A Human–Robot-Environment Interactive Reasoning Mechanism for Object Sorting Robot

Yunhan Lin, Huasong Min, Haotian Zhou, and Feilong Pei

***Abstract*—In this paper, we design an object sorting robot system, which is based on robot operating system distributed processing framework. This system can communicate with human beings; can percept the 3-D environment by Kinect sensor; has the ability of reasoning; can transfer the natural language intention to machine instruction to control the movement of manipulator. In particular, in order to improve the intelligence and usability of our robot, we propose a human–robot-environment interactive reasoning mechanism. In our method, a "dialogue and 3-D scene interaction module" is added into the traditional case-based reasoning–belief-desire-intention mechanism. Our proposed mechanism not only realizes the traditional function of map matching but also achieves the function of desire analysis and guidance. When the user's desire is incomplete and/or mismatched with the actual scene, our robot will take the initiative to guide users through dialogue, and the user's input information will be used to replenish the user's desire. Experimental results prove the advantages of our mechanism.**

***Index Terms*—Belief-desire-intention (BDI) mechanism, case-based reasoning (CBR) mechanism, human–robot-environment interaction, object sorting robot, reasoning mechanism.**

I. Introduction

MANY new kinds of intelligent service robot systems are emerging in these days. These systems have a common characteristic that the robot could collaborate with humans to accomplish tasks in an unstructured environment with a flexible manipulator. Such a system needs to be capable of environmental perception, human–robot interaction (HRI), target recognition, knowledge reasoning, trajectory planning and grabbing, etc. Among the requirements, the reasoning mechanism is the core of the entire system, the brain of an intelligent robot that determines the degree of that intelligence.

The classic reasoning mechanisms include rule-based reasoning (RBR), procedural reasoning system, knowledge-based reasoning, and case-based reasoning (CBR). Among them, the research on CBR mechanism has become a hot topic in the field of artificial intelligence because of its advantages in short reasoning time and easy case learning ability. Some works applied CBR to HRI [1], [2] and multirobot coordinated navigation [3]. Our previous work [4] and [5], applied CBR to the real-time autonomous navigation and obstacle avoidance of a mobile robot. The robot can reuse previous experience to obtain suitable information to solve the current problems by using CBR technology, which gives the robot a certain degree of intelligence. However, it does not have the capacity of real-time interaction with the external environment, while this ability is an important indicator of robot's intelligence.

In order to improve the intelligence of the CBR mechanism, Olivia *et al.* [6] proposed CBR-belief-desire-intention (CBR–BDI) technology, which combined the CBR mechanism with a belief-desire-intention (BDI) model. The BDI model is a kind of deliberative agent. Belief means the information about the environment and the internal state the agent may hold. Desire means the original motivation and their potential preferred behavior to act. Intention means a set of actual executable action sequence, which is a detailed executing procedure of the belief. The combination of CBR with BDI addressed the BDI model's shortcoming of inability to learn; adding BDI in CBR also increased the autonomous interactive ability of the CBR system.

The mechanism of CBR–BDI was used in various intelligent systems these years. In 2012, Bajo *et al.* [7] used CBR–BDI as a decision support tool in a multiagent system, which can contribute to detect potential risky situations and to avoid them by acting on the tasks that compose each of the activities of the business. In 2013, Dalal *et al.* [8] applied CBR–BDI to imitate the entities of supply chain system, this system helps managers to analyze the business policies with respect to different situations arising in the supply chain. In 2014, Fraile *et al.* [9] used CBR–BDI in their planning home care system, which is capable of reacting automatically when faced with dangerous or emergency situations, replanning any plans in progress and sending alert messages to the system. The use of CBR–BDI in these systems enables the reuse of previous experience to obtain suitable information to solve current problems and also enables the reply to events, interactions with the environment, taking the initiative according to goals. However, those systems were based on the hypothesis that the user's utterance is complete and fluent; otherwise, human's commands will not

Manuscript received December 24, 2016; revised March 15, 2017 and May 8, 2017; accepted May 16, 2017. Date of publication May 23, 2017; date of current version September 7, 2018. This work was supported by the National Natural Science Foundation of China under Grant 61673304. *(Corresponding author: Huasong Min.)*

The authors are with the Institute of Robotics and Intelligent Systems, Wuhan University of Science and Technology, Wuhan 430081, China (e-mail: yhlin@ustc.edu; mhuasong@wust.edu.cn; zhtwust@163.com; 576467670@qq.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2017.2706975

be recognized or only partly recognized. It will lead to wrong instruction recognition and execution failed.

To address these shortcomings, many studies have been carried out to solve the incomplete issue. Ros *et al.* [10] proposed a set of strategies that allow a robot to identify the referent when human refers to an object giving incomplete information. This system uses an ontology to store and reason on the robot's knowledge to ease clarification. Wang and Zhang [11] proposed a multicriteria decision-making method to handle multicriteria fuzzy decision-making problems in which the information about criteria's weights is incomplete. Banerjee and Dubois [12] proposed a logic for reasoning about incomplete knowledge. However, all the reasoning mechanisms of these systems are based on RBR. RBR has some obvious disadvantages [13]:

1) the rules which acquiring from expert interviews is cumbersome and time-consuming;
2) the maintenance of rule bases becomes a difficult process as the size of the rule base increases;
3) a rule-based system is not self updatable, in the sense that there is no inherent mechanism to incorporate experience acquired from dealing with past problems.

Considering that the system we designed is an object sorting system for nonprofessional users. There are two problems in the process of human–robot-environment interactive reasoning:

1) there is a situation that the user's utterance is incomplete or not fluent. if the user's utterance is not fluent (pause time is too long), only part of the utterance can be recognized;
2) the utterance styles of the nonprofessional users are not unified, there is a situation that the presence of the same expression can be similarly expressed by different users, and there is uncertainty in this similar expression.

If RBR is used, there will be problems, such as reasoning rules are difficult to obtain, solutions are not easy to generate and the system is difficult to maintain. However, CBR can effectively achieve the correct reasoning of different similar cases, which uses past similar experience in the case database to solve current problem. Unlike RBR, CBR only need to design some basic cases in the casebase.

In order to solve the two problems that may arise in the process of human–robot-environment interaction and reasoning, and to further improve the level of robot's intelligence, we proposed a human–robot-environment interactive reasoning mechanism. In our mechanism, a "dialogue and 3-D scene interaction module" is added into the traditional CBR–BDI reasoning mechanism to implement the correct interaction and reasoning of the situations when the user's desire is incomplete and/or mismatched with the actual scene. Our mechanism is based on CBR–BDI mechanism, which can reuse previous experiences to solve current similar problems, can reply to events, interact with the environment and take the initiative according to goals.

The rest of this paper is organized as follows. In Section II, we introduce the hardware and software structure of our designed object sorting system. Section III introduces the traditional CBR–BDI reasoning mechanism and Section IV explains our human–robot-environment interactive reasoning mechanism in detail. Experiment and analysis are discussed in Section V. Finally, in Section VI, we state the conclusion and future work.

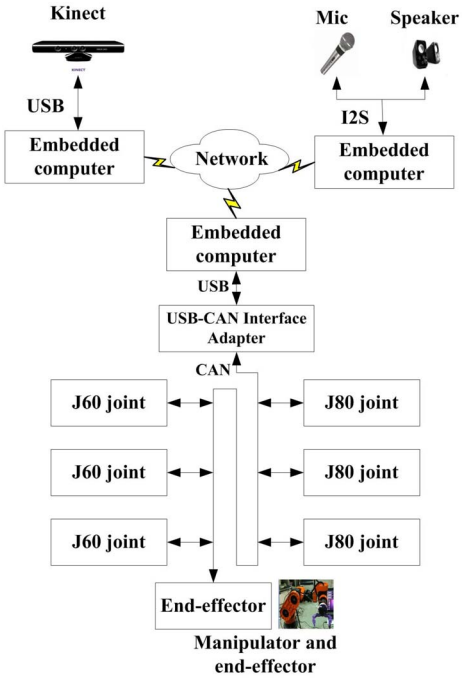


Fig. 1. Hardware structure of our designed object sorting robot system.

