

The Intelligent Techniques in Robot *KeJia* – The Champion of RoboCup@Home 2014

Kai Chen^(✉), Dongcai Lu, Yingfeng Chen, Keke Tang, Ningyang Wang,
and Xiaoping Chen^(✉)

Multi-Agent Systems Lab, Department of Computer Science and Technology,
University of Science and Technology of China, Hefei 230027, China
{chk0105,ludc,chyf,kktang,wny257}@mail.ustc.edu.cn,
xpchen@ustc.edu.cn
<http://wrighteagle.org/en/robocup/atHome>

Abstract. In this paper, we present the details of our team WrightEagle@Home’s approaches. Our *KeJia* robot won the RoboCup@Home competition 2014 and accomplished two tests which have never been fully solved before. Our work covers research issues ranging from hardware, perception and high-level cognitive functions. All these techniques and the whole robot system have been exhaustively tested in the competition and have shown good robustness.

1 Introduction

The RoboCup@Home league was established in 2006 and aims at developing intelligent domestic robots which can perform tasks safely and freely in human daily lives. Our team WrightEagle@Home first participated in @Home league in 2009 and we got two 2nd places in 2011 and 2013. With continuous development in these years, we won the competition in 2014 with a clear advantage.

In this paper, we will first describe the setup of RoboCup@Home 2014. Then we will cover all the aspects of our *KeJia* robot and present our latest research progress. Section 3 gives an overview of *KeJia*’s hardware. The low-level functions for the robot are described in Sect. 4. Section 5 presents the techniques of complicated task planning while Sect. 6 introduces the grounding technique used in the open demonstrations. In Sect. 7 we report the performance of our robots at 2014 competition. Finally we conclude in Sect. 8.

2 RoboCup@Home 2014 Design and Setup

In RoboCup@Home competition, each team has to pass through two stages and a final demonstration to achieve the top places. In the 2014 competition, stage I has five tests while stage II has four. Each stage contains one open demonstration, fully open in stage I while scoped in stage II.

Each regular test in stage I focuses on some certain abilities of the robot. For example, the *Robo-Zoo* tests the robot’s visual appearance and interaction with

ordinary audience, *Emergency Situation* require the robot to detect abnormal status of human and take measures to prevent potential hazard. *Follow Me* test requires more abilities, including human detection, tracking and recognition, as well as dynamic obstacle avoidance in unknown environment.

While stage I tests focus on some certain but limited abilities, stage II tests expect the robot to handle difference situations and accomplish a complex task. Take the test *Restaurant* for example, during the limited time of this test, the robot is guided by one of the teammates through a completely unknown environment, which is or is like a restaurant settings. The guiding person takes the robot to three ordering places and two object locations (from where to retrieve a certain category of object, e.g. food or drinks), then return to the starting point. After the guiding phase, the robot is ordered to deliver three objects to two tables, simulating the robot being a waiter. So the robot is expected to drive to the object locations, look for and grasp the ordering objects, and take them to the correct tables. In such test, robots are requested to demonstrate more than one abilities, including human tracking, on-line SLAM, object recognition as well as grasping, making it a very challenging test.

After the two stages, the best five teams advance into the final, where each team is requested to perform an open demonstration and will be evaluated by both executive committee and an outer-league jury. More details of the competition are described in [9].

3 Hardware

The *KeJia* service robot is designed to manipulate common daily objects within an indoor environment. In order to move across narrow passages and avoid collisions with furniture, *KeJia* is equipped with a small sized chassis of 50×50 cm. The chassis is driven by two concentric wheels, each fixed with a motor of opposite direction. Another omni-directional wheel is mounted to keep balance and makes the robot able to turn. A lifting system is mounted on the chassis, attached with the robot's upper body. The vertical range of the lifter is about 74 cm, allowing the robot to manipulate objects in different heights, ranging from the floor to adult chest height. When fully lifted, the *KeJia* robot is about 1.6m high. Assembled with the upper body is a five degrees of freedom (DOF) arm. It is able to reach objects over 83 cm far from mounting point and the maximum payload is about 500 g (fully stretched). This arm enables the robot to grasp common daily items such as bottles and mugs. The robot's power is supplied by a 20Ah battery, which guarantees the robot a continuous running of at least one hour.

