Natural Language Command of an Autonomous Micro-Air Vehicle

Albert S. Huang*, Stefanie Tellex*, Abraham Bachrach*, Thomas Kollar*, Deb Roy, Nicholas Roy
Massachusetts Institute of Technology
Cambridge, MA

***Abstract*— Natural language is a flexible and intuitive modality for conveying directions and commands to a robot but presents a number of computational challenges. Diverse words and phrases must be mapped into structures that the robot can understand, and elements in those structures must be grounded in an uncertain environment.**

In this paper we present a micro-air vehicle (MAV) capable of following natural language directions through a previously mapped and labeled environment. We extend our previous work in understanding 2D natural language directions to three dimensions, accommodating new verb modifiers such as *go up* and *go down*, and commands such as *turn around* and *face the windows*. We demonstrate the robot following directions created by a human for another human, and interactively executing commands in the context of surveillance and search and rescue in confined spaces. In an informal study, 71% of the paths computed from directions given by one user terminated within 10 m of the desired destination.

I. Introduction

Micro-air vehicles (MAVs) have many applications for surveillance and search-and-rescue in indoor environments. Aerial vehicles have the unique ability to reach vantage points inaccessible to most ground vehicles, a valuable asset in environments with elevated features. However, operating an indoor micro-air vehicle currently requires a specially trained and highly skilled operator.

We present a system that relaxes this requirement by providing a natural language interface for specifying motion trajectories to the MAV. If the human issues a command such as "Fly up to the windows," the robot infers and executes a corresponding path through the environment. This interface allows the operator to flexibly and naturally describe a three dimensional path. In such scenarios, language has a strong advantage over other interaction modalities. Humans need less training to interact with the robot and can keep their hands and eyes free for other tasks. Our system supports interactions with the robot such as the following:

Operator: Fly past room 124 and look at the windows.
Robot: *Takes off, flies to the windows past room 124.*
Operator: Go up.
Robot: *Ascends, while transmitting video.*
Operator: Go back down.
Robot: *Descends back to the operator's level.*
Operator: Come back towards the tables and chairs.
Robot: *Flies towards the tables and chairs.*

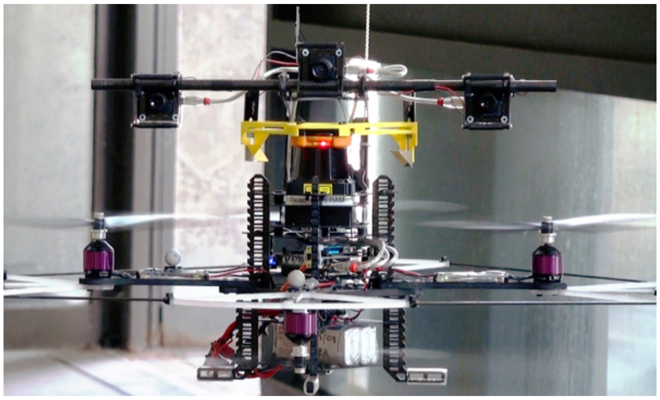

Fig. 1. Our research platform is an autonomous quad-rotor helicopter outfitted with on-board sensors, actuators, and computers.

The primary challenge in understanding such language arises from requiring the system to take input from untrained users. However, restricting this interaction (e.g., by limiting users to a small vocabulary or grammar) would forfeit much of the advantage gained by using natural language. We therefore seek to create a system that is able to accommodate as diverse a range of inputs as possible.

Building robust language understanding systems that can actually robustly understand diverse language in realistic situations requires a number of advanced capabilities. These include speech recognition, mapping and localizing within the environment, parsing and grounding language symbols, and planning in uncertain environments. In this paper, we focus on understanding natural language directions in the context of commanding a MAV in three-dimensional environments.