II. Hardware and Software Structure of Object Sorting Robot System

A. Hardware Design for Object Sorting Robot System

The hardware structure of our designed object sorting robot system is shown in Fig. 1. Our system uses three embedded computers to serve as the core of the system. One of the computers is connected to a Kinect sensor for 3-D environment recognition, which processes the RGB-D point data from Kinect to generate a scene semantic map file. Another one is connected to the microphone and speaker to implement natural language interaction and reasoning, which act as the brains of our system. The third embedded computer is connected to the modular manipulator and end-effector, which is responsible for control tasks.

The system's robot body is a six degree of freedom serial modular manipulator having six modular joints. Each modular joint includes reducer, motor, encoder, control circuit board, and servo board separately. Communication between the control system and modular joints is achieved through a controller area network (CAN) bus.

B. Software Design for Object Sorting Robot System

This system is based on a robot operating system (ROS) distributed processing framework. The human–robot-environment interactive reasoning mechanism is at the core. The object sorting robot system's software is shown in Fig. 2. The system is divided into many executable nodes, which include point

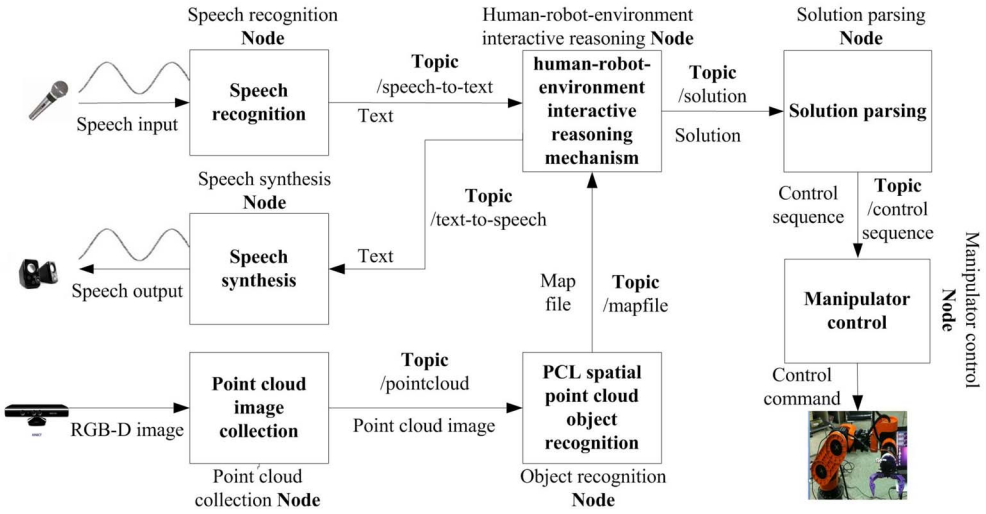


Fig. 2. Software structure of our designed object sorting robot system.

cloud collection node, object recognition node, speech recognition node, speech synthesis node, human–robot-environment interactive reasoning node, solution parsing node, and manipulator control node. Nodes are loosely coupled and communications between nodes are asynchronous data communication based on topic.

The point cloud collection node gathers the depth and RGB data of scene by Kinect, and fuses them together to generate 3-D point cloud data. The output point cloud image is published to point cloud topic.

The object recognition node subscribes point cloud topic to obtain data, then uses this data to generate a 3-D scene semantic map through preprocessing, key point extraction, descriptor extraction, and 3-D feature matching, etc. Point cloud collection node and object recognition node run on the same embedded computer.

The speech recognition node uses the input speech to generate text through the process of input speech signal noise reduction, mel-frequency cepstral coefficients feature extraction [14], and speech decoding by combining the hidden Markov model acoustic model [15] and N-gram language model [16]. The output text is published to the speech-to-text topic.

The human–robot-environment interactive reasoning node subscribes the speech-to-text topic and map file topic, then use these data as input to realize interaction and reasoning. The robot's feedback information or questions are sent to text-to-speech (TTS) topic in the form of text. In this node, the system will generate a solution and publish it to the solution topic when the user's desire is complete.

The speech synthesis node makes use of TTS technology [17] to process the subscribed text, which consists of text analysis, prosody modeling, and speech synthesis. Speech recognition node, speech synthesis node, and human–robot-environment interactive reasoning node run on the same embedded computer.

The solution parsing node subscribes solution topic to obtain solution, parse the solution, extract coordinate information,

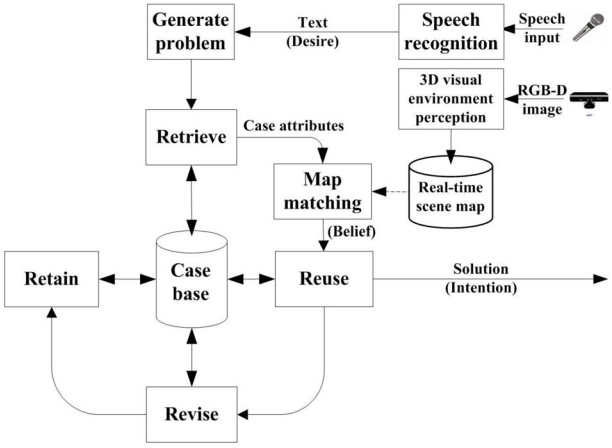


Fig. 3. Structure of traditional CBR–BDI reasoning mechanism.

calculate the best path through trajectory planning, the system will output the control sequences and publish them to the control sequence topic.

The manipulator control node subscribes control sequence topic and gets the sequence of control information, converts the control sequence to the rotation angle of each joint and end-effector, controls the motion of manipulator via USB to CAN interface adapter. Solution parsing node and manipulator control node run on the same embedded computer.

III. Traditional CBR–BDI Reasoning Mechanism

The structure of a traditional CBR–BDI reasoning mechanism which applied to our object sorting robot system is shown in Fig. 3. The user's input information is defined as desire, system gets the desire and builds up a problem, then matches the problem with casebase by a case retrieve algorithm and extracts the case attributes of the problem. Finally, belief is generated after map matching and intention is obtained

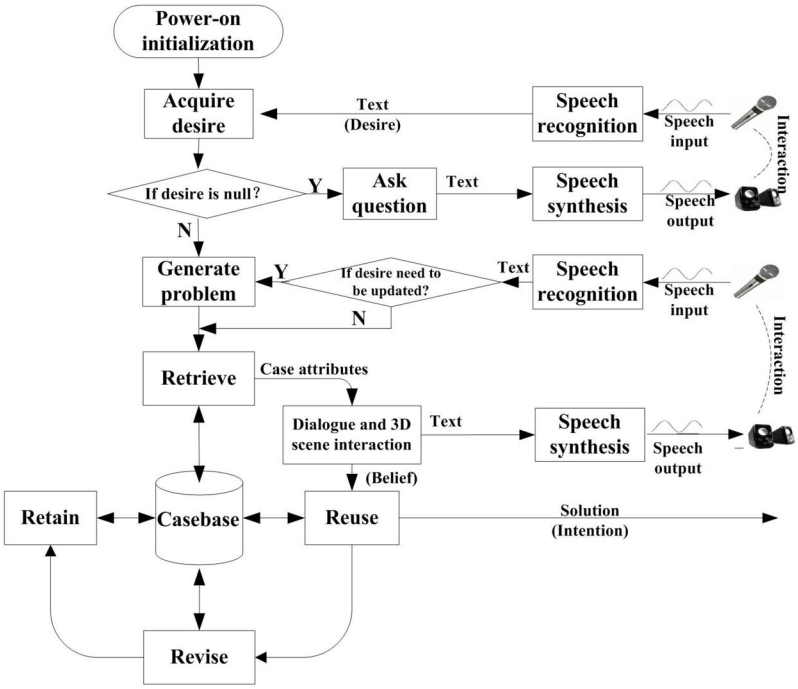


Fig. 4. Structure of human–robot-environment interactive reasoning mechanism.

through case reuse. For instance, there are two apples (one is green, the other is red) in front of robot and a basket on the left side of robot. User's input information is "grab the red apple and put it into the basket." As the above definition of BDI, system gets the user's input information as desire and generates a problem: grab the red apple and put it into the basket. After casebase retrieving and case attributes extracting, system communicates with the real 3-D environment and gets the real matched information of objects (object's size, shape, color, coordinate, etc.). The case attributes with the actual object information is defined as belief. At last, system invokes the actions which corresponding to the case attributes in the casebase and combines the actions with target object information to generate a set of actual executable action sequence (intention).

