneously collecting sound, thus reducing
the noise. In addition to Kinect for XBOX, Microsoft also launched a product called Kinect for Windows.
It differs from Kinect for XBOX in that it adds a near mode to achieve a stable depth map from 0.4 to
0.8 meters, while Kinect for XBOX only works in the default mode, ie Stable depth image information
is only available in the range of 0.8 to 4 meters

![Fig. 1 Diagram of Kinect](image)

1.2. Human interaction
The human-computer interaction in this paper has three corresponding concepts in English: one is
Human Machine Interaction, the other is Human Computer Interaction, and the third is Human Robot
Interaction. Interactive) 17J. The term “human-computer interaction” in this paper refers to the last
concept, that is, the interaction between humans and robots. It is the development of human-machine
interaction in the field of robotics, but also through the interaction of people and computers. The design
principles of human-computer interaction systems should meet the following requirements: user control,
ease of use, intuitiveness, simplicity, and visibility. As an important part of the field of computer science research, human-computer interaction has experienced more than half a century, and has made great progress and improvement. How to realize natural, convenient and ubiquitous human-computer interaction is the intersection of multidisciplinary research in information science, mathematics, intelligent science and psychology. It is one of the hot trends of computer research at present and in the future [5].

1.3. Kinect Human Skeleton Model

The optical part of the Kinect consists of two main components: an infrared emitter and an infrared CMOS camera, which form a 3D depth sensor. The 3D depth sensor emits a "laser" through the infrared emitter to cover the entire Kinect's visual range, and the CMOS camera receives the reflected light to obtain a "depth field" of the image. Shotton et al. use Kinect to mark various parts of the human body in the depth image, and use large and rich training data to ensure that the decision tree classifier does not differ in personal, clothing, posture, etc. when evaluating various parts of the body. Caused a classification error of $15^\circ$. After completing the division of each part of the body, Kinect will generate a human skeleton model containing three-dimensional information of the joints of the body based on the 20 joint points traced. The skeleton tracking system in the Kinect SDK has four main objects, namely the skeleton data, Skeleton Stream, Skeleton Frame, Skeleton, and Joint [6].

The Skeleton Stream is used to obtain the skeleton frame data. It is normally closed by default and needs to be started when used. The smoothing parameters in the bone data stream are used to eliminate bone node jitter. It includes the following four attributes:

- **Correction value property**, accept a floating point type from 0.1. The smaller the value, the more the correction.
- **Jitter Radius property**, set the radius of the correction, if the joint point "jitter" exceeds the set radius, it will be corrected to this radius. This property is floating point and the unit is meters.
- **Max Deviation Radius property**, which is used together with the jitter radius to set the maximum boundary of the jitter radius. Any point above this radius will not be considered to be jittery and will be considered a new point. This property is floating point and the unit is meters.
- **Predict the frame size d**, attribute, return the number of bone frames needed for smoothing.

Smoothing property, set the amount of smoothing when processing bone data frames, accept a floating point value of 0-1, the larger the value, the more smooth. 0 means no smoothing is performed.

Smoothing bone joint points can have performance overhead. The more smoothing, the more performance is consumed. Setting smoothing parameters has no experience to follow and requires constant testing and debugging to achieve the best performance and results. Different smoothing parameters may need to be set at different stages of the program run. [7] The bone tracking system is capable of tracking and acquiring 20 joint point information for each operator. Each joint has a Position property of type Skeleton Point, which describes the position of the joint point by three values of x, Y, and Z. The X, Y values are relative to the bone plane space. The Kinect SDK provides some coordinate transformation methods to convert the bone coordinate points into corresponding depth data images. In a human skeleton model consisting of 20 joint points, the palm of a person is represented by a joint point, which does not include contour information of the palm and fingertip information of the finger. For static gesture recognition, the contour information of the gesture is often very important, so this article will segment the palm information based on the human skeleton model provided by Kinect SDK.

Using Kinect to obtain the three-dimensional coordinate information of 20 nodes in the skeleton model, select 4 pairs of the 20 skeleton nodes as the feature processing objects of the dynamic gesture, which are left and right hand nodes, left and right wrist nodes, left and right elbow nodes and left and right shoulder nodes. As shown in Fig 2.
The feature vector that can represent a dynamic gesture at this time can be expressed as

\[ V_n = (x_1, y_1, z_1, \ldots, x_b, y_b, z_b) \]  

(1)

Where \( n \) is the number of feature vectors contained in a gesture. In the actual dynamic gesture recognition process, the movement of the shoulder is relatively small, so the position of the shoulder can be used as the calculation of the reference point. Define the reference point \( B (x_b, y_b, z_b) \) is the center of the shoulder line, after defining the reference point, define

\[ B(x_b, y_b, z_b) = \frac{1}{2} (x_7 + y_7, y_7 + z_7, z_7) \]  

(2)

\[ V' = \frac{|V - B|}{|L-R|} \]  

(3)

Where: \( B \) is the corresponding vector of the calculated reference point in the original coordinate system; \( V \) is the vector corresponding to the normalized pre-skeleton node; \( V' \) is the vector corresponding to the bone node after normalization; \( L \) is the corresponding shoulder of the left shoulder Vector; \( R \) is the vector corresponding to the right shoulder. Due to the relative fixation of the left shoulder and the right shoulder, the distance between the left shoulder and the right shoulder is used as the standard length. By the above method, the original coordinate system can be converted into the origin of the shoulder center point. The two-hand coordinate system, in which the reference point \( B \) is used as the origin of the two-hand coordinate system. This basically eliminates the influence of the human body size and the distance from the Kinect on the feature vector, thereby obtaining the pre-processed gesture feature vector.

