Go, Look, and Tell: Natural Language Communication with a Ballbot

Matthew Wilson[1], Jean Oh[2], Ralph Hollis[2]

***Abstract*— Household and service industry robots will become much more useful when they can effectively communicate with humans and better understand the world. A natural speech interface and understanding of the world could enable robots to take on more general purpose use, as users could interactively specify tasks, and robots could provide feedback and semantic knowledge of the world in an intuitive way.**

In this paper, we describe an initial formulation and results for a natural speech interface on the ballbot, an agile and safe robot that balances on a single spherical wheel. We develop an end-to-end system for perceiving the world, responding to user questions, and receiving natural language commands to navigate in the world. We then demonstrate this system through an example indoor service scenario.

I. Introduction

Household and service industry robots will become much more useful when they can effectively communicate with humans and better understand the world. In contrast to the domain-specific tasks that service robots are limited to now, such as vacuuming, robots with a speech interface could be commanded to accomplish a wide variety of tasks. Speech could be used as an intuitive and high bandwidth interface between user and robot. A user could specify tasks; the robot could ask for clarifications of an ambiguous task, could provide feedback about the task, and could answer questions about world observations and other derived semantic knowledge of the world. With advances in natural language interaction, service robots could achieve more general purpose use—an analogous step up from a calculator to a general purpose computer.

Few robots today offer an intuitive interface for users to interact with them; completing tasks almost always involves explicit programming. This either limits the benefits of robots to expert users, or else only enables a trivial set of functionality. Many human-sized mobile service robots now have robust ability to autonomously navigate in indoor spaces. With growing object detection capabilities and semantic understanding of scenes being developed in the computer vision field [1], robots could offer great utility to users. However, currently, most advanced functionality is often unavailable to users because there are no comprehensive natural interfaces.

A promising method for giving commands and receiving information in a human-centric way is through speech, or natural language. Recent advances in voice recognition technology and its widespread adoption in the household, with products such as the Amazon Echo and Google Home, make this an exciting and promising area of research for service robotics and human robot interaction (HRI). Speech is a natural means of communication for humans, and has two large benefits over other methods for commanding robots: 1) a speech interface is intuitive; untrained users can quickly learn to use unfamiliar systems, and 2) it requires little cognitive load; it can be used with little active attention. With a comprehensive speech interface, non-expert users could interact with robots to complete useful tasks and gain world information, without having to be trained and without having to think much about it. These properties, coupled with a capable HRI robot platform, could lead to wide-spread and frequent daily use of household service robots.

Fig. 1: We develop a system to allow a user to naturally communicate with the ballbot (called Shmoobot, or Shmoo) to issue commands and receive information about the world.

This paper focuses on the system integration of a speech interface, mobility, object detection, and world modeling systems, on a dynamically stable mobile robot to enable control through natural language and communication of world observations. The system we develop enables the user to interact with the robot in a natural way to give commands and gain world information. This is meant to serve as an initial formulation of a more extended system that we believe, with more probabilistic modeling and data-driven components, could be used to generate a widely useful, general purpose service robot.

II. Related work

Most approaches to using natural language to command robots consist of three primary components: **1)** natural lan-

[1]Matthew Wilson is an undergraduate student at the University of Utah and conducted this research as part of the Robotics Institute Summer Scholars at Carnegie Mellon University, Pittsburgh, PA 15213 USA matthew.b.wilson@utah.edu