IV. Human–Robot-Environment Interactive Reasoning Mechanism

The proposed structure of our human–robot-environment interactive reasoning mechanism is based on traditional CBR–BDI mechanism. As shown in Fig. 4. First, the robot checks whether the desire exists after power-on initialization; if the desire is null, robot will take the initiative to ask questions to request user to provide task information; user's input will be defined as desire. After getting the user's desire, the desire will be treated as an input problem of case-based reasoning. Then, matches the problem with casebase by case retrieve algorithm and extracts the case attributes of the problem. Finally, belief is generated after dialogue and 3-D scene interaction; intention is obtained through case reuse. Herein, comparing with the traditional CBR–BDI mechanism, the great improvement of our mechanism is the introduction of our proposed dialogue and 3-D scene interaction module. Our proposed dialogue and 3-D scene interaction module not only realizes the traditional function of map matching but also achieves the function of desire analysis. The robot is capable of analyzing the user's desire. When the user's desire is incomplete, the robot will take the initiative to guide users through dialogue, and the user's input information will be used to replenish its former desire (give logic to belief, to achieve autonomous reasoning).

The reasoning process of dialogue and 3-D scene interaction module is shown in Fig. 5. It has three main parts: 1) map matching; 2) desire analysis; and 3) guidance. Map matching is used to implement the matching of user's desire with real-time scene map. Desire analysis is used to calculate the complete of the user's desires. Guidance is used to generate guidance solutions.

The following content of this section will describe the working principle of our proposed human–robot-environment interactive reasoning mechanism. In particular, this mechanism is applied to Chinese natural language interaction as an example. In fact, the overall structure of our proposed mechanism can also be applied to other languages with small modifications in speech recognition, speech synthesis, and case retrieval algorithms.

A. Case Representation and Retrieval

1) Case Representation: A case can be stored in a variety of forms, such as framework representation, object-oriented representation, and predicate representation [18]. Because of its features of applicability, summarizing, structuring, and reasoning, in this paper, we chose the framework as the representational format. Meanwhile, considering the requirement

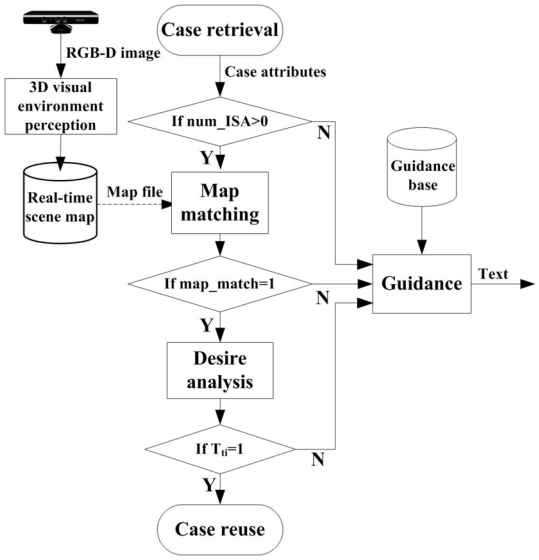


Fig. 5. Reasoning process of dialogue and 3-D scene interaction module.

of our object sorting robot system, the representation of a case can be described as

$$\text{Case base} = <\text{Case}_1, \text{Case}_2, \ldots, \text{Case}_i> \quad (1)$$

$$\text{Case}_i = <\text{Initial\_state}_i, \text{Solution}_i, \text{Final\_state}_i>. \quad (2)$$

Here

Initial_State: The initial attributes set of a case.

Solution: The solution of a case, it is a set of actions. Suppose that we want to grab a target object in one place and put it to another place, the solution could be described as follows: move end-effector of manipulator to approach the target object which is vertical along z-axis 100 mm (coordinate: 0.350, 0.100, 0.660); open gripper; end-effector arrive at the target point (coordinate: 0.350, 0.100, 0.560); close gripper; raise up end-effector (coordinate: 0.350, 0.100, 0.660); end-effector move to another place (coordinate: 0.700, −0.300, 0.555); open gripper; back to original point (coordinate: 0.000, 0.000, 0.789); close gripper. Each action of the solution above is corresponding to a serial of machine instruction which is used to control the robot.

Final_State: The final attributes set of a case which is generated after the robot, human, and environment interactions and reasoning.

In this paper, consider the trait of our object sorting system that the employed vocabulary is task-based utterance. The structure of a task topic tree is selected to describe the logical relationship among the state attributes, as shown in Fig. 6. There are three types of nodes: 1) topic node; 2) intermediate node; and 3) leaf node:

1) each of the topic nodes is the root node of a topic tree, which represents the type of the topic and relevant knowledge base;
2) intermediate nodes give the logical relationship among its child nodes, which include two kinds of logical operators: "and" and "or";
3) leaf nodes are used to store the information about the topic, associated with particular semantics. The value of leaf nodes is the knowledge state of its information items.

Each node has a Boolean valid state symbol. The valid state symbol of the leaf node indicating that the node is valid or not, that is, whether the attribute of the node is confirmed by the user. The valid state symbol of the topic node and intermediate node depends on the logic operation result of their child node.

Each node has a corresponding dialogue generating function, which constitutes the guidance base. The system will get a different guidance output under the different system status. Every function is only responsible for its own corresponding node. The dialogue generating function is independent of each other in the design and modification.

According to the definition above, in this paper, *state* attributes of object sorting robot system are shown in Fig. 7. The topic node consists of two kinds of topics (asking topic and sorting topic). The selection of topic is decided by the key words.

In Fig. 7:

1) *location* represents the location user asks in the desire command;
2) *interrogative words* represents the interrogative words user asks in the desire command;
3) *Obj_qty* represents the quantity of object;
4) *Obj_name* represents the name of object;
5) *Obj_location* represents the location of object;
6) *Obj_color* represents the color of object;
7) *Obj_size* represents the size of object;
8) *Des_name* represents the name of destination;
9) *Des_location* represents the location of destination.

2) Process of Case Retrieval: For the input text, the retrieval process includes the following steps.

Step 1: Extract the words which include obvious semantic information in the input text, then obtain the similar topics which are matched with the user's desire.

Step 2: Find the similar cases by traversing all the similar topics. In this paper, a relative position offset-based Chinese sentence similarity algorithm is employed to calculate the similarity of cases. In this algorithm, first, after word segmentation and part-of-speech (POS) tagging, central word of the sentence is extracted by the POS and grammatical rules, and the global qualifiers of the sentence are eliminated. Second, the relative position of words is marked according to the central word, and the position offset of all words relative to the center word is calculated. At last, the length different information of sentence, shallow hierarchical structure information and semantic information are integrated together to calculate sentence similarity. The similarity between two sentences $S_1 = \{W_{11}, W_{12}, \ldots, W_{1m}\}$ and $S_2 = \{W_{21}, W_{22}, \ldots, W_{2n}\}$ are defined as

$$\text{Sim}(S_1, S_2) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} \left(\text{Sim}\left(W_{1i}, W_{2j}\right) W_{\text{pos}}\right)}{\text{Max}(m, n)} \quad (3)$$

where, $\text{Sim}(W_{1i}, W_{2j})$ represents the "HowNet"-based word similarity between W_{1i} and W_{2j} [19], and W_{pos} is defined as

$$W_{\text{pos}} = 1 - \frac{\left|\text{pos}(W_{1j}) - \text{pos}(W_{2j})\right|}{\text{Max}(m, n)} \quad (4)$$

