



## Article

# Fusing Hand Postures and Speech Recognition for Tasks Performed by an Integrated Leg–Arm Hexapod Robot

Jing Qi <sup>1</sup> , Xilun Ding <sup>1</sup>, Weiwei Li <sup>1,2</sup>, Zhonghua Han <sup>3</sup> and Kun Xu <sup>1,\*</sup> 

<sup>1</sup> Robotics Institute, School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China; qijing@buaa.edu.cn (J.Q.); xlding@buaa.edu.cn (X.D.); lww1994@buaa.edu.cn (W.L.)

<sup>2</sup> Beijing Institute of Computer and Electronics Application, Beijing 100191, China

<sup>3</sup> First Research Institute of the Ministry of Public Security of People's Republic of China, Beijing 100191, China; zhuanghuahan@126.com

\* Correspondence: xk007@buaa.edu.cn

Received: 30 July 2020; Accepted: 5 October 2020; Published: 7 October 2020



**Abstract:** Hand postures and speech are convenient means of communication for humans and can be used in human–robot interaction. Based on structural and functional characteristics of our integrated leg–arm hexapod robot, to perform reconnaissance and rescue tasks in public security application, a method of linkage of movement and manipulation of robots is proposed based on the visual and auditory channels, and a system based on hand postures and speech recognition is described. The developed system contains: a speech module, hand posture module, fusion module, mechanical structure module, control module, path planning module and a 3D SLAM (Simultaneous Localization and Mapping) module. In this system, three modes, i.e., the hand posture mode, speech mode, and a combination of the hand posture and speech modes, are used in different situations. The hand posture mode is used for reconnaissance tasks, and the speech mode is used to query the path and control the movement and manipulation of the robot. The combination of the two modes can be used to avoid ambiguity during interaction. A semantic understanding-based task slot structure is developed by using the visual and auditory channels. In addition, a method of task planning based on answer-set programming is developed, and a system of network-based data interaction is designed to control movements of the robot using Chinese instructions remotely based on a wide area network. Experiments were carried out to verify the performance of the proposed system.

**Keywords:** hand postures recognition; speech recognition; human–robot interaction (HRI); hexapod robots; manipulation

## 1. Introduction

Robots are being used increasingly in activities in our daily lives; thus, robots need to interact with people who are not experts in robotics. To make robots to be used conveniently and efficiently, good human–robot interaction plays an important role. Human–robot interaction (HRI) based on command lines requires that a technician operates the robot. Although HRI based on the graphical user interface has made this possible for non-expert users, it does not satisfy the requirements of natural interaction. To solve this problem, the means that humans employ to communicate with each other are introduced into human–computer interaction [1].

Humans obtain information through vision and hearing and can communicate with one another. Robots have similar capabilities: they can acquire information through visual and auditory sensors, analyze the data, and hence interact with humans naturally. In daily life, people usually communicate with one another using language and gestures and choose an adaptable manner of

communicating depending on the task and objective at hand. To render human–robot interaction natural, some researchers have used hand postures/gestures and natural language to interact with robots.

Vision-based hand gestures can be generally classified into static gestures (hand postures) and dynamic gestures (hand gestures) [2]. Hand shapes are hand postures (static gestures), while hand movements are hand gestures (dynamic gestures) [3]. Some researchers utilized dynamic gestures (hand gestures) and natural language to interact with robots. Stiefelhagen et al. [4–9] built a multimodal system for human–robot interaction based on the humanoid robots ARMAR II and ARMAR III. Burger et al. [10] used speech and hand gestures to control the movement of the mobile robot Jido. Liu et al. [11] integrated voice, hand motions and body posture into a multimodal interface by using a deep learning-based method to control an industrial robot.

However, some researchers used static gesture (hand postures) and speech to control movements of robots. A multimodal system of interaction has been proposed for generating a map of the environment, where hand postures and natural language are used to help a wheeled mobile robot to generate a map of the environment [12]. Hand postures and speech recognition are used to command the assistant robot ALBERT to perform simple tasks [13]. Hand postures and natural language were used to assist a lay user in a pick-and-place application [14,15]. We focus the work that use hand postures and voice commands to control movements of the robot.

In this work, we focus on tasks of reconnaissance, rescue and other public security applications, based on characteristics of our integrated leg–arm hexapod robot. At present, robots can achieve known tasks in a structural environment autonomously, while they cannot fulfill unknown tasks in an unstructured environment autonomously. However, unknown tasks in an unstructured environment are often accomplished by public security personnel. In this case, human knowledge and experiences are utilized, a supervised pattern and demonstration pattern are combined, and a method of linkage of movement and manipulation of robots is proposed based on the visual and auditory channels.

Our robot has multiple ways of movement, such as “3 + 3” gait, “2 + 4” gait, and “1 + 5” gait. Additionally, the robot has two manipulators, and different tools can be installed in the end effectors of the manipulators, such as a clamp and scissors. Thus, it requires multiple interactivities. Moreover, it needs natural interaction between a human and the robot. The difficulty of linkage of multiple movements and manipulations of the robot is how to design an interaction system based on hand postures and speech recognition, so that public security personnel can perform tasks naturally, conveniently and efficiently. To solve this problem, we combine a supervision pattern and demonstration pattern, and propose an interactive method of auditory-visual modality, to achieve tasks of reconnaissance and rescue in public security applications. Specifically, tasks that require a robot to perform are mainly divided into two categories: regular tasks and complex tasks. Regular tasks, such as move forward, backward, left and right, are achieved by speech commands, and the video captured by the camera installed on the robot is used to supervise the operations of the robot, to make certain that the robot perform the desired action. However, complex tasks, such as unknown tasks in an unstructured environment, cannot be accomplished by the robot only using speech instructions; this is because a robot cannot fully “understand” and execute complex speech instructions. A good solution is learning from demonstration. For example, the position of a specific hand posture in an image is used to control the position of a manipulator of the robot, in other words, the position of an end effector of a manipulator changes with the position of a specific hand posture in an image.

In addition, we focus on reconnaissance and rescue tasks in public security applications. When reconnaissance tasks are performed, only hand postures are used to control the movements of the robot, because of the concealment of reconnaissance. Thus, the proposed system of the modalities of vision and hearing has three modes, i.e., hand postures, speech, and a combination of them. The appropriate mode is chosen depending on the task at hand. The speech mode is used to query the path and control the movement of the robot. In a combination of the hand posture and speech modes, the combination of deictic hand posture and speech can be used to avoid ambiguity. A semantic understanding-based task slot structure is developed by using the visual and auditory channels.

Furthermore, we describe an auditory-visual system for reconnaissance and rescue tasks through the interaction of a human and our integrated leg–arm hexapod robot. The system consists of several modules, including a speech module, hand posture module, fusion module, 3D SLAM (Simultaneous Localization and Mapping) module, path planning module, mechanical structure module and a control module. Furthermore, to solve the problem of information loss, which is caused by demonstrative words with ambiguous reference of speech commands, a semantic understanding-based task slot structure is developed by using the visual and auditory channels. In addition, structural language commands can be used to express key information, but there is a deviation between the structural language commands and the corresponding instructions that the robot can execute. To solve this problem, a method of task planning based on answer-set programming is developed. Moreover, a system of network-based data interaction is designed, to control movements of the robot using Chinese instructions remotely based on a Wide Area Network (WAN).

The remainder of the paper is organized as follows: An overview of the proposed system that combines vision and hearing channels is presented in Section 2. The use of CornerNet-Squeeze to recognize hand postures for reconnaissance is described in Section 3. A semantic understanding-based task slot structure through the visual and auditory channels is presented in Section 4. Experiments to verify the proposed system are presented in Section 5, and the conclusions of this study are given in Section 6.

## 2. System Architecture

The leg–arm hexapod robot is shown in Figure 1, and the architecture of the proposed system is shown in Figure 2. The results of hand posture and Chinese natural language recognition are transmitted to a control layer via a human–robot interaction layer. The hexapod robot is then controlled to perform a specific task using the control layer. Environmental information is also obtained through an environmental perception layer.

A path planning module and a 3D SLAM module constitute the environmental perception layer. The former features an efficient hierarchical pathfinding algorithm based on a grid map of the indoor environment in which the integrated leg–arm hexapod robot operates [16]. The latter is an effective approach to SLAM based on RGB-D images for the autonomous operation of the robot [17].

The speech module is responsible for semantic understanding, task planning, and data interaction based on the network's submodules. The semantic understanding submodule is based on a task-oriented method of semantic understanding [18] using the characteristics of the integrated leg–arm hexapod robot and Chinese instructions, a semantic understanding algorithm, and a structural language framework based on verbs. Natural language was thus transformed into regular commands in the structural language.

A method of task planning based on answer-set programming was developed in the task planning submodule. Structural language commands can be used to express key information, but there is a deviation between the structural language commands and the corresponding instructions that the robot can execute. To solve this problem, a combination of information concerning the robot's state, environment, structural commands, executable actions, and the optimal objective was used. Furthermore, the answer-set rule was designed, and the commands in the structural framework are converted into a sequence of actions that a robot can perform.

In data interaction based on the network submodule, a remote connection between the host computer and the robot is established for exchanging data through a Wide Area Network (WAN). In this way, the host computer can remotely obtain real-time information on the robot and send commands to guide the robot's movements. Furthermore, a real-time communication connection between the front and back ends of the network is established based on the Django framework in the host computer so that it can obtain services from the front end of the network and guide the robot to specific locations on the outdoor map. By designing the interface of the host computer, it becomes convenient for users to obtain information about the robot in real time.

The system of interaction has three modes: the hand posture mode, the speech mode, and a combination of the two. In the hand posture mode, hand postures are used to control the robot. In the speech mode, Chinese natural language is used to control the robot. Information on both hand postures and speech is used to control the robot in the combination of the hand posture and speech modes.



Figure 1. Our leg-arm hexapod robot.

