



Research on multimodal human-robot interaction based on speech and gesture[☆]

Deng Yongda, Li Fang*, Xin Huang

South China University of Technology, Panyu District, 511400 Guangzhou, Guangdong, China



ARTICLE INFO

Article history:

Received 15 June 2018

Revised 10 September 2018

Accepted 12 September 2018

Keywords:

Human robot interaction

Fusion of speech and gesture

Interval Kalman Filter (IKF)

Maximum entropy classification

Natural language understanding

ABSTRACT

This paper presents a multimodal human-robot interaction based on fusion of speech and gesture. In the interface, a robot control command system is designed, which can transform the speech and gesture of users into commands that the robot can execute. Microsoft speech SDK is used in this system to collect the speech of the operator. Then, a corpus-based algorithm of maximum entropy classification for natural language understanding is employed to generate commands. Leap Motion is employed to capture the gesture of operator in this system. Interval Kalman Filter (IKF) is used to estimate the measured data to reduce the inherent noise of the sensor. The advantage of the proposed method is that the combination of speech and gesture makes the human-robot interaction more convenient and direct. Finally, a series of experiments were carried out to validate our method, and proved that it performed better than the other proposed methods.

© 2018 Published by Elsevier Ltd.

1. Introduction

In the future of the world, the robot will become a good helper of mankind. X. Li et al. [1] proposed a multi-modal control scheme for rehabilitation robotic exoskeletons. Therefore, the communication between human beings and robots is inevitable. Human beings get used to using gesture and speech to communicate with each other in daily life. So, people naturally think of using gesture and speech to interact with robots.

Using gestures for human-robot interaction is an ideal way, for most of the industrial robots and service robots which have the same structure as human hand. Compared with other forms of interaction, gestures express rich semantics and are easy to identify. Most human-robot interaction methods based on gesture recognition technology are Vision-based HCI (Human-Computer interaction) or Wearable HCI. The vision-based method of gesture capture is more natural than the wearable method, no matter whether the methods have markers or not. There is an occlusion problem with the application of markers that the markers may hinder the hand motion to some extent. Thereby, the proposed method employs markerless vision-based method of gesture capture. X. Suau et al. [2] put forward a three-dimensional gesture estimation algorithm based on 2.5 D data. A.R. Varkonyi-Koczy et al. [3] proposed a fuzzy estimation method of posture and hand gesture which applied to intelligent interactive environments. G. Du et al. [4] proposed a markerless human-robot interface to make the manipulator copies the movements of human hands. G. Du et al. [5] proposed a three-dimensional gesture tracking recognition algorithm based on hybrid sensors. G. Du et al. [6] optimized the accuracy of human-robot interaction using techniques

[☆] Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Guanglong Du.

* Corresponding author.

E-mail address: fl_scut@163.com (L. Fang).

such as hybrid filtering and precision optimization algorithms. However, using gestures alone lacks consideration about naturalness in the process of operation. In this case, it is difficult to apply to human-robot collaboration. Human hand tremor, psychological fluctuation can lead to the decrease of accuracy. Even some unintentional action from human led to wounding action of the robot.

The speech interaction based on natural language understanding is the most direct and convenient interactive way [7]. When different instructions were sent to the robot, the traditional natural language understanding used the keyword to match method [8]. Natural speech understanding interaction has a richer instruction library than traditional speech recognition interaction methods. Y. Shimizu et al. [9] proposed the Port-to-Port path instruction resolution system and applied it to map path planning. D.K. Misra et al. [10] from Cornell University have implemented a system that can perform specific tasks in a changing indoor environment with the user's natural language instructions. Most of above researches focus on the use of natural language to assist robots in better perform tasks, so most commands are simple. For this shortcoming, there are some improvements. W. Wang et al. [11] designed a series of effective robot control commands to control service robot, whose instruction matching accuracy is more than 90 percent. J. Fasola et al. [12] proposed a global attribute dynamic spatial relationship (DSR) method as part of a method to make the robot follow the non-expert users of natural language commands. The method particularly concerned about the development of space language primitives. In order to achieve the desired depth of analysis, M. Eppe et al. [13] adopted the module and the frame of the Embodied Construction Grammar (ECG) developed a natural language interface for human robot interaction, realizing the deep meaning in natural language. But, as the natural language has a great ambiguity, especially while controlling the robot, the device instructions given may be uncertain, sometimes even cause discrepancy [14].

Inspired by the fact that human beings communicate with each other using multi-sensory channel, we employ the combination of gestures and speech to control the robot. In our method, speech acts as a natural interaction way to control robot while gesture serves as a complement to the speech. In this case, the instructions that cannot be expressed in words can be given by gestures which could capture the user's real intentions. The other instructions can be directly expressed by speech which is natural and convenient. Furthermore, for those complicated instructions we can combine the instructions of gestures and speech to complete it which improves the accuracy of instructions. The combination of gestures and speech can reduce the shortcoming of using gestures or speech alone and make the communication between human and robots more natural, efficient and accurate.

The remainder of this paper is organized as follows. Section II gives an overview of the multimodal human-robot interaction. Section III describes the recognition of hand motion. Section IV introduces the speech recognition. And Section V describes the combination of gesture and the speech to control the robot. Section VI details the experiment. Section VII makes a conclusion.

2. Overview

Using Speech to control the robot is natural and convenient, especially when there is no need to meet the keywords. Speech commands can instruct the robot to move, even ask the robot to cover a precise distance with just a few words. However, if use gesture commands, the operator maybe should do a series of gestures to make the robot understand. But inevitably there are some things that cannot be expressed clearly by words such as some non-special direction, some actions that cannot be described easily and so on. Instead, an intuitive gesture can make the robot know the intention of the operator. Therefore, our method combines the speech and the gesture to make the human-robot interface more convenient and accurate.

