2014 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS 2014)
September 14-18, 2014, Chicago, IL, USA

# Using Spatial Language to Drive a Robot for an Indoor Environment Fetch Task

Zhiyu Huo, Tatiana Alexenko, Marjorie Skubic

***Abstract*—This paper proposes a system that allows the use of natural spatial language to control a robot performing a fetch task in an indoor environment. The system processes spatial referencing language and extracts a tree structure of language chunks. The spatial language system is then grounded to a robot navigation instruction in the form of a sequence of actions based on spatial references to furniture and room structure; the best navigation instruction is selected by scoring. In addition, the Reference-Direction-Target (RDT) model is proposed to represent indoor robot actions. To control the robot for the fetch task, a behavior model is designed based on the RDT model. An assistive robot has been designed and programmed based on this system. The proposed spatial language grounding model and robot behavior model are tested experimentally in three sets of experiments. Results show that the system enables a robot to follow spatial language commands in a physical indoor environment even if the referenced furniture items are re-positioned.**

## I. INTRODUCTION

The aging population is becoming a challenge that will continue to stress the care of seniors in the future. The old-age dependency ratio in the United States was 0.20 in 2012 and will increase to 0.35 at 2050 [1][2]. In other countries, the situation is more severe. For example, the old-age dependency ratio in Japan was 0.39 in 2012 and is forecast to be 0.74 in 2050, which means four Japanese workers per three retired older people (not considering children) [1][2]. A shortage of labor leads to shortages of healthcare staff. This creates a need for assistive devices such as robots [3]. Surveys have shown that older adults would consider assistive robots for household tasks such as fetching and searching for missed objects [4]. Furthermore, older adults also prefer natural language rather than other communication methods for robot interaction. In this paper, we propose natural spatial language interface methods for communicating with a robot performing the fetch task. Here, we focus on the language translation and navigation of the fetch task. The grasping component is not included [13].

Robot spatial language understanding has been explored previously. Matuszek [5] proposes an idea to convert natural language commands to logic descriptions. Tellex et al. developed a probabilistic graphical model, named generalized grounding graphics, to derive the best grounding solution from natural language commands. It is realized on a forklift robot as a sequence of robot actions [6][7]. Kollar et al. developed an imitation learning policy to convert natural spatial language commands to sequential actions in an unknown environment. The method is tested on a simulation platform [8]. Fasola et al developed a model to generate a global path from using dynamic spatial relation references in a semantic map [9]. They assume the robot has a global knowledge of the working environment. Our work differs from the previous work in that different language structures are supported and we do not assume complete knowledge of the scene. Also, in this paper, we report test results with a real (non-simulated) robot in a physical environment.

Details of our proposed system are included. The next section discusses spatial language grounding, i.e., how to ground natural language chunks to a robot navigation instruction. The Reference-Direction-Target (RDT) model is proposed; a scoring procedure is used to find the best robot navigation instruction from the chunked natural language description. In addition, we introduce robot behavior models that support both dynamic and static spatial descriptions. Dynamic spatial descriptions use sequential actions such as "*go forward*" and "*turn left*" to navigate a robot to a target location. Static spatial descriptions use objects as references to describe a target location, i.e., "*behind the couch*" or "*on the table next to the bed*". The third section shows the design of an assistive robot and its perceptual capabilities. The fourth section presents the experiment and results in a one-bedroom and one-living room apartment environment. The experiment was run in the physical world, which means the performance is affected by both the spatial language grounding model, as well as the robot's perception and navigation capabilities. Finally, we conclude with discussion and future work.