Our approach extends previous research in understanding natural language directions [1] to infer three-dimensional paths. We implement this system on an autonomous MAV, so that a user can command the vehicle using phrases such as "Fly up to the windows" (Fig. 1). The user can issue a long sequence of commands, or interactively and iteratively command the vehicle to move through its environment. We evaluate the system with an informal user study and identify both its successes and limitations.

II. System Overview

Our system takes as input a natural language string consisting of directional commands and translates the string into a

*The first four authors contributed equally to this paper.

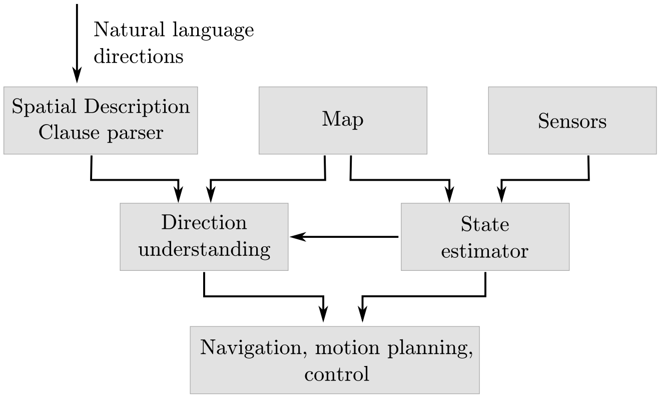
Natural language directions
Spatial Description Clause parser
Map
Sensors
Direction understanding
State estimator
Navigation, motion planning, control

Fig. 2. System diagram. Using on-board sensors and a previously acquired map, our system infers and executes a trajectory from natural language directions. Arrows indicate information flow.

sequence of motion primitives to be executed by the vehicle. We divide this task into a number of separate steps (Fig. 2).

First, the commands are parsed into *spatial description clauses* (SDCs), semantic structures that robustly capture the meaning of spatial directions [1]. Next, a probabilistic direction understanding module uses the SDCs together with a map of the environment to infer the maximum likelihood path conveyed by the natural language directions. This path, expressed as a series of 3D waypoints, is used by the navigation and motion planning modules to plan and execute an obstacle-free route that reaches the desired destination.

Our current work assumes that the robot is operating in a previously visited space, has acquired a map of the environment and its salient objects, and is continuously estimating its pose within this map using appropriate exteroceptive sensors. Extensions to online exploration and direction understanding are discussed in Sec. VII.

Micro-Air Vehicle

Our quad-rotor helicopter, shown in Fig. 1, is the AscTec Pelican manufactured by Ascending Technologies GmBH. We outfitted the vehicle with both LIDAR and camera sensors, which allows us to obtain accurate information about the environment around the vehicle.

In previous work [2] we developed a suite of sensing and control algorithms that enable the vehicle to explore unstructured and unknown GPS-denied environments. Here, we leverage that system to localize and control the vehicle in a previously explored, known environment [3], [4].

To compute the high-precision, low-delay state estimates needed to control the vehicle, we employ a 3-level sensing and control hierarchy, distinguishing processes based on the real-time requirements of their respective outputs. A fast, accurate, and robust scan matching algorithm generates relative vehicle position estimates. These are fused with inertial measurements using an extended Kalman filter (EKF) to estimate the full vehicle state, including velocity. A separate Monte Carlo localization algorithm provides lower frequency global position estimates within a free-space gridmap, which are periodically fused with the EKF estimates.

The waypoints produced by the direction understanding modules are input into a trajectory planning system, which plans the actual path for the vehicle. The planned trajectory accounts for obstacles and traversability constraints using a modified version of the navigator module from the CARMEN Robotics toolkit [5].

The real-time state estimation and control algorithms are run onboard the vehicle. The computationally intensive direction understanding, waypoint planning, and Monte Carlo localization modules run on a laptop base-station, which relays information to the vehicle via a wireless link.