Fig. 1 shows the process of the proposed human-robot interface system. The user issues a speech command which could be obtained and converted into text by Microsoft Speech SDK. The text is processed by the maximum entropy model, which is able to understand the intention of the user through speech. Then our robot first determines if there is a gesture indicating in the speech. If the speech contains some information that suggests the control of robot is related to the operator's gestures, the gestures of the user will be captured by Leap Motion. There are three infrared radiation (IR) LED and two IR cameras installed internally to generate frames. Leap Motion is a motion detecting device, and the infrared frames obtain the hand position data in its workspace. Due to the inherent noise of the sensor, Interval Kalman Filter is employed to estimate the measured data from Leap Motion controller. The processed data will be regarded as part of the intention of the operator. Then the gesture instructions and speech instructions will be combined to make a complete instruction for the robot control. If the speech instruction itself is a completed command, the speech instruction will work alone to control the robot. The robot will recognize key words in the speech instructions and do what the instructions indicate.

3. Hand gesture recognition

3.1. Hand tracking

The Leap Motion sensor can detect and track palms, fingers and tools which are similar to fingers. Leap Motion software analyzes objects within the visible range of the device, assigning a unique ID to palms, fingers and tools. The position, gestures and motion of palms, fingers, tools can be queried in real time. If the object exists in the current frame, then the query function returns a reference to the object. If the object does not exist, then the query function returns a special

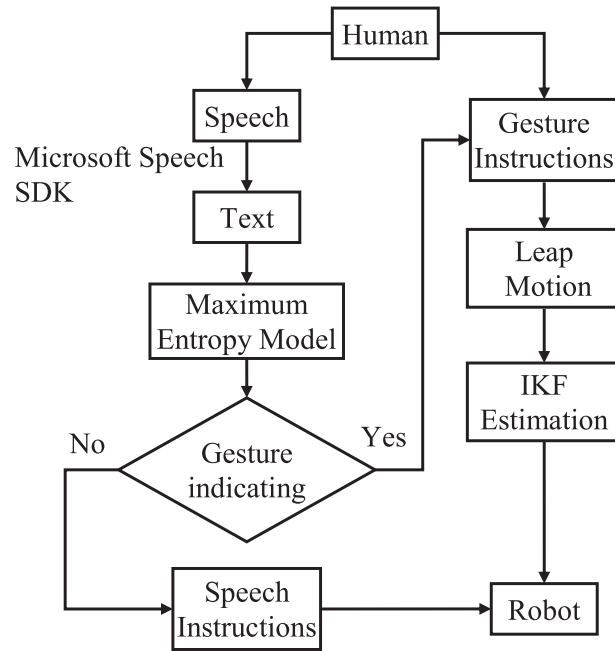


Fig. 1. Human-robot interface system process.

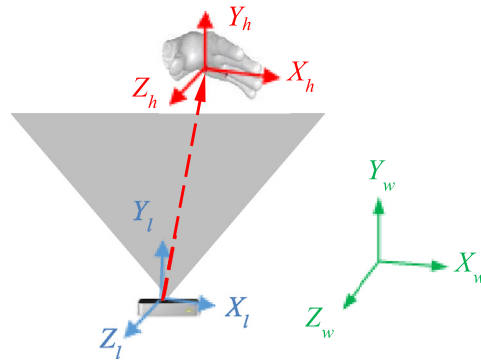


Fig. 2. Coordinate of hand tracking system.

invalid object. The acceleration and orientation of the palms can be obtained from the measured data with the intrinsic hand gesture recognition algorithm [15].

Leap Motion uses the right-hand Cartesian coordinate system. The origin of the Leap Motion controller is in the center of itself, while the X-axis and Z-axis are on the device's horizontal plane. The X-axis is parallel to the long side of the device. The Y-axis is vertical, with the positive value increases in the direction of upwards. Leap Motion's visual range is an inverted cone whose vertex is at the center of the device. The working range of the Leap Motion controller is about 25 mm to 600 mm above the device. In order to define the hand position in the global coordinate system, three coordinate systems can be defined: world coordinate system, hand coordinate system and Leap Motion coordinate system. As shown in Fig. 2, the world frame is defined as $X_wY_wZ_w$, which is fixed. The hand frame is defined as $X_hY_hZ_h$. The Leap Motion frame is defined as $X_lY_lZ_l$. In order to control the robot, the hand data can be converted to the global coordinate system, since the hand data obtained by Leap Motion is based on the Leap Motion coordinate system.

In this study, gesture serves as a complement to the speech, which is mainly used to indicate the direction. Each finger has four joints. Fig. 3 shows the direction indicating gesture, which is recognized as the instruction to get direction. The direction is indicated by a line with number 1 joint and the number 4 joint lying on. The Leap Motion controller can detect the position of the number 1 joint and the number 4 joint. The positions of the two joints are estimated by Interval Kalman Filter that will be introduced in the following subsection. When the operator makes a direction indicating gesture, the operator can use the speech to remind the robot to pay attention to the hand motion of the operator at the same time. The estimated data will be regarded as a part of the intention of the operator.

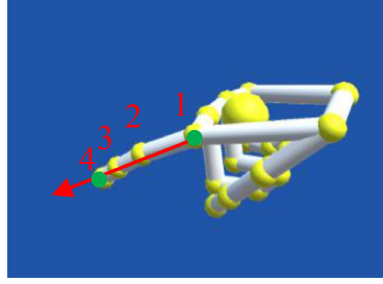


Fig. 3. Direction indicating gesture.

3.2. Direction estimation using IKF

There are some methods to perform tracking control. Y. Tang et al. [16] examined the problem of tracking control of networked multi-agent systems with multiple delays and impulsive effects. More frequent methods used to perform tracking control is based on sensors. In our paper, Leap Motion is employed to measure the position, velocity and acceleration. However, the noise of the sensor will increase over time. If the measured data is not processed, the proposed system will be of low accuracy and reliability. Therefore, it's important to reduce the noise. S. Li et al. [17] presented an ∞ observer design scheme to solve the state estimation problem such that the resulting error dynamic system is stochastically stable. G. Du et al. [18] proposed an approach that incorporates a Kalman filter (KF) and a particle filter to reduce errors while estimating the position and orientation of the manipulator. X. Qing et al. [19] presented a decentralized unscented Kalman Filter (UKF) method based on a consensus algorithm for multi-area power system dynamic state estimation. The noise of our sensor is the inherent noise. Therefore, Interval Kalman Filter (IKF) [20] is used to estimate the data in this study with an attempt to improve the precision of the human-robot interface. IKF can deal with the situation with statistical parameters and inaccurate dynamics compared to standard Kalman Filter [21]. It is because parameters uncertainties are expressed as intervals in IKF.

