Integrated Intelligence for Human-Robot Teams

Jean Oh[1], Thomas M. Howard[2], Matthew R. Walter[3], Daniel Barber[4], Menglong Zhu[5], Sangdon Park[5], Arne Suppe[1], Luis Navarro-Serment[1], Felix Duvallet[7], Abdeslam Boularias[6], Oscar Romero[1], Jerry Vinokurov[1], Terence Keegan[9], Robert Dean[9], Craig Lennon[10], Barry Bodt[10], Marshal Childers[10], Jianbo Shi[5], Kostas Daniilidis[5], Nicholas Roy[8], Christian Lebiere[1], Martial Hebert[1], and Anthony Stentz[1]

[1] Carnegie Mellon University, Pittsburgh, PA
[2] University of Rochester, Rochester, NY
[3] Toyota Technological Institute at Chicago, Chicago, IL
[4] University of Central Florida, Orlando, FL
[5] University of Pennsylvania, Philadelphia, PA
[6] Rutgers University, New Brunswick, NJ
[7] Ecole Polytechnique Federale de Lausanne, Switzerland
[8] Massachusetts Institute of Technology, Cambridge, MA
[9] General Dynamics Robotic Systems, Westminster, MD
[10] U.S. Army Research Laboratory, Adelphi, MD

Abstract. With recent advances in robotics technologies and autonomous systems, the idea of human-robot teams is gaining ever-increasing attention. In this context, our research focuses on developing an intelligent robot that can autonomously perform non-trivial, but specific tasks conveyed through natural language. Toward this goal, a consortium of researchers develop and integrate various types of intelligence into mobile robot platforms, including cognitive abilities to reason about high-level missions, perception to classify regions and detect relevant objects in an environment, and linguistic abilities to associate instructions with the robot's world model and to communicate with human teammates in a natural way. This paper describes the resulting system with integrated intelligence and reports on the latest assessment.

1 Introduction

As robots become commonplace in a variety of domains ranging from manufacturing to the military, there has been growing interest in the development of intelligent robots that can support humans not only as tools, but also as teammates. To be a competent teammate, *e.g.*, to perform a screening mission illustrated in Figure 1, a robot needs to have basic cognitive abilities including perceiving the semantics of its environment, reasoning about spatial relationships, and communicating with natural language. In this context, while the subfields of robotics and artificial intelligence have been extensively evaluated according to standard metrics accepted within each research community, little work has been done to-date that gauges the current state-of-the-art for an intelligent robot with cognitive abilities. For example, the computer vision community has mainly focused on improving performance on benchmark data sets as opposed to addressing the types of real world challenges faced in robotics [21, 12].

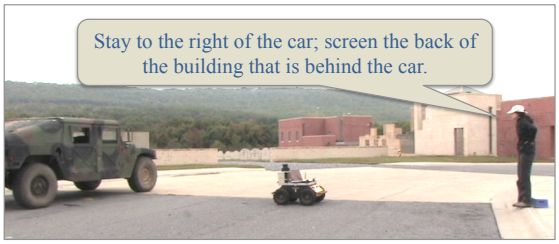


Fig. 1: An example showing a Clearpath^{TM} Husky unmanned ground vehicle working with a human teammate on a screening mission in an unknown environment.

As a result of such disconnections, the majority of existing works in intelligent (or cognitive) robotics includes simplifying assumptions, *e.g.*, ideas are verified in simulated environments or a robot's perception is assumed to be perfect or is simplified in order to measure the intelligence without including errors due to imperfect perception [22, 10, 9, 17, 6]. In our work, we aim to assess where the technology stands and where technology gaps are in the development of an intelligent robot teammate by integrating various pieces of technologies needed for a robot to perform tactical behaviors autonomously without adding simplifying assumptions. In this paper, we focus on semi-urban outdoor navigation and search behavior.

Toward this goal, we develop an intelligence architecture and integrate relevant technologies including state-of-the-art perception modules on a robot platform to assess robot intelligence at the tactical behavior level. Specifically, the capabilities that have been integrated to support intelligence are the following: 1) multi-modal interface to support rich interaction with humans,* 2) semantic world model, 3) high-level mission planning, 4) object detection,* 5) door detection,* 6) human detection and tracking,* 7) scene classification, 8) building (stuff) detection, 9) object prediction beyond sensor ranges, 10) natural language grounding,* 11) object symbol grounding, 12) (global and local) path planning, 13) imitation learning for navigation modes, and 14) an interaction layer for mobile robots. We note that the architecture builds on our prior work [19], augmented with new capabilities (marked with *). We describe our approach and share the lessons we have learned from recent assessment.