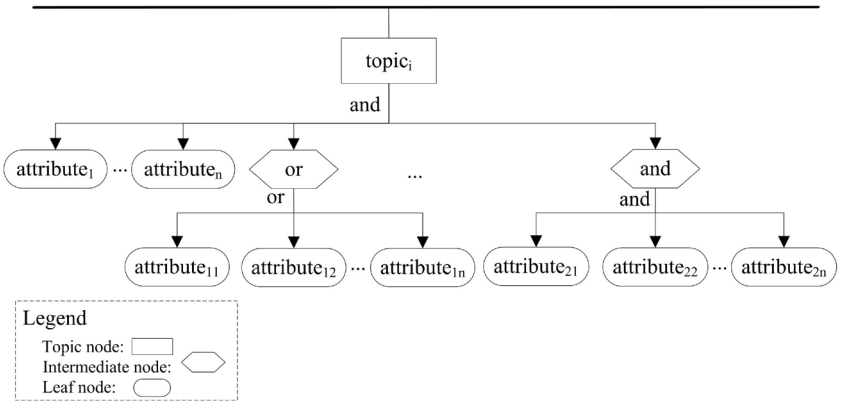


Fig. 6. Definition of *state* attributes.

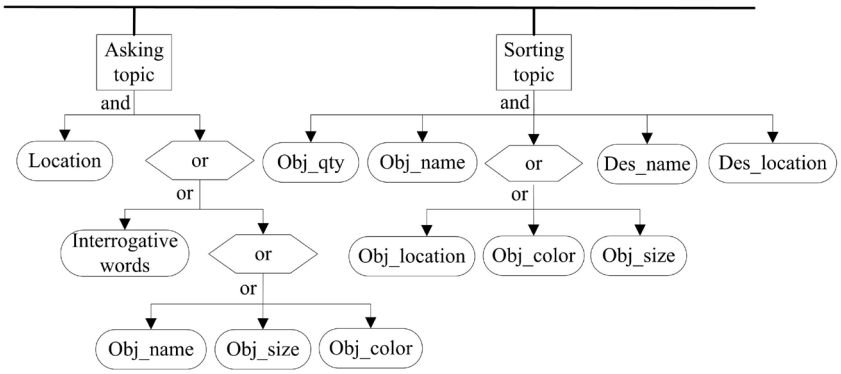


Fig. 7. Definition of *state* attributes in object sorting robot system.

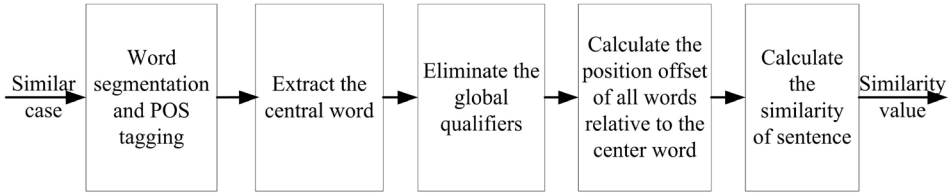


Fig. 8. Pipeline of Chinese sentence similarity algorithm.

where, pos(W) represents the relative position of words W which is related to the central word.

The pipeline of Chinese sentence similarity algorithm is shown in Fig. 8.

Step 3: Sort these cases by the similarity value which was calculated in step 2.

Step 4: Under the preset similarity threshold, select the most similar case as the best matched case.

3) Case Attributes Obtaining: For each input text, retrieves the similar case in the casebase. If there is no similar case in the casebase, builds up new case case$_{n+1}$<Initial_state$_{n+1}$, Solution$_{n+1}$, Final_state$_{n+1}$>, assigns input information to Initial_state$_{n+1}$; meanwhile, assigns null to Solution$_{n+1}$ and Final_state$_{n+1}$. Else returns the similar case Case$_i$<Initial_state$_i$, Solution$_i$, Final_state$_i$>. In the new built or returned similar case, system calculates the number of initial_state attributes (num_ISA), If num_ISA>0, it means that the user's input is valid, system will go next to map matching, else it means that the user's input is invalid, system will guide the user to provide correct information through asking question.

B. 3-D Environmental Perception

The system needs to obtain high quality environment semantic map information through 3-D environmental perception before map matching. The 3-D environmental perception pipeline in this paper is shown in Fig. 9. It consists of three parts: 1) object modeling; 2) object recognition and pose estimation; and 3) semantic map file generation.

1) Object Modeling: The process of object modeling with a Kinect is achieved offline. First, the synchronized RGB and depth images of an object are captured with a Kinect. The foreground object is then segmented out from the background and represented in a 3-D point cloud using the corresponding RGB and depth data. The partial 3-D point clouds from different views are then registered together to form a complete 3-D point cloud for the object. In this paper, the modeling

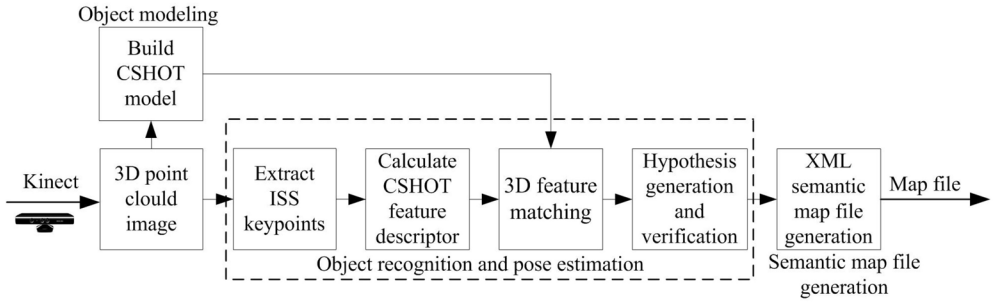


Fig. 9. 3-D environmental perception pipeline in this paper.

TABLE I
QUALITATIVE COMPARISON OF SIX KINDS OF THE STATE-OF-THE-ART DETECTOR

Algorithm	Computational efficiency	Repeatability	Robustness to noise	Distinctiveness
LSP	Good	Normal	Normal	Normal
ISS	Good	Good	Normal	Good
KPQ	Normal	Normal	Good	Good
LBSS	Bad	Normal	Normal	Normal
MeshDoG	Normal	Good	Normal	Good
KPQ-SI	Normal	Good	Normal	Good

method which proposed by Yang *et al.* [30] is employed to build our object models. With this method, objects dense 3-D point cloud model can be obtained simply by placing the object on a plane table, and collecting 10 to 20 frames data around the object.

When object's 3-D point cloud models are obtained, the next step is to find an appropriate object surface representation method (descriptor extract algorithm) to describe the salient feature of an object. Normally, the descriptor can be classified as global descriptor and local descriptor [20]. The 3-D object recognition based on global descriptor is not suitable for multiobject scenes due to some of the information object may be lost with issues, such as occlusion, clutter, and changes of viewpoint. It makes the calculation of the global descriptor not accurate, which will result in recognition errors. The 3-D object recognition based on local descriptor characterizes the objects by its partial 3-D geometry information, which only has little effect to the accuracy of local feature calculation and recognition. In this paper, the objects to be recognized in our system are rigid body, smooth surface, significant color differences, and the objects may be affected by the light, occlusion and viewpoint. So the local descriptor is chosen to describe the salient feature of an object. The first step for every 3-D environmental perception pipeline based on local descriptors is represented by the extraction of 3-D keypoints from data, and then the subsequent step follows the description of extracted keypoints by descriptor to characterize geometric information around them. So here we need to select a robust and efficient keypoint detection algorithm (detector). Tombari *et al.* [21] conducted a performance evaluation of the state-of-the-art 3-D keypoint detector, its evaluation result can be qualitative summarized as shown in Table I.

As can be seen from Table I, the ISS detector [22] appears to be the best overall detector according to the different aspects evaluated throughout the comparison. So, in this paper, ISS detector is selected to detect the keypoints of the object in the scene.