The Kalman Filter algorithm assumes that the prior probability of processing noise and the measurement noise are consistent with the Gaussian distribution. Moreover, at the beginning the initial state of the system x_0 , measurement noise covariance matrix R_k and system noise covariance matrix Q_k are known. The calculation process is as follows:

- (1) Compute a priori state estimate:

$$\tilde{x}_k = A_k \cdot \tilde{x}_{k-1} + B_k \cdot u_{k-1} \quad (1)$$

- (2) Compute a priori estimate error covariance:

$$\tilde{R}_k = A_k \cdot \tilde{P}_{k-1} \cdot A_k^T + Q_{k-1} \quad (2)$$

- (3) Compute the Kalman gain:

$$K_k = \tilde{P}_k \cdot \tilde{H}_k^T \cdot [H_k \cdot \tilde{P}_k \cdot \tilde{H}_k^T + R_k]^{-1} \quad (3)$$

- (4) Update a posteriori estimate error covariance:

$$\hat{P}_k = [1 - K_k \cdot H_k] \cdot \tilde{P}_k \quad (4)$$

- (5) Update a posteriori state estimate:

$$\hat{x}_k = \tilde{x}_k + K_k \cdot (z_k - H_k \cdot \tilde{x}_k) \quad (5)$$

The KF model comprises the measurement model and the system state model, and we define t_k as a period of time with respect to the subscript k . The related formulas can be expressed as:

$$\begin{aligned} x_k &= A_k \cdot x_{k-1} + B_k \cdot u_{k-1} + b_{k-1} \\ z_k &= H_k \cdot x_k + v_k \end{aligned} \quad (6)$$

where x_k represents the state vector, u_{k-1} is the deterministic input, z_k means the measurement vector, b_{k-1} and v_k are the random variables which represent the process and measurement noise. A_k is the system transformation matrix from time t_{k-1} to time t_k , B_k is the input matrix, H_k means the measurement matrix.

The center of palm coordinates in the world frame are represented by $P(p_x, p_y, p_z)$. The IKF estimates the position state P in the position estimation process. The Leap Motion measures position component data in the hand frame. Defined M_{H2W} as the transformation matrix from the hand frame to the world frame, and then M_{H2W} has the form:

$$M_{H2W} = \begin{bmatrix} m_{x_x} & m_{y_x} & m_{z_x} \\ m_{x_y} & m_{y_y} & m_{z_y} \\ m_{x_z} & m_{y_z} & m_{z_z} \end{bmatrix} \quad (7)$$

where the angle between the i -axis in the hand frame and the j -axis in the world frame is $m_{ij} = \cos(\theta_{ij})$ and θ_{ij} ($i, j \in (X, Y, Z)$).

In the world frame, the hand acceleration can be computed by:

$$\begin{aligned}\dot{V}_x &= m_{X_x} \bullet A_x + m_{Y_x} \bullet A_y + m_{Z_x} \bullet A_z \\ \dot{V}_y &= m_{X_y} \bullet A_x + m_{Y_y} \bullet A_y + m_{Z_y} \bullet A_z \\ \dot{V}_z &= m_{X_z} \bullet A_x + m_{Y_z} \bullet A_y + m_{Z_z} \bullet A_z - |g_l|\end{aligned}\quad (8)$$

where A is the calculated acceleration components, the subscript represents the corresponding axis in the hand frame. And $|g_l|$ is the modulus of the local gravity vector. Then the velocity component V can be got in the local frame (the subscripts represent the corresponding axis):

$$V_x = \dot{p}_x \quad V_y = \dot{p}_y \quad V_z = \dot{p}_z \quad (9)$$

the state x'_k of the estimated position by IKF can be written as:

$$x'_k = [p_{x,k}, V_{x,k}, A_{x,k}, p_{y,k}, V_{y,k}, A_{y,k}, p_{z,k}, V_{z,k}, A_{z,k}] \quad (10)$$

where at the time t_k , the state x'_k is determined by the variables values. Because of Eqs. (8) and (9), the state-transition matrix A_k can be expressed as:

$$A_k = \begin{bmatrix} 1 & t & m_{X_x} \bullet t^2/2 & 0 & 0 & m_{Y_x} \bullet t^2/2 & 0 & 0 & m_{Z_x} \bullet t^2/2 \\ 0 & 1 & m_{X_x} \bullet t & 0 & 0 & m_{Y_x} \bullet t & 0 & 0 & m_{Z_x} \bullet t \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & m_{X_y} \bullet t^2/2 & 1 & t & m_{Y_y} \bullet t^2/2 & 0 & 0 & m_{Z_y} \bullet t^2/2 \\ 0 & 0 & m_{X_y} \bullet t & 0 & 1 & m_{Y_y} \bullet t & 0 & 0 & m_{Z_y} \bullet t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & m_{X_z} \bullet t^2/2 & 0 & t & m_{Y_z} \bullet t^2/2 & 1 & t & m_{Z_z} \bullet t^2/2 \\ 0 & 0 & m_{X_z} \bullet t & 0 & 0 & m_{Y_z} \bullet t & 0 & 1 & m_{Z_z} \bullet t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

The acceleration is decided by gravitational force and the direction of the Y-axis is the same with that of the gravity vector. The input matrix can be expressed as:

$$B_k \bullet u'_{k-1} = [0, 0, 0, 0, 0, 0, -|g_l| \bullet t^2/2, -|g_l| \bullet t, 0]^T \quad (12)$$

where $|g_l|$ is the modulus of the local gravity vector.

$$w'_k = [0, 0, w'_x, 0, 0, w'_y, 0, 0, w_z]^T \quad (13)$$

where (w'_x, w'_y, w_z) is the process noise of the hand acceleration.

The observation matrix for the position estimation can be defined as:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

At time k , the most appropriate value for the position of the human hand can be represented by the determined position $P(p_{x,k}, p_{y,k}, p_{z,k})$.