## II. SPATIAL LANGUAGE GROUNDING

When using natural language for a spatial oriented task, people prefer to use relative spatial references rather than precise quantitative terms. For instance, to describe the position of a cellphone, people may say "*the cellphone is in the living room on the right on the table behind the couch*" rather than *"the cellphone is 3.21 meter from the living room door at a 45 degree direction"*. However, it is not as easy for the robot to understand such human-like descriptions. We refer to the procedure to translate a natural language description into a robot-understandable navigation instruction as grounding. In this paper, the natural language robot fetch command is first grounded (i.e., translated) to a robot navigation instruction and then executed by a pre-defined robot behavior model. To ground natural language to robot navigation instructions, we first use the method

Zhiyu Huo is with University of Missouri, Columbia, MO 65211 USA (e-mail: zhdy7@mail.missouri.edu)
Tatiana Alexenko is with University of Missouri, Columbia, MO 65211 USA (e-mail: ta7cf@mail.missouri.edu)
Marjorie Skubic is with University of Missouri, Columbia, MO 65211 USA (e-mail: SkubicM@missouri.edu)

discussed in [11] to extract a tree structure in the form of the best robot command match for each chunk and connect them together to form a robot action sequence.

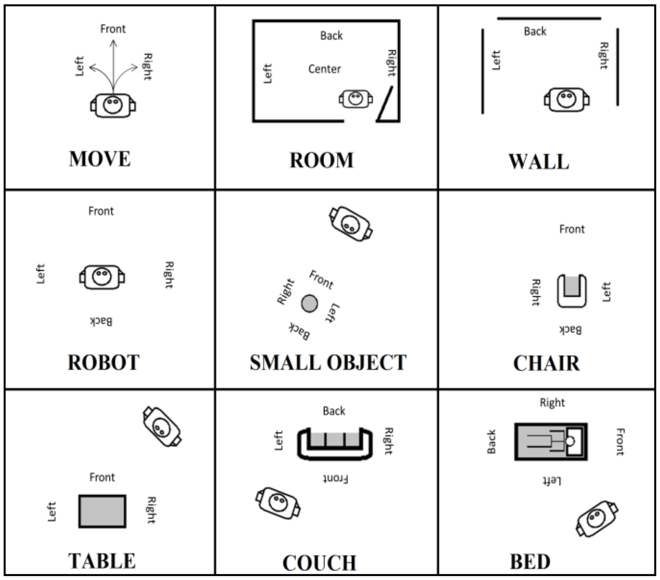


Fig. 1 Reference types and their corresponding directions, defined from human subject experiment [13]

### A. Fetch Task Model

In our fetch task, the robot is assumed to have prior knowledge of the room structure as this is fixed. However, we assume that the placement of furniture and daily objects inside the room is not known to the robot. In the fetch task, a human speaker stands in a hallway outside the target rooms and gives the robot a spatial description of the target object. The robot addressee then starts from the hallway, moves to the designated room, and then moves to the target object. The target objects are assumed to be on the surface of furniture items so there is no need to search inside furniture. The robot uses its local perception for navigation and object recognition in this task. The fetch task process is divided into three sub-tasks which represent three types of groundings which is a bridge between the natural spatial language description and the robot action:

(1) *Target Room*: Determine the target room and enter the correct room,

(2) *Inside-room Navigation instruction*: Move close to the target object by following the spatial description.

(3) *Target Object*: Find the target object designated by the speaker.

### B. Reference-Direction-Target (RDT) Model

The most difficult part of the task is to navigate the robot within the target room because the robot has no *a priori* information of furniture placement within the room, which may be changed by people who live there. The robot will use its own perception for navigation and object recognition. Guided by the human spatial language description, a robot can find a target object more efficiently than aimless language chunks. Then we use a scoring procedure to find searching. The Reference-Direction-Target (RDT) model is proposed which converts the inside-room spatial description into a series of actions with navigation instructions (grounding type 2, above).

In the RDT model, *Reference* refers to objects in the room, furniture or even room structure, e.g. wall and door. It can also be a label that informs the robot about the behavior type it should perform. Dynamic commands are defined as a special kind of reference type which has no real reference object but rather uses a sequence of moves, e.g., turn left, go forward. Such reference types are different from static command behaviors that need perception to find an object used for reference in navigation. Several types of references are used in the fetch task, as described below. These references are collected based on human subject experiment [13].