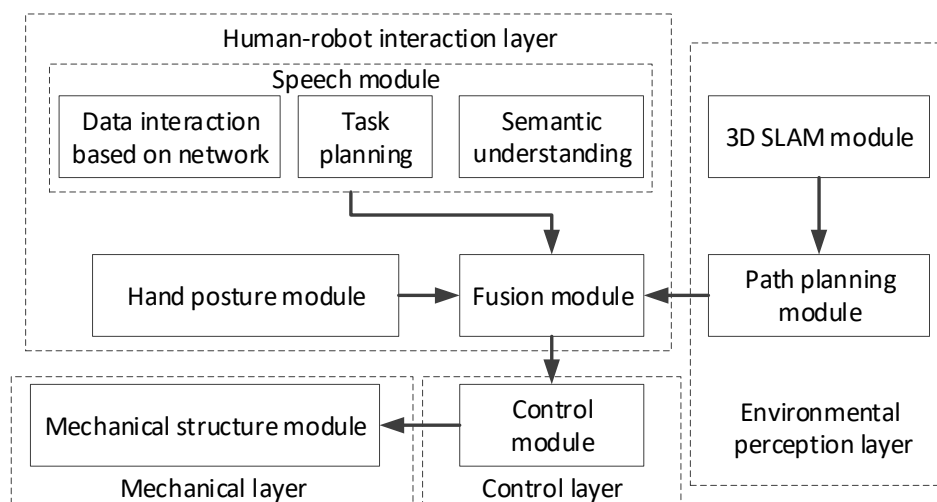


Figure 2. Framework of the proposed system.

### 3. Hand Posture Recognition

Typical users are not experts in robotics, and natural interaction between human and robots is required for many tasks. Because the reconnaissance task has the feature of concealment, speech is not suitable for the user to interact with the robot. However, hand postures are intuitive, non-verbal, and natural, and require no sound for interaction. They are thus chosen for the reconnaissance task in this study.

Knowledge gained and learned from humans is transferred to the proposed hand posture module to enhance HRI. The hand postures are regarded as graphics, and their maps are regarded as knowledge representations. Hand postures are used to control the movements of the leg-arm hexapod robot in this study.

The hand posture module of our leg-arm hexapod robot is designed to enable the robot to perform reconnaissance, rescue, and counterterrorism tasks. CornerNet-Squeeze [19] is also used in the system.

Several types of hand posture were designed according to daily communication-related actions, and based on the requirements of the tasks and the characteristics of our robot. Moreover, both a



mapping from a given hand posture to the corresponding motion/manipulation of the robot, and one from the hand posture to the user's corresponding intention were predefined. Furthermore, part of the mapping is shown in Tables 1–3.

Datasets of hand postures, confirmation of the user's intention, and the movement and manipulation of the hexapod robot were designed based on the structural and functional characteristics of the leg–arm hexapod robot, and according to the principle of natural interaction between humans and their robot partner. The latter two data sets were also mapped to the former.

Images of hand postures were captured to form our dataset. CornerNet-Squeeze was then used to train our model to recognize the hand postures.

**Table 1.** Mapping manipulations of the hexapod robot.

Number	Manipulation of the Robot	Type of Hand Posture
1	lift the left integrated limb	Thumb and forefinger are stretched out straight; the forefinger points upward, and the thumb points to the left.
2	open the clamp	All fingers are stretched straight and spread wide.
3	put the right integrated limb down	Thumb and forefinger are stretched out straight; the forefinger points downward, and the thumb points to the right.
4	close the scissor	Forefinger and middle finger are stretched straight, and touch.
5	protrude the speculum	Thumb, forefinger, and pinkie are stretched straight.

**Table 2.** Mapping the movements of the hexapod robot.

Number	Movement of the Robot	Type of Hand Posture
1	forward	Only the forefinger is stretched out straight, pointing upward, and the palm of the hand faces forward.
2	left	All fingers are stretched out straight and close together, and the fingertips point to the left, or only the forefinger is stretched out straight and its tip points to the left.
3	stop	All fingers are stretched out straight and close together, the palm of the hand faces forward, and the fingertips point upward.
4	stand up	All fingers are stretched out straight and close together, the palm of the hand faces backward, and the fingertips point upward.
5	hunker down	All fingers are stretched out straight and close together, the palm of the hand faces backward, and the fingertips point downward.

**Table 3.** Mapping the confirmation of user intentions.

Number	User's Intention	Type of Hand Posture
1	yes	Middle finger, pinkie, and ring finger are stretched straight, and only the tips of the thumb and the forefinger touch.
2	satisfaction	Thumbs up.
3	uncertainty	Only the tips of all fingers touch, and the palm of the hand faces forward.
4	repeat	Index finger and pinkie are stretched straight.
5	switch to voice mode	Four fingers are stretched straight.

#### 4. Semantic Understanding-Based Task Slot Structure through Visual and Auditory Channels

Primitive operations are independent operations by users that cannot be divided but can be recognized by devices. Primitive operations include the minimal information transmitted through each channel, where this is required to analyze a specific task. A task is divided into independent primitive operations. In the hand posture and speech modes, the channels are independent, and the entire task is divided into collaborations using different channels. Only speech is entered into the primitive operation through the auditory channel. In other words, users convey commands using natural language. Only images are entered into the primitive operation through the visual channel; thus, users issue commands using hand postures.

Hand postures are intuitive and expressive, and their ideographic meaning is concise, whereas speech is abstract and rich in connotations. To efficiently interact using hand postures and speech, both auditory and visual channels should be used. For example, “Go there!” a user said, pointing in a particular direction. Moreover, the information conveyed by hand postures and speech is complementary. Multichannel interaction is designed to solve the problem of coordinating information from different channels to describe a complete task. To solve it, a semantic understanding-based task slot structure through the visual and auditory channels is proposed here.

The multichannel integration model for user tasks fills the task slot. Once the slot has been filled with multichannel data, a complete command is formed. As different data are needed to achieve different goals, different task slots are designed for different tasks. However, too many parameters for tasks can lead to complex operations, which affect the ease of user operation. To fulfill the requirements of specific tasks, the characteristics of the given task and commands conveyed from hand postures and speech need to be analyzed. A structural language framework is used to form the structure of our task slot concisely so that users can easily understand it.

The standard structure of a task slot is as follows:

$$\text{ActionsForTask} + \text{parameter1} + \text{parameter2} + \dots + \text{parameterN}$$

Different parameters are needed to perform different tasks. However, an interaction task generally features actions for the task, objects for the action, and the corresponding parameters. The general structure of a task slot is as follows:

$$\text{ActionsForTask} + \text{ObjectsForAction} + \text{parameters}$$

If the above structure is used, a definite object of the action should be given. This not only increases the workload and makes the application interface complex, but also burdens the operation and memory of users. Furthermore, this structure cannot satisfy the requirements of tasks based on our leg–arm hexapod robot. Because the robot is multifunctional and involves complex motion, many types and numbers of commands can be given using hand posture and speech. Moreover, Chinese natural language has a large vocabulary, and there are many means of expression. Differences between verbs are sometimes subtle even though they represent significantly different tasks. To enable the robot to understand the meanings of deictic hand postures and speech, the relevant information is integrated into the overall interaction-related information, and the semantic understanding-based task slot structure using visual and auditory channels is employed.

Integrated information on deictic hand postures and speech is converted to fill a semantic understanding-based task slot through visual and auditory channels as below.

Primitive operations through the visual and auditory channels are executed simultaneously, and data on the deictic hand postures and speech are obtained. The relevant posture and speech are then recognized and converted into corresponding semantic text, as shown in Figure 3. Following this, a grammatical rule is designed based on the category of the given word, and the corresponding semantic components are selected from the semantic texts. The structural language framework is

thus filled, such that the commands conveyed by the hand posture and speech are integrated into a synthesized command.

#### 4.1. Extracting Semantic Texts for Hand Posture- and Speech-based Commands Using Semantic Understanding

Our leg–arm hexapod robot is multifunctional, and can perform many types of actions. A semantic understanding module is designed to facilitate this. By analyzing characteristics of the commands used to control the robot, a semantic understanding algorithm is proposed to covert commands in Chinese into those in the structural language. Commands conveyed by hand postures and speech are converted into their respective semantic texts, as shown in Figure 3.

Data on the hand postures and speech are first obtained through the visual and auditory channels, respectively, and the relevant posture and speech are recognized. Based on the predefined map of hand postures, the results of recognition are converted into text conveying the relevant command in Chinese. Following this, semantic texts of the commands pertaining to the hand posture and speech are extracted.

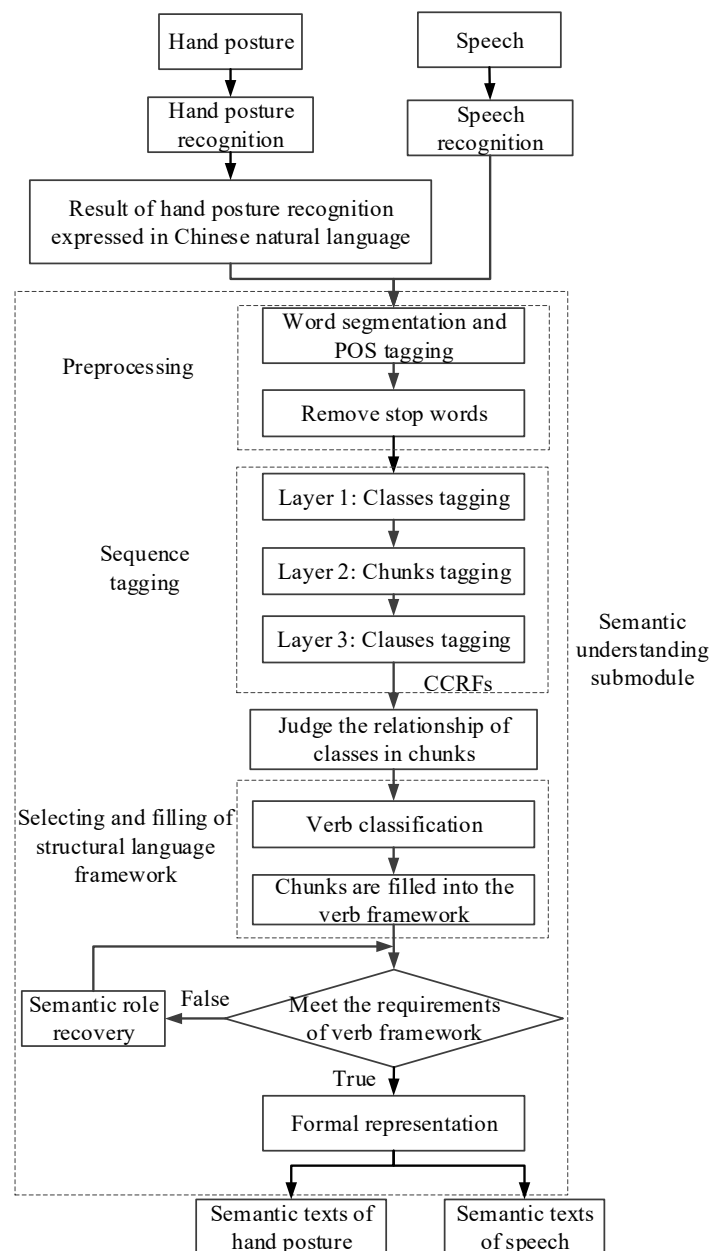


Figure 3. Flowchart for the extraction of semantic texts from hand postures and speech.