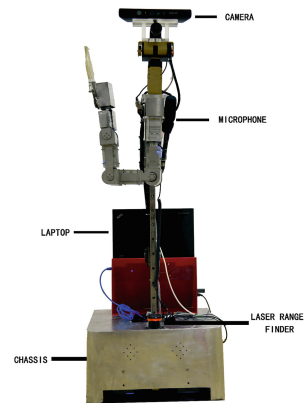


Fig. 1. Our robot KeJia

As for perception demand, our robot is equipped with a Kinect and a high-resolution CCD camera. These two cameras are mounted on a Pan-Tilt Unit, which provides two degrees of freedom to ensure the cameras to acquire different views. Two 2D laser range finders are installed for self-localization and navigation. These two sensors are mounted in different height, one close to floor, another one about 15 cm above ground. This configuration makes our robot able to detect and avoid small obstacles, e.g. bottles on the floor. A working station laptop is used to meet the computation needs. An image of our *KeJia* robot is shown in Fig. 1.

4 Perception

4.1 Self-localization and Navigation

Precise indoor localization and navigation is an important prerequisite for domestic robots. We use the 2D laser scanners and the Kinect as main sensors for localization and navigation. First, a 2D occupancy grid map [5, 7] is generated from the laser scanning results and odometry data. The scanning results are collected frame by frame and labeled with corresponding odometry information. In order to retrieve a full map of the working environment, our robot is driven by human control to visit all the rooms beforehand to get the full map. Each pixel in the grid map has a probability of occupancy, making it adaptive to potential changes in the environment. This feature is important as it allows the robot to deal with movable obstacles, such as chairs and walking people. With this pre-built occupancy map, a probabilistic matching technique is employed for localization. Here we use the adaptive Monte Carlo localization method [4], which uses a particle filter to track the pose of a robot against a known map.

As such representation does not provide any information of the topological structure of the environment, e.g. the rooms and their connective doors, we extend this grid map representation to contain such information. After the grid map is built, we annotate the different structures such as rooms, doors, furniture and other interested objects according to their semantic information. After the annotation, a topological map is automatically generated. This map is modeled as an undirected graph, vertices in the graph are rooms while the edges represent their connective doors or passages. With such topology, a layered path planning is implemented. First, the global path planner finds a connective path in the topological graph, starting from the start room (usually the current room the robot is in) to the destination room. This path consists of a series of key points. Between the key points within a room, a local path planner is used to search for the shortest path in the grid map. This is implemented via a heuristic A star algorithm. At last, during the navigation phase, VFH+ [12] is adopted for local obstacle avoidance, including both static and dynamic ones. A labeled map of our laboratory is shown in Fig. 2.

Beside the 2D localization and navigation, we also use a 3D depth camera, e.g. the Kinect, to avoid those obstacles not visible from the laser scanners. Here we use octo-tree [5] as the representation of 3D map. The acquired depth



Fig. 2. A labeled map of our laboratory.

images are aligned with the 2D localization results and converted into octo-tree structure, and is used for avoiding local obstacles during navigation.

4.2 Object Recognition

In order to achieve fast yet robust object recognition, our recognition system is designed as a pipeline approach. Two cameras are used, a high resolution CCD camera and a Microsoft Kinect, to obtain aligned RGB-D images as well as high quality RGB images. Both cameras are calibrated so we can directly get the correspondence between the images. Figure 3 shows the pipeline of our approach.