2 Technical Approach

Figure 2 shows an architectural diagram of our intelligence system for a robot teammate. In this section, we briefly illustrate how various modules contribute and interact within this architecture to support high-level robot intelligence.

Because our goal is focused on robots that can work with humans, it is important that robots be able to communicate in ways that are natural and efficient to humans. In our system, the interaction between a robot and a human is supported by a Multi-Modal Interface (MMI). Using this interface, a human teammate can issue commands through natural language speech and hand gestures, and review the robot's reasoning process via annotated camera images and semantic maps.

The world model is a central storage of information that is accumulated and merged from various modules. The information stored in a world state includes robot pose data, sensor data, semantic objects, multi-layered cost maps, com-

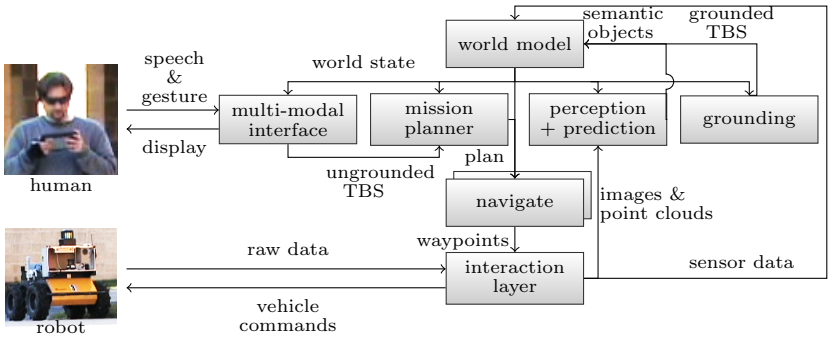


Fig. 2: An architectural diagram of integrated intelligences for human-robot teams.

mands, and various action status. The world model supports a query interface for the modules to look up relevant information.

The mission planner takes a command and reasons about pre- and post-conditions of available actions to find a plan that will accomplish the task specified by the command. For instance, given a command "Screen the back of the building," a set of actions needs to be performed in a sequential order; *i.e.*, the robot needs to navigate to the back of the building, locate a door in the back of the building, and then monitor the area near the door to report upon anyone's egress from the building.

The core of the intelligence system consists of *perception*, *prediction* and *language understanding*. These units contribute to the robot's understanding of its environment and enable it to interpret and execute a given natural language command. The *perception* module translates the raw data from the robot's sensors into semantically meaningful information (e.g., semantic scene classifier, an object detector, a door detector, and a human detector). The *prediction* module enables the robot to infer a world model for the unseen parts of the environment, effectively compensating for limitations in the robot's sensing range (as well as possible perception errors) by using prior information about object models or descriptions of objects specified in the natural language command. The *language understanding* module translates a spoken utterance into a structured representation, known here as Tactical Behavior Specification (TBS) [19], that formally represents the task and its constraints, and computes symbol grounding results [3]. Combined together, these modules enable the robot to robustly perform complex tasks in an unknown environment.

2.1 Human-Robot Interface (HRI)

The effectiveness of human-robot teams is intrinsically linked to the efficiency of bi-directional communication. Robots must be able to transform human forms of expression (*e.g.*, language and gesture) into a meaningful representation *and* communicate their understanding and actions to humans in order to share a cognitive model of mission goals and objectives. To address these challenges, we developed a MMI based on a Toughpad FZ-M1 tablet (Figure 3). This device enables a human teammate to command the robot through a combination of speech and gestures and receive robot status from the visual display and auditory cues. The MMI represents instructions to the intelligence architecture using the TBS lexicon.

Fig. 3: An illustration of the Multi-Modal Interface (MMI) for human-robot interaction. The MMI accepts input in the form of speech and/or gesture and visualizes the state of the intelligence architecture. The MMI Visual Display illustrates a "screen the back of the building" command. The robot status shown in the COMMANDS and STATUS sections indicate the command is still running and the robot is currently searching.