The covariance of the model error and observation error in the IKF are defined as:

$$\begin{cases} Q_k^I = [Q_k - \Delta Q_k, Q_k + \Delta Q_k] \\ R_k^I = [R_k - \Delta R_k, R_k + \Delta R_k] \end{cases} \quad (15)$$

where ΔQ_k and ΔR_k are two constant perturbation matrices. It can be considered as constant if the environment does not change for disturbances from the measured White Gaussian Noise. Therefore, two constant perturbation matrices are nonnegative and bounded.

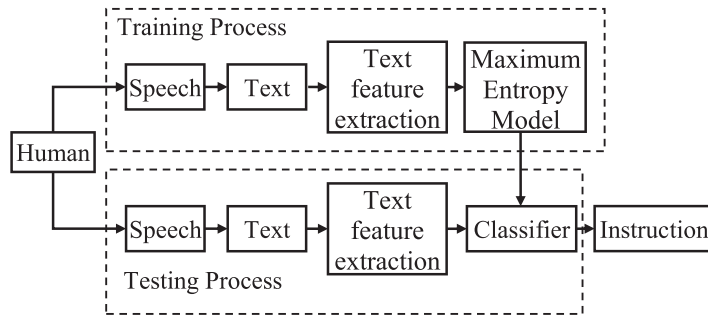


Fig. 4. Speech understanding process.

4. The speech recognition

Fig. 4 shows the process of the speech comprehension process. In the training process, Microsoft Speech SDK gets the operator's speech and converts it to text. Text features are extracted from the training texts, and then the weights are added to the text feature, which is expressed as the feature vector of the text. Then the maximum entropy classification model is trained by the text feature vector and the corresponding class labels. This classification model will be used for the process's test as the decision strategy of the classifier. In the testing process, still, Microsoft Speech SDK acquires the operator's speech and converts it into text. The testing text is expressed as a text feature vector. Then on the basis of the maximum entropy model established in the training process, the testing text is classified by the existing maximum entropy classifier, and then corresponding robot control instructions are gained.

The human-robot interaction based on natural language comprehension stresses the understanding of the operator's some complicated natural speech commands. The difficulty of the natural language comprehension method for human-robot interaction lies in the transformation between the purpose expressed in the speech and the corresponding robot control commands. It can be treated as a classification problem. The classifier based on the maximum entropy model [11] is adopted in this paper. The speech recognition and text extraction can be achieved through the available mature speech recognition algorithm. The natural language comprehension framework used for human-robot interaction includes two modules: the design of control instruction corpus and text analysis (extracting control instruction).

4.1. Control instruction corpus

According to the literature [11] and analysis of a number of control corpus, we designed the robot-control-command by introducing four attributes variables (V_{dir} , V_{opt} , V_{val} , V_{unit}) in this paper. The robot control commands give the definitions and descriptions of four attribute values to makes the instructions more systematic. The robot control command system can also prevent ambiguity and promote the robot's performance. In the four attribute variables, V_{dir} represents operational orientation keywords, namely, up, down, front, back, left, or right in the speech. However, with the help of gestures, the robot can move in any direction which the operator noted, and the variables is showed as \vec{p} or $[x, y, z]$ to represents a direction vector. V_{opt} and V_{val} are a pair of operational descriptions, which represent operating keywords and operating values severally. V_{unit} is an operating unit. The ambiguous expressions are automatically replaced by some of the certain descriptions.

For instance, if the instruction of UP 3 mm sent by the operator, it will be translated to the attribute variables [Up, Move, 3, mm]. If the operator wants to control the robot to UP a little, the instruction will be translated to [Up, Move, a little, Null]. Then the command [Up, Move, a little, Null] will be automatically transformed into the command [Up, Move, 1, mm] by our system. In this case, the user's natural speech can be combined into an amount of particular executable robot commands. Due to the control instruction corpus, robot executes more effectively.

4.2. Text analysis

For text analysis, it is important to extract text information automatically. In our method, we select TF-IDF (Term Frequency-Inverse Document Frequency) to complete this task. Text representation uses vector space model [22]. Based on the training corpus, all the occurrences of the word are counted. If a corpus includes N words, then the feature vector of N dimension can be used to express the text. TF-IDF is used for classifying the feature vectors before classification in this paper. TF value is a local variable, while IDF is a global variable. Combine the two aspects of the global and local features into the feature vector's weight as indicated in the formula:

$$TF_{i,j} = n_{i,j} / \sum_k n_{k,j} \quad (16)$$

$$IDF_i = \log(|D| / |\{j : t_i \in d_j\}|) \quad (17)$$

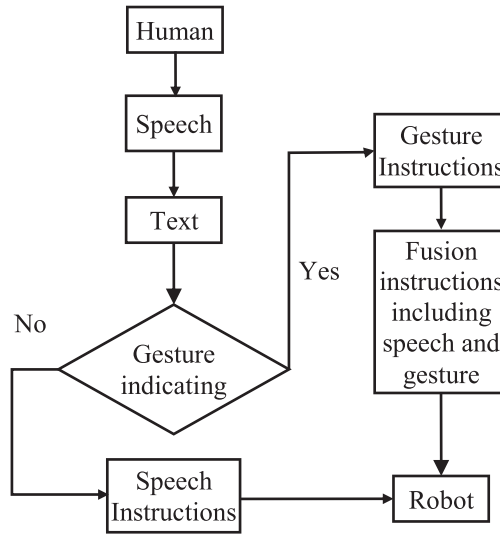


Fig. 5. Combination of speech and gesture to control the robot.

$$TFIDF_{i,j} = TF_{i,j} * IDF_i \quad (18)$$

In formula (16), n_{ij} denotes the occurrences' number of the word in the corpus text, $\sum_k n_{k,j}$ denotes the number of all words which corpus text contain. In formula (17), $|D|$ denotes texts' number in the training corpus, $\{j: t_i \in d_j\}$ denotes the number of corpus texts containing the word.

There are a lot of examples of statistical modeling in the natural language processing. It usually uses the maximum entropy model technology to treat English statistical methods since the maximum entropy model is simple, versatile and easy to transplant [23].