III. Related Work

Our previous work building systems for understanding natural language directions modeled directions as a sequence of landmarks [6] while accounting for spatial relations and a limited set of verbs [1]. The notion of spatial description clauses is influenced by many similar formalisms for reasoning about the semantics of natural language directions [7], [8], [9], [10], [11], [12]. The structure of the spatial description clause builds on the work of Landau and Jackendoff [13], and Talmy [14], providing a computational instantiation of their formalisms.

Previous work in our group [2] and many others [4], [15], [16] has sought to develop MAVs capable of flight in unstructured indoor and urban canyon GPS-denied environments. The primary challenges addressed thus far have been in developing state estimation and control solutions to enable flight in confined and cluttered spaces. Little attention has been given to the human interface with a MAV, and how a human operator can issue commands.

Commonly used unmanned aerial vehicle interfaces include direct control with a joystick or a graphical interface with an integrated aerial map view. The former demands constant operator attention, while the latter is awkward in confined environments with significant 3D structure. For high flying vehicles operating in outdoor environments, altitude is generally a secondary concern, set based on desired viewing resolution or stealth concerns. However for indoor MAVs, the paths must explicitly guide the vehicle above and below objects in the environment, requiring an effective method for specifying 3D vehicle paths.

In this work, we leverage our experience building direction understanding systems to develop a natural language interface for issuing task-constrained natural language directions to a MAV. In addition to demonstrating an integrated platform, we build on our previous language model by incorporating language relating to three-dimensional environments and provide real-world demonstrations that the overall system can follow natural language directions.

IV. Spatial Description Clauses

To follow natural language commands, the system uses a semantic structure called a spatial description clause (SDC) that exploits the structure of language typical to directions. In our previous work, we formalized this structure by modeling each sentence in a set of directions as a hierarchy

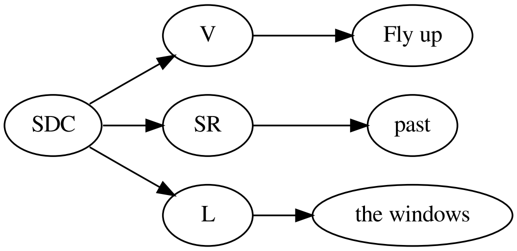

Fig. 3. SDC for the sentence “Fly up past the windows.” Here, *V* is the verb, *SR* is the spatial relation, *L* is a landmark, and the figure is implicit.

of structured clauses [1]. Each SDC consists of a *figure* (the subject of the sentence), a *verb* (an action to take), a *landmark* (an object in the environment), and a *spatial relation* (a geometric relation between the landmark and the figure, e.g., past, through, up, etc.). Any of these fields can be unlexicalized and therefore specified only implicitly. For example, in the sentence “Fly up past the windows,” the figure is an implicit “you,” the verb is “fly up”, the spatial relation is “past” and the landmark is “the windows” (Fig. 3).

V. Direction Understanding

Our system uses SDCs to follow natural language directions by finding a path that maximizes the joint distribution of paths and SDCs, given mapped objects. To do so, it first extracts SDCs from text input, and then grounds the SDC components in the environment. Text input can come from either a speech recognizer operating on spoken instructions, or from text received directly via a keyboard interface. We use an extended version of our previously introduced methods [1]. For brevity, we summarize the existing approach and focus on the novel extensions when appropriate.

A. SDC parser

We automatically extract SDCs from text by using a conditional random field (CRF) [17]. The CRF labels each word in each sentence with one of the four possible fields (*figure*, *verb*, *spatial relation* and *landmark*), or none. The CRF was trained on a different corpus of route instructions from the one used in our evaluation [1]. A greedy algorithm groups continuous chunks together into SDCs.

B. Map Layers

Our system uses a previously acquired map of the environment that consists of three layers: free-space, a three-dimensional topology, and known objects. The free-space layer, represented as a gridmap, encodes the regions traversable by the robot and is generated offline from LIDAR scans.