After getting the ISS keypoints, it requires a descriptor to characterize the geometric information around the ISS keypoints. Alexandre [23] conducted a comparative evaluation of the most state of the art descriptors. Their main conclusions are as follows:

1) increasing the number of key points improves recognition results at the expense of size and time;
2) since there are big differences in terms of recognition performance, size, and time requirements, the descriptor should be matched to the desired task;
3) a descriptor that uses color information should be used instead of a similar one that uses only shape information;
4) the color signature of histograms of orientations (CSHOT) descriptor [24] presents a good balance between recognition performance and time complexity.

Based on the conclusion above, in this paper, the CSHOT descriptor is selected to characterize the geometric information around the ISS keypoints.

The CSHOT descriptor adds color information (based on the CIELab color space) to the SHOT descriptor [25] resulting in a 1344 value descriptor (plus nine values to describe the local reference frame). The SHOT descriptor is based on obtaining a repeatable local reference frame using the eigenvalue decomposition around an input point. Given this reference frame, a spherical grid centered on the point divides the neighborhood so that in each grid bin a weighted histogram of normals is obtained. The descriptor concatenates all such histograms into the final signature. It uses nine values to encode the reference frame and the authors propose the use of 11 shape bins and 32 divisions of the spherical grid, which gives an additional 352 values. The object model which is constructed in this paper is shown as

$$C_n = \{(x_i, y_i, z_i, f_i), \text{label}, \ \text{size}, \ \text{color}\}, (C_n \in C) \quad (5)$$

where x_i, y_i, and z_i represent the 3-D coordinates of each keypoint, f_i represents the 1344+9 dimensional CSHOT descriptor of each keypoint. Each model has its corresponding label, size, and color.

2) Object Recognition and Pose Estimation: The process of object recognition and pose estimation is running online. The algorithms of ISS keypoints extraction and CSHOT feature descriptor calculation are the same as object modeling stage. CSHOT descriptors are obtained on the extracted keypoints and these form a set that is used to represent the input real-time scene point cloud. This set is matched against sets already present in the object model database and the one with largest similarity (smallest distance) is considered the match for the real-time input point cloud.

Many different methods can be used to match the set of descriptors that represents a given input point cloud against the available sets that represent previously registered object (these are typically part of an object model database). In this paper, we will use the following set distance: find the centroid of each set plus the standard deviation for each dimension (coordinate) of each set and return the sum of the L_1 distances between them

$$D(A, B) = L_1(C_A, C_B) + L_1(\text{std}_A, \text{std}_B) \tag{6}$$

where C_A and C_B are the centroids of sets A and B, and

$$\text{std}_A(i) = \sqrt{\frac{1}{|A|}\sum_{j=1}^{|A|}(a_j(i) - C_A(i))^2}, i = 1, \ldots, n \tag{7}$$

and likewise for std_B; n is the size of the descriptor used in the sets. The L_1 distance between two descriptors a and b is given by

$$L_1(a, b) = \sum_{i=1}^{n}|a(i) - b(i)|. \tag{8}$$

The pose estimation method is based on least-squares fitting of two 3-D point sets [26]. Suppose that there are two 3-D point sets $\{{}^{Q}P\}$ and $\{{}^{C}P\}$, the determination of the relative pose of a rigid object with respect to a reference can be calculated as

$${}^{Q}P = {}^{Q}_{C}R \cdot {}^{C}P + {}^{Q}_{C}T \tag{9}$$

where R is a 3×3 rotation matrix, T is a translation vector (3×1 column matrix). Our target is to find R and T to minimize

$$E = \sum_{i=1}^{n}\left\|\left({}^{Q}_{C}R \cdot {}^{C}P + {}^{Q}_{C}T\right) - {}^{Q}P\right\|^2. \tag{10}$$

The pose matrix ${}^{C}_{K}M$ of the object models relative to Kinect coordinate system can be calculated by the checkerboard calibration algorithm, and the pose matrix ${}^{Q}_{C}M$ of the scene objects to the object models can be calculated by the method mentioned above. Multiply those two matrices together can get the pose matrix of the scene objects relative to Kinect coordinate system as

$${}^{Q}_{K}M = {}^{C}_{K}M \cdot {}^{Q}_{C}M. \tag{11}$$

In order to improve the outcome of the recognition, it can undergo an optional post-processing stage of hypothesis generation and verification. Combine the gotten pose of objects with the results of the cluster segmentation, merge the clusters which belong to the same kind, and finally get the accurate recognized object's label and pose.

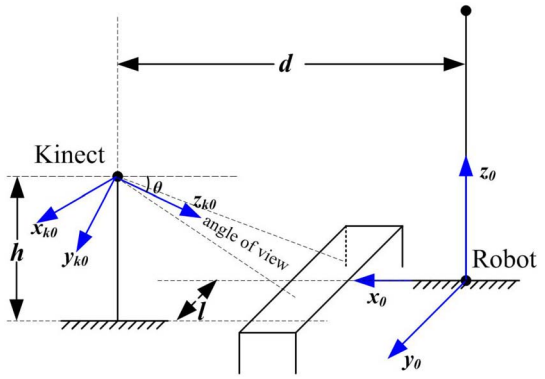


Fig. 10. Relationship between Kinect coordinate system and robot coordinate system.

3) Semantic Map File Generation: The coordinates of the object we got on the previous section is under the Kinect reference coordinate system, which need to be converted to the coordinates of robot arm (x, y, z). The relationship between Kinect coordinate system and robot coordinate system is shown in Fig. 10, the conversion formula is shown as

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 & \sin\theta & -\cos\theta \\ 1 & 0 & 0 \\ 0 & -\cos\theta & -\sin\theta \end{bmatrix}\begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} + \begin{bmatrix} d \\ l \\ h \end{bmatrix}. \tag{12}$$

Among them, x_k, y_k, and z_k represent object's coordinate relative to Kinect, θ represent Kinect's tilt angle which relative to the horizon. d, l, and h represent the relation between the origin of robot coordinate system and the origin of Kinect coordinate system which along the x, y, and z-axes of the robot coordinate system, respectively.

Next, the object's identification and geometric information are written to the semantic map file. In this paper, the format of semantic map file which we use is eXtensible markup language (XML). XML is a user-defined markup language, which is a set of specifications created by World Wide Web consortium and designed to transport and store data [27]. It is a new generation of Web language that uses tree structure, adapts for multiobject architectures, and would be able to describe the relationship between the data. XML language has good data storage format, is high-structured, and is easy for network transmission. The expansibility and reusability of XML are better than traditional relational databases. The structure of XML map file which we use to store semantic information is defined as

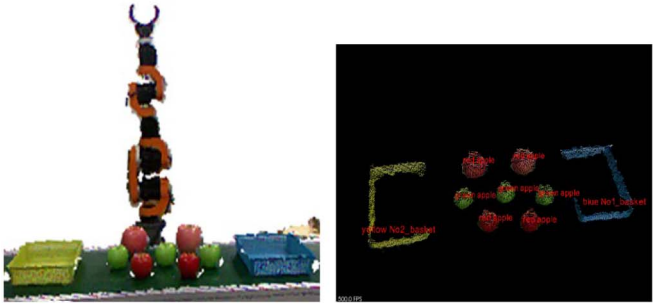
$$\text{object}_i = \{\text{obj ID, name, color, shape, } x, y, z, \text{size}\}. \tag{13}$$

Herein

objID	ID number of object in scene map;
name	name of object;
color	color of object;
shape	shape of object;
x, y, z	coordinate of object in the manipulator reference coordinate system;
size	size of object.

The key steps of 3-D environmental perception procedure are illustrated in Fig. 11.