#### 4.1.1. Preprocessing

There are spaces between English words, while Chinese characters are closely arranged, and there are no obvious boundaries between Chinese words. However, words are generally the smallest semantic unit, so the first step is Chinese word segmentation for Chinese natural language processing. The NLPPIR (Natural Language Processing and Information Retrieval) Chinese lexical analysis system (NLPPIR-ICTCLAS) is used for this and part-of-speech (POS) tagging. This is because some words in instructions are unrelated to semantic content, for example, “please” and “probably”. To delete these words and simplify structures of commands, stop words are then removed as they are not related to semantic content.

Although ICTCLAS can segment Chinese words, two problems arise in the results of word segmentation when ICTCLAS is used. To explain the problems clearly, a few examples were given, which are shown Table 4. As shown in Table 4, each Chinese instruction was given in the top line of the left column, and was translated from Chinese into English, which was listed below the corresponding Chinese instruction, so that the Chinese instructions can be read easily. As shown in Table 4, ICTCLAS achieves fine granularity in this case, which leads to the first problem, i.e., a word is sometimes divided into more than one part. For example, New Main building, room a306 and speculum are words, respectively; however, they are divided into two parts. The second problem is that the result of word segmentation is sometimes not consistent with the contextual meaning of the relevant sentences, such as, (in the tank) and (into the door).

**Table 4.** Results of the word segmentation and POS (part-of-speech) tagging of Chinese instructions.

Speech Instruction	Results of the Word Segmentation and POS Tagging
Put the dangerous goods, which is in the a306 room in the New Main building, into the explosion-proof tank.	with/pba New/a (Main building)/n a306/n room/n in/f of/ude1 (dangerous goods)/n put/v into/v proof/v explosion/v (into the tank)/s
Put the speculum into the crevice of the door slowly.	with/d peep/vg glass/ng slowly/ad stretch/v (into the door)/vn crevice /n into/f

#### 4.1.2. Sequence Tagging of Instructions

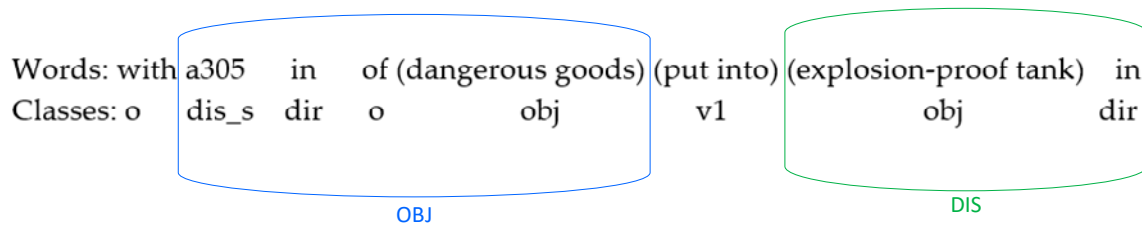
To solve the first problem, linear chain Conditional Random Fields are used to tag words in instructions. Furthermore, the words of these instructions are classified by analyzing the characteristics of the commands, which is shown in Table 5. To tag “dis\_g”, “dis\_s” and “obj” classes (Which is shown in Table 5) correctly, a pre-established lexicon and Amap API (the application programming interface provided by AutoNavi) are used to identify their types.

Because the motion of the integrated leg–arm hexapod robot is complex, speech instructions in human–robot interaction are more complex than ordinary mobile robots, and key information cannot be extracted from instructions directly. Thus, the chunk analysis method is applied to analyze the instructions, and we find that the effective information, which is not affected by syntactic structure, can be divided into several chunks: motion type “V”, motion direction “DIR”, stop position “DIS”, motion speed “VE”, moving gait “BT” and operation objects. Moreover, the operation objects are divided into two categories: body of the robot “USE” and external object “OBJ”. Additionally, instructions are tagged by the chunks above, and classes compose chunks.

To elaborate chunks clearly, a result of chunk analysis of an instruction (Put the dangerous goods, which is in the a305 room, into the explosion-proof tank.) was given, which is shown in Figure 4. As shown in Figure 4, a Chinese instruction is on the top row of the table, the classes of Chinese words are listed on the bottom row of the table. For the sake of an English reader’s convenience, Chinese words were translated from Chinese into English, which are listed below the corresponding Chinese words. The words and classes in the blue bounding box belong to external object chunk “OBJ”, while the words and classes in the green bounding box belong to stop position chunk “DIS”.

**Table 5.** Classes of words.

Classes	Example	Description	Classes	Example	Description
v1	Move	Verbs that should be executed in order	pre	to	Preposition
v2	speed up	Verbs that should be executed immediately	dir	front	Direction
obj	dangerous goods	Entity can be recognized through the image recognition and pronoun	t	2 min	Time
dis	5 m	Distance	tb	4 + 2	Gait
dis_g	Vision Hotel	Spots on the outdoor map	ve	fast	Speed
dis_s	elevator room	Spots on the indoor map	wj	,	Punctuation
use	right hand	Robot body	o	at	Others

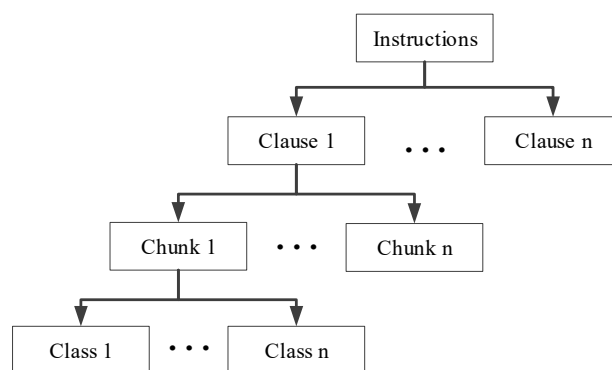
**Figure 4.** Result of chunk analysis of an instruction.

Because an instruction often has multiple clauses, if punctuation is used as the basis for segmentation (speech is recognized as text, so, there is punctuation in the text), the problems are that a clause may imply multiple actions and that two verbs in different clauses form one action. Some examples are shown in ③④. To solve this problem, the instruction is segmented into clauses based on the assumption that only one action is conveyed in a clause.

③ Go to room a306 to get inflammable substances.

④ Move forward until you reach the Weishi Hotel.

After three steps of tagging, instructions are decomposed into clauses, the clauses are composed of chunks, and chunks are comprised of classes. A diagram of structure of instructions is shown in Figure 5. It can be seen that each word in an instruction can be described accurately.

**Figure 5.** Diagram of structure of instructions.

Conditional Random Fields (CRFs) is a kind of undirected graphical model [20], in which context features are considered, and all features are normalized to obtain a global optimal solution. Thus, CRFs suit to label sequence data. Moreover, the size of our corpus is relatively small, for which it is suitable to use a supervised learning algorithm such as CRFs. Thus, CRFs were selected in this work.



Specifically, linear CRFs were used, in which  $y = (y_1, y_2, \dots, y_n)$  denotes the probability of a label sequence, and  $x = (x_1, x_2, \dots, x_n)$  represents an observation sequence, which is shown as follows:

$$P(y|x) = \frac{1}{Z(x)} \exp \left( \sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \right) \quad (1)$$

$$Z(x) = \sum_y \exp \left( \sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \right)$$

where  $t_k$  is a transition feature function,  $s_l$  is a state feature function,  $\lambda_k$  and  $\mu_l$  are weights, and  $Z(x)$  is the normalization factor.

Three steps of instructions tagging both use CRFs for sequence tagging, and the three steps constitute the Cascaded Conditional Random Fields (CCRFs), as shown in Figure 6. The input variables of the upper layer contain not only the observation sequences, but also the recognition results of the lower layers, which increases the types of features of upper layers, and it is helpful to complete the complex semantic sequence tagging.

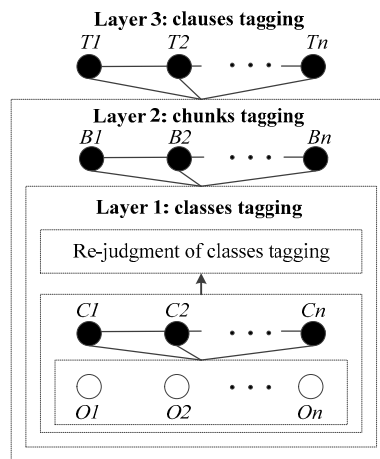


Figure 6. Diagram of cascaded conditional random fields.

To solve the second problem, letter-based features instead of word-based features are adopted to correct the position of segmentation of the words. To get the position of the letter in word and class, we utilize “BMEWO” to mark the position. Specifically, in “BMEWO”, “B” denotes the letter that is in the first place of the word or class, “M” represents the letter that is in the middle of the word or class, “E” means the letter that is in the end of the word or class, “W” denotes the single letter that constitutes the word or class, “O” means the letter that does not belong to any class.