The concept of maximum entropy was first proposed by Jaynes [24] and was first applied in natural language processing in Berger [25]. Now the maximum entropy model is widely used in various natural language processing tasks and the original model based on the calculation of the continuous optimization. The main idea of the maximum entropy model is to follow the principle of maximum entropy to model, that is, to select the model with the largest entropy in the model. As a distinguishing model, one of the advantages of the maximum entropy model is that it can fuse multiple features in one model and model these features directly to the posterior. In addition, the distribution of the maximum entropy model is exponential family distribution, with good analytical properties to facilitate the calculation.

The core idea of the maximum entropy model is when predicting the probability distribution of a random variable, we do not make any assumptions about the unknown conditions while all known conditions are satisfied. At this time, the information entropy of the probability distribution is the maximum, which preserves all kinds of possibilities and minimizes the risk of prediction.

Assuming that x is a text feature vector, the corresponding intent output tag is y ($y \in Y$, Y is a finite set of intent labels). The maximum entropy algorithm is to model the conditional probability $P(L \times y)$, and get the most uniform distribution model, which is an optimization problem. We introduce the conditional entropy $H(P)$ to measure the uniformity of conditional probability $P(L \times y)$ distribution. According to Shannon's definition of entropy, the calculated formula of $H(P)$ is as the formula (19):

$$H(p) = - \sum_{x,y} \tilde{p}(x) p(y|x) \log p(y|x) \quad (19)$$


Through the maximum entropy model, it is possible to identify the interaction instructions contained in the text extracted by the user's speech. The text then is converted into a robot control instruction to control the robot movement.

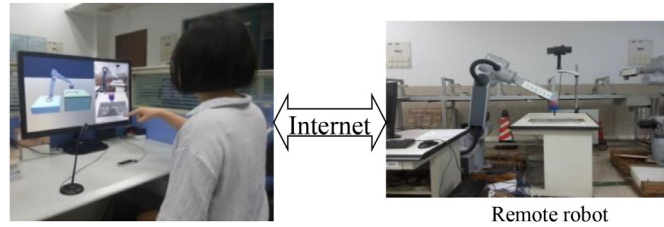
5. The combination of gesture and speech

In the proposed method, the natural multimodal human-robot interaction includes two parts: the speech and gestures as shown in Fig. 5. After extracting the control description of the robot, the robot will first analyze the four attributes variable transformed by Microsoft Speech SDK. If the robot gets the specific description of orientation and distance of some action, the robot will know the speech instruction can work alone without gestures' aid. On the contrary, if the four attributes variable doesn't contain the attributes of orientation and distance, then the Leap Motion will capture gestures of the operator. Gestures will be analyzed and transformed into some attribute of the variable transformed from speech command. Some

Table 1

Different types of speech commands and their corresponding robot control instructions.

Instruction type	Speech	Vague speech	Speech with static gesture	Speech with dynamic gesture
Interaction ways	Speech	"Move 1 mm Toward the direction of the X axis"	"Up a little"	"Move 2 mm in the direction"
	Gesture	–	–	 direction: \vec{p} or $[x, y, z]$
Instruction attributes variables	V_{dir}	X	Up	\vec{p} or $[x, y, z]$
	V_{opt}	Move	Move	Move
	V_{val}	1	1	2
	V_{unit}	mm	mm	mm

**Fig. 6.** The environment of experiments.

gestures even will be transformed into a series of four attributes variables to control the robot. In general, speech control can express some action control and the detailed distance conveniently while gesture control is the aid of speech control, mainly used in the case that the speech cannot describe the task clearly such as accurate azimuth.

Table 1 shows different types of speech commands and their corresponding robot control instructions. For the instruction of speech with static gesture, the user issues a command like "move 2 mm in the direction and points to a certain direction with a finger. Then it is necessary for robot to understand the natural language, that is, the operation is move, the orientation is the direction of the user's finger (\vec{p} or $[x, y, z]$), the distance value is 2 and the unit is mm. In this case, the natural language understanding model doesn't get the specific description of orientation. So Leap Motion will capture the direction of the operator's finger to help generate the command. According to the control form described above, the four attributes variable robot control commands will be represented as ($V_{dir} = \vec{p}$ (or $[x, y, z]$), $V_{opt} = move$, $V_{val} = 2$, $V_{unit} = mm$). In this case, the corpus used is not merely a simple speech control command corpus, but also includes a gesture command corpus. The gesture command corpus tells the robot that a part of the robot control commands should be acquired by three-dimensional gesture. The combination of the gesture and the speech produce a complete control command. Furthermore, the user can employ the method of speech with dynamic gesture to control robot for completing a more direct interaction between human and robot. For example, the user says "Follow my hand" and then moves the hand to draw a trajectory. Then the following robot control commands will be produced by the transformation of the speech and dynamic gesture.

$$\begin{aligned}
 (V_{dir} &= \overrightarrow{p_0 - p_{now}}, V_{opt} = move, V_{val} = |\overrightarrow{p_0 - p_{now}}|, V_{unit} = m), \\
 (V_{dir} &= \overrightarrow{p_1 - p_0}, V_{opt} = move, V_{val} = |\overrightarrow{p_1 - p_0}|, V_{unit} = m), \dots, \\
 (V_{dir} &= \overrightarrow{p_n - p_{n-1}}, V_{opt} = move, V_{val} = |\overrightarrow{p_n - p_{n-1}}|, V_{unit} = m)
 \end{aligned}$$

As we can see, it is convenient and straightforward to use speech to process communication between human beings and robots while gestures are usually used in situations where it is difficult to express merely by the speech. According to the combination of speech and gesture, the four attribute variables is prone to be constructed and the human-robot interaction becomes more natural and efficient.

6. Experiment

6.1. The content and steps of the specific experiment

Our experimental platform is built using the GOOGOL GRB3016 robot. To control the robot, we send speech and gesture commands to the robot. These commands will be passed to the human-computer interaction system, through which the natural language will be transferred to the corresponding execution instructions. And then the computer parses the operator's commands and remotely controls the robot over the network as shown in Fig. 6. To verify the effectiveness of the proposed method, four experiments were carried out.