(a) (b)



(c) (d)

```
<?xml version="1.0" encoding="UTF-8"?>
- <map>
  - <object>
      <objID>1</objID>
      <name>Left_basket</name>
      <x>0.000</x>
      <y>0.600</y>
      <z>0.430</z>
      <size>10000</size>
    </object>
  - <object>
      <objID>2</objID>
      <name>apple</name>
      <color>red</color>
      <shape>cuboid</shape>
      <x>0.475</x>
      <y>-0.0019</y>
      <z>0.588</z>
      <size>1274.463</size>
    </object>
```

(e)

Fig. 11. (a) RGB image of actual scene, (b) point cloud image of actual scene, (c) point cloud image after preprocessing, (d) clustering objects after point cloud segmentation, and (e) XML 3-D semantic map (here, we only captured a part of XML semantic information as an example).

C. Map Matching

First, the system calculates the sum of user-expected object (sum_expobj) in XML map file

$$\text{sum\_expobj} = \sum_{i=1}^{\text{Max\_objID}} s\big(\text{expobj} \cap \text{object}_i\big) \tag{14}$$

wherein, expobj represents the expected object information. This expobj is a single object attribute set, which is defined as same as the attributes in the object_i, and $s(x)$ is defined as

$$s(x) = \begin{cases} 1, & \text{for } x = \text{expobj} \\ 0, & \text{for } x \neq \text{expobj}. \end{cases} \tag{15}$$

Then, calculate the matching status of user's expected objects with map file (map_match)

$$\text{map\_match} = \frac{\text{sum\_expobj}}{\text{expected Qty}} \tag{16}$$

wherein, expected Qty represents the number of user-desired objects.

When

map_match = 0 it means no expected object in the scene;

$0 < \textit{map\_match} < 1$ it means that the number of objects in the scene is less than the number of user's expectation;

map_match = 1 it means the number of objects in the scene is equal to the number of user's expectation;

map_match>1 it means the number of objects in the scene is more than the number of user's expectation.

Here, the desire is valid when *map_match* = 1, which means the user's desire is determined exactly, and the system can go next to desire analysis. Otherwise, the system will guide the user to provide correct information through asking a question. For example, there are two small green apples in the actual scene; if you want to grab one small green apple, then *map_match* > 1, system will start the guidance mode and ask the question: "there are two small green apples, which one do you want to grab?" Then, the user's reply information will be used to replenish the user's former desire.

D. Desire Analysis

As described in section "case representation," each node of the state attribute has a Boolean valid state symbol (T_{xi}) which is either "0" or "1." Herein, 1 represents true and 0 represent false. All the nodes are divided into three types: 1) the state symbol of topic node (T_{ti}); 2) the state symbol of topic node's child node (T_{tci}); and 3) the state symbol of intermediate node's child node (T_{mci}). The value of T_{xi} can be calculated by

$$T_{xi} = \begin{cases} 1, & \text{for expobj} = \text{node}_{xi} \cap \text{expobj} \\ 0, & \text{for expobj} \neq \text{node}_{xi} \cap \text{expobj}. \end{cases} \tag{17}$$

Here, node_{xi} represents the attribute set of topic node, topic node's child node, and intermediate node's child node, $x \in \{t, tc, mc\}$, $1 \leq i \leq n$.

Calculate the value of topic node valid state symbol. The formula is shown as

$$T_{ti} = \bigcap_{i=1}^{n} T_{\text{tci}} \tag{18}$$

$$T_{\text{tci}} = \bigcup_{i=1}^{n} T_{\text{mci}}. \tag{19}$$

If T_{ti} equals 1, it means that user's desire is complete. Else, the system will guide the user through asking a question.

E. Guidance

As described in section case representation, each node has a corresponding dialogue generating function, which constitutes Guidancebase. The definition of Guidancebase is shown as (20) and (21). The guidance solution (guidance_solution) is decided by the state set (node_state) of the binary valid state

$$\text{Guidancebase} = (\text{GC}_1, \text{GC}_2, \ldots, \text{GC}_n) \tag{20}$$

$$\text{GC}_i = \big(\text{node\_state}_i, \text{guidance\_solution}_i\big). \tag{21}$$

Algorithm 1 Dialogue and 3-D Scene Interaction Algorithm

```
procedure MyProcedure
Top:
    Awaiting user's input.
    Acquire/replenish desire from user and save it as state.
    Retrieve the similar case in the Casebase.
    if there is no similar case in the Casebase then
        {Build up new case Case_{n+1}
         < Initial_state_{n+1}, Solution_{n+1}, Final_state_{n+1} >;
        Initial_state_{n+1} ← state; }
    else fetch similar case Case_i
         < Initial_state_i, Solution_i, Final_state_i >;
    endif
    Calculate the number of initial_state attribute (num_ISA).
    if num_ISA > 0 then
        Map Matching: Calculate the value of map_match.
        if map_match == 1 then
            Desire analysis: Calculate the value of T_tx.
            if T_tx == 1 then
                Case Reuse: Intention ← Solution.
            else goto Guidance
            endif
        else goto Guidance
        endif
    else goto Guidance
    endif
    return Intention.
Guidance:
    {Invoke guidance_solution in the Guidancebase by the state set (node_state);
    Transfer the guidance solution to voice;
    goto Top }
```

F. Complete Intention Generation and Case Reuse

The initial incomplete desire will be replenished step by step through one or more times of guidance, and finally, generates the complete desire. Retrieve the most similar case in the casebase which matches with the complete desire. Reuse the solution of the most similar case after real-time 3-D environment matching, and generate a sequence of actions (intention) to complete specific job tasks.

The Pseudo code of proposed dialogue and 3-D scene interaction is shown in Algorithm 1.

With the introduction of the "dialogue and 3-D scene interaction" module, the system not only realizes interaction and reasoning among human, robot, and environment, but also achieves the function of desire analysis; the robot is capable of analyzing the user's desire; when user's desire is incomplete, the robot will take the initiative to guide users through dialogue, and the user's input information will be used to replenish user's desire; finally, reuse the solution of the most similar case and get the standardization of complete intention.

We have to point out that our human–robot-environment interaction is based on the reaction of human information is logical. The un-logical reaction of human information, such as the user's speech is self-contradicting or contradicts with previous information, is not considered in our mechanism. That because the un-logical reaction of human information has its uncertainty, which would be another research area of HRI.

Let us take a look at an example to better understand the overall process. Suppose there are a big red apple, two small red apples, and two small green apples on the platform, two baskets on the left and right side of the platform, respectively. There are three basic case in the casebase: 1) how many small green apples on the platform; 2) what is that on the platform? and 3) grab two small green apples and put it into the right basket.

[Robot]: I am Wuster, what can I do for you? (First, the desire is null, system will take the initiative to ask question).

[User]: What is that on the production line? (This sentence of desire is similar to the case "What is that on the platform?" which is predefined in the casebase.)

[Robot]: There are a big red apple, two small red apples, and two small green apples on the platform. (Reuse the solution of similar case "What is that on the platform?" then tell the human user what is that on the platform.)

[User]: I want small green apples. (The keyword "want" determines that this is a sorting topic, then system achieves the attributes of this sentence that *Obj_size* "small"; *Obj_color* "green"; *Obj_name* "apple." As the definition of state attributes in Fig. 7, it can be seen that system still need the information of *Obj_qty*, *Des_name*, and *Des_location* to achieve the value of the topic node to 1. So, system will invoke the predefined question in the guide base to guide user to give more detailed information for replenishing the users desire.)

[Robot]: How many small green apples do you want to grab?

[User]: Grab three. (In this sentence, the user gives the attribute of *Obj_qty* "three" to replenish the user's desire, however, there are not so many small green apples in the actual scene after scene matching. So system will guide the user through asking question.)

[Robot]: There are not enough small green apples in the scene.

[User]: How many small green apples in the scene? (Reuse the solution of similar case "How many small green apples on the platform?" then tell the human user what is that on the platform.)

[Robot]: There are two small green apples.

[User]: Take two small green apples. (The user gives the attribute of *Obj_qty* "two" to replenish the user's desire, and it is valid which is matched with actual scene. Now, system still need another two pieces of information of *Des_name* and *Des_location* to achieve the value of the topic node to 1. So system will guide the user through asking question.)