Features used for each layer of CRFs are shown in Table 6, in which W is a letter, P is the comprehensive feature of the position of the letter in the word and the POS tagging (like “B\_v”); T is the result of the layer 1 and U is the result of the layer 2; W(0) is the current letter, W(1) is the next letter, W(2) is the second letter after W(0), W(-1) is the letter preceding W(0) and W(-2) is the second letter before W(0). The definition of P(n), T(n) and U(n) are the same to W(n).

#### 4.1.3. Judgment on Relationships between Classes in Chunks

After the sequence tagging of instructions, the number of the class “obj”, class “dis\_g” and class “dis\_s” in the chunk “DIS”, chunk “OBJ” and chunk “DIR” is greater than one. Moreover, there are some relationships between classes. Thus, it is difficult to extract key information of the chunks directly. Some examples are shown in ①②③. There are three kinds of relationship between classes in the chunk: the latter modifies the former, such as ①; the former modifies the latter, such as ②; and the former is juxtaposed with the latter, such as ③.

- ① DIS: the flammable substance is beside the table
- ② DIS: beside the table in a306 laboratory
- ③ DIS: between the window and the table

**Table 6.** Temples of cascaded conditional random field (CCRF) features.

	Layer	Temples of Features
1	$W(n) (n = -2, -1, 0, 1, 2)$	$P(n) (n = -2, -1, 0, 1, 2)$
	$W(n-1)/W(n) (n = 0, 1)$	$P(n-1)/P(n) (n = -1, 0, 1, 2)$
	$P(n-2)/P(n-1)/P(n) (n = 0, 1, 2)$	
2	$W(n) (n = -2, -1, 0, 1, 2)$	$T(n) (n = -2, -1, 0, 1, 2)$
	$W(n-1)/W(n) (n = 0, 1)$	$T(n-1)/T(n) (n = -1, 0, 1, 2)$
	$T(n-2)/T(n-1)/T(n) (n = 0, 1, 2)$	
3	$W(n) (n = -2, -1, 0, 1, 2)$	$T(n) (n = -2, -1, 0, 1, 2)$
	$U(n) (n = -2, -1, 0, 1, 2)$	$W(n-1)/W(n) (n = 0, 1)$
	$T(n-1)/T(n) (n = -1, 0, 1, 2)$	$T(n-2)/T(n-1)/T(n) (n = 0, 1, 2)$
	$U(n-1)/U(n) (n = -1, 0, 1, 2)$	$U(n-2)/U(n-1)/U(n) (n = 0, 1, 2)$

To extract information easily, it is necessary to distinguish the target class and the class that modifies the target class from the classes in the chunk. Therefore, a support vector machine (SVM) is used to judge the relationship between classes. Because there are more than two relationships among classes, “one-against-one” strategy is utilized to solve the multi-classification problem. For example, a chunk is (Dangerous goods are on the first floor of the library). We need to judge the relationship between “dangerous goods” and “library”, the relationship between “dangerous goods” and “the first floor” and the relationship between “library” and “the first floor”.

Based on analyzing the relationships between classes in the chunks, five types of features are summarized and quantified, which is shown in Table 7.

**Table 7.** Features that are put into the support vector machine (SVM).

Features	Quantification	
	Ture	False
There is a word “of” between the two classes and there is not irrelevant adjective before “of” such as class “col”	1	0
The front class is followed by the class “dir”	1	0
The front class is followed by the conjunction such as “or” and “of”	1	0
The front class is followed by the words such as “in”, “lie” and “locate”	1	0
If the front class is followed by the class “dir” and “dir” is followed by word “of”	1	0

The modified relationships between the classes in the chunks are obtained, and the classes are rearranged based on the assumption that the former modifies the latter. Therefore, the target class can be extracted from the end of the chunk.

#### 4.1.4. Framework Design Based on Verbs

Because the motion of the integrated leg–arm hexapod robot is complex, which leads to complex speech instructions, it is difficult to extract key information from instructions. So, verbs in the instructions are classified by matching in the predefined action lexicon. To convert natural language into structural language, a semantic framework based on verbs is presented in this work.

After the instructions are segmented by CCRFs, it should be transformed into a framework to describe a task. Because verbs are key information of tasks, a rule based on verbs is designed in this work, to covert natural language into structural language. Verbs are divided into v1 (Verbs that should be executed in order) and v2 (Verbs that should be executed immediately), based on the meaning of

tasks. Furthermore, according to types of tasks, v1 (Verbs that should be executed in order) are divided into 18 types, and v2 (Verbs that should be executed immediately) are divided into six types, which are shown in Tables 8 and 9. Based on each type of verbs, a semantic framework for each type of verbs is designed.

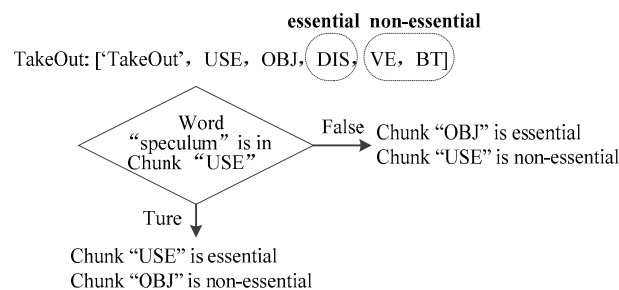
**Table 8.** Categories of v1 (verbs that should be executed in order).

Types	Examples	Types	Examples	Types	Examples
TakeOut	take out	MoveBack	retreat	PickUp	pick up
PutDown	lay down	Leave	leave	Keep	keep
Take	put in	Turn	turn	Loosen	loosen
PutUp	lift up	MoveIn	move in	Clamp	clamp
Cut	cut	TurnAround	turn round	Move	move
Observe	observe	TakeBack	take back	Stretch	stretch

**Table 9.** Categories of v2 (verbs that should be executed immediately).

Types	Examples	Types	Examples	Types	Examples
Change	change	IncreaseSpeed	accelerate	IncreaseSpeedTo	increase the speed to
Stop	stop	ReduceSpeed	reduce the speed	ReduceSpeedTo	reduce the speed to

Furthermore, chunks are divided into essential chunks and non-essential chunks. Essential chunks are necessary for tasks to perform, while non-essential chunks are not necessary for core tasks. In other words, if the framework lacks essential chunks, tasks cannot be performed, while if the framework lacks non-essential chunks, the core tasks can be still performed. Because there are many types of verbs, an example of the framework is given, which is shown in Figure 7.



**Figure 7.** Example semantic framework.

#### 4.2. Structure of Proposed Task Slot Based on Structural Language Framework

When a user wants the robot to go somewhere, "Go there" he/she may say, pointing in particular direction. This speech instruction lacks necessary direction information, meaning that the robot cannot perform the task. However, the deictic hand posture can supply the direction information. The information provided by speech and deictic hand posture is complementary under the circumstances. If the two kinds of information are used, a complete task can be formed. This section focuses on the cases, in which deictic hand postures can provide direction information, which speech commands lack.

To clarify demonstrative words in the vocal commands, the information on the deictic hand posture is added. Deictic words can be divided into two categories: adjectives and pronouns. Some chunks in the semantic results of the deictic hand postures are inserted into appropriate places in the semantic results of the vocal instructions to form a complete interactive task. To this end, a grammar rule based on POS is designed, and chunks in the semantic results of hand postures and speech are selected to fill the blanks in the integrated instruction using the structural language framework.

Figure 8 shows a flowchart for the selection and filling of the framework of structural language based on the grammatical rule of the POS.

Consider a demonstrative word  $T_p$  in the results of speech recognition. It is identified as the appropriate part of speech,  $T_{pw}$ . The appropriate strategy is then used to fill the task slot, divided into two kinds according to situation. If the part-of-speech of the demonstrative word, i.e.,  $T_{pw}$ , is an adjective, direction information is extracted from the semantic text of the hand posture and converted into an adjective chunk  $c_a$ . Moreover, the noun chunk  $c_{na}$ , which is qualified by  $T_p$ , is identified in semantic text of the voice commands. The position before that of  $c_{na}$  is filled by  $c_a$ , so that instructions of the deictic hand posture and speech are integrated into a complete instruction containing information on both. If the demonstrative word  $T_{pw}$  is a pronoun, the number of verbs  $num_v$  is counted in the semantic text of the speech. An instruction sometimes contains multiple verbs, the demonstrative word is generally related to the first verb in Chinese speech instructions, when deictic hand postures and speech are used. Therefore, it is assumed that the demonstrative word is related to the first verb in an instruction. Thus, if  $num_v$  is greater than one, the direction chunk  $c_d$  is extracted from the semantic text of deictic hand posture and inserted in the position before the second verb of the semantic text of voice command. Instructions of the deictic hand posture and speech are thus integrated into a synthesized instruction. If  $num_v$  is one, the end of the semantic text of the voice command is filled with the direction chunk of the semantic text of deictic hand posture.

Some examples are shown in ⑤⑥. In ⑤, a demonstrative word  $T_p$  in the results of speech recognition is “yonder”, and the part of speech of the demonstrative word  $T_{pw}$  is an adjective, then direction information “left” is extracted from the semantic text of the hand posture and converted into an adjective chunk  $c_a$  [left, dir]. Next, the adjective chunk  $c_a$  [left, dir] is inserted into the position before the noun chunk  $C_{na}$ , i.e., [‘wire’, ‘obj’]. So, a synthesized instruction integrating hand postures with speech is formed in this way, which is [[‘Cut’, 0, [[[[‘left’, dir], [‘wire’, ‘obj’]]]], 0]]. In ⑥, a demonstrative word  $T_p$  in the results of speech recognition is “that”, and the part-of-speech of the demonstrative word  $T_{pw}$  is a pronoun, and the direction chunk  $C_d$  [[[[‘towards’, ‘pre’], [‘right’, ‘dir’]]]] is extracted from the semantic text of the deictic hand posture, then it was inserted into the bottom of the semantic result of speech recognition ([[‘Observe’, 0, [[[[‘flames’, ‘obj’]]]], 0]]). So, a semantic result of posture-speech combination is formed, i.e., [[‘Observe’, 0, [[[[‘flames’, ‘obj’]]]], 0]] [[[[‘towards’, ‘pre’], [‘right’, ‘dir’]]]].