As shown in Fig. 3, first we use the Line-MOD [8] template matching method to find possible candidates in the RGB-D image. Using the non-maximal suppression (NMS), we select a smaller set of candidates in different regions of the image. As the Line-MOD method is characterized by a high recall low precision rates, this procedure can easily reject those regions where no possible targets are present. After this, all the regions are projected into the high quality RGB image, from where SURF [2] features are extracted and computed. These features are then matched against the database using a pre-built K-d tree. The matched features are grouped and filtered using RANSAC to check their geometry correspondence with the features extracted from known objects. If the number of inliers after RANSAC passes a certain threshold, then the object is treated as possible present. However, to overcome the weakness that SURF does not contain

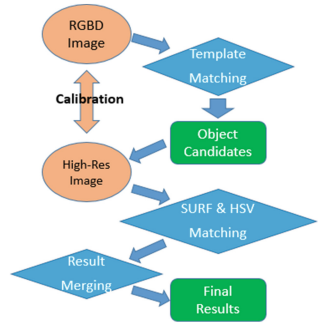


Fig. 3. The object recognition pipeline



Fig. 4. Left: results after template matching. Right: final results

color information, we compute a histogram from the object’s bounding box in the HSV color space, and use it to further decrease false detections. Please note that our recognition pipeline does not rely on surface extraction, which makes it able to detect objects in human hands or even in another robot’s gripper. An example matching result is shown in Fig. 4.

4.3 People Detection and Tracking

We developed a fast walking people detection method to efficiently detect standing or walking people. First the depth image is transformed into the robot’s reference frame, where the origin is the center point of the two wheels projected to floor. After the transformation, each pixel’s height in the depth image is its real height against the floor. Since standing people usually compose a fixed shape, we can use spatial information to quickly obtain such candidates. So we remove the floor plane and other uninterested areas in the depth image, leaving out isolated regions. Then we adopt a graph labeling algorithm to segment the image into multiple connected components based on the relative distance between pixels. Figure 5 shows the output of segmentation. After that, a pre-trained HOD [10] upper body detector is used to classify each component human or not. As the segmentation procedure filters out many irrelevant regions, the classifier does not need to be run on the whole image, gaining a significant acceleration. As for sitting people, we use HAAR [13] face detector to detect and localize human faces. If present, the *VeriLook SDK* is used to identify each face.

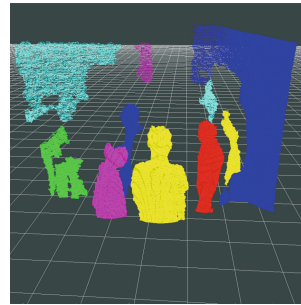


Fig. 5. A segmentation result

4.4 Speech Recognition

For speech synthesis and recognition, we use a software from iFlyTek¹. It is able to synthesis different languages including Chinese, English, Spanish etc. As for recognition, a configuration represented by BNF grammar is required. Since

¹ <http://www.iflytek.com/en/index.html>.

each test has its own set of possible speech commands, we pre-build several configurations to include all the possible commands for each test.

5 Integrated Decision-Making

One of the most challenging tests in the RoboCup@Home competition is the *General Purpose Service Robot*, where a robot is asked to fulfill multiple requests from an open-ended set of user tasks. In the *KeJia* robot, the integrated decision-making module is implemented using Answer Set Programming (ASP), a logic programming language with Prolog-like syntax under stable model semantics originally proposed by Gelfond and Lifschitz (1988). We pre-define and implement a set of atomic actions, where each action is implemented by a parameterized pre-defined program. All these actions are designed as primitives for the task planning module. With the specification of atomic actions, the division between the task and motion planning modules is clearly defined. Some of the atomic actions are listed in Table 1.