2. DTW algorithm

DTW is a classical algorithm in speech recognition. The algorithm uses the idea of dynamic programming to solve the problem of isolated word recognition in speech recognition [651]. In speech recognition, people's speaking speed is different and the tones are similar. Similar to speech recognition, people in dynamic gesture recognition have different motion speeds and similar trajectories. The DTW algorithm can calculate the similarity between two sets of time-dependent sequences. The lengths of the two sequences can be different, but the sampling time is the same. For the dynamic gesture recognition system in this paper, each time the sample is acquired, the normalized feature vector in the gesture preprocessing is obtained [8].

In the DTW algorithm, the eigenvector space is defined as \( T \), and the time series \( X = (x_1, X_2, \ldots, X_N) \) of length \( N \) and the time series \( Y = (y_1, Y_2, \ldots, y_M) \) of length \( M \) are known, where \( N \) and \( M \) are not necessarily equal. In order to compare the similarities between two time series, a time warping function \( f \), known as the distance, is defined. Among them:

![Fig. 2 Skeleton model obtained by Kinect](image)
The value of $f$ is a real number greater than zero. When the similarity of the two time series is higher, the smaller the distance between the two features vectors, the smaller the value of $f(x, y)$. Conversely, when the similarity of the two time series is lower, the larger the distance between the two feature vectors, the larger the value of $f(x, y)$. The distance between each of the feature vector groups $(x, Y_m)$ in $X$ and $Y$ is calculated to obtain a time warping matrix $F$. Where $F \in \mathbb{R}^{N \times M}$, $F(n,m)=f(x_n, Y_m)$. After the time warping matrix $F$ is obtained, the problem of finding the similarity of two time series can be converted into finding a regular path in the $F$ matrix that minimizes the distance between $X$ and $Y$. Define the regular path as the sequence $G, G = (G_1, 62, ..., G_K)$, where $G_k = (n_k, m_k), G_k \in [1:N] \times [1:M], k \in [1:K]$.

The sequence $G$ should satisfy the following three conditions:

(1) Boundary conditions: $G_1 = (1, 1), G_K = (N, M)$;
(2) Monotonic conditions: $n_1 \leq n_2 \leq \cdots \leq n_k, m_1 \leq m_2, \leq \cdots \leq m_k$
(3) Step condition: $G_{k+1} - G_k \in \{(1,0), (0,1)(1,1)\}$, where $k \in [1:K-1]$.

The role of the boundary condition is to determine that the $X$ sequence corresponds to the first element of the $Y$ sequence, and the last element corresponds to the full sequence comparison of the $X$ sequence and the $Y$ sequence. The effect of the monotonic condition is to ensure that the direction of the regular path proceeds in the forward direction. The effect of the step size condition is that each element in the $X$ and $Y$ sequences should be compared and there will be no duplication in the course of the course.

A path that satisfies the above three conditions can be called a legal regular path. The general regularization function $f_c(X, Y)$ of a legal path $G$ between the $X$ sequence and the $Y$ sequence can be defined as:

$$f_G(X, Y) = \sum_{k=1}^{K} f(x_{n_k}, y_{m_k})$$

(5)

How to find the optimal path in all paths, that is, the path with the smallest $f(x, y)$ is the problem that the DTW algorithm needs to solve. Defining the distance between the $X$ sequence and the $Y$ sequence as $D(X, Y)$, then the distance corresponding to the optimal path $G^+$ is the DTW distance of $X$ and $Y$.

$$DTW(X, Y) = \min(f_G(X, Y))$$

(6)

The specific calculation method of the DTW distance between two time series is as follows: define a part of $X$ and $Y$ sequence $X(1:n)=\{x(1), X(2), ..., X(n)\}$ and $Y(1:m)=\{y(1), Y(2), ..., Y(m)\}$, where the threshold $[1:IV], m[1:M]$. The cumulative regularity matrix $D(n, m)$ of $X$ and $Y$ is calculated using Equation 4-7.

$$D(n, m) = DTW(X(1:n), Y(1:m))$$

(7)