⑤ A demonstrative word  $T_p$  “yonder” is an adjective

A result of hand posture recognition: Turn left

A semantic result of the hand posture recognition: [[‘Turn’, [[[[‘towards’, ‘pre’], [‘left’, ‘dir’]]]], 0, 0]]

A result of speech recognition: Cut the wire over there

A semantic result of the speech recognition: [[‘Cut’, 0, [[[[‘wire’, ‘obj’]]]], 0]]

A semantic result of gesture–speech combo: [[‘Cut’, 0, [[[[‘left’, dir], [‘wire’, ‘obj’]]]], 0]]

⑥ A demonstrative word  $T_p$  “that” is a pronoun

A result of hand posture recognition: Turn right

A semantic result of the hand posture recognition: [[‘Turn’, [[[[‘towards’, ‘pre’], [‘right’, ‘dir’]]]], 0, 0]]

A result of speech recognition: Check for flames over there

A semantic result of the speech recognition: [[‘Observe’, 0, [[[[‘flames’, ‘obj’]]]], 0]]

A semantic result of gesture–speech combo: [[‘Observe’, 0, [[[[‘flames’, ‘obj’]]]], 0]] [[[[‘towards’, ‘pre’], [‘right’, ‘dir’]]]]

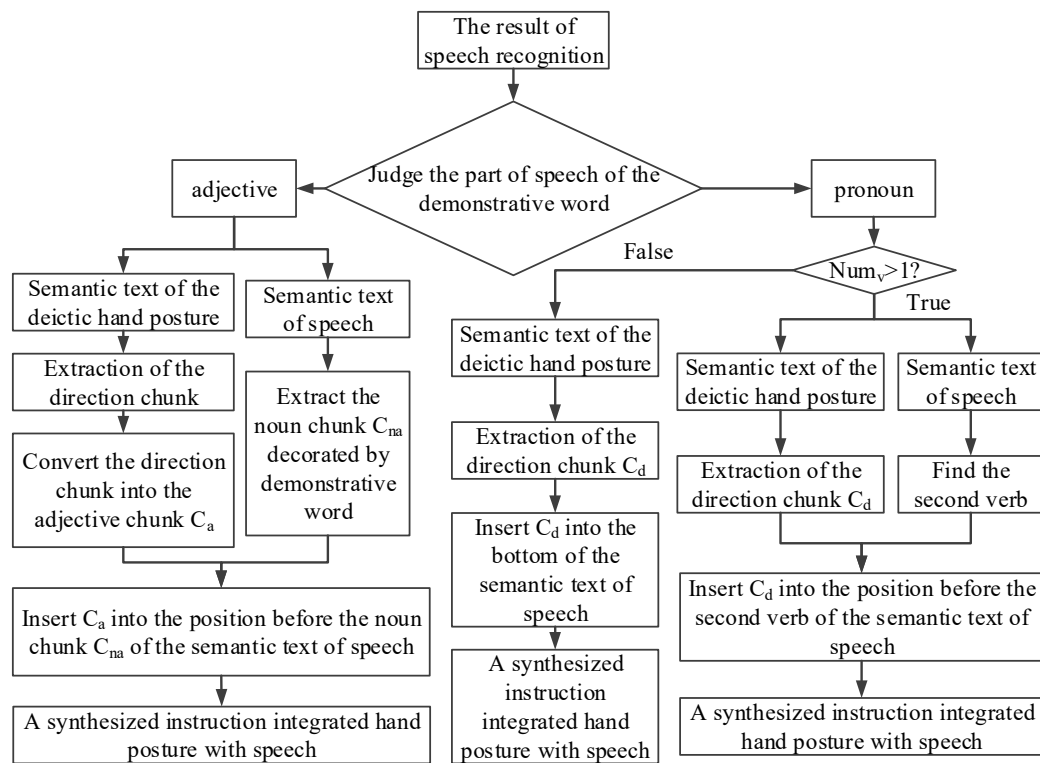


Figure 8. Flowchart for the integration of hand postures and speech based on semantic understanding.

## 5. Results of Experiments on Human–Robot Interaction

### 5.1. Experiments on Hand Posture Interaction

Twenty-five hand type postures were designed to enable the robot to perform reconnaissance and manipulation tasks. Maps were designed based on the movement/manipulation of our hexapod robot and the user's intention and the hand postures.

A dataset was constructed to evaluate the performance of the proposed method of hand posture recognition. It consisted of 7500 images in 25 classes of postures. The dataset featured three scenes: a conference room, a laboratory, and a corridor. A total of 2500 hand postures were captured for each scene, and featured the use of both the left and right hands. There were 100 hand postures in each scene, and each type of hand posture was different with respect to its position, rotation, and distance between the camera and the gesturing hand. The postures are captured by a laptop camera, which was used to remotely operate the robot. The size of each image in our dataset was  $1290 \times 720$  pixels. We designed our hand posture recognition system from a human-centric perspective. When the user interacted with the robot through hand postures, he/she looked at the camera and could see the captured image. Some of the images from our dataset are shown in Figure 9.

CornerNet-Squeeze was used to detect and recognize the hand postures. Thirty images of each type of posture made using the left and right hands for each scene were used to train the model, and 10 images of each type of posture for each hand were used to evaluate it. The remaining 10 images for each of the left and right hands were used to test the trained model. Hence, 4500 images were used for training, and 1500 images were used to evaluate the model, and 1500 images were used to test it.

Experiments are performed on a workstation (Precision 7920 Tower Workstation produced by Dell Inc.). The processor of the workstation is Intel(R) Xeon(R) Gold 6254\*2, and the graphic card of the workstation is NVIDIA TITAN RTX\*2. Moreover, Python 3.7.1 is utilized on the workstation. Some results of hand posture recognition are shown in Figures 10–16. As shown in Figures 10–16, the locations of hand are shown using the bounding boxes, and the categories of hand postures are listed on the top of the bounding boxes. Furthermore, an image in the test dataset is incorrectly classified,



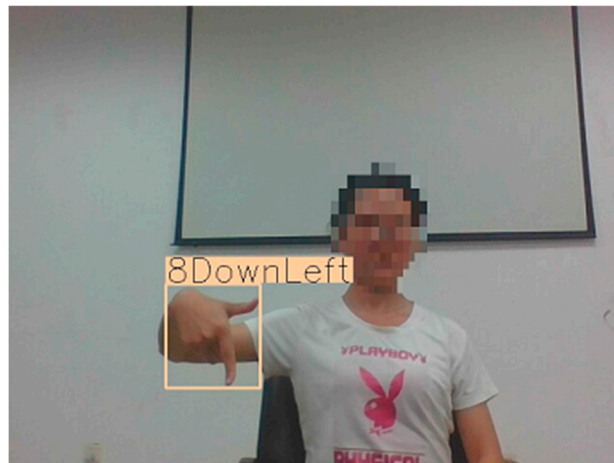
which is shown in Figure 15a. In addition, the performance of the hand posture recognition in the test dataset is shown in Table 10. The method of CornerNet-Squeeze has good performance, and the average accuracies of hand posture recognition is 99.9%. The results demonstrate the effectiveness of the method of hand posture recognition.



**Figure 9.** Sample images from our dataset. (a–e) are hand posture images with short sleeves captured in a conference room; (f–j) are hand posture images with short sleeves captured in a lab; (k–o) are hand posture images with long sleeves captured in a conference room; (p–t) are hand posture images with long sleeves captured in a lab; (u–y) are hand posture images captured in a corridor.



**Figure 10.** Result of recognition of hand postures in the conference room.



**Figure 11.** Result of recognition of hand postures in the conference room.



**Figure 12.** Result of recognition of hand postures in the lab.



**Figure 13.** Result of recognition of hand postures in the lab.



**Figure 14.** Result of recognition of hand postures in the corridor.



**Figure 15.** Example of comparison when the system correctly classifies the hand posture and when it fails. (a) The hand posture is incorrectly classified; (b) the hand posture is correctly classified.



**Figure 16.** Results of hand posture recognition. (a–m) are hand posture images which are used to control movement/manipulation of the robot.



**Table 10.** Confusion matrix for twenty-five hand postures using our dataset.

Types	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Accuracy (%)	98	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

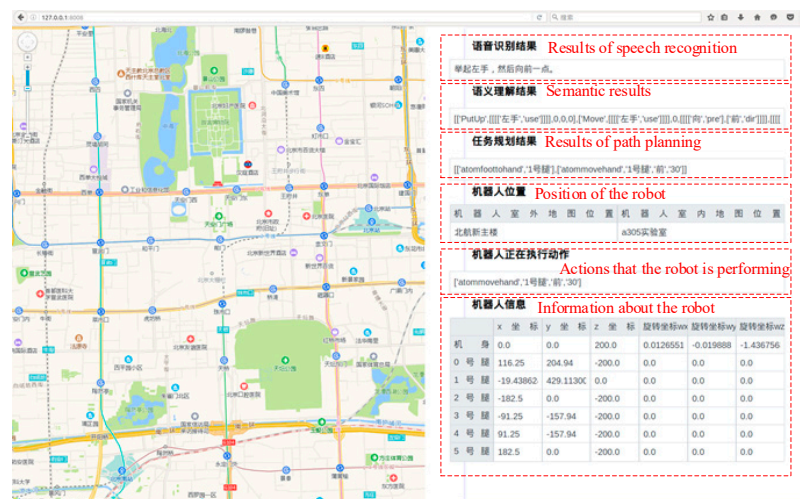
A speculum was installed on the leg–arm hexapod robot, which can assist public security personnel in reconnoitering. The task of reconnaissance was divided into a series of subtasks, and hand postures were used to control the robot to perform the subtasks. Furthermore, the results of recognition of hand postures for the subtasks are shown in Figure 16.