Grounding natural language to a TBS in the MMI is performed by the Hierarchical Distributed Correspondence Graph (HDCG) [4, 11, 2]. This model searches a pair of graphical models to efficiently translate natural language into a TBS command. The first graphical model is used to infer a set of rules to construct a more efficient representation of a second graphical model that is used to infer a distribution of the physical meaning of each phrase. To characterize the performance of the HDCG in this application, we measured the average run-time of symbol grounding for the natural language expression "screen the back of the building that is behind the car." Over 100 queries on a MacBook Pro with a 2.6 GHz Intel Core i7 processor, we observed that the model required 0.131 seconds on average to correctly translate the expression to a valid TBS command.

2.2 Common World Model (CWM)

The Common World Model (CWM) [5] defines and instantiates the data model for the intelligence architecture, providing a common, centralized intelligent data storage services. The world model is divided into three main concepts: Metric (sensor data and aggregates), Semantic (class descriptions and instances), and Self Information–data relative the robot, *e.g.*, pose data. At the Semantic level, objects represent symbolic information, enabling abstract reasoning needed for intelligent behavior. Here, CWM maintains semantic information from perception modules, and provides methods for client modules, *e.g.*, the navigate action, to search for semantic objects that are relevant to a specific mission context with a set of filtering criteria.

2.3 Mission Planner

The goal of the mission planner is to take commands in the mission vernacular from a teammate (via the MMI) and convert them into a sequence of actions (TBSs). We leverage recent work in ACT-R [1] on models of instruction following in the form of decision graphs, where the decisions themselves are made based on examples of past decisions in the form of Instance-Based Learning [13]. This research uses a single model of decision-making in which more instructions and examples can be included in the system in the form of "chunks"—ACT-R representations of semantic information. The goal of this new model is to provide

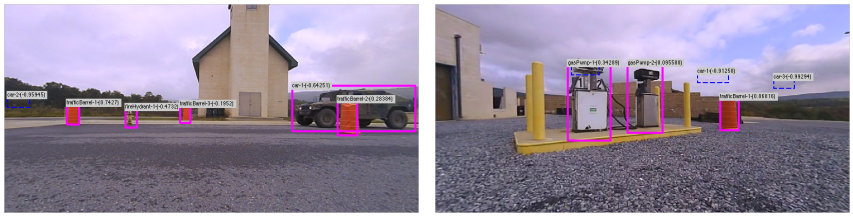


Fig. 4: Examples of object detection. Final detections are shown as red solid rectangles and rejected false positives as blue dashed rectangles.

increased flexibility in adding new examples to the model, which, in turn, allows the model to plan for new missions, as well as in combining generalizations from multiple examples.

2.4 Perception

We first describe four sensor-based perception modules in our system. Additionally included in this section is perception through prediction.

Semantic Classifier An online scene labeler is used to find buildings, vehicles, traffic barrels, and fire hydrants, and to classify background, *e.g.*, trees, asphalt, concrete, gravel, or grass as shown in Figure 6. Our approach builds on the Hierarchical Inference Machine [18], a scene labeling method that decomposes an image into a hierarchy of nested superpixel regions. Rather than perform inference on a graphical model, which can be expensive, we instead train a decision forest regressor with 10 trees and the segmentation hierarchy of depth 7 for predicting label distribution. We use SIFT [16], LAB colorspace statistics, and texture information derived from convolving the image with a bank of spatial filters, in addition to statistics on the size and shape of a superpixel region. We process a 640×384 image in approximately 2 seconds on a dedicated quad-core i7-3615QM at 2.3 GHz, with feature extraction being the dominant cost.

Object Detector We employ an Active Deformable Part Models (ADPM) method [24] for on-board object detection on our system. ADPM is an accelerated DPM that dynamically schedules parts and prunes locations in a cascade framework. With the current MATLAB/C++ implementation, ADPM simultaneously detects 5 classes on a 10MP image at 0.5Hz on a modern CPU. ADPM employs a sliding window approach at multiple image scales to detect objects at different positions and distances. In order to reduce the number of false positives, the detection hypotheses are further pruned using LADAR measurements as shown in Figure 4.