[2]Jean Oh and Ralph Hollis are with the Robotics Institute of Carnegie Mellon University, Pittsburgh, PA 15213 USA (jeanoh, rhollis)@cs.cmu.edu

guage interpretation, **2)** plan generation and execution, and **3)** world modeling [2]. In natural language interpretation, the recognized text is processed to determine the action or intent that the user specified, and to identify symbols, or words in the phrase that represent objects in the real world. These symbols are then "grounded" [3], where they are mapped to concrete objects (groundings) in the world. Much work has been done to develop methods for efficiently grounding symbols in large search spaces [4]–[8]. Next, plan generation and execution involves determining a way to execute a user specified action with respect to the grounded symbols in the command. For a navigation action, this would entail grounding the destination symbol, and any spatial constraint objects (*e.g.*, to the right of object X), to known objects in the world and planning a route through the environment. Finally, world modeling addresses the problem of maintaining a persistent representation or state of relevant concepts in the world. In many cases, this involves probabilistic beliefs of object classes and locations, such as in [9]. The world model is populated by processing sensor inputs, such as object detections from a vision system, and can be used to facilitate grounding by providing search candidates for the symbols referenced in natural language commands.

Another large thread of work has been done on indoor service robots interacting with people. Some classic examples include Shakey the Robot [10] and the museum robots, RHINO [11] and Minerva [12]. Several other works, beginning with [13], used gesture recognition for commanding robots to do certain tasks. Other interfaces commonly used are web applications and on-board touch screens. One recent example is the CoBot [14], a robot deployed for an office service scenario, that autonomously navigates through a building, and has a web interface and touch display for commanding it to do tasks, such as driving to different offices to deliver small items such as papers. These robots interact with users via traditional interfaces, but the lack of a speech interface downgrades their capability to facilitate effective communication with users. The museum robots have spoken components for guided tours, but they do not facilitate robust interaction. The CoBot has a few screen based interfaces, which the user must click through to issue commands.

At the intersection of these two threads, natural language interaction combined on indoor mobile service robots, there are also several mentionable works. In [13], they develop a system to provide natural language descriptions of the environment, for a robotic assistive wheelchair that is manually driven, so that it can form a semantic map of the environment. Unlike the current work, speech is just used to provide descriptions of the current environment of the robot so that it can learn groundings. In [15], they use natural language to command a PR2 robot to do various complex household tasks, such as pouring a cup of tea or making a bowl of ice cream. In this case, they use much more complex action primitives for doing tasks, and can represent more complicated sequence of actions. They, however, do not discuss methods for interactive communication.

This paper differs from previous work in two important ways: **1)** it utilizes speech to not only facilitate giving robot commands, but also for receiving the robot's gained knowledge of the environment, and **2)** it is done with a safe and agile robot, extremely well suited for HRI. None of the works we mention effectively explore the interaction of two-way communication for mobile indoor service robots. In the past, focus has primarily been on developing methods to issue commands, or use natural language to develop better semantic mapping and symbol grounding. This work develops methods for users to not only specify navigation commands to a robot, but also for the robot to answer queries about knowledge gained from commands. The user can specify areas to go to and observe, and additionally interface with the robot's world model to question the robot for knowledge that it has gained, such as where certain objects are or what is in certain locations. Conducting the work on the ballbot platform is also significant, because as discussed in Section III, the ballbot provides many advantages over other robots for interacting with humans.

TABLE I: Comparison of human, original ballbot, and Shmoobot dimensions

	Human (average male)	Original ballbot	Shmoobot (this paper)
Height (m)	1.7	1.7	1.2
Shoulder width or diameter (m)	0.45	0.40	0.28

III. Background on the ballbot

Critical in a robot that communicates and remains in close contact to humans is an ability to move safely in human environments. The system this paper describes is developed on a ballbot, an omnidirectionally compliant human-sized robot.

The ballbot, introduced in [16], is a robot that balances on a ball driven by two sets of parallel rollers. The drive system is termed Inverse Mouse Ball (IMB) drive, as inversely to old computer mouses which used a free-rolling mouse ball to move roller encoders and determine mouse displacement, the IMB uses four rollers to actuate the ball and keep the ballbot balanced. During operation, the ballbot uses an Inertial Measurement Unit (IMU) to sense the lean angle and motors to drive the rollers and actively keep its center of gravity over the point of contact with the ground.