Table 1. Samples of atomic actions

Action	Function
Goto a location	Drive to the target location
Pick up an item	Pick up the assigned item
Put down at a position	Put down the item in hand at the assigned position
Search for an object	Search for the assigned object

An important feature of *KeJia*'s atomic actions is underspecifiedness, which provides flexibility of representation. In fact, for example, all phrases semantically equivalent to "search for an object" are identified by *KeJia* as instances of *findobj* action. Generally, people cannot afford to explicitly spell out every detail of every course of action in every conceivable situation. For example, the user may express a query like "bring me the drink on a table". If *KeJia* knows more than one table or drink in the environment, it will ask the ordering person for further information. To realize these features, *KeJia*'s task planning module should be able to *plan under underspecification* to generate underspecified high-level plans, and provides *KeJia* with the possibility of acquiring more information when necessary. Once new information is received, the task planning module should update its world model and re-plan if needed. For this purpose, non-monotonic inference is required. This is one of the main reasons that we choose ASP as the underlying inference tool. Also, ASP provides a unified mechanism of handling commonsense reasoning and planning with a solution to the frame problem. There are many ASP program solver tools which can produce a solution to an ASP program. The solver *KeJia* uses is *iclingo* [6]. A more detailed explanation of the decision-making module could be found in [3].

6 Grounding

In this section, we present our grounding system. The goal of this system is to map natural language queries to their referents in a physical environment, we decompose it into two sub-problems: *semantic parsing*, i.e. learning the semantics of natural language query, and *spatial relations processing*, i.e. learning to recognize objects and extracting a knowledge base containing nouns and spatial relations. Both kinds of knowledge are necessary to achieve one query’s grounding. For example, Fig. 6(a) and (b) show the input and results of the two processing modules. Through *spatial relations processing* module, the visual input of Fig. 6(a) is processed, the objects labeled and referred as numerical IDs. A knowledge base is also extracted which shown in Table 2. The query sentence in Fig. 6(b) is parsed by the *semantic parsing* module, which transforms the natural language query into logical form (semantic in Fig. 6(b)). Finally, both pieces of information are combined in the grounding system through logical rules which only consist of conjunctions and existential quantifiers to extract the grounding result (Fig. 6(c)). In case of ambiguity, the robot would ask the querying person to provide more information.

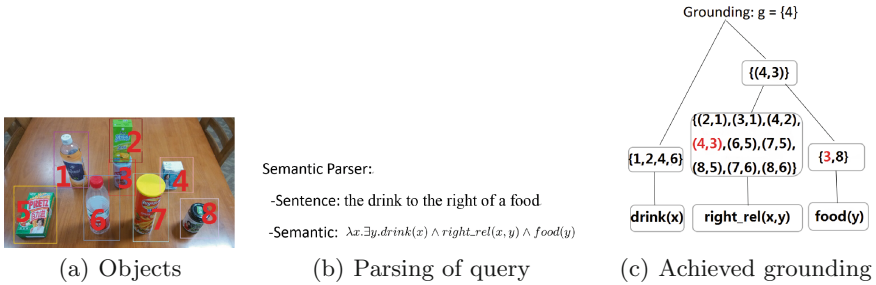


Fig. 6. An example process of the grounding system

6.1 Semantic Parser

The Semantic Parser is used to transform natural language query to internal logical form that the planning module can handle. The queries in our training data are annotated with expressions in a typed lambda-calculus language [1]. Our Semantic Parser is based on CCG [11]. A CCG specifies one or more logical forms for each sentence that can be parsed by the grammar. The core of any CCG is the semantic lexicon. In any lexicon, a word with its syntactic category and semantic form is defined. For example, the lexicon for our Semantic Parser would be as follows:

$$\begin{aligned}
 \text{box} &:= N : \lambda x. \text{box}(x) \\
 \text{food} &:= N : \lambda x. \text{food}(x) \\
 \text{right} &:= N/PP : \lambda f. \lambda x. \exists y. \text{right_rel}(x, y) \wedge f(y)
 \end{aligned}$$

Table 2. Extracted knowledge base

Object	Categories	Relations
1(amsa)	drink, bottle	behind:{5 ,6}, left:{2, 3, 4}
2(ice tea)	drink, box	behind:{6, 7}, left:{4}, top:{3}, right:{1}
3(porridge)	food, can	behind:{6, 7}, left:{4}, under:{2}, right:{1}
4(acid milk)	drink, box	behind:{7, 8}, right:{2, 3, 1}
5(pretz)	snack, box	before:{1}, left:{6, 7, 8}
...