It can be seen that the cumulative regular matrix $D$ is a matrix of $N*M$, $D(N, M) = DTW(X, Y)$. In the cumulative warping matrix, each matrix unit records the DTW distance of a subsequence of $X$ and $Y$. The cumulative warping matrix satisfies the following conditions:

$$D(n, 1) = \sum_{i=1}^{n} f(x_i, y_1) \quad n \in [1:N]$$

(8)

$$D(1, M) = \sum_{i=1}^{m} f(x_1, y_i) \quad m \in [1:M]$$

(9)

$$D(n, m) = \min(D(n-1)(m-1),D(n-1)(m),D(n)(m-1)) + f(x_n, y_m)$$

(10)
Equation 4.10 embodies the choice of the local optimal path, as shown in Figure 3. The local optimal path refers to the path that minimizes the current cumulative distance when calculating the current cumulative distance under path constraints. After the local optimal path is determined, the initial matching point can be found from the matching end point along the local optimal path to obtain the global optimal path.

**Fig. 3** Diagram of the adjacent nodes

### 3. Real-time and robust verification

In Figure 4, the robustness of the method under different illumination intensities and complex backgrounds was verified. Experiments were carried out in a sunny day with a light intensity of 300 Lx and a cloudy day with a light intensity of 40 Lx. The results showed that in these environments under the above, the recognition result can be obtained effectively, but the influence of the reduction of illumination on the recognition speed and recognition accuracy is more obvious than the complexity of the background environment.

**Fig. 4** Physical experiment of robustness verification

In order to verify the impact of environmental changes on real-time performance, the right hand and the right hand and the left hand are placed as test gestures to test the speed comparison of 20 identical gestures in the strong and weak environment. The results are shown in Figure 5.

In order to verify the influence of lighting environment changes on the robustness of the system, the left hand and the right hand and the right hand are placed as test gestures. The two gestures are in the environment of sunny indoor (300 Lx) and cloudy indoor (40 Lx). Tested 100 times, the recognition rate
results are shown in Figure 6. From the comparison results, it can be seen that the same gestures in the sunny room (300 Lx) will be relatively less time-consuming than the cloudy room (40 Lx).

The recognition rate will be improved because the pre-processing stage of the algorithm is based on Kinect's RGB colour camera. When the light intensity is lower than 90 Lx, the image quality will be seriously degraded. Therefore, the light intensity is lower. When low, the gesture pre-processing process in the algorithm will be affected to some extent, resulting in longer processing cycles and lower precision, but low illumination has no effect on Kinect's depth camera, so the dynamic gesture information can still be well recognized. It shows that the method has good real-time and robustness in both environments, and meets the expected requirements. [9]

Fig. 5 Comparison results of the time-consuming. Fig. 6 Comparison results of robustness verification

4. Conclusion
The integration of vision-based human gesture recognition into the robot control simplifies the complexity of robot control and improves the interaction between humans and robots. With the release of the Kinect somatosensory device, Kinect-based human motion and gesture recognition technology can effectively support robot control. This paper analyzes the current research status of human-computer interaction, human motion recognition and gesture recognition, and summarizes the current methods of gesture recognition. In the dynamic gesture recognition, the improved DTW algorithm is used for dynamic gesture recognition. The combination of dynamic and static gesture recognition not only improves the accuracy of dynamic gesture recognition, but also increases the number of gesture combinations, thereby increasing the richness of gesture commands.

The application of Kinect-based gesture recognition to robot control enriches the control technology of the robot and also reflects the application value of Kinect.

References
[1] Dong Shihai. The progress of human-computer interaction and the challenges it faces [J]. Journal of Computer-Aided Design & Computer Graphics, 2004, 16(1):1-13.
[2] Kong Fanzhen. Research on static gesture recognition algorithm based on vision [D]. Lanzhou Jiaotong University, 2009.
[3] Chang Yanan. HMM-based dynamic gesture recognition [D]. Guangzhou: South China University of Technology, 2012
[4] Yang Chengzhi, Zhang Yunsheng. Intelligent human-computer interaction in industrial integrated automation [J]. Journal of Kunming University of Science and Technology: Science and Engineering Edition,1999, 24(6): 6-12
[5] Takahashi T, Kishino F-Hand gesture coding based on experiments using a hand gesture Interface device[J]. ACM SIGCHI Bulletin, 1991, 23(2):67.74
[6] Yang M H, Ahuja N. Recognizing hand gestures using motion trajectories[M]//Face
Detection And Gesture Recognition for Human—Computer Interaction. Springer US, 2001:53—11.

[7] Cho K, Cho H, Um K. Human action recognition by inference of stochastic regular Grammars [M]// Structural, Syntactic, and Statistical Pattern Recognition. Springer Berlin Heidelberg, 2004: 388-396.

[8] Schlenzig J, Hunter E, Jain R. Recursive identification of gesture inputs using hidden Markov Models [C]// Applications of Computer Vision, 1994. Proceedings of the Second IEEE Workshop on. IEEE. 1994:187—194.

[9] JONES S, SHAO L. Computer analysis of images and patterns [M]. Berlin Heidelberg: Springer, 2013: 20-27.