The original ballbot is an omnidirectional robot, equipped with arms, and with dimensions approximating that of humans (see Table I). The ballbot used in this paper, called Shmoobot, is a slightly smaller version of the original ballbot, and currently does not have any manipulators.

The core advantages of the ballbot over other robots in human spaces are **dynamic stability** and **omnidirectionality**. Dynamic stability refers to the property of the ballbot that it must actively work to keep balance by actuating the rollers. Humans are also considered dynamically stable, as while standing, their muscles must actively work to keep them from falling over. In contrast to this, statically stable robots such as

Fig. 2: The smaller ballbot, called Shmoobot (1.2 m height, 0.28 m diameter) is shown in the foreground, without the camera or Amazon Echo Dot attached. The original ballbot (1.7 m height, 0.40 m diameter) is in the background.

the PR2 rest on a wide base with several supporting wheels and can maintain their pose without any effort. For basic robotic service tasks that have been the focus of much prior human robot interaction (HRI) work, dynamic stability adds an unnecessary control challenge. However, for interacting with humans in more complex ways, dynamic stability is largely beneficial. It enables a robot direct control over its center of gravity, allowing it to perform tasks impossible for many statically stable robots. By being able to shift its weight and by having omnidirectional mobility, the original ballbot, equipped with arms, can do tasks such as:

- **Help people out of chairs [17]**. This is infeasible for most robots with static bases, as they cannot lean or sustain large enough forces to help somebody out of a chair, without tipping over.
- **Lead someone by hand [18]**. This is possible with other robots, but not in a natural way, as they cannot move omnidirectionally or react to large forces, such as from somebody losing balance and applying a sudden jerk.

The Shmoobot does not have not any actuators to manipulate the world, so the advantages of the ballbot platform directly relevant to this paper are:

- **Safety.** While they are moving through an environment, ballbots can easily be pushed away with only a finger.[1] In collisions, other robots are susceptible to tipping or else can hurt a person by running into them.
- **Moving in most spaces a humans can.**[2] Ballbots can navigate in cluttered human environments [19], [20], can rotate in place, and move omnidirectionally. Other robots can easily be trapped in cluttered environments and be unable to rotate their wide bases.
- **Similar profile to human.** As previously discussed, the ballbots have similar dimensions to humans, allowing them to move easily in narrow human spaces and still interact at human height. Statically stable robots with a similar profile would be susceptible to tipping.
- **Moving in crowds of people.** Some work has been done in [18]. Other robots cannot move compliantly through a crowded area, and if they are stationary, become a static obstacle. Ballbots, if stationary, can easily be pushed out of the way.

[1] https://youtu.be/8BtDuzu2WeI?t=1m55s

[2] The ballbot can handle slopes and bumpy terrain, but is currently limited to approximately level spaces (no stairs, etc.).

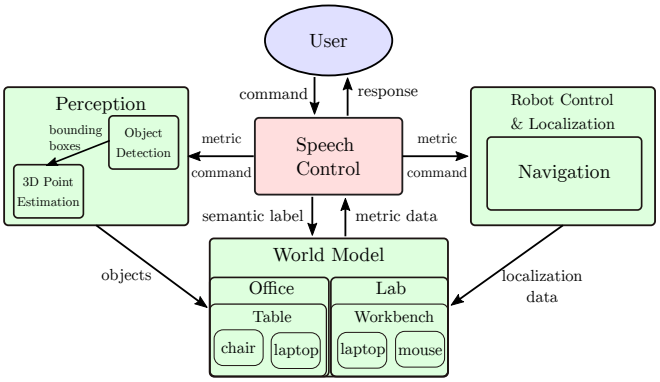


Fig. 3: Interaction of system components

Ballbots offer many of the same advantages that a fully humanoid bipedal robot does, and they have, at present, a more reliable mobility system for indoor spaces. Ballbots are a versatile platform for physical Human Robot Interaction (pHRI) and play a key component in the formulation of this system.