[Robot]: Okay, where are you going to put them?

[User]: Put them into the right basket. (At this time, the user provides the attributes of *Des_location* "right," *Des_name* "basket," and those attributes are valid which are matched with the actual scene. All the attributes of this topic are confirmed

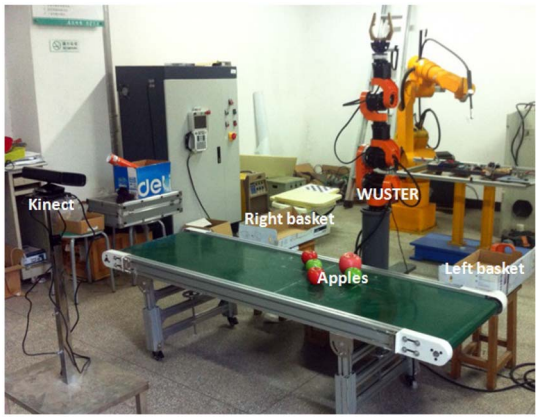


Fig. 12. Experimental platform of apple sorting robot system.

till now, and the value of topic node is equal to 1, which indicates that the desire is complete.)

[Robot]: Okay, grab two small green apples and put them into the right basket, correct? (Reconfirm the complete desire with human).

[User]: Yes.

[Robot]: Okay.

V. Experiment and Analysis

In this section, the verification experiments are conducted in the scenario of apple sorting in our designed object sorting system to verify the effectiveness and practicality of our proposed human–robot-environment interactive reasoning mechanism. An open source speech recognition system PocketSphinx [28] which was developed by University of Carnegie Mellon is used to convert speech to text, and an open source speech synthesis system Ekho [29] is used to convert text to speech. The robot body is a modular manipulator (WUSTER), and each function module runs on Ubuntu OS (version 12.04). There are a big red apple, two small red apples, and two small green apples on the conveyor. The image collection device Kinect is fixed in the front of the conveyor as shown in Fig. 12.

The initial basic cases in casebase are preset manually. All of them are complete desire which consists of five asking topic cases and ten sorting topic cases. Four datasets were designed to conduct a series of performance tests for our proposed mechanism from two aspects: 1) user desire's completeness and 2) user desire's matching performances. All the human utterances can be summarized as the following four types of desires: 1) complete and matching desire; 2) complete but mismatching desire; 3) incomplete but matching desire; and 4) incomplete and mismatching desire.

Dataset 1: Complete and matching test case set. This dataset totally includes 30 sentences of desire which consists of two types of desires.

1) The 15 complete desires which were predefined in the casebase.
2) The 15 synonymous similar desires which are impromptu expressed by the tester according to the predefined 15 cases in the casebase, (a different expression of synonymous desire). The user's desire in each case contains all case attributes of its topic and its case attributes match with the actual scene.

Dataset 2: Complete but mismatching test case set. On the basis of the dataset 1 test case set, randomly mismatch one or several case attributes with the actual scene.

Dataset 3: Incomplete but matching test case set. On the basis of the dataset 1 test case set, randomly drop one or several attributes of each desire to construct this test case set.

Dataset 4: Incomplete and mismatching test case set. On the basis of the dataset 2 test cases set, randomly drop one or several attributes of each desire to construct this test case set.

According to the experimental setup above, we test the performances of speech recognition and speech synthesis in a limited set which is suitable for our designed system. After a large number of training and testing, the correct rate of speech recognition can reach at 93%, and the correct rate of speech synthesis reaches at 99%. However, in order to avoid the impacts which are generated by speech recognition and speech synthesis errors, we monitor the intermediate contents of TTS and automatic speech recognition during our test. Once there is an error, the test will be judged as invalid and it will be retested again until the speech recognition or speech synthesis is correct.

In CBR the correct rate of reasoning is directly determined by the ability that correctively calculates the similarity of cases. In this paper, before our comparison experiment, we conduct a test for our employed correcting offset-based Chinese sentence similarity calculation algorithm. thirty undergraduate students were invited as testers to test the 15 synonymous similar desires in dataset 1 type 2) which is impromptu expressed by the tester according to the predefined 15 cases in the casebase. Each sentence of desire is tested three times, totally test 1350 times for this test. Evaluation criteria of the corrective similarity calculation is the correct rate, which shown in (22). The test result of this test is 86.81%

$$S = \frac{R}{M} \times 100\% \tag{22}$$

wherein, R is the quantity of correct calculation, M is the quantity of all tests.

Based on those four datasets, a comparison experiment is conducted among our proposed method, the traditional CBR–BDI method and the rule-based method [10]. For rule-based method, We considered all the possible rules of the 15 basic cases in casebase and converted them to knowledge base which consists of a set of "IF...THEN..." rules. For each dataset, 30 undergraduate students were randomly invited as testers to test above four datasets, respectively. Each user's desire tests three times, totally test 2700 times for each dataset. Evaluation criteria of robot reasoning result is the correct rate, which are shown in (22).

The test results are shown in Table II. As can be seen from Table II, the correct rate of our proposed reasoning mechanism all reach around 87% in four datasets which is equivalent to the correct rate of our used similarity algorithm 86.81%. However, as for the traditional CBR–BDI mechanism, the correct rate

TABLE II
CORRECT RATE COMPARISON AMONG OUR PROPOSED MECHANISM, THE TRADITIONAL CBR–BDI MECHANISM AND RULE-BASED MECHANISM

Dataset No.	CBR-BDI mechanism	Rule-based mechanism	Our proposed mechanism
Dataset 1	87.07%	50.00%	87.11%
Dataset 2	1.15%	0.00%	86.33%
Dataset 3	1.12%	0.00%	86.48%
Dataset 4	0.85%	0.00%	86.93%

reaches at 87% only when the user's desires is complete and matched, in the other three cases, the correct reasoning only happens in a small probability rate. For the rule-based mechanism, the reasoning correct rate is 50% in dataset 1, however, in the other three dataset, they are unable to reason out the correct results. That is because that the establishment of knowledge base only by taking into accounts the rules which are completely certain and predefined. In this experiment, the 15 synonymous similar desires in dataset 1 type 2) which is impromptu expressed by the tester according to the predefined 15 cases in the casebase, cannot be involved in knowledge base in advance. RBR mechanism can only achieve effective reasoning of the situation that rules have been already preset in the knowledge base in advance, and for the similar desires, the attributes missing desires and the attributes mismatch with actual scene desire, which are not predefined in the knowledge base will result in an error.

So we can conclude that our proposed algorithm comparing to CBR–BDI mechanism and rule-based mechanism has obvious advantages in three kinds of situations: 1) complete but mismatching desire; 2) incomplete but matching desire; and 3) incomplete and mismatching desire; meanwhile, in the situation of complete and matching desire, the reasoning ability of CBR–BDI-based reasoning mechanism is better than the RBR mechanism.

The illustrated experiments with actual robot are available on the website below: http://v.youku.com/v_show/id_XMTY0ODcxNjc5Ng==.html.

VI. CONCLUSION

In this paper, we design an ROS-based object sorting system, it includes the function of language recognition and synthesis, RGB-D-based point cloud 3-D environment perception, human–robot-environment interactive reasoning, and natural language automatic programming. In particular, we proposed a human–robot-environment interactive reasoning mechanism. In our mechanism, a dialogue and 3-D scene interaction module is added into the traditional CBR–BDI reasoning mechanism, which not only realizes the traditional function of map matching but also achieves the function of desire analysis. The robot is capable of analyzing the user's desire, and when a user's desire is incomplete, the robot will take the initiative to guide the user through dialogue, and the user's input information will be used to replenish the user's desire. Our proposed reasoning mechanism is based on the CBR–BDI mechanism, which can reuse previous experiences to solve current problems, and also enables the reply to events, interactions with the environment, taking the initiative according to goals. The verification experiments are conducted in the scenario of apple sorting in our designed object sorting system to verify the effectiveness and practicality of our proposed reasoning mechanism. Four datasets were designed to conduct a series of performance tests for our proposed mechanism from two aspects: 1) user desire's completeness and 2) user desire's matching performances. Experimental results show that our proposed mechanism could implement the correct interaction and reasoning of the situation when the user's desire is incomplete and/or mismatched with the actual scene. However, the traditional CBR–BDI and rule-based mechanism do not have those functions.