The space of all possible paths through the environment free-space is intractably large for typical search algorithms, so the system creates a 3D topological roadmap from the free-space gridmap. This topological roadmap, represented as a graph, is used during path inference to reduce the search space on desired paths.

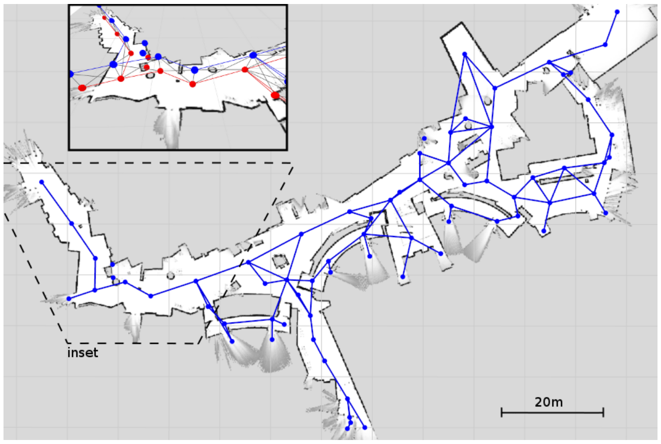

Fig. 4. Overhead free-space and topological maps for the environment used in the evaluation.. A perspective view of the dashed region is shown inset, and illustrates the 3D structure of the topology.

The three-dimensional roadmap is created by repeating a two-dimensional roadmap at multiple heights. The two-dimensional roadmap is, in turn, created by automatically segmenting the free-space based on visibility and detected objects, then extracting a topology of the environment from this segmentation [18]. Connections in the three-dimensional roadmap are created according to the original two-dimensional topology: by adding connections between the current topological node and its neighbors (at the current height and the levels directly above and below). Fig. 4 shows the free-space and topological layers of a map. Each node contains four viewpoints, facing in each of the four cardinal directions. Additional height resolution could be achieved by adding additional levels, at the cost of graph complexity.

Finally, a third layer contains a listing of objects and their positions in the environment. This layer provides the basis for grounding SDC components. In our reported results, the objects are manually annotated in the map. Our preliminary experiments with automatic object detection from on-board camera imagery suggest that fully automatic creation of this map layer is feasible for future work.

C. Model

We formulate the problem of understanding natural language directions as inferring a path (a sequence of viewpoints v_i) given a set of natural language directions (a list of SDCs $sdc_1 \dots sdc_M$). We can factor this distribution into a component for the path and a component for the observed SDCs. In particular, we assume that an SDC depends only on the current transition v_i, v_{i+1}, and that the next viewpoint v_{i+1} depends only on previous viewpoints. When P is the path, S is the sequence of the SDCs, and O are the detected objects, we have the following factorization:

$$p(P,S|O) = \left[\prod_{i=1}^{M} p(\mathrm{sdc}_i|v_i, v_{i+1}, O)\right] \times \left[\prod_{i=1}^{M} p(v_{i+1}|v_i \dots v_1)\right] \times p(v_1) \tag{1}$$

We model the transition probabilities in the second term of Eq. (1) as uniform among connected viewpoints in the topological map, together with a constraint that disallows backtracking. This constraint means that the path is not allowed to revisit any location that it has previously visited.

The most important part of our model is the observation probability, $p(\text{sdc}_i|v_i, v_{i+1}, O)$. To compute this probability, we break down the SDC into its component parts: the figure, f, the verb or action, a, the spatial relation, s, and the landmark, l. Given that v_i is the ith viewpoint and o_k is the ith detected object, we obtain the following distribution:

$$\begin{aligned} p(\text{sdc}_i|v_i, v_{i+1}, O) &= p(f_i, a_i, s_i, l_i, |v_i, v_{i+1}, O) \qquad (2)\\ &\approx p(f_i|v_i, v_{i+1}, o_1 \ldots o_K) \times p(a_i|v_i, v_{i+1}) \times \\ &\quad p(s_i|l_i, v_i, v_{i+1}, o_1 \ldots o_K) \times p(l_i|v_i, v_{i+1}, o_1 \ldots o_K) \end{aligned}$$

At this point, we have factored the distribution into four parts, corresponding to each field of the SDC, plus transition probabilities.