IV. Approach

To develop this system, we integrated and developed interfaces for perception and navigation components, using a world model to maintain relationships of metric and semantic labels so that they could be accessed by a central speech interface. A diagram illustrating user interaction and communication within the system can be seen in Figure 3. The user interacts directly with the speech interface using natural language, with semantic queries and commands such as "go the office." The command verbs (*e.g.*, "go") get mapped to a discrete set of robot actions in the speech control, and the semantic information (*e.g.*, "office") in these interactions is resolved into metric information by the world model (*e.g.*, coordinates $(x = 3, y = 3)$). Thus, the user can give natural commands that get executed with respect to the metric data of discrete objects in the world.

A. Platform

The Shmoobot is running the Robot Operating System (ROS) [21] and this is used to facilitate communication between all software components. For navigation, the Shmoobot is equipped with a Hokuyo UTM-30LX laser rangefinder and uses the ballbot navigation stack, developed in [19], [20]. For vision, the Shmoobot is equipped with an on-board Orbbec Astra RGB-D camera, which produces 640×480 size RGB and depth images. For a speech interface,

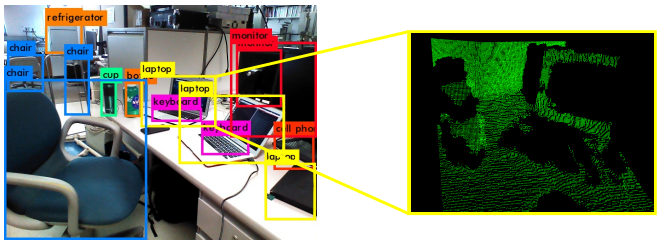


Fig. 4: The bounding box for the laptop, provided by the object detector [22], is used to segment the corresponding 3D point cloud data. These segmented 3D points, shown in green, are then used as an estimated 3D location of the object, where the centroid of the points is assumed to be the center of the laptop. This method has some inaccuracies when bounding boxes include pixels not part of the object, or when the detected bounding boxes do not directly map to the point cloud points, as seen with points from the water bottle, table, and wall, being included in the calculation of the laptop centroid.

it uses an on-board Amazon Echo Dot to send raw audio waveforms to Amazon servers and receive processed text via a custom Amazon Alexa skill.

B. Natural language processing

The speech interface receives processed text from an on-board Amazon Echo Dot and parses this processed text to a discrete set of **actions** or **queries**, along with **semantic parameters**. Actions specify the tasks that the robot can execute, *e.g.*, move and observe. Queries are questions that the robot can respond to. Semantic parameters are natural labels for concepts in the world (such as "office" or "keyboard") and are resolved by the robot's world model into metric data that the central speech control can use to execute actions to detect objects, move in the environment, or respond to user queries. The speech interface is the primary point of contact between the user and the knowledge structure and capabilities of the robot.

C. World model

For the robot to be able to reason and create more intelligent relations of the world, we developed the following hierarchical world model, including **1) locations**, **2) surfaces**, and **3) objects**. For example, objects such as a laptop would be on the surface of the table, inside the location of the office. This is one possible natural representation for humans and supports intuitive commanding and querying. A user can tell the robot to go to a specific room and look what is on a table in that location.

The representations of these objects, namely the semantic label and information associated with underlying metric data, are saved in a non-relational database structure. This representation allows programmatic querying by either the metric data (such as location, or radius of locations) or by semantic label (*e.g.*, bottle, office, etc.). The world model is key to integrating the perception and speech components with the control of the robot system. It resolves the semantic labels in the speech interface to metric data that is used to plan trajectories and to change the robot's orientation.

D. Perception

The perception module uses the on-board RGB-D camera to detect objects and generate estimates of 3D locations of objects to add to the robot's world model. The module uses the YOLOv2 object detector [22] to detect a discrete set of the objects from the MSCOCO data set [23]. The output of the detector is a set of labels, associated probabilities of the labels, and bounding boxes in a 2D image. The bounding boxes are then used to segment point cloud data from the RGB-D camera to obtain the matching points for each object, as shown in Figure 4. The corresponding 3D points from each bounding box are used to calculate an estimated centroid for each object. This allows the object detections to be grounded in global 3D space. The advantage of this approach over other common methods is that it does not require full 3D models of all objects. The disadvantage, illustrated in Figure 4, is that the segmentation based on bounding boxes is imperfect and points from other objects or the background can be included in centroid estimation.