the N/N	drink N	to (S/N)/N	the N/N	right N/PP	of PP/N	a N/N	food N
$\lambda f.f$	$\lambda x.drink(x)$	$\lambda f.\lambda g.\lambda x.g(x)\wedge f(x)$	$\lambda f.f$	$\lambda f.\lambda x.\exists y.right-rel(x,y)\wedge f(y)$	$\lambda f.f$	$\lambda f.f$	$\lambda x.food(x)$
						N: $\lambda x.food(x)$	
						PP: $\lambda x.food(x)$	
					N: $\lambda x.\exists y.right-rel(x,y)\wedge food(y)$		
					N: $\lambda x.\exists y.right-rel(x,y)\wedge food(y)$		
N: $\lambda x.drink(x)$		S/N : $\lambda g.\lambda x.\exists y.g(x)\wedge right-rel(x,y)\wedge food(y)$					
S : $\lambda x.\exists y.drink(x)\wedge right-rel(x,y)\wedge food(y)$							

Fig. 7. An example parse of “the drink to the right of a food”. The first row of the derivation retrieves lexical categories from the lexicon, while the remaining rows represent applications of CCG combinators.

In the above lexicon, *box* has the syntactic category *N* which stands for the linguistic notions of noun, and the logical form denotes the set of all entities *x* such that *box* is true. In addition to the lexicon, a CCG also has a set of *combinatory rules* to combine syntactic categories and semantic forms which are adjacent in a string. The basic combinatory rules are the functional application rules:

$$\begin{aligned}
 X/Y : f \quad Y : g &\Rightarrow X : f@g \quad (>) \\
 Y : g \quad X \setminus Y : f &\Rightarrow X : f@g \quad (<)
 \end{aligned}$$

The first rule is the forward (>) application indicating that a syntactic category *X/Y* with a semantic form *f* can be combined with a syntactic category *Y* with a semantic form *g* to generate a new syntactic category *X* whose semantic form is formed by λ -applying *f* to *g*. Symmetrically in the second rule, backward (<) application, generates a new syntactic category and semantic form by applying the right one to its left. Figure 7 illustrates how CCG parsing produces a syntactic tree *t* and a logical form *l* using combinatory rules.

When defining a CCG parser, we make use of a conditional log-linear model to select the best scored parses. It defines the joint probability of a logical form *Z* constructed with a syntactic parse *Y*, given a sentence *X*:

$$P(Y, Z|X; \theta, \Lambda) = \frac{\exp[\theta \cdot f(X, Y, Z)]}{\sum_{Y', Z'} \exp[\theta \cdot f(X, Y', Z')]} \quad (1)$$

Where Λ is the semantic lexicon, $f(X, Y, Z)$ is a feature vector evaluating on the sub-structures within (X, Y, Z) . In this paper we make use of the lexical features. Each lexical feature counts the number of times that a lexical entry is used in y . With the probabilistic model, the semantic parsing then turn to compute:

$$\operatorname{argmax}_Z P(Z|X; \theta, \Lambda) = \operatorname{argmax}_Z \sum_y P(Y, Z|X; \theta, \Lambda) \quad (2)$$

In this formalism, the distribution over the syntactic parses y is modeled as a hidden variable. The sum over y can be calculated efficiently by dynamic programming (i.e., CKY-style) algorithms. In addition, the beam-search during parsing is employed, in which sub-derivations with low probability under some level are removed.

6.2 Spatial Relations Processing

Given a preposition and landmark object from object recognition module, the *spatial relations processing* module first calculates the probability of each target object, and extract the 3D point cloud segment which is located at the given preposition in relation to the landmark object. Then a logical knowledge Base τ containing object instances and spatial relations is constructed.