MOVE – This reference represents dynamic spatial language commands in which there is not a real reference object. For example, "turn right" or "go forward". There may be a target object for this reference type.

ROOM – This reference uses fixed room information in navigation, e.g., "move halfway in" or "to the left part of the room". The Direction component shows the possible part or direction of a room as destination. Because the room structure is not changed, by using a compass, odometry and prior knowledge of the room structure, the robot can move to the target area. In the experiment, the robot has a semantic room map with walls, and doors for the navigation.

WALL – A wall is used as the reference to define target position, e.g., "to the back wall". The robot will start searching once a target is in the RDT node.

ROBOT – When using the robot itself as the reference, it does not directly appear in the description, but rather ego-centric references are used, e.g., "behind you".

FURNITURE – A furniture item is used as a reference object. Based on previous work [10][11][12][13], the direction of the furniture reference follows human interpretation. For example, "in front of the chair" is defined using the intrinsic frame of the chair while the direction of the table and couch is defined by the viewing angle.

*Direction* represents the position relationship between objects. It tells the robot where it should move to search for the target. The direction of each reference type is defined based on human speakers' intentions. It may not be based on object ego-centric coordinates. For MOVE and ROBOT, the direction uses robot ego-centric coordinates. For FURNITURE, the direction sometimes uses viewer angle. For different types of navigation instructions, the reference frame for direction may be defined very differently. The directions used in the robot fetch commands include: front, left, right, back, central, side, and between. The Histogram of Forces (HoF) is used to represent the direction reference [19][20]. Fig. 1 shows the different reference types and their corresponding directions.

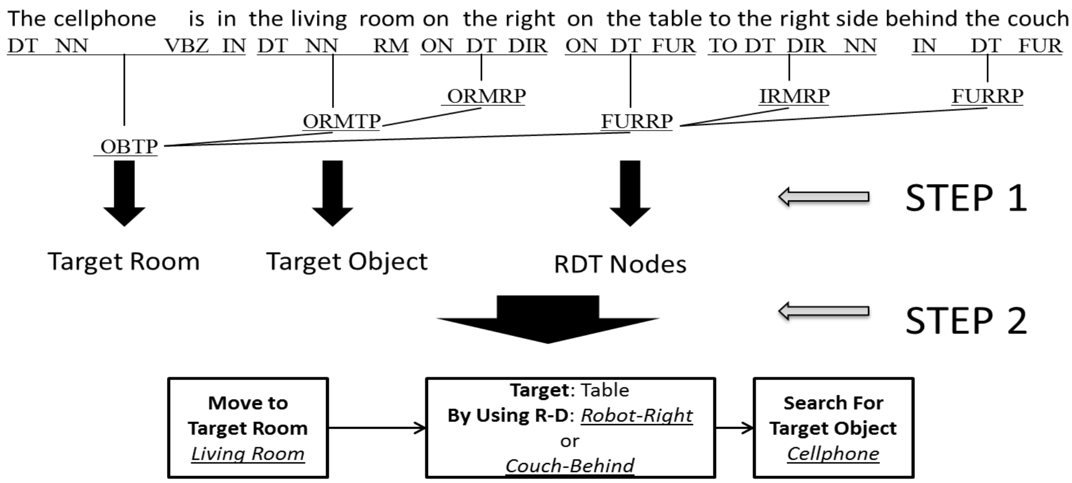


Fig. 2 Spatial Language chunking tree example

*Target* indicates the target furniture or target object in the navigation instruction. Sometimes there is not a target furniture word in the spatial language chunk. Often, the target furniture can be derived from content or human intention, usually, a table.There is a natural assumption that people usually put small objects on table-like furniture. An RDT node is built based on a target. It has one target and one or more reference–direction pairs because a speaker may use more than one reference to describe a target position. The FURTP chunk in Fig. 2 shows a multiple reference-pair example. If more than one reference-direction pair is given, the robot will skip remaining pairs once the target is found.