E. Navigation

For a grounded location in a command, the system provides coordinates of a goal position and sends this to the ballbot navigation stack. The navigation stack takes in the current pose of the robot and a goal position, and plans a path to execute trajectories of continuous, smooth motion, taking into account the unique dynamics of the ballbot. The navigation can quickly replan trajectories and is robust to dynamic obstacles [20].

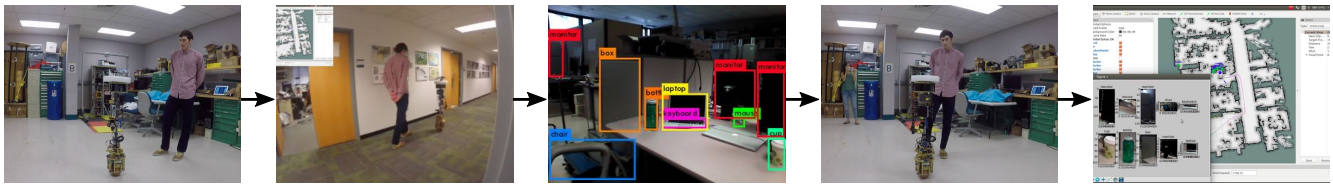
Several experiments were conducted where the ballbot received commands with respect to locations in the world. Location groundings were straightforward deterministic mappings, stored in the world model, that could be told to the robot (*e.g.*, by saying "you are in lab"), or be loaded from a configuration file. We did not conduct any experiments where the ballbot would navigate to objects it has previously seen in the world, but this is well within the capabilities of the system.

V. Experiment

We develop an indoor service scenario as a proof of concept demo and showcase of the system. This scenario consists of a user commanding the robot, in spoken English, to go to another room, look what is on a table there, and come back to tell the user what it saw. The flow of each of these commands through the system is described below.

A. "Go to the office"

The first command is spoken to the robot and enters the system through the speech interface. The speech interface maps the phrase "go to", to the navigation action. It then passes the semantic parameter "office" to the world model to resolve into metric coordinates. If no "office" is found in the world model, the system will give immediate feedback



(a) User commands Shmoobot to, "Go to the office, looks what's on the table. Come back to the lab."

(b) Shmoobot autonomously navigates to the office. The person is only following in case of a navigation system failure.

(c) The robot takes a picture and depth cloud image of the table, and uses the YOLOv2 object detector to detect bounding boxes.

(d) The robot autonomously navigates back to the lab. The user asks the robot, "Shmoo, what have you seen?"

(e) The robot responds with a list of items ("I have seen a bottle, 2 monitors, a keyboard, etc."). The system displays images and 3D coordinates of the objects.

Fig. 5: Indoor service scenario

to the user, by speaking, "I don't know where the office is." Also, because the navigation may be a long running goal, it is added to a command queue and subsequent commands in sequence can be given to the robot to execute.

B. "Look what is on the table"

The second user command follows a similar flow through the system. The phrase "look what" is mapped to an observe action. The phrase "on the table" is passed as a semantic parameter which similarly gets resolved by the world model to metric coordinates. With these coordinates, the robot turns its body in that direction and takes a picture to detect objects on the table and place them into the world model. Similarly, it will give immediate feedback if no surface, such as a table, is known.

C. "Come back to the lab. That's all."

Just as in the first command, this is interpreted as a navigate action, and the semantic parameter "lab" is resolved into the coordinates of the lab in the world model—in this case, near the starting place of the robot.