In the experiment 1, the operator could remotely control the robot to perform placing peg in a hole experiment on the target metal board, which verifies the feasibility of the proposed method. At the same time, we also compared our method

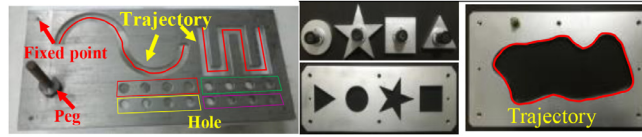


Fig. 7. Experimental plates and metal objects. (a) Steel plate. (b) The workpieces and corresponding holes. (c) Irregular trajectory.

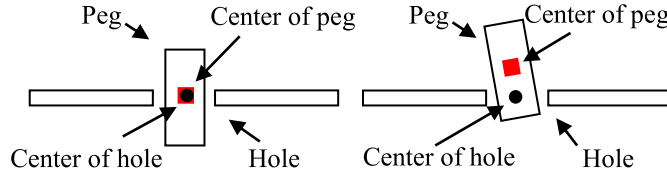


Fig. 8. Definition of 3D errors.

with method [6] and method [13] to prove that our method has many advantages. As shown in Fig. 7(a), we performed sixteen operations in the experiment. Each operation was placing a peg in a hole in the steel plate. Among them, the size of the steel plate is 300 mm × 500 mm, the diameter of the punched nail is 7.5 mm, and the radius of the hole is slightly larger than 8 mm.

Experiment 2 corresponds to Fig. 7(b), which is an operation experiment for placing metal blocks. In this experiment, round, star, square and triangular metal blocks were used. Corresponding to this, there was a steel plate capable of accommodating four holes of corresponding shapes. The operator needs to remotely manipulate the robot by speech commands to place four differently shaped metal blocks into corresponding holes. Speech commands are converted to a classification model through the Microsoft Speech SDK and later translated into instructions that the robot can execute.

In Experiment 3, we used speech commands to control the robot to move along the red sine curve and “W” curve as shown in Fig. 7(a). In Experiment 4, it was moved along an irregular pattern as shown in Fig. 7(c) which is different from Experiment 3. These two experiments verify the accuracy of our method. In order to avoid accidental errors, we repeated each of the experiment 3 and the experiment 4 twice.

6.2. Results

As can be seen from Fig. 8, prior to the experiment, the center coordinates of the hole were measured by the depth camera. After inserting the hole, the insertion depth and direction are determined by the depth camera by detecting the edge of the plug. Next, the coordinates of the nail relative to the center of the camera frame can be obtained. Define it as the center of the nail and the center of the hole. Fig. 8 gives a 3D position error. The resolution of the calibration camera is 1280 (H) × 960 (V). The formula for calculating the error E is:

$$E = \sqrt{(x_0 - x_t)^2 + (y_0 - y_t)^2 + (z_0 - z_t)^2} \quad (20)$$

where (x_0, y_0, z_0) as the center of the peg and (x_t, y_t, z_t) as the center of the hole.

Define E_k is the error of the k^{th} hole, the N punching tests mean absolute error about position was:

$$E_m = \sum_{k=1}^N E_k / N \quad (21)$$

Table 2 shows the 3D error of 16 holes among our method, method [6] and method [13]. From Table 2, we can discover that the 3D error caused by our method is the smallest among the three methods. The average 3D error is 1.26 mm, which is smaller than methods [6] and [13]. This shows that our method is more accurate than the other two methods.

We organized a total of five volunteers to participate in the experiment. These volunteers needed to perform four operations to put four different shapes of metal blocks into the corresponding holes in the metal plate. The number of successes and the average operating time are shown in Table 3. The success time refers to the time when the volunteers successfully placed the metal block into the corresponding hole. The average operating time refers to the average execution time of the four tests. Seen in Table 3, although the average execution time of method [6] is the smallest of the three methods, it has the lowest accuracy and the average number of successes is only 3.4. The four tests of our method and method [13] can be completed. However, the average running time of our method performed by 5 different volunteers was 92.8 s, which was less than the 123.8 s value of method [13]. From the results of placing the metal block experiments, we can see that our method can meet the requirements of efficiency and accuracy at the same time.

Table 4 shows the data records in Experiment 3. We used our method, method [6] and method [13] to control the robot tracking sine curve and “W” curve, respectively, and perform two operations each time. It can be seen that our method performs best in both aspects, and method [13] is the most frequent. When tracking the “W” curve in Test 1, our method

Table 2

3D errors of the 16 holes for three methods (mm).

Hole number	Method [13]	Method [6]	Our method
1	1.46	1.82	1.39
2	1.37	1.78	1.31
3	1.52	1.65	1.28
4	1.51	1.76	1.24
5	1.36	1.77	1.18
6	1.48	1.92	1.15
7	1.34	1.66	1.22
8	1.61	1.53	1.29
9	1.54	1.76	1.33
10	1.49	1.89	1.35
11	1.57	1.67	1.26
12	1.52	1.68	1.17
13	1.45	1.72	1.15
14	1.39	1.83	1.23
15	1.69	1.87	1.31
16	1.55	1.88	1.35
Mean error	1.49	1.76	1.26

Table 3

The placing workpieces task (V: Volunteer, St: success times, Tm: the mean of operation time (s)).

Volunteer number		V1	V2	V3	V4	V5	Mean
Our method	St	4	4	4	4	4	4
	Tm	91	90	92	93	98	92.8
Method [13]	St	4	4	4	4	4	4
	Tm	121	122	125	123	128	123.8
Method [6]	St	3	3	4	3	4	3.4
	Tm	72	74	73	73	70	72.4

Table 4

Regular curve tracking errors of the three methods. (T1: test 1, T2: test 2, Me: mean error (mm), Mx: max error (mm)).