### C. *Grounding from Chunking Tree to RDT model*

The RDT model can support either dynamic or static spatial descriptions with the same framework. The input is a chunking tree extracted by part-of-speech tagging [11]; see Fig. 2. The tree is parsed by a forward direction traversing process through the tree. The result is an action queue.

To convert the chunking tree to a grounded navigation instruction (a sequence of robot actions), a scoring method is designed to find the maximum likelihood match for each chunk. It scores a chunk by two steps. First, it recognizes the grounding type. The grounding types include the target room, the inside-room navigation instruction, and the target object which represent the fetch sub-tasks. Then for the inside-room navigation case, the second step finds the *reference*, *direction* and *target* information (RDT node) of the chunk. Fig. 2 shows the procedure of a grounding example "***The cellphone is in the living room on the right on the table to the right side behind the couch***".

The scoring model is trained using spatial descriptions from a template corpus which summarizes the structure of 1024 collected spatial language descriptions for a robot fetch task [14]. There are 101 unique chunks which cover all the words for six target object fetch description sets. First, we manually label the grounding information of each element in the training chunks. Chunk elements include chunk tag, chunk text, parent chunk tag and children chunk tags.

Extracting the grounding type can be viewed as a classification problem. The final result is the grounding type with the highest score. For a sample of chunk $s$, the scoring equation for grounding type classification is:

$$T = Maximum_{tp}\ (\ P(\ t_p \mid tag_s\ )\ S(\ tx_s\ ,\ TX_{tp})\ )$$

$T$ is the result of the grounding type classification. $P(t_p \mid tag\ )\ S(\ tx\ ,\ TX_{Tp})$ is the score of the $t_p$ grounding type. $tag_s$ is the chunk name of the sample chunk $s$, and the definition of each kind of chunk name can be found in [26]; $txt_s$ is the text of the sample chunk $s$; $TX_{tp}$ is a corpus of template chunk text with chunk name$tag_s$ and belongs to grounding type $t_p$. $S(\ TX_s\ ,\ TX_{tp})$ is the degree of membership for $tx_s$ to $TX_{Tp}$ by weighted Levenshtein distance (WLD).

$$S(\ tx_s\ ,\ TX_{tp})\ ) = 1 - Min(\ WLD(tx_s\ ,\ tx_{tp})\ )$$

The result of the first step is shown in Fig. 2.

The following scoring equation is used in step 2.

$$G = Maximum_{gd}\ (\ P(\ g_d \mid tag_s\ )\ S(\ tx_s\ ,\ TX_{tp})\ \ldots$$
$$P(\ g_d \mid prt\_Tag_s\ )\ P(\ g_d \mid phn\_Tag_s\ )\ )$$

$G$ is grounding result. This equation can be used on all of the groundings in the RDT node, including reference, direction and target furniture/object. $g_d$ is all the possible groundings in a grounding type $T$ which is derived from step 1. $P(\ g_d \mid Tag_s\ )\ S(\ tx_s\ ,\ TX_{tp})\ P(\ g_d \mid prt\_tag_s\ )\ P(\ g_d \mid chn\_tag_s\ )$ is the score value of the grounding type. $prt\_tag_s$ is the parent tag of the sample chunk and $chn\_tag_s$ is the child tag of the sample chunk. Fig.2 shows the second step result for the example command.

A RDT node may be built using more than one chunk. In the example in Fig. 2, the FURTP chunk and its two nested child chunks build a single RDT node. The FURTP chunk is grounded to "target: table". The IRMRP is grounded to "reference: robot + direction: right". The FURRP is grounded to "reference: couch + direction: back". The groundings of IRMRP chunk and FURRP chunk both describe the position of the target table in FURTP.