"That's all" is one of the phrases to command the ballbot to execute a series of queued commands. Upon hearing this command, the robot will ask for confirmation by speaking back the queued commands. If the user confirms the actions, the robot will then execute them.

D. "What have you seen?"

Finally, when the robot completes its tasks and returns to the lab, the user can ask what the robot has observed. The phrase "What have you seen" is mapped to a *query* with no parameters. This query can also be invoked at any time. This pulls from the world model all the objects that have been detected. These are spoken to the user (in a phrase something like "I have seen two monitors, a laptop, etc."), and images of objects and their corresponding 3D locations on a map can be displayed on an offline computer screen. In this example, the query did not have any parameters, but the system also allows the user to query with semantic parameters, such as an object label or room. For example, "Did you see a laptop?" or "Have you seen a laptop?" will map to a query with parameter "laptop," and cause the robot to respond with "I have seen one laptop" and show an image and map location of the detected laptop. One can imagine this functionality could be used to find lost things, or a number of related tasks.

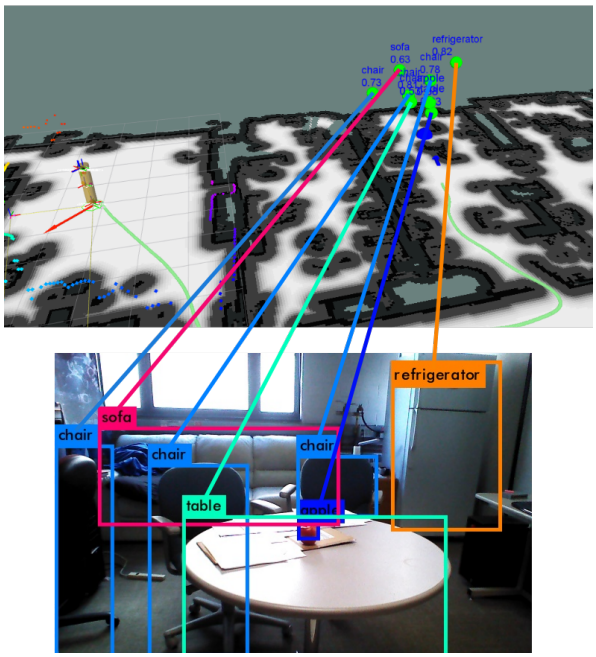


Fig. 6: Detected objects, along with their estimated global 3D locations, are added to the world model. The world model keeps track of the locations and other derived metadata of the objects, such as the semantic location (determined by metric coordinates). The user can then query the world model for these objects at any time.

VI. Conclusion

We have developed an end-to-end system for a dynamically stable mobile robot to receive natural language commands and communicate world observations. This system, unlike previous work, allows the user to command a robot to actively search an environment and to ask questions about gained world knowledge. It is also done in an indoor service environment on a robot well suited for safe interactions with humans. The shmoobot ballbot used in this study is particularly adept since it can move and navigate through the environment at a fast walking speed, has omnidirectional compliance; has a gearless low noise friction drive; and, unlike a traditional statically stable mobile robot, it is gravity-referenced which affords a stable platform for video input.

This is an initial formulation of the system, and with the

framework outlined, there are many potential extensions of this work.

VII. Future Work

Whereas the demonstrations were simple, in that the surfaces and locations were pre-loaded into the world model, the system could allow more intelligent sources of input. Further methods for determining room location and location of a table as well as probabilistic representations and methods for interpreting speech commands could be implemented, as in [4]–[8].

A. Language grounding and spatial reasoning

The system described in this work is limited to discrete, deterministic mappings between symbols in commands and objects in the real world. With a way to handle ambiguities arising from multiple objects of the same label and of representing the spatial relation of objects in the world, a more natural interface could be developed. The user could specify actions with phrases, such as "the table to the right of the door," using learned spatial relations as in [24].

More intricate label grounding could also be used in navigation commands. With probabilistic grounding of symbols and learned spatial relations applied to a cost map as in [24], the robot could be commanded "go to the left of the person, to the table."