		Minimum (mm)		Mean (mm)		Maximum (mm)		Time (s)	
		T1	T2	T1	T2	T1	T2	T1	T2
Our method	“W”	0.21	0.18	0.32	0.38	0.58	0.61	18	19
	Sine	0.31	0.28	1.31	1.28	2.35	2.37	78	77
Method [13]	“W”	0.48	0.45	1.48	1.45	2.63	2.66	112	118
	Sine	0.44	0.47	1.44	1.47	2.72	2.73	135	137
Method [6]	“W”	0.36	0.38	1.36	1.38	2.77	2.73	68	65
	Sine	0.33	0.39	1.33	1.39	2.68	2.69	78	79

Table 5

Irregular curve tracking errors of the four methods. (T1: test 1, T2: test 2).

		Minimum (mm)		Mean (mm)		Maximum (mm)		Time (s)	
		T1	T2	T1	T2	T1	T2	T1	T2
Our method		1.21	1.26	2.45	2.35	11.41	10.44	118	116.5
Method [13]		1.47	1.48	3.62	3.58	13.98	14.92	231.5	223.5
Method [6]		1.35	1.36	3.73	2.78	13.28	13.47	121.5	123

has the shortest running time, which is nearly 6 times shorter than method [13] and 4 times shorter than method [6], and the error is much lower than the other two methods. Therefore, the data in Table 4 demonstrates that our method performs better than methods [6] and [13] in tracking curves both from accuracy and speed.

Fig. 9 shows a trace of a regular pattern such as a sine curve. The enlarged part of the trace trajectory is shown in a small window. Black represents the reference path, red, purple, and blue are the trajectories formed by our method, method [6], and method [13], respectively. From the figure we can see that the trajectory formed by our method is more smooth and closer to the reference path, which means our method is more accurate than the other two methods.

Table 5 shows the results of the robot tracking irregular curves. In order to reduce accidental errors caused by human factors, we repeated each character twice, recording as Test 1 and Test 2, respectively. Our method has smaller average error

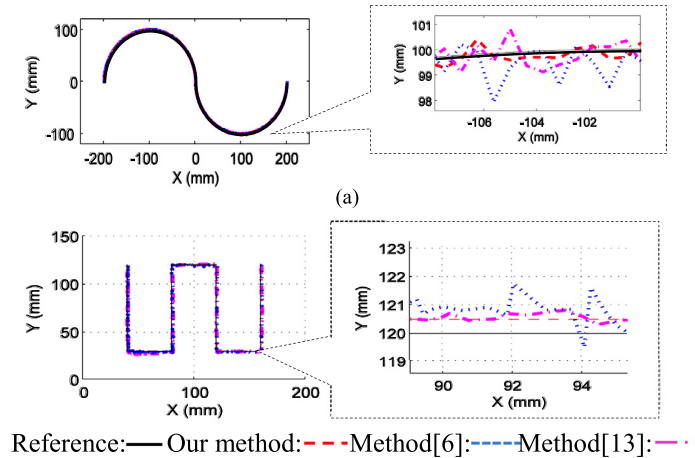


Fig. 9. Tracking results of regular trajectory tracking experiment. (a) The tracking results of sine curve. (b) The tracking results of “W” curve.

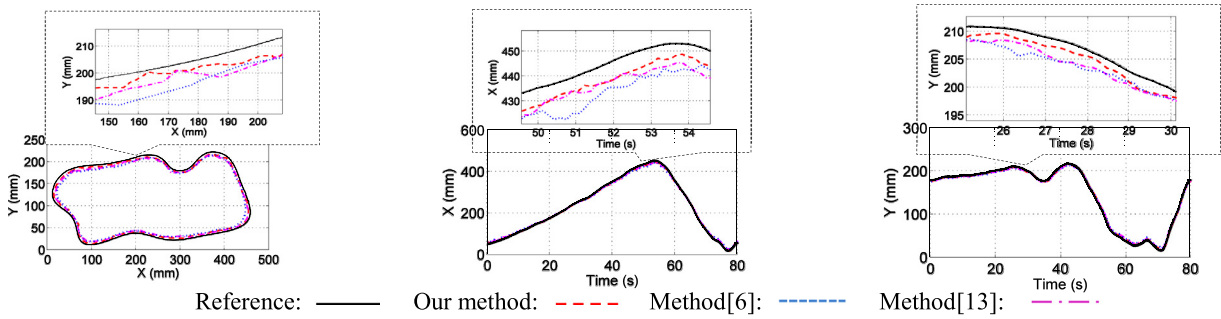


Fig. 10. Tracking results of irregular trajectory. (a) The trajectory of the robot's movement. (b) The trajectory in the x direction. (c) The trajectory in the y direction.

in Test 1 and Test 2 than methods [6] and [13], and the operating time of our method is also the shortest. This shows that our method has higher accuracy and efficiency than methods [6] and [13] while tracking irregular curves.

Fig. 10(a) shows the plane movement trajectory of the robot EE of Experiment 4. Fig. 10(b) and Fig. 10(c) are trajectories developed on the X-axis and the Y-axis, respectively. A small window shows the portion of the magnified track. The black solid line represents the reference path and the red dotted line represents our method. The blue dashed and purple dashed lines represent method [6] and method [13], respectively. From these figures, it can be seen that the trajectory of our method is the closest to the reference path, which means that our method is the most accurate and stable.

From the above experiments, we could conclude that while executing some simple commands, our method is more accurate and efficient. In the four experiments, in terms of both operational error and execution time, our method performs better.

7. Conclusion

In this paper, we have comprehensively considered the advantages and disadvantages of previous research, complementing each other, and proposed an improved human-robot interaction. The movements of robot are controlled by both the operator's speech and gesture. In order to recognize the speech command, the maximum entropy classifier is used herein to classify the speech command, which is mapped with the control instructions of the robot. Furthermore, we use Leap Motion to measure hand data at the same time. Due to the noise of the Leap Motion controller, we use Interval Kalman Filter to estimate the measured data of the operator's hand, which increases the accuracy of the proposed system. In order to verify the effectiveness of the method, we invited several volunteers to conduct different experiments, including placing peg in a hole, placing metal blocks and tracking a path. The results proved that the proposed method is better than the method only based on speech or gesture. The fusion of speech and gesture in this method indeed improves the naturalness, efficiency and accuracy of the method. It also makes non-professionals have the ability to perform human-computer interaction easily.