### D. Robot Behavior Model

The robot behavior model is built using the result of the spatial language grounding. The basic behavior of the robot is to compute the best point that fulfills the navigation instruction requirement and then let the robot move to it. The higher tier is a global sequence of three subtasks. The lower tier is for the navigation within the room by RDT nodes. Dynamic descriptions and static ones are distinguished by using different state machine strategies. Because dynamic descriptions use no furniture as references or targets, they do not need furniture searching and detection behaviors. By using odometry with prior knowledge about the house structure and basic obstacle avoidance by range sensing (e.g., sonar), it is possible for the robot to move to the target location. However, the static command strategy requires the robot to search and recognize the reference and target items and because of the limitation on perception, the robot sometimes should move to an intermediate position to get a better view to improve its perception confidence. The system will try reference-direction pairs sequentially until the target is detected when there is more than one reference for a target. This is an improvement than previous work because it reduced the ambiguous in target searching and it then brought higher success rate in experiment results.

## III. Robot Design

### A. Robot Design

A mobile robot with the intelligence to navigate in an indoor environment and interact with a human has been designed and built to validate the performance of the method discussed in this paper. The robot has a differential drive chassis with an RGB-Depth camera.

A Pioneer 3-DX (P3DX) robot was used as the robot chassis and driving component [22]. The robot has a 16 unit sonar array, eight in front and eight in the back. The tower frame is made of light aluminum and holds a Kinect camera and a laptop computer. The Kinect is popular because it can provide high quality synchronized color and depth data [23]. Usually its effective detection range is from 0.5 meter to 8 meters which is adequate for indoor work. The controller of the robot is a laptop which runs the perception, robot behavior and human-robot interaction programs. The robot uses the Robot Operating System (ROS) [24][25], as a software platform. ROS provides libraries and tools to help software developers create robot applications. For robot navigation, we manually constructed map of room structure for robot and allowed it to use odometry for localization. However, the robot has no information about furniture items inside the room and it has to use visual perception to explore the furniture map.

### B. Visual Perception

The spatial language corpus collected for the fetch task uses furniture items as reference objects [11][14]. Thus, the robot perception for the fetch task consists of two parts which are furniture recognition and furniture pose detection. Our previous work discusses details on the furniture recognition and furniture orientation methods [11].

#### 1) Furniture Recognition

Furniture recognition provides category classification of a furniture sample, whereas furniture pose detection identifies the position and the orientation of a furniture piece. We used features of shape, size, height and color for the furniture recognition challenge, using both RGB and depth images. The furniture recognition results are good but not perfect, which lends a realistic perspective to the experiments.

#### 2) Furniture Pose Detection

Furniture pose detection includes both position and orientation. We use a grid map to represent the furniture positions and the robot because it retains the size and shape information which is used for the HoF-based spatial relations computation. The orientation is defined according to the different furniture categories. The orientation of table-shaped furniture is defined by the orientation of the short visible edge [11]. The orientation of chair-shaped furniture is defined by the orientation of the chair back [11].