## 5.2. Speech-Based Interaction Experiments

By combining speech recognition, semantic understanding, task planning, network information interaction, and remote data interaction, a speech-based interaction system was designed to remotely control the leg–arm hexapod robot. An experiment verified the validity of the semantic understanding algorithm, task planning method, system design, reliability of communication, and coordination among different modules.

We focus on reconnaissance and rescue tasks, and designed several Chinese instructions based on structural and functional characteristics of our robot. Tasks that the robot can execute are divided into four categories: motion of the whole robot, leg/arm/body/motion, object detection and autonomous navigation indoors. Twenty instructions are used in each category, and each instruction is repeated five times, so, there are four hundred experiments in total. Some Chinese instructions are shown in Table 11. For the sake of reader's convenience, each Chinese instruction is translated from Chinese into English, and the English instruction is listed in parentheses below the corresponding Chinese instruction. Furthermore, sometimes the robot is required to scout around for specific dangerous goods to achieve reconnaissance and rescue tasks, so some instructions are designed, such as (check for dangerous goods), (check conditions indoor). The commands are given by a user through the host computer, and the accuracy of different types of speech instructions is shown in Table 12. Moreover, the average accuracy of the speech-based interaction experiments is 88.75%. Because the speech-based interaction utilizes the voice dictation API (Application Programming Interface) provided by Iflytek Co., Ltd to recognize speech, and if results of voice dictation are wrong, the following processing procedure is directly affected. Moreover, noise influences accuracies of speech recognition in real application scenarios. If speech is not correctly recognized, commands cannot be parsed.

When the robot performs a task, which is given by a speech instruction, which is (Raise your left hand, then move forward a little), Figure 17 shows the movement, state, and position of the robot and the results of speech recognition, semantic understanding and task planning of the robot.

**Figure 17.** Graphical user interface of host computer.

**Table 11.** Results of speech control.

Type of Task	Instruction	Number of Experiments	Accuracy (%)
Motion of the whole robot	Move forward 5 m quickly	5	100
	Turn left, and move forward two steps.	5	100
	Rotate 60 degrees clockwise; then move back half a meter	5	100
	Move 3 m southeast at a speed of 100 millimeters per second	4	80
	Turn around	5	100
Leg/arm/body/motion	Lift up leg No. 5	5	100
	Move forward a little	5	100
	The body of the robot rotates 5 degrees clockwise, and then moves forward a little	5	100
	Lift the left hand and hold for 5 s; then put it down	5	100
	Lift the left hand; then move forward a little	5	100
Object detection	Check for dangerous goods	5	100
	Check for flames indoor	5	100
	Check conditions indoor	5	100
Autonomous navigation indoors	Go to the elevator of the new main building of Beihang University.	5	100
	Quickly check Room a506 for dangerous goods quickly	4	80

**Table 12.** Average accuracies of different types of speech instructions.

Types	Motion of the Whole Robot	Leg/Arm/Body/Motion	Object Detection	Autonomous Navigation Indoors
Accuracy (%)	88	85	96	86

### 5.3. Human–Robot Interaction Using Hand Postures and Speech

Experiments were conducted to evaluate the effectiveness of the proposed semantic understanding-based task slot structure through visual and auditory channels. Some results are shown in Figures 18 and 19. Figures 18 and 19 show the results of hand posture recognition and speech recognition, and corresponding semantic result of hand posture recognition, speech recognition and gesture–speech combination. To make it understand easily, each Chinese instruction has been translated from Chinese into English, and it was listed below the corresponding Chinese instruction.

Speech commands used in the gesture–speech combination can be divided into two categories, and five speech commands are given in each category, and each speech command is combined with twelve images of hand postures. These images are captured in a conference room, in a lab, and a corridor, respectively. There are four types of hand posture images in each scene, and each type of hand posture was different with respect to its position, rotation, and distance between the camera and the gesturing hand. Some examples are shown in Figure 20. The experimental results are shown in Table 13. The average accuracy is 83.3%. The proposed method is based on hand posture and speech recognition. Once the hand posture or speech is not recognized correctly, the following process goes wrong.

Because the proposed method is influenced by hand posture and speech recognition, to obtain results that suggest the proposed method is affected by hand posture recognition, we assumed that all speech instructions are recognized correctly. In this way, the factor of speech recognition is excluded from factors which influence the proposed method. Thus, the experiments that interact using hand posture and the text of speech instruction are designed. The difference between the experiment using voice commands and hand postures and the experiment using hand postures and the text of speech instructions is the input method of speech instructions; specifically, the former is a voice command, while the latter is the text of a speech instruction, and the others are the same. The result of the experiment using hand posture and the text of speech instructions is shown in Table 14. The average accuracy is 98.3%. Furthermore, the confusion matrix of deictic hand postures is shown in Table 15. Because the accuracy of the proposed method of gestures–speech combination is greatly affected by accuracies of hand posture recognition and speech recognition, if some images of hand postures



are not detected, as shown in Figures 21a and 22a, incorrect results of gesture–speech combination are obtained.

Furthermore, to show the results of the integration of deictic hand postures and speech more intuitively, a graphical user interface based on Qt was designed. Moreover, additional images are captured in another scene which is not in the training data set. Some results are shown in Figures 23–26. In addition, if hand postures are incorrectly classified, deictic hand postures and speech cannot be correctly integrated, as shown in Figures 27–30, perhaps because the training dataset does not include images in this scene, which are shown in Figures 27–30. Moreover, appearances of the same type of hand postures vary greatly, and appearances of different kinds of hand postures are similar. Specifically, although hand postures in Figure 27a,b are the same type of hand postures, they differ in appearances greatly. However, the appearances of hand postures in Figures 27a and 28b are similar.

**Instruction 1:**  
 The result of hand posture recognition:  
 Turn left  
 The semantic result of hand posture recognition:  
 [['Turn',[[['towards', 'pre'],['left','dir']]],0,0]]  
 The result of speech recognition:  
 Go there, and cut the electric wires.  
 The semantic result of speech recognition:  
 [['Move',0,0,0,0],['Cut',0,[[['electric wires', 'obj']]],0]]  
 The semantic result of the integration of hand posture and speech:  
 [['Move',0,0,0,0],[[['towards', 'pre'],['left','dir']]],['Cut',0,[[['electric wires', 'obj']]],0]]  
**Instruction 2:**  
 The result of hand posture recognition:  
 Turn right  
 The semantic result of hand posture recognition:  
 [['Turn',[[['towards', 'pre'],['right','dir']]],0,0]]  
 The result of speech recognition:  
 Check if there are any dangerous goods over there quickly.  
 The semantic result of speech recognition:  
 [['Observe',0,[[['dangerous goods', 'obj']]],[[['quickly','ve']]]]]  
 The semantic result of the integration of hand posture and speech:  
 [['Observe',0,[[['dangerous goods', 'obj']]],[[['quickly','ve']]]],[[['towards', 'pre'],['right','dir']]]]

**Figure 18.** Results of the integration of deictic hand posture and speech instructions when the demonstrative word was a pronoun.

**Instruction 1:**  
 The result of hand posture recognition:  
 Turn left  
 The semantic result of hand posture recognition:  
 [['Turn',[[['towards', 'pre'],['left','dir']]],0,0]]  
 The result of speech recognition:  
 Quickly drop the package over there into the explosion-proof tank.  
 The semantic result of speech recognition:  
 [['Take',0,[[['package', 'obj']]],[[['explosion-proof tank', 'obj'],['into','dir']]],[[['quickly','ve']]],0]]  
 The semantic result of the integration of hand posture and speech:  
 [['Take',0,[[['left, dir'],['package', 'obj']]],[[['explosion-proof tank', 'obj'],['into','dir']]],[[['quickly','ve']]],0]]  
**Instruction 2:**  
 The result of hand posture recognition:  
 Turn right  
 The semantic result of hand posture recognition:  
 [['Turn',[[['towards', 'pre'],['right','dir']]],0,0]]  
 The result of speech recognition:  
 Put the speculum into the crevice of the door slowly.  
 The semantic result of speech recognition:  
 [['Stretch',[[['speculum', 'use']]],[[['crevice of the door', 'obj'],['中','dir']]],[[['slowly','ve']]]]]  
 The semantic result of the integration of hand posture and speech:  
 [['Stretch',[[['speculum', 'use']]],[[['right, dir'],['crevice of the door', 'obj'],['into','dir']]],[[['slowly','ve']]]]]

**Figure 19.** Results of the integration of deictic hand posture and spoken instructions when the demonstrative word was an adjective.



**Figure 20.** Sample images used in gesture–speech combination.

**Table 13.** Experimental results using voice commands and hand postures.

POS of the Demonstrative Word	Speech Instructions	No. of Experiments	Accuracy of Gesture–Speech Results (%)
pronoun	Go over there quickly	12	91.7
	Go there	12	100.0
	Check whether there are dangerous goods over there quickly	12	75.0
	Check for flames over there	12	91.7
	Move over there	12	83.3
adjective	Cut the wire over there	12	50.0
	Put the speculum into the crevice of the door over there slowly	12	41.7
	Examine the circumstance over there	12	100.0
	Check the situation over there	12	100.0
	Go to the elevator over there	12	100.0

**Table 14.** Experimental results using hand postures and the text of voice commands.

POS of the Demonstrative Word	Instructions	Accuracy of Posture Recognition (%)	Accuracy of Gesture–Speech Combination Results (%)
pronoun	Go over there quickly	91.7	91.7
	Go there	100.0	100.0
	Check whether there are dangerous goods over there quickly	100.0	100.0
	Check for flames over there	100.0	100.0
	Move over there	91.7	91.7
adjective	Cut the wire over there	100.0	100.0
	Put the speculum into the crevice of the door over there slowly	100.0	100.0
	Examine the circumstance over there	100.0	100.0
	Check the situation over there	100.0	100.0
	Go to the elevator over there	100.0	100.0