B. Multi-hypothesis probabilistic world model

Another important extension to improve the robustness and adaptiveness of this system, is in probabilistic world modeling. The object detector and location pipeline produced some uncertainties in both the labels and positions of objects which were largely ignored in our system. These data could be used to maintain a multi-hypothesis state of the system as in [9]. This would be more adaptive to movement of objects and time variance. Furthermore, it would be necessary in integrating continuous real-time object data.

Acknowledgments

This work was supported by an NSF REU award and by NSF grant IIS-1547143. Thank you to Michael Shomin and Roberto Shu for their work on the ballbots and help in conducting experiments.

References

[1] A. Karpathy, "Connecting images and natural language," Ph.D. dissertation, Stanford University, 2016.

[2] R. Liu and X. Zhang, "A review of methodologies for natural-language-facilitated human-robot cooperation," *arXiv preprint arXiv:1701.08756*, 2017.

[3] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.

[4] T. Kollar, S. Telle1, M. R. Walter, A. Huang, A. Bachrach, S. Hemachandra, E. Brunskill, A. Banerjee, D. Roy, S. Teller, and N. Roy, "Generalized grounding graphs: A probabilistic framework for understanding grounded language," *Artificial Intelligence Research*, 2013.

[5] T. M. Howard, S. Tellex, and N. Roy, "A natural language planner interface for mobile manipulators," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[6] I. Chung, O. Propp, M. R. Walter, and T. M. Howard, "On the performance of hierarchical distributed correspondence graphs for efficient symbol grounding of robot instructions," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[7] R. Paul, J. Arkin, N. Roy, and T. Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators," in *Proc. Robotics: Science and Systems*, 2016.

[8] A. Boteanu, T. Howard, J. Arkin, and H. Kress-Gazit, "A model for verifiable grounding and execution of complex natural language instructions," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[9] J. Elfring, S. van den Dries, R. van de Molengraft, and M. Steinbuch, "A model for verifiable grounding and execution of complex natural language instructions," in *Proc. IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[10] N. J. Nilsson, "Shakey the robot," *SRI International*, 1984.

[11] W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1-2, pp. 3–55, 1999.

[12] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *International Journal of Robotics Research*, vol. 19, no. 11, pp. 972–999, 2000.

[13] M. Walter, S. Hemachandra, B. Hamberg, S. Tellex, and S. Teller, "A framework for learning semantic maps from grounded natural language descriptions," *International Journal of Robotics Research*, vol. 31, 2004.

[14] M. Veloso, J. Biswas, S. Rosenthal, S. Brandao, T. Mericli, and R. Ventura, "Symbiotic-autonomous service robots for user-requested tasks in a multi-floor building," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[15] D. Misra, J. Sung, K. Lee, and A. Saxena, "Tell me dave: Context-sensitive grounding of natural language to manipulation instructions," in *Proc. Robotics: Science and Systems (RSS)*, 2014.

[16] T. Lauwers, G. Kantor, and R. Hollis, "A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[17] M. Shomin, J. Forlizzi, and R. Hollis, "Sit-to-stand assistance with a balancing mobile robot," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[18] M. Shomin, "Navigation and physical interaction with balancing robots," Ph.D. dissertation, Carnegie Mellon University, May 2016.

[19] M. Shomin and R. Hollis, "Differentially flat trajectory generation for a dynamically stable mobile robot," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[20] ——, "Fast, dynamic trajectory planning for a dynamically stable mobile robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[21] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[22] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[23] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollr, "Microsoft coco: Common objects in context," 2014.

[24] J. Oh, A. Suppe, F. Duvallet, A. Boularias, J. Vinokurov, L. Navarro-Serment, O. Romero, R. Dean, C. Lebiere, M. Hebert, and A. Stentz, "Toward Mobile Robots Reasoning Like Humans," in *Proc. AAAI Conf. on Artificial Intelligence*, 2015, pp. 1371–1379.