Here six spatial relations are used: {above, below, in_front_of, behind, to_the_left_of, to_the_right_of}, where the positions are described in the robot's perspective, e.g. object A *in_front_of* object B means A is closer to the robot. We model these spatial relations using a probabilistic distribution that predicts the identity of a target object (or 3D point) T conditioned on a preposition W , and landmark object L . To obtain the spatial relations between a target object and a landmark object (except itself), our module computes the maximum probability on each relation: $\operatorname{argmax}_W P(T|W; L)$. A 3D point cloud segment is then extracted and its center position calculated as the pose of landmark object, which is denoted as (x, y, z) . So the probability of this spatial relation is:

$$\operatorname{argmax}_W P(T|W; L) = \operatorname{argmax}_W P((x, y, z)|W; (x', y', z')) \quad (3)$$

In the Eq. 3, (x, y, z) is the 3D pose of object T , (x', y', z') is the 3D pose of object L . we assume that \mathbf{V} is a six-vector represent {in_front_of, behind, to_the_left_of, to_the_right_of, below, above}. For example, $\mathbf{V} = (1, 0, 0, 0, 0, 0)$ is represent the $W = \textit{in_front_of}$, we also assume that $\overrightarrow{\textit{delta}} = ((x' - x), (x - x'), (y' - y), (y - y'), (z' - z), (z - z'))$. Now we can calculate the right side of the Eq. 3:

$$\operatorname{argmax}_W P((x, y, z)|W; (x', y', z')) = \operatorname{argmax}_W \mathbf{V} * \overrightarrow{\textit{delta}} \quad (4)$$

After the module obtains all spatial relations between landmark objects, a logical knowledge base τ is constructed. The knowledge base produced by perception module is a collection of ground predicate instances and spatial relations (see Table 2).

6.3 Demo in the Open Challenge

In the *Open Challenge* test, the grounding system was used to fulfill a task where ordering person did not know the names of the objects. The person used descriptive expressions to describe his intention, e.g. “The food to the left of a drink.” If such denotation was not clear, the robot would ask more questions to eliminate ambiguity. Finally, the robot understood human’s intention and retrieved the correct object for him. Here the logical rules described in Sect. 6 are as follows:

$$\begin{aligned}
 & \text{if } l = \lambda x.c(x) \text{ then } g = g^c \\
 & \text{if } l = \lambda x.\lambda y.r(x,y) \text{ then } g = g^r \\
 & \text{if } l = \lambda x.l_1(x) \wedge l_2(x), \text{ then } g(e) = 1 \text{ iff } g_1(e) = 1 \wedge g_2(e) = 1 \\
 & \text{if } l = \lambda x.\exists y.l_1(x,y), \text{ then } g(e_1) = 1 \text{ iff } \exists e_2.g_1(e_1, e_2) = 1
 \end{aligned}$$

7 Competition Results at RoboCup@Home 2014

At last we discuss our team’s performance at the competition. The scores of most tests are shown in Table 3. In the *Robo-Zoo* test, *KeJia* failed to attract most audience due to a not-that-friendly appearance, thus resulting in a very low score. As for *Basic Functionalities Test*, our robot met some problems in entering the door, but finished the other two sub-tasks afterwards. In the *Emergency* test, almost all teams failed to enter the door, making a nearly blank score column. But starting from the test *Follow Me*, *KeJia* began to show better performance. In the *Follow Me* test, *KeJia* succeeded in following the guiding person till the very end. It passed around intercepting human, entered and left elevator, and find the person again after she sneaked through a group of unknown people. This was the first time this test has been fully solved by participating team in RoboCup@Home. Team Nimbro from Germany also completed this test, but us finishing in shorter time resulted in a higher score. In the *Open Challenge* in stage I, our robot performed the grounding demo described in Sect. 6.3, showing it able to not only recognize objects, but also understand location relationships and reason with such knowledge. During the *General Purpose Service Robot* test in Stage II, *KeJia* successfully completed one command in which important information was left out, but failed to recover from audience noise when executing the second command. Then in the *Cocktail Party*, *KeJia* almost finished the whole test, showing the abilities of robust speech recognition, human detection and recognition, object recognition and grasping, as well as navigation through crowded rooms. *KeJia* achieved 1750 points from a total of 2000 points in this test, making another record of best single test score since *Cocktail Party*