**Table 15.** A confusion matrix of deictic hand postures.

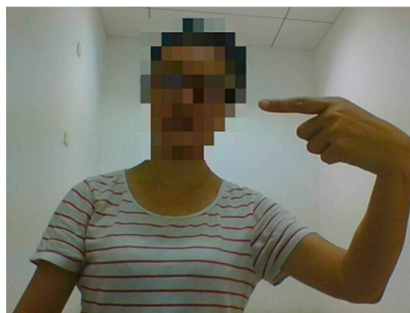
Hand Postures	Prediction				Accuracy (%)
	1Left	1Right	PalmLeft	PalmRight	
1Left	30	0	0	0	100.0
1Right	0	28	0	0	93.3
PalmLeft	0	0	30	0	100.0
PalmRight	30	30	0	30	100.0



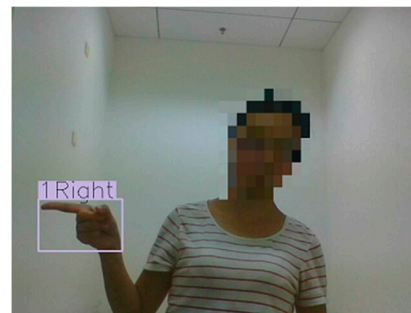
(a)



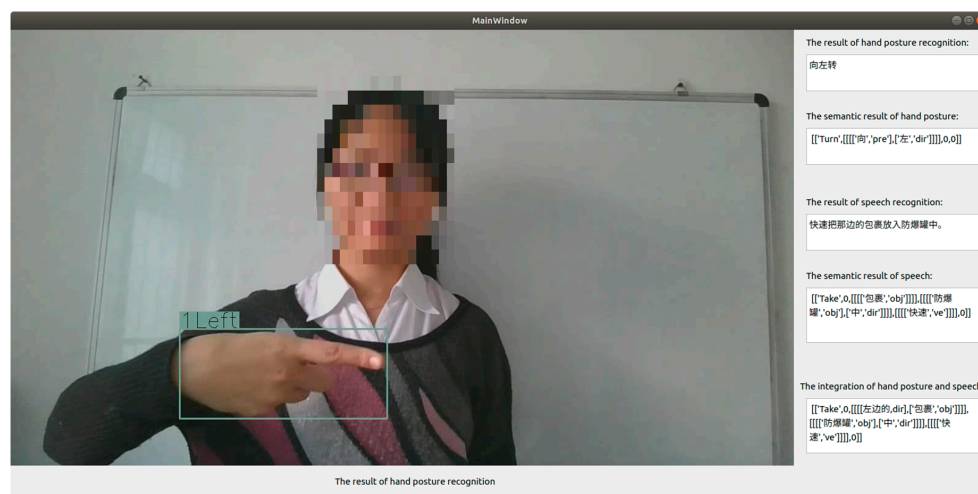
(b)

**Figure 21.** Example of comparison when the system correctly classifies the hand posture and when it fails. (a) The hand posture is not detected and recognized; (b) the hand posture is correctly classified.

(a)



(b)

**Figure 22.** Example of comparison when the system correctly classifies the hand posture and when it fails. (a) The hand posture is not detected and recognized; (b) the hand posture is correctly classified.**Figure 23.** Results of the integration of deictic hand posture and voice commands.

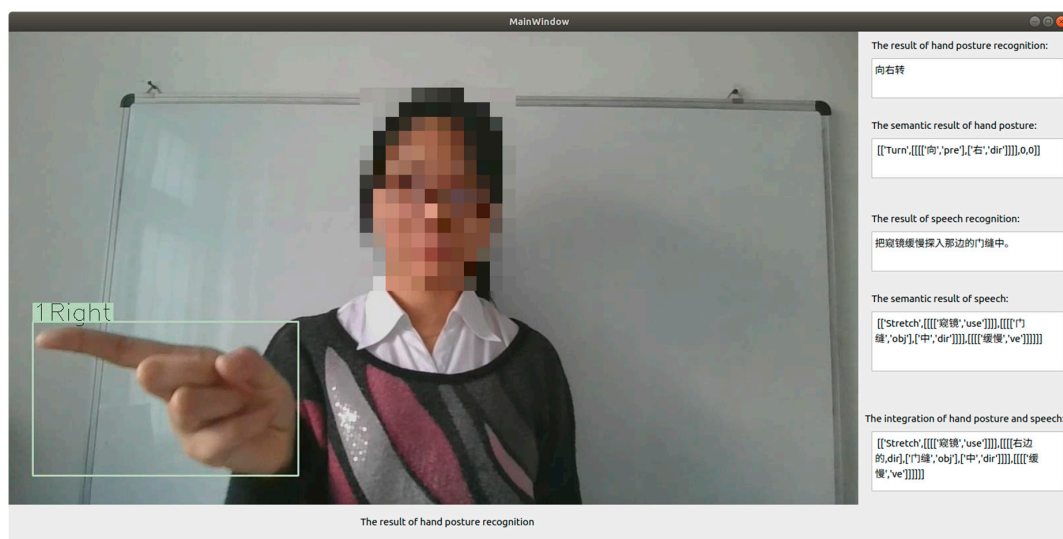


Figure 24. Results of the integration of deictic hand posture and spoken instructions.

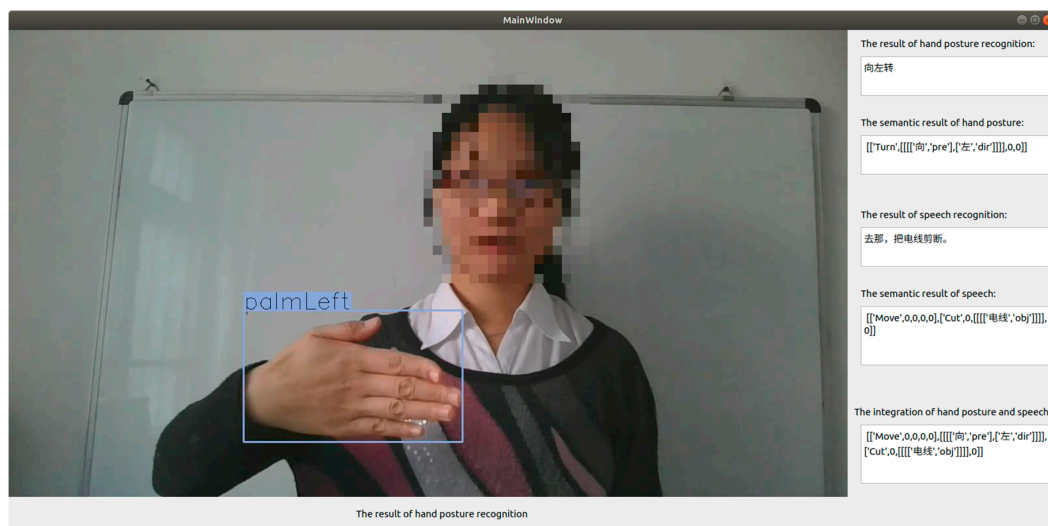


Figure 25. Results of the integration of deictic hand posture and voice commands.

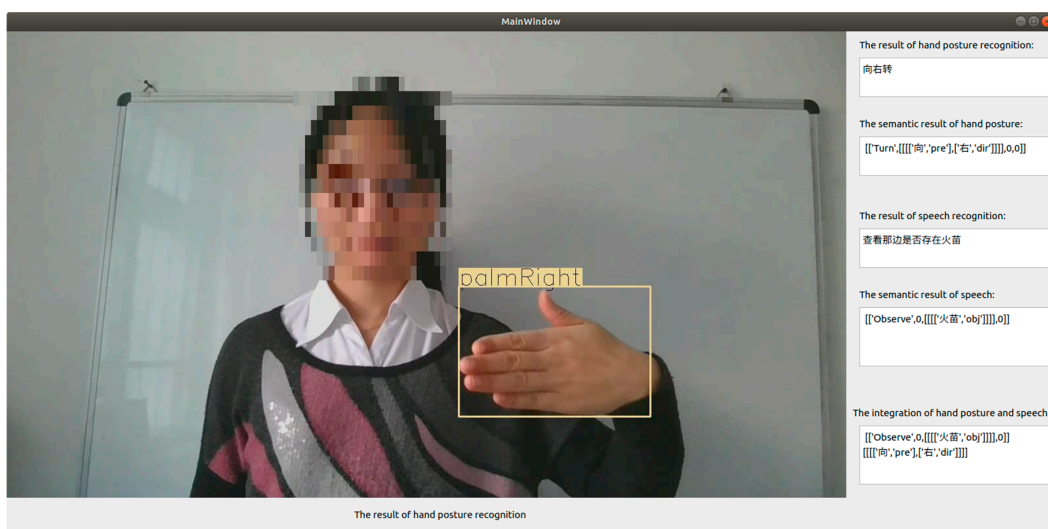
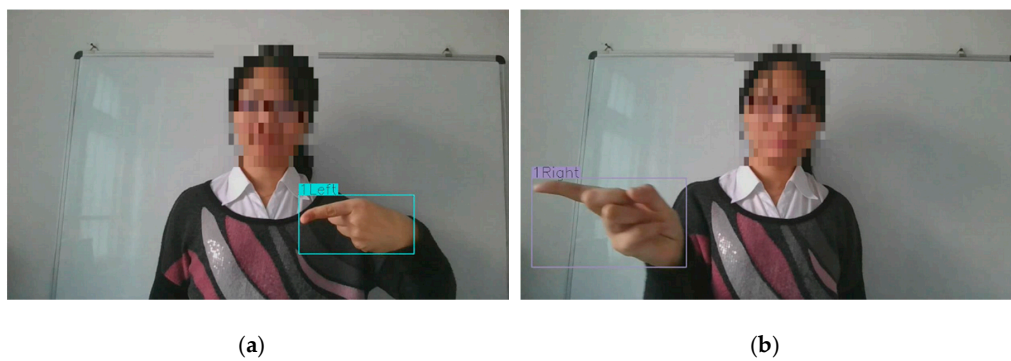
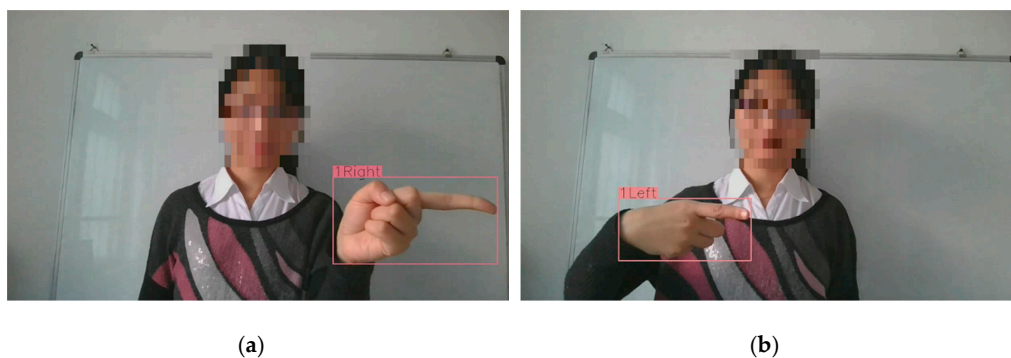


Figure 26. Results of the integration of deictic hand posture and spoken instructions.