Door Detection Detecting doors imposes a unique challenge because doors undergo severe perspective distortion under different viewpoints. Based on the intuition that doors should be seen as a rectangle at a frontal (canonical) viewpoint, each façade candidate is mapped to the image domain according to the known calibration of each sensor. We preprocessed each candidate façade for door detection as follows: façade regions in the image are rectified using the estimated plane orientation in 3D and resized to a fixed scale such that the rectified façades are (virtually) observed at a fixed distance. Due to the canonicalization,

Fig. 5: Examples of door detections are shown. Façade detection and door candidates are shown on the left. Final detection output is shown on the right.

the pose and scale variation of doors in the façades can be eliminated. On top of the rectified façades, a Deformable Part Model based door detector [8] is applied. Since the façades are standardized in canonical view and fixed distance, detection can be performed online because searching in a single scale space is sufficient to detect doors as seen in Figure 5.

Human Detection One of the main objectives of the human-robot team is to identify potential human threats, which would feed directly into the observe action as the architecture is currently laid out. A tree-structured Deformable Part Model [23] was chosen as the state-of-the-art algorithm to perform this task. Given a rectified image, the algorithm reports the locations of 26 individual parts for each detected person. Our contribution is to port the feature pyramid processing code to run on a FPGA or GPU while the rest of the code runs as a module on a separate laptop. Using the current system architecture, streaming 1020×768 images from the camera, and processing all scales within the human detection algorithm runs at a 0.5 Hz processing rate. Additional LADAR processing is included within the observe action to better discriminate humans from other arbitrary objects.

2.5 Object Prediction

In addition to those approaches that use actual sensors to detect objects or humans in the robot's environment, we also utilize language inputs to perceive objects, primarily in the part of the environment that the robot has not directly explored. The current approach hypothesizes an object when two conditions are met: symbol grounding fails to map a symbol to an object in the world model; and there are areas that satisfy the spatial constraints but have not been explored by the robot. Given a language phrase l that describes a target object with spatial constraints relative to a reference object o, we sample a set of candidate locations from a discretized 2D map defined in $X \times Y$ space. A predicted object is created in an unseen location (x, y) that best satisfies the given spatial constraints: $(x, y) = \arg\max_{(x',y') \in X \times Y} k(x', y')\phi(x', y', l, o)$, where $k(x, y)$ is a binary indicator with value 0 for free space (i.e., no detection) that has been visited, 1 otherwise; and ϕ is a function that represents how well a given location (x, y) satisfies the spatial constraints l relative to a reference object o.

2.6 Structured Command Grounding

The symbol grounding algorithm takes as inputs a TBS command and a set of semantic objects in the world model, and grounds each object symbol referenced

in the TBS to an object instance in the world model. Spatial constraints specified in the TBS are evaluated in a robot-centric manner, *i.e.*, a spatial relationship relative to the position of the robot at the time when the command was given.

We first use a log-linear model to represent the probability that an object in the environment satisfies a given spatial relation. Given an object, this probability is defined as a function ϕ of weighted sum of the object's spatial feature values. The spatial features used here include the distances and the angles between the centers of objects and the robot. A weight vector of each relation is learned by maximizing the log-likelihood of all the training examples using gradient descent with the l_1 regularization. For details, we refer to previous work [3].

2.7 Actions: Tactical Behaviors

An action implements a specific tactical behavior of a robot. Currently supported actions include: navigate (Figure 6), search, observe (Figure 7), bump, go-to-xy, and wait; here, we describe *navigate* as an example.

Navigate Semantic navigation [20] differs from path planning with regards to the expressiveness of its command, as shown in Figure 6. In contrast to the go-to-xy action, for instance, where a goal is specified in map coordinates, a destination can be described using its spatial relationships with landmarks in an environment. Additionally, a navigation mode can also be specified to instruct a robot to move quickly or more covertly depending on the characteristics of a mission.

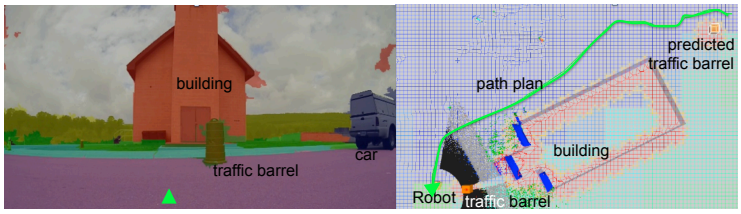


Fig. 6: Navigate: Given a command, "Stay to the left of the building; navigate quickly to the back of a traffic barrel that is behind the building," a robot navigates to the left of the building toward a hypothesized goal, a traffic barrel in the back of the building.