## IV. Experiment and Evaluation

### A. Spatial Language Grounding Experiment

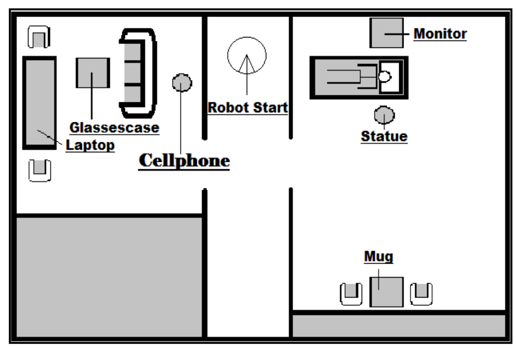


Fig. 3 Experimental environment with furniture and object placement

Three experiments are used to evaluate the proposed system. The first experiment evaluates the translation (grounding) of the spatial language chunking tree to robot navigation instructions. We used the template corpus with 149 spatial descriptions, which summarizes the structure of 1024 collected spatial language fetch descriptions [14].Even our ultimately goal is to let the robot can interact with human by natural talking, all the commands are input in text form in this experiment so that the evaluation can be independent from speech recognition. The descriptions were categorized by major syntactic differences across instruction type (how/where) and a function of landmark type (none, goal, path). The *how* commands were mainly dynamic (sequential actions), whereas the *where* commands contained more static descriptions [14]. For different landmark conditions, *none* means no furniture reference was used. A *goal* landmark included a spatial reference description of a table where the target was located. A *path* landmark means there was furniture used as reference in the description along the path to the target [14]. The object names and positions are shown in Fig. 3. The ground truth for this experiment was manually edited. We used 36 commands for training (six for each of 6 target objects) which were representative of both dynamic and static descriptions. All 149 descriptions were used for testing. The result is shown in TABLE I.

TABLE I Spatial language grounding experiment results (in %)

| Types and Landmarks | How vs. Where | | Goal vs. Path vs. None | | | Total |
|---|---|---|---|---|---|---|
| | How | Where | Goal | Path | None | |
| Successful Rate | 89.4 | 81.0 | 89.5 | 72.54 | 100.0 | 87.9 |

## B. *Robot Behavior Test*

The robot behavior model was evaluated in a two-room environment which has the same structure used for collecting the spatial descriptions. The room map and furniture and object placement are shown in Fig. 3. After the spatial language grounding procedure, there are 33 unique robot instruction combinations generated. To evaluate the robot behavior separately from the spatial language translation, this experiment used manually generated navigation instructions. In a fetch task, the robot is required to start from the hallway, enter the target room, then move along a path to the target furniture and take a picture of the target object. The robot state in each frame for each trial is recorded. An RGB image is taken with the robot's Kinect at the end of each trial. The criterion of success is that the target object is recognizable on the camera picture at the end of the trial. We ran both simulation and real robot experiments with improved robot behavior model. The results are displayed by landmark type in TABLE II with a comparison to the previous simulation experiment [14].

TABLE II Robot behavior experiment results (in %)

| Experiment Type | Goal | Path | None | Total |
|---|---|---|---|---|
| Simulation (Previous) | 89.5 | 40.0 | 98.0 | 84.6 |
| Simulation | 89.5 | 86.0 | 98.0 | 90.1 |
| Real Robot | 50.0 | 78.6 | 100.0 | 81.3 |

## C. *Robot Behavior Model Robustness*

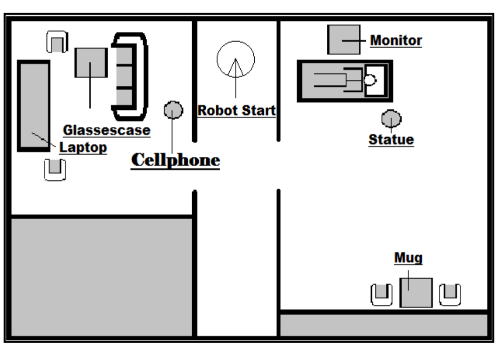


Fig. 4 Modified room placement for Experiment 3

The robot behavior model was further evaluated for robustness by changing the furniture placement in the scene. In real life, the furniture position may be changed a little bit without notice by people. Such a change usually does not affect spatial relations between furniture objects. Therefore, a robot should have the ability to keep an accurate spatial understanding with a slight furniture position change. A modified furniture placement of the rooms, as shown in Fig.4, was used to test the robot behavior model again using the same navigation instructions. The results of the two experiments are compared in TABLE III.

TABLE III Robot behavior model robustness experiment (in %)

| Experiment Type | Goal | Path | None | Total |
|---|---|---|---|---|
| Original Placement | 50.0 | 78.6 | 100.0 | 81.3 |
| Modified Placement | 33.0 | 78.6 | 100.0 | 78.1 |

Fig. 5 shows pictures of the scene and robot view for the fetch description: "***Go into the room on the left. Move about halfway in and then turn right. Go forward to the table against the wall with the chairs and there is the mug***." Fig.6 shows the robot path of the fetch task. The path consists of a set of purple short lines. Each line represents a robot position in the path. The slope of each short line is the robot orientation.