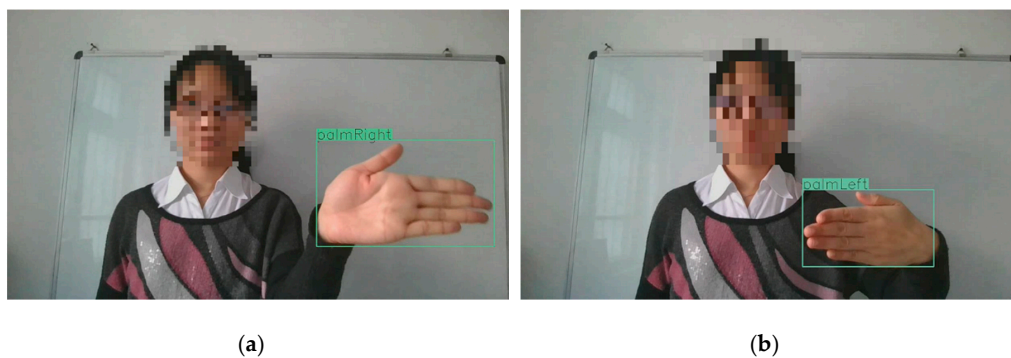




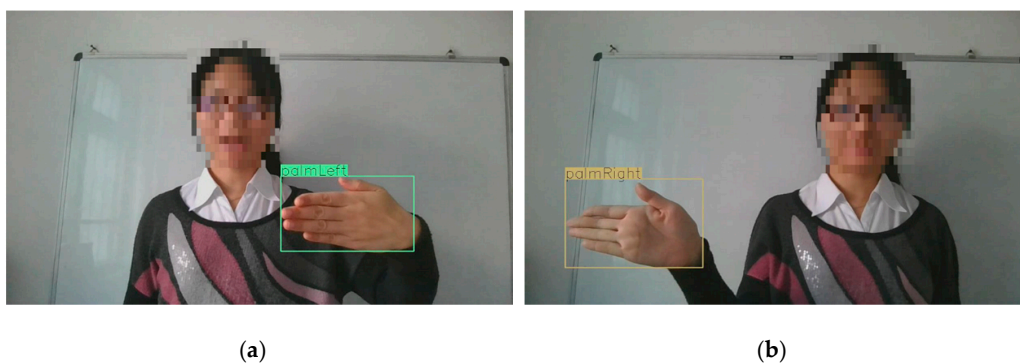
**Figure 27.** Example of comparison when the system correctly classifies the hand posture and when it fails. (a) The hand posture is incorrectly classified; (b) the hand posture is correctly classified.



**Figure 28.** Example of comparison when the system correctly classifies the hand posture and when it fails. (a) The hand posture is incorrectly classified; (b) the hand posture is correctly classified.



**Figure 29.** Example of comparison when the system correctly classifies the hand posture and when it fails. (a) The hand posture is incorrectly classified; (b) the hand posture is correctly classified.



**Figure 30.** Example of comparison when the system correctly classifies the hand posture and when it fails. (a) The hand posture is incorrectly classified; (b) the hand posture is correctly classified.



## 6. Conclusions

To perform certain particular tasks related to reconnaissance, rescue, and counterterrorism, a system of interaction was designed in this study for a leg–arm hexapod robot. It consisted of hand posture-based, speech-based interaction, and an integration of the two.

CornerNet-Squeeze was used to identify hand postures. Certain types of hand posture were designed and a dataset was created based on the requirements of specific tasks and characteristics of our robot. To ease the memory-related burden on the user, only a few hand postures were designed. A mapping from the hand posture to the corresponding movement/manipulation of our robot and one from the hand posture to the user's intention were predefined. CornerNet-Squeeze was then used to train our model to recognize the hand postures, and enabled non-expert users to interact naturally with the robot during reconnaissance and rescue tasks.

In the combination of hand posture and speech modes, deictic hand postures and voice commands were used simultaneously to improve the efficiency of interaction, but the demonstrative words used in the voice commands were ambiguous. To correctly understand the user's intention, the directional information of the deictic hand postures was used to complement the information conveyed by the demonstrative words, and a grammatical rule based on the types of words used was designed. A semantic understanding-based task slot structure using the visual and auditory channels was thus proposed. This structure was based on an expansion of the structural language framework. The results of experiments proved the effectiveness of the proposed method.

**Author Contributions:** K.X. and X.D. have conceived the project. K.X. revised the manuscript and acquired funding. Z.H. provided requirements of special robots for reconnaissance and rescue. W.L. performed the speech module. J.Q. wrote the manuscript and performed hand posture recognition and the integration of hand posture and speech. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by the Natural Science Foundation of Beijing Municipality (3192017) and the National Natural Science Foundation of China (Grant No. 51775011 and Grant No. 91748201).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Pavlovic, V.; Sharma, R.; Huang, T. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 677–695. [\[CrossRef\]](#)
2. Badi, H.S.; Hussein, S. Hand posture and gesture recognition technology. *Neural Comput. Appl.* **2014**, *25*, 871–878. [\[CrossRef\]](#)
3. Chang, C.C.; Chen, J.J.; Tai, W.K.; Han, C.C. New approach for static gesture recognition. *J. Inf. Sci. Eng.* **2006**, *22*, 1047–1057.
4. Stiefelhagen, R.; Fogen, C.; Giesemann, P.; Holzapfel, H.; Nickel, K.; Waibel, A. Natural human-robot interaction using speech, headNatural human-robot interaction using speech, head pose and gestures. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004.
5. Stiefelhagen, R.; Ekenel, H.K.; Fügen, C.; Giesemann, P.; Holzapfel, H.; Kraft, F.; Nickel, K.; Voit, M.; Waibel, A.; Ekenel, H.K. Enabling multimodal human–robot interaction for the karlsruhe humanoid robot. *IEEE Trans. Robot.* **2007**, *23*, 840–851. [\[CrossRef\]](#)
6. Nickel, K.; Stiefelhagen, R. Visual recognition of pointing gestures for human–robot interaction. *Image Vis. Comput.* **2007**, *25*, 1875–1884. [\[CrossRef\]](#)
7. Nickel, K.; Stiefelhagen, R. Real-time person tracking and pointing gesture recognition for human-robot interaction. *Comput. Vision Hum. Comput. Interact.* **2004**, *3058*, 28–38.
8. Nickel, K.; Stiefelhagen, R. Pointing gesture recognition based on 3D-tracking of face, hands and head orientation. In Proceedings of the 5th International Conference on Multimodal Interfaces, Vancouver, BC, Canada, 5–7 November 2003.

9. Seemann, E.; Nickel, K.; Stiefelhausen, R. Head pose estimation using stereo vision for human-robot interaction. In Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea, 19 May 2004.
10. Burger, B.; Ferrané, I.; Lerasle, F.; Infantes, G. Two-handed gesture recognition and fusion with speech to command a robot. *Auton. Robot.* **2011**, *32*, 129–147. [[CrossRef](#)]
11. Liu, H.; Fang, T.; Zhou, T.; Wang, Y.; Wang, L. Deep learning-based multimodal control interface for human-robot collaboration. *Procedia CIRP* **2018**, *72*, 3–8. [[CrossRef](#)]
12. Ghidary, S.S.; Nakata, Y.; Saito, H.; Hattori, M.; Takamori, T. Multi-modal interaction of human and home robot in the context of room map generation. *Auton. Robot.* **2002**, *13*, 169–184. [[CrossRef](#)]
13. Rogalla, O.; Ehrenmann, M.; Zollner, R.; Becher, R.; Dillmann, R. Using gesture and speech control for commanding a robot assistant. In Proceedings of the 11th IEEE International Workshop on Robot and Human Interactive Communication, Berlin, Germany, 27 September 2002.
14. Van Delden, S.; Umrysh, M.A.; Rosario, C.; Hess, G. Pick-and-place application development using voice and visual commands. *Ind. Robot. Int. J.* **2012**, *39*, 592–600. [[CrossRef](#)]
15. Van Delden, S.; Umrysh, M.A. Visual detection of objects in a robotic work area using hand gestures. In Proceedings of the 2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Montreal, QC, Canada, 17–18 September 2011.
16. Chen, X.; Qin, S. Approach to high efficient hierarchical pathfinding of indoor mobile service robots based on grid map and Floyd-Warshall algorithm. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017.
17. Chen, X.; Qin, S. An effective approach to SLAM toward autonomous operation for a Leg/Arm composite mobile robot in unknown environment based on RGB-D images. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017.
18. Li, W.; Xu, K.; Qi, J.; Ding, X. A Natural Language Processing Method of Chinese Instruction for Multi-legged Manipulating Robot. In Proceedings of the 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 12–15 December 2018.
19. Law, H.; Teng, Y.; Russakovsky, O.; Deng, J. CornerNet-Lite: Efficient Keypoint Based Object Detection. *arXiv* **2019**, arXiv:1904.08900.
20. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, 28 June–1 July 2001.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).