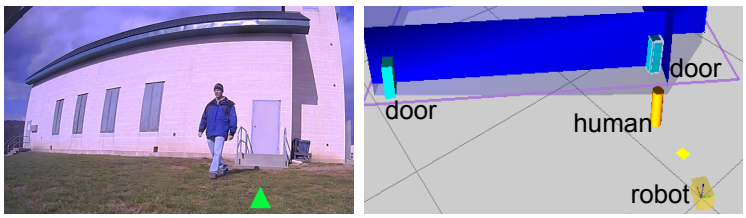


Fig. 7: Observe: a static, focused action where the robot registers human detections and reports them to the world model. Once the observe action starts running, it begins listening to the output from the human detector that is already sending human detection messages.

3 Experimental Results

To assess the ability of the intelligence architecture to use different capabilities, the system was tested in various mission scenarios. A human teammate used speech and gestures to command each mission through the MMI, and the robots performed the mission autonomously for the entire duration. We evaluated the robot's performance both via human assessment and via comparisons against human performance on similar tasks.

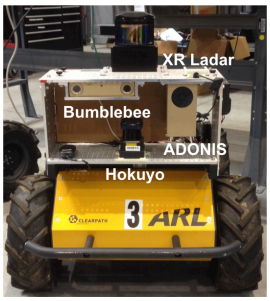


Fig. 8: An experimental setup: Two replica of Clearpath™ Husky unmanned ground vehicles equipped with the General Dynamics XR 3D LADAR sensor and Adonis camera were used.

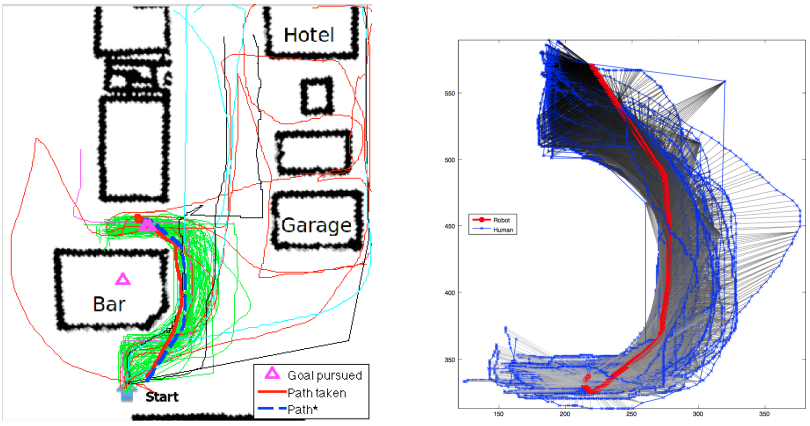
IDs	Runs	Site	Task (%)	Time (min)	Dist. (m)	Weather	Errors
V1	6	Bar	87	5.8	36.4 ± 0.5	3 rain, 3 sun	2 comm.
V2	4	Church	80	5.5	52.7 ± 2.1	3 sun, 1 cloud	grounding
V3	4	Church	75	3.5	23.0 ± 0.0	1 sun, 3 cloud	2 software
V4	3	Bar	93	5.7	31.3 ± 1.5	cloud	1 battery
2013	20	Various	50			snow, ice	various

Table 1: Results on the four vignettes involving navigation (against results from 2013).

3.1 Evaluation by human experts

Performances on screening missions: The complete runs involved two building sites, the Church and the Bar, requiring the robot to navigate 20–60 meters to achieve the mission. Total of 17 runs were graded on a 0–100 scale by increments of 20. Table 1 contains the overall human evaluation. When compared with an earlier performance, there has been a significant improvement. In previous results, on a similar set of navigation tasks, the average completion rate was 50% (where only 30% received full scores) [14]. Overall, the system consistently executed the screening mission, with 11 of 17 runs scored at 100%. Of the remaining runs, 3 failed due to low batteries or software crashes, 2 because of the communication system, and 1 because of a symbol grounding error.

Performances on semantic navigation: Table 2 summarizes the experiments from two distinct outdoor environments. The first set of experiments was conducted as part of a larger system assessment in a physically simulated town with 12 buildings in $1\,\mathrm{km}^2$ outdoor space in a military training facility. A qualitative summary from this set of experiments was reported in [15]. This set of experiments consisted of 57 runs that are 2 replications of 30 commands divided into 12 vignettes–*i.e.*, world configurations.



(a) The paths chosen by a robot and 82 human users are shown.