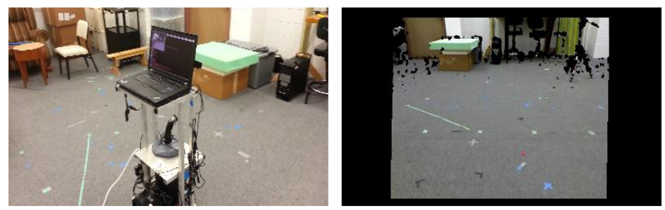

Fig. 5 The left image is a scene photo taken by an external camera. The right image show a view from robot Kinect camera.

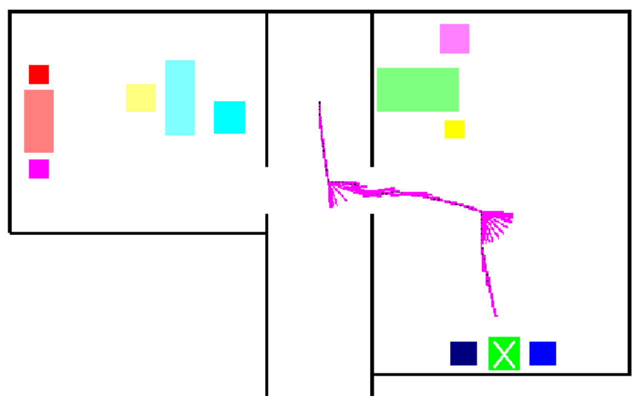

Fig. 6 The robot path for the fetch task, the X is the position of the target furniture.

# V. Conclusion

The work in this paper enables the robot to follow spatial language descriptions for a fetch task. The methods proposed here can be expanded to a standard spatial language understanding model. In the paper, we discuss a method to ground chunked spatial descriptions to robot navigation instructions. We defined a Reference-Direction-Target model which supports both dynamic and static spatial descriptions for indoor navigation instructions. With the RDT model and the HoF, which models spatial relations, the robot behavior can be built dynamically. A robot system was built to evaluate the system introduced in this paper. The spatial language grounding experiment shows good results for both dynamic and static descriptions, and includes improvement over the work in [14]. The robot behavior model experiment evaluated the basic method in a real world environment. However, the perception of the real world robot yielded a lower performance overall compared to the simulation experiment. In our result, the "*Path*" and "*None*" landmark type result is better than simulation due to an improved grounding algorithm in RDT node building. An analysis of the robot trace shows that the robot sometimes incorrectly detects furniture which results in the wrong reference and direction selection. This decreased the performance in the "*Goal*" landmark case. The robot behavior model robustness experiment demonstrated the robot is robust on small furniture position changes that retain basic spatial relationships between furniture items.

We will continue to improve the spatial language grounding system and the corresponding robot behavior model. Future plans include an experiment on a larger corpus collected from older adults rather than the templates. The grounding system will be improved for the end-user. Moreover, we will also improve the perception by building more precise furniture models to solve occlusion problems. We will continue simulated and real robot experiments to evaluate the robustness of the system in new room structures and with varying object placement. Our ultimate goal is to build reliable robot to assist elderly people in the home environment.

## ACKNOWLEDGEMENT

The authors gratefully acknowledge Laura Carlson, Jared Miller and Xiaoou Li for their contributions to our early work and human spatial description data collection.

## REFERENCES

[1] *Old-age dependency ratios*. The Economist. May 9th, 2009

[2] Age dependency ratio, old (% of working-age population). World Bank Data.

[3] J. Beer, M. Jenay, L. C. Tiffany, P. Akanksha, L. M. Tracy, C. K. Charles, & A. R. Wendy, "The domesticated robot: design guidelines for assisting older adults to age in place", *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on*, pp. 335-342, IEEE, 2012.

[4] M. Scopelliti, V. Giuliani, and F. Fornara, "Robots in a Domestic Setting: A Psychological Approach," *Universal Access in the Information Society, 4,* pp. 146-155, 2005.