However, there are also some defects. Current experiments are simple so that we cannot analyze how it behaves under more complex instructions. Secondly, we do not think about the complexity of the method. But in fact, in a real environment, robots do face more complex tasks and instructions, which inevitably requires lower complexity of the entire algorithm. Therefore, in the future work, we will try more complex instructions and observe the performance of our method to carry

out a further assessment. At the same time, to make our methods more practical, we will consider simplifying our method to reduce the complexity of the method.

References

- [1] Li X, Pan Y, Chen G, Yu H. Multi-modal control scheme for rehabilitation robotic exoskeletons. *Int. J. Robot. Res.* 2017;36(5–7):759–77.
- [2] S Xavier R, Javier C, Josep R. Detecting end-effectors on 2.5D data using geometric deformable models: Application to human pose estimation. *Comput. Vis. Image Underst.* 2013;117(3):281–8.
- [3] V Annamária R, Balázs T. Human–computer interaction for smart environment applications using fuzzy hand posture and gesture models. *IEEE Trans. Instrum. Meas.* 2011;60(5):1505–14.
- [4] Guanglong D, Ping Z, Xin L. Markerless human-manipulator interface using leap motion with interval Kalman filter and improved particle filter. *IEEE Trans. Ind. Inform.* 2016;12(2):694–704.
- [5] Guanglong D, Ping Z, Di L. Human–manipulator interface based on multisensory process via Kalman Filters. *IEEE Trans. Ind. Electron.* 2014;61(10):5411–18.
- [6] Guanglong D, Ping Z. A markerless human-robot interface using particle filter and Kalman Filter for dual robots. *IEEE Trans. Ind. Electron.* 2015;62(4):2257–64.
- [7] Laurence D, Marie T, Amine S, Agnes D. Inference of human beings' emotional states from speech in human–robot interactions. *Int. J. Soc. Robot.* 2015;7(4):451–63.
- [8] Michael S, Brian A. Method and apparatus for multiple tiered matching of natural language queries to positions in a text corpus. US 2003.
- [9] Yasushi S, Tomomichi S. Efficient path planning of humanoid robots with automatic conformation of body representation to the complexity of environments. *IEEE-Ras Int. Conf. Humanoid Robots IEEE* 2013:755–60.
- [10] Dipendra KM, Jaeyong S, Kevin L, Ashutosh S. Tell me Dave: Context-sensitive grounding of natural language to manipulation instructions. *Int. J. Robot. Res.* 2014;35(1–3).
- [11] Wen W, Qunfei Z, Tehao Z. Research of natural language understanding in human–service robot interaction. *Microcomput. Appl.* 2015;31(3):45–8.
- [12] Juan F. Modeling dynamic spatial relations with global properties for natural language-based human-robot interaction. *Ro-man IEEE* 2013:453–60.
- [13] Manfred E, Sean T, Jerome F. Exploiting deep semantics and compositionality of natural language for Human-Robot-Interaction. *IEEE/RSJ Int. Conf. Intel. Robots Syst.* 2016:731–8.
- [14] Hans K, Uwe R. From discourse to logic: introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory. *Language* 1993;71(4).
- [15] Frank W, Bachmann D, Rudak B, Fisseler D. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors* 2013;2013(13):6380–93.
- [16] Tang Y, Xing X, Karimi HR, Kocarev L, Kurths J. Tracking control of networked multi-agent systems under new characterizations of impulses and its applications in robotic systems. *IEEE Trans. Ind. Electron.* 2016;63(2):1299–307.
- [17] Li S, Xiang Z, Lin H, Karimi HR. State estimation on positive Markovian jump systems with time-varying delay and uncertain transition probabilities. *Inf. Sci.* 2016;369:251–66.
- [18] Guanglong D, Ping Z. Online Serial Manipulator Calibration Based on Multisensory Process Via Extended Kalman and Particle Filters. *IEEE Trans. Ind. Electron.* 2014;61(12):6852–9.
- [19] Qing X, Karimi HR, Niu Y, Wang X. Decentralized unscented Kalman filter based on a consensus algorithm for multi-area dynamic state estimation in power systems. *Int. J. Electrical Power Energy Syst.* 2015;65:26–33.
- [20] Xiufeng H, Yang L, Wendong X. MEMS IMU and two-antenna GPS integration navigation system using interval adaptive Kalman filter. *IEEE Aerosp. Electron. Syst. Mag.* 2013;28(10):22–8.
- [21] Kaci B, Benjamin L, Walter S. A fault tolerant architecture for data fusion: A real application of Kalman filters for mobile robot localization. *Robot. Auton. Syst.* 2017;88:11–23.
- [22] Salton G, Wong A, Yang CS. A vector space model for automatic indexing. *ACM* 1975.
- [23] Skut W, Brants T. A Maximum-Entropy Partial Parser for Unrestricted Text. In: *Proceedings of the 6th ACL Workshop on Very Large Corpora (WVLC)*, Montreal; 1998.
- [24] Chun-hung L, Lee CK. Minimum cross entropy thresholding. *Pattern Recognit.* 1993;26(4):617–25.
- [25] Adam LB, Stephen A Della P, Vincent J Della P. A Maximum Entropy approach to Natural Language Processing. *Comput. Linguist.* 1996;22:39–71.

Deng Yongda is a Research Scholar in school of computer science and engineering of South China University of Technology, Guang Zhou, China. He is a student in computer science and engineering department of South China University of Technology. His current research interests include human-computer interaction and Intelligent Robotic control.

Li Fang received her Ph.D. degree in mechanical and automotive engineering department of South China University of Technology, Guang Zhou, China. She is a teacher in computer science and engineering department of South China University of Technology. Her current research interests include embedded system development approach, Cyber-physical system development, Intelligent Robotic control.

Xin Huang received his Ph.D. and M.S. degrees in school of mechanical and electrical engineering of Central South University, Chang Sha, China. He is a S.E. in Guangzhou Start To Sail Industrial Robot Co., LTD. His current research interests include industrial robot control, motion planning, embedded system control, intelligent manufacturing.