Table 3. Scores of most tests (except for *Emergency*)

	Robo-Zoo	Follow-Me	BFT	Open	GPSR	Cocktail-Party	Restaurant	Demo
Our score	87	811	600	1507	750	1750	2600	450
Average	211.2	368.8	415	763.4	195	410	442.5	225
Best	500	811	800	1507	750	1750	2600	750

was set. In the last most challenging regular test *Restaurant* which has never been even half-solved, our robot was able to carry out the whole test smoothly, taking a big lead against all the other teams. This was the third record that *KeJia* broke during the competition. Finally, we got a total of 9305 points in the first two stages, with a clear advantage over the former champion team Nimbro (Germany, 5701 points) and the third team TU/e (Netherlands, 5656 points).

At last in the *Final*, two *KeJia* robots collaborated to serve the ordering person a cup of coffee with sugar, in which two robots executed parts of the task in parallel and cooperated with each other when opening the cap of a sugar can. This demonstrated *KeJia*'s ability of multi-agent task planning, motion planning and precise motion control. In the end we achieved a total normalized score of 97 points, followed by TU/e (79 points) and Nimbro (74 points).

8 Conclusion

In this paper we present our contributions of our team WrightEagle@Home to the RoboCup@Home competition 2014. Our robot *KeJia* is designed to accomplish tasks involved in environment perception, human robot interaction, speech understanding and reasoning. In the competition, *KeJia* has shown great abilities and good robustness thus winning the competition with a big lead. Furthermore, we have demonstrated our robot's high level understanding and reasoning functions, which means a good integration of AI techniques into robotics. In future work, we plan to increase the generality of *KeJia*'s object recognition and grasping abilities. We also aim at further increase *KeJia*'s cognition functions by utilizing internet open knowledge.

Acknowledgement. This work is supported by the National Hi-Tech Project of China under grant 2008AA01Z150, the Natural Science Foundations of China under grant 60745002, 61175057, and USTC 985 project. Team members beside the authors are Zhe Zhao, Wei Shuai and Jiangchuan Liu. We also thank the anonymous reviewers for their valuable comments and suggestions on the manuscript of this work.

References

1. Carpenter, B.: Type-Logical Semantics. MIT Press, Cambridge (1997)
2. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)

3. Chen, X., Xie, J., Ji, J., Sui, Z.: Toward open knowledge enabling for human-robot interaction. *J. Hum.-Rob. Interact.* **1**(2), 100–117 (2012)
4. Dellaert, F., Fox, D., Burgard, W., Thrun, S.: Monte carlo localization for mobile robots. In: *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1322–1328 (1999)
5. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. *Computer* **22**(6), 46–57 (1989)
6. Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Thiele, S.: Engineering an incremental ASP solver. In: Garcia de la Banda, M., Pontelli, E. (eds.) *Logic Programming. LNCS*, vol. 5366, pp. 190–205. Springer, Heidelberg (2008)
7. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE Trans. Robot.* **23**(1), 34–46 (2007)
8. Hinterstoisser, S., Holzer, S., Cagniard, C., Ilic, S., Konolige, K., Navab, N., Lepetit, V.: Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In: *IEEE International Conference on Computer Vision* (2011)
9. Holz, D., del Solar, J. R., Sugiura, K., Wachsmuth, S.: On robocup@home - past, present and future of a scientific competition for service robots. In: *Proceedings of the RoboCup International Symposium* (2014)
10. Spinello, L., Arras, K. O.: People detection in RGB-D data. In: *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3838–3843. IEEE (2011)
11. Steedman, M.: *Surface Structure and Interpretation*. MIT Press, Cambridge (1996)
12. Ulrich, I., Borenstein, J.: Vfh+: reliable obstacle avoidance for fast mobile robots. In: *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1572–1577 (1998)
13. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518 (2001)