For future works, our improvement efforts will focus on two aspects. First, in the process of testing, we found that the proposed case retrieval algorithm cannot effectively reflect its semantic information in the calculation of long text case, so we will consider adding more semantic information to further improve the accuracy of case retrieval. Second, we plan to conduct research on the feature descriptor algorithm of 3-D environmental perception to improve the accuracy of object recognition.

REFERENCES

[1] R. Swanson and A. S. Gordon, "Say anything: Using textual case-based reasoning to enable open-domain interactive storytelling," *ACM Trans. Interactive Intell. Syst.*, vol. 2, no. 3, pp. 1–32, 2012.

[2] V. L. Salazar, E. M. E. Cabeza, J. L. C. Pena, and J. M. Z. Lopez, "A case based reasoning model for multilingual language generation in dialogues," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7330–7337, 2012.

[3] J. M. Peula, C. Urdiales, I. Herrero, and F. Sandoval, "Implicit robot coordination using case-based reasoning behaviors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, 2013, pp. 5929–5934.

[4] H. Min, Y. Lin, S. Wang, F. Wu, and X. Shen, "Path planning of mobile robot by mixing experience with modified artificial potential field method," *Adv. Mech. Eng.*, vol. 7, no. 12, pp. 1–17, 2015.

[5] S. Wang and H. Min, "Experience mixed the modified artificial potential field method," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Tokyo, Japan, 2013, pp. 4823–4828.

[6] C. Olivia, C.-F. Chang, C. F. Enguix, and A. K. Ghose, "Case-based BDI agents: An effective approach for intelligent search on the world wide Web," in *Proc. AAAI Spring Symp. Intell. Agents*, 1999, pp. 22–24.

[7] J. Bajo, M. L. Borrajo, J. F. De Paz, J. M. Corchado, and M. A. Pellicer, "A multi-agent system for Web-based risk management in small and medium business," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6921–6931, 2012.

[8] S. Dalal, G. Tanwar, and N. Alhawat, "Designing CBR-BDI agent for implementing supply chain system," *System*, vol. 3, no. 1, pp. 1288–1292, 2013.

[9] J. A. Fraile, Y. De Paz, J. Bajo, J. F. De Paz, and B. Perez-Lancho, "Context-aware multiagent system: Planning home care tasks," *Knowl. Inf. Syst.*, vol. 40, no. 1, pp. 171–203, 2014.

[10] R. Ros *et al.*, "Which one? Grounding the referent based on efficient human-robot interaction," in *Proc. 19th Int. Symp. Robot Human Interactive Commun.*, Viareggio, Italy, 2010, pp. 570–575.

[11] J.-Q. Wang and H.-Y. Zhang, "Multicriteria decision-making approach based on Atanassov's intuitionistic fuzzy sets with incomplete certain information on weights," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 3, pp. 510–515, Jun. 2013.

[12] M. Banerjee and D. Dubois, "A simple logic for reasoning about incomplete knowledge," *Int. J. Approx. Reason.*, vol. 55, no. 2, pp. 639–653, 2014.

[13] J. Prentzas and I. Hatzilygeroudis, "Categorizing approaches combining rule-based and case-based reasoning," *Expert Syst.*, vol. 24, no. 2, pp. 97–122, 2007.

[14] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, "Speaker recognition using Mel frequency Cepstral coefficients (MFCC) and vector quantization (VQ) techniques," in *Proc. 22nd Annu. Int. Conf. Electron. Commun. Comput.*, Puebla, Mexico, 2012, pp. 248–251.
[15] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[16] J. Saraiva, C. Bird, and T. Zimmermann, "Products, developers, and milestones: How should I build my N-Gram language model," in *Proc. ACM 10th Joint Meeting Found. Softw. Eng.*, 2015, pp. 998–1001.
[17] M. P. Couper, P. Berglund, N. Kirgis, and S. Buageila, "Using text-to-speech (TTS) for audio computer-assisted self-interviewing (ACASI)," *Field Methods*, vol. 28, no. 2, pp. 95–111, 2016.
[18] S. C. K. Shiu and S. K. Pal, "Case-based reasoning: Concepts, features and soft computing," *Appl. Intell.*, vol. 21, no. 3, pp. 233–238, 2004.
[19] Q. Liu and X. Gu, "Study on HowNet-based word similarity algorithm," *J. Chin. Inf. Process.*, vol. 24, no. 6, pp. 31–37, 2010.
[20] A. Aldoma *et al.*, "Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 80–91, Sep. 2012.
[21] F. Tombari, S. Salti, and L. Di Stefano, "Performance evaluation of 3D keypoint detectors," *Int. J. Comput. Vis.*, vol. 102, nos. 1–3, pp. 198–220, 2013.
[22] Y. Zhong, "A shape descriptor for 3D object recognition," in *Proc. ICCV Workshop 3DRR*, vol. 6. Beijing, China, 2009, pp. 689–696.
[23] L. A. Alexandre, "3D descriptors for object and category recognition: A comparative evaluation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. Workshop Color-Depth Camera Fusion Robot. (IROS)*, vol. 1. Vilamoura, Portugal, 2012, pp. 1–6.
[24] F. Tombari, S. Salti, and L. Di Stefano, "A combined texture-shape descriptor for enhanced 3D feature matching," in *Proc. 18th IEEE Int. Conf. Image Process.*, Brussels, Belgium, 2011, pp. 809–812.
[25] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 356–369.
[26] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.
[27] M. Chen, "Factors affecting the adoption and diffusion of xml and Web services standards for e-business systems," *Int. J. Human Comput. Stud.*, vol. 58, no. 3, pp. 259–279, 2003.
[28] D. Huggins-Daines *et al.*, "PocketSphinx: A free, real-time continuous speech recognition system for hand-held devices," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 1. Toulouse, France, 2006, pp. 185–188.
[29] C. Wong. (2015). *Ekho—Chinese Text-to-Speech Software*. Accessed on Dec. 12, 2016. [Online]. Available: http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/
[30] Y. Yang, Q. Cao, X. Zhu, and P. Chen, "A 3D modeling method for robot's hand-eye coordinated grasping," *Jiqiren(Robot)*, vol. 35, no. 2, pp. 151–155, 2013.

Yunhan Lin received the B.Eng. degree from the Wuhan University of Science and Technology, Wuhan, China, in 2007, and the M.S. degree from the University of Science and Technology of China, Hefei, China, in 2013. He is currently pursuing the Ph.D. degree with the Institute of Robotics and Intelligent Systems, Wuhan University of Science and Technology.

His current research interests include robot 3-D environmental perception and intelligent reasoning.

Huasong Min received the B.Eng. and M.S. degrees from the Wuhan University of Water Transportation Engineering, Wuhan, China, in 1990 and 1999, respectively, and the Ph.D. degree from Wuhan University, Wuhan, in 2006.

From 2008 to 2010, he was with the Robotics Institute, Beijing University of Aeronautics and Astronautics, Beijing, China, as a Post-Doctor Researcher. He is currently a Full Professor with the Institute of Robotics and Intelligent Systems, Wuhan University of Science and Technology, Wuhan. His current research interests include embedded system and intelligent robotics.

Haotian Zhou received the B.Eng. and M.S. degrees from the Wuhan University of Science and Technology, Wuhan, China, in 2013 and 2016, respectively, where he is currently working toward the Ph.D. degree with the Institute of Robotics and Intelligent Systems.

His current research interest includes intelligent robot reasoning and SLAM.

Feilong Pei is currently pursuing the undergraduation degree with the Institute of Robotics and Intelligent Systems, Wuhan University of Science and Technology, Wuhan, China.

His current research interest includes embedded system and intelligent robot.