[5] C. Matuszek, E. Herbst, L. Zettlemoyer, & D. Fox, "Learning to parse natural language commands to a robot control system", *In Proc. of the 13th Int'l Symposium on Experimental Robotics (ISER),* 2012

[6] S. Tellex, T. Kollar, S. Dickerson, M. Walter, A. Banerjee, S. Teller & N. Roy, "Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation," *Proc., Conf. on Artificial Intelligence (AAAI),* 2011.

[7] S. Tellex, P. Thaker, J. Joseph, and N. Roy, "Learning perceptually grounded word meanings from unaligned parallel data," *Machine Learning,* pp. 1–17, 2013.

[8] F. Duvallet, T. Kollar, A. Stentz, (2013, May), "Imitation learning for natural language direction following through unknown environments", *In Robotics and Automation (ICRA),* pp. 1047-1053. 2013 IEEE.

[9] J. Fasola, & J. Mataric, "Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots", *In Intelligent Robots and Systems (IROS),* pp 143-150, 2013 IEEE/RSJ.

[10] M. Skubic, Z. Huo, L. Carlson, X. Li, J. Miller, "Human-Driven Spatial Language for Human-Robot Interaction", *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[11] M. Skubic, T. Alexenko, Z. Huo, L. Carlson, J. Miller, "Investigating Spatial Language for Robot Fetch Commands", *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*. 2012.

[12] M. Skubic, L. Carlson; X. Li, J. Miller, Z. Huo, "Spatial language experiments for a robot fetch task", *Human-Robot Interaction (HRI), 2012 7th ACM/IEEE International Conference on. IEEE,* 2012.

[13] L. A. Carlson, M. Skubic, J. Miller, Z. Huo, and T. Alexenko. "Strategies for human-driven robot comprehension of spatial descriptions by older adults in a robot fetch task", *In Proc*.

[14] M. Skubic, Z. Huo, T. Alexenko, L. Carlson, and J. Miller, "Testing an assistive fetch robot with spatial language from older and younger adults." *In RO-MAN, 2013 IEEE*, pp. 697-702. IEEE, 2013.

[15] G. A. Radvansky, S. A. Krawietz, and K. T. Andrea, "Walking through doorways causes forgetting: Further explorations", *The Quarterly Journal of Experimental Psychology 64, no. 8 (2011),* pp. 1632-1645, 2011.

[16] G. A. Radvansky and D. E. Copeland, "Walking through doorways causes forgetting: Situation models and experienced space", *Memory & cognition 34.5 (2006),* pp. 1150-1156, 2006.

[17] C. R. Arkin, "Behavior-based robotics", *MIT press*, 1998.

[18] L. A. Carlson, and P. L. Hill, "Formulating spatial descriptions across various dialogue contexts", *Spatial Language and Dialogue 1.9 (2009)*, pp. 89-10, 2009.

[19] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, Magda Bugajska, and D. Brock, "Spatial language for human-robot dialogs", *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 34.2 (2004)*, pp. 154-167, 2004.

[20] P. Matsakis, and L. Wendling, "A new way to represent the relative position between areal objects", *Pattern Analysis and Machine Intelligence, IEEE Transactions on 21.7 (1999)*, pp. 634-643, 1999.

[21] R. Siegwart, and I. R. Nourbakhsh. "Introduction to autonomous mobile robots", *MIT press*, 2004.

[22] P3DX Robot Introduction, Internet: http://www.mobilerobots.com-/researchrobots/pioneerp3dx.aspx

[23] Leyvand, Tommer, et al. "Kinect identity: Technology and experience."Computer 44.4 (2011): 94-96.

[24] Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on open source software. Vol. 3. No. 3.2. 2009.

[25] ROS Wiki, Internet: http://wiki.ros.org

[26] Z. Huo. "Robot methods for human-robot spatial language interaction", Master Thesis, University of Missouri--Columbia, 2013.