(b) Frechét distance between the paths chosen by a robot and humans

Fig. 9: Given a command "Navigate to the back of the building," this example compares a robot's navigation path against those of 82 human users.

The second set of experiments was carried out in a parking lot of a large, irregular-shaped building. The background in this environment was natural but highly cluttered. In the vignette where the robot was facing the large building, the robot performed poorly because there were many unknown objects on which the recognition algorithm had not been trained. The performance in those vignettes involving known objects was highly reliable, resulting in the average completion of 100% and 86% in the complete and the incomplete information cases, respectively.

Environment	Complete information	Incomplete information
Simulated Town	94 ± 13 (18)	81 ± 20 (36)
Building Outdoor	100 ± 0 (7)	86 ± 26 (13)

Table 2: Outdoor semantic navigation completion rate (%) with complete vs. incomplete information (The number of runs is in parenthesis).

3.2 Evaluation against human performance on similar tasks

According to our preliminary data collection on 20 subjects, human interpretation of a verbal instruction can vary significantly. Given a simple command, "go to a barrel that is in the back of the building," 20% (4 out of 20) of the subjects interpreted the command differently from the commander's intention, and the paths chosen by the majority who chose similar goal positions as the commander also varied.

Motivated by this result, we have collected a larger set of user data on interpreting navigation directions. We created a human intelligence test (HIT) on Amazon Mechanical Turk to collect the navigation paths selected by humans

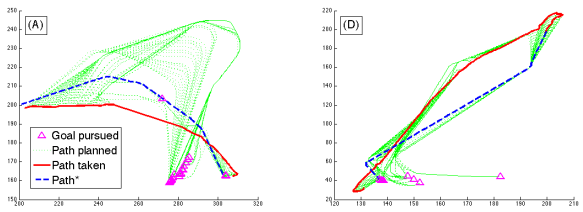


Fig. 10: Navigation paths with complete vs. incomplete information.

for a similar set of problems to the robot's. Two out of 84 data entries were eliminated due to incompleteness.

To compare the paths generated by a robot against that by a human, we used Frechét distance [7] that measures the distance between two curves. We sort the entries according to their choice of a goal landmark and their mode of navigation, *e.g.*, left or right of a building. We computed Frechét distance between the robot's path and the paths taken by the group of users who had made the same grounding decisions as the robot. We note that, in all 6 examples, the robot's grounding choice agreed with that of the human majority.

Path comparison: For each human turker, we computed Frechét distance between the path chosen by the human and that of the robot. In addition, we randomly selected another human turker and computed the distance between the paths chosen by the two human participants. The mean and the standard deviation of the Frechét distance for the example shown in Figure 9 between the paths chosen by a robot and 69 human users who have chosen the same building as their landmarks (drawn in green lines) are 56.79 ± 14.37, whereas those between human users in that same group were 67.70 ± 83.19. The t-test failed to reject the null hypothesis that there is no significant difference when a human generated path is compared against that of a robot or a human; the confidence interval at the 0.05 significance level was $[-34.29, 12.48]$ on the example in Figure 9.

Task-level performance comparison: When evaluated based on the intended goal and landmark groundings, the accuracy of human participants was 68.9%. People performed better on path constraints, reaching 86.9% in accuracy. We also asked the participants to evaluate the paths generated by a robot given the same set of navigation commands. Based on the evaluation of 82 participants, the robot scored 86.0%.

4 Main Experimental Insights

Our approach takes advantage of additional information conveyed within verbal commands by a human teammate to improve a robot's perception. Figure 10 shows progressive changes in the robot's navigation plans as the robot drives from a partially known world to a known world by gradually acquiring information through perception. The blue dotted line shows the path that the robot would have taken if it had complete information about the environment at the time when the command was given; the red line is the actual path that the robot has taken; the green lines and magenta triangles show the paths and the goals, respectively, that the robot pursued during execution. In these runs, the robot's

early goals may not be precisely correct (because they were the hypothesized goals as opposed to those perceived) but generally guide the robot to a proper direction so that the robot can revise its plan for the actual goal when detected. These examples illustrate that the paths taken by the robot under incomplete information strongly resemble those that would have been taken under complete information. Our experimental results show that, in outdoor navigation, semantic understanding of an environment is still challenging and exploiting information from verbal directions can compensate significantly.

In our previous experiments, the performance has been assessed only in terms of task completion as shown in Table 1. Here, we also evaluated the robot performance by surveying human participants on similar navigation tasks. Our experiments suggest that the paths generated by the robot resemble closely those generated by humans and that the robot performs comparably with humans.

5 Conclusion

In this paper, we present an intelligence architecture for human-robot teams that has been fully integrated into a mobile robot platform. During extensive assessments on various screening missions, the system performed consistently and robustly, demonstrating the strength of integrated intelligence. We conclude that combining the latest perception technologies and reasoning about complex surroundings with additional capabilities, such as natural language understanding to follow instructions from teammates or predicting unseen environments beyond the ranges of sensors, can lead to a viable robot teammate for implementing high-level intelligence in real environments.

Acknowledgments

This work was conducted in part through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016, and in part by ONR under MURI grant "Reasoning in Reduced Information Spaces" (no. N00014-09-1-1052). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. J. R. Anderson, D. Bothell, M. D. Byrne, S. A. Douglass, C. Lebiere, and Y. Qin. An integrated theory of the mind. *Psychological Review*, 111:1036–1060, 2004.
2. D. Barber, T. M. Howard, and M. R. Walter. A multimodal interface for real-time soldier-robot teaming. In *Proc. SPIE 9837, Unmanned Systems Technology XVIII*, 2016.
3. A. Boularias, F. Duvallet, F. Oh, and A. Stentz. Grounding spatial relations for outdoor robot navigation. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 1976–1982, 2015.
4. 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. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 5247–5252, 2015.
5. R. Dean. Common world model for unmanned systems. In *Proc. SPIE 8741, Unmanned Systems Technology XV*, 2013.

6. F. Duvallet, M. R. Walter, T. M. Howard, S. Hemachandra, J. Oh, S. Teller, N. Roy, and A. Stentz. Inferring maps and behaviors from natural language instructions. In *Proc. Int'l. Symp. on Experimental Robotics*, pages 373–388. Springer, 2015.
7. T. Eiter and H. Mannila. Computing discrete frechet distance. Technical report, Christian Doppler Laboratory, Vienna University of Technology, 1994.
8. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part Based Models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.
9. D. Golland, P. Liang, and D. Klein. A game-theoretic approach to generating spatial descriptions. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, pages 410–419, 2010.
10. N. Hawes, M. Klenk, K. Lockwood, G. S. Horn, and J. D. Kelleher. Towards a Cognitive System that Can Recognize Spatial Regions Based on Context. In *Proc. AAAI Conf. on Artificial Intelligence*, 2012.
11. S. Hemachandra, F. Duvallet, T. M. Howard, N. Roy, A. Stentz, and M. R. Walter. Learning models for following natural language directions in unknown environments. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, pages 5608–5615, 2015.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Neural Information Processing Systems*, pages 1097–1105, 2012.
13. C. Lebiere, F. Jentsch, and S. Ososky. Cognitive models of decision making processes for human-robot interaction. In *Proc. Int'l Conf. on Virtual, Augmented and Mixed Reality*, pages 285–294, 2013.
14. C. Lennon, B. Bodt, M. Childers, R. Dean, J. Oh, and C. DiBerardino. Assessment of navigation using a hybrid cognitive/metric world model. Technical Report ARL-TR-7175, Army Research Labs, Jan. 2015.
15. C. Lennon, B. Bodt, M. Childers, R. Dean, J. Oh, C. DiBerardino, and T. Keegan. RCTA Capstone Assessment. In *Proc. SPIE 9468, Unmanned Systems Technology XVII*, 2015.
16. D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
17. C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox. Learning to Parse Natural Language Commands to a Robot Control System. In *Proc. Int'l Symp. on Experimental Robotics*, 2012.
18. D. Munoz. *Inference Machines: Parsing Scenes via Iterated Predictions*. PhD thesis, The Robotics Institute, Carnegie Mellon University, 2013.
19. 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.
20. 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*, pages 1371–1379, 2015.
21. S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Neural Information Processing Systems*, 2015.
22. S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. J. Teller, and M. Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proc. AAAI Conf. on Artificial Intelligence*, pages 1507–1514, 2011.
23. Y. Yang and D. Ramanan. Articulated Pose Estimation using Flexible Mixtures of Parts. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1385–1392, 2011.
24. M. Zhu, N. Atanasov, G. J. Pappas, and K. Daniilidis. Active deformable part models inference. In *Proc. European Conf. on Computer Vision*, pages 281–296. Springer, 2014.