D. Grounding each SDC component

In order to convert a natural language command to robot action, we need to ground each term in the observation probability in terms of sensor data available to the robot.

a) Verbs: The verb component models the probability of verbs such as "up" and "down" given two viewpoints v_i and v_{i+1} which define a path segment $[v_i, v_{i+1}]$. In our previous work [1], we modeled three verbs: "turn left," "turn right," and "go straight." Extending our model to three-dimensional interactive commands required modeling the additional verbs (and verb satellites): "up," "down," "turn around," and "face."

In order to compute this term, the system extracts the type of the verb based on keywords in the verb field of the SDC; the default type is "straight." Given the verb, we compute $p(a_i|v_i, v_{i+1})$ according to the type of the verb. We use a single feature that describes the probability of each path segment. For "left," "right," and "straight," the probability of the verb is computed using the feature corresponding to the total amount of change in orientation required to travel between two viewpoints. We assume natural robot motion: in order to move from one viewpoint to another the robot must first turn to the destination, drive there, and then turn to its final orientation. The total turn amount corresponds to how much the robot must turn in order to achieve the desired change in orientation. For "left," the desired change in orientation is $90°$; for "right," it is $-90°$. For "straight," it is $0°$ and for "turn around," it is $180°$.

In order to model the new verbs such as "go up" we use the feature of whether or not the second viewpoint has a higher elevation than the first viewpoint. For "down" we compute the probability using the feature of whether the second viewpoint has a lower elevation than the first viewpoint. For "turn around", as we did with "right" and "left" we compute the probability using expected turn amount of $180°$ along with a feature that biases the robot to stay at the same location. For "face" (as in "face the windows") we use a feature that is set to change uniformly to any orientation at the same topological location. Then the landmark term computes the probability that "the windows" are visible from each particular orientation.

b) Landmarks: The landmark component models the probability of a landmark in an SDC given a viewpoint transition $[v_i, v_{i+1}]$ and detected objects O. For example, we might want to compute the probability that we would see a "computer" given that a "monitor" and a "keyboard" occur along this particular path segment. This problem is challenging because people refer to a wide variety of objects in natural language directions (in diverse ways), and not all of these objects may be in our map. However, using the notion of object-object context, we can still reason about and ground unmapped objects [19]. Conceptually, many objects are statistically likely to co-occur (e.g., a computer is probably present when a monitor and keyboard are also present). Using an object co-occurrence model built from over a million labeled images downloaded from an online photo-sharing website, our system can reason about many more objects than are present in its object map.

c) Spatial Relations: The spatial relation component models the probability of a particular spatial relation given a landmark type, landmark location, and a path segment $[v_i, v_{i+1}]$. For example, we need to compute the probability of how well a phrase such as "past the door" describes a particular path segment and landmark polygon. In order to evaluate this component, the viewpoints are converted into a sequence of points. In addition, wherever objects have been detected are converted into polygons that represent the geometry of a landmark.

The system extracts features from this schematic representation that capture the semantics of spatial prepositions. These features are functions of the geometry of the path and landmark. For example, one of the features utilized for the spatial preposition "to" is the distance between the end of the path and the landmark's location. In order to learn this distribution we use a dataset of hand-drawn examples of paths that matched a natural language description such as "through the door." Given this dataset and a set of features (along with a target class such as "through"), we use Naive Bayes to model the distribution $p(s_i = \text{past}|\text{landmark} = o_i, \text{path} = v_i, v_{i+1})$. Features are described in detail in [20]. We have trained classifiers for eleven spatial prepositions: "across", "along," "through," "past," "around," "to," "out," "towards," "down,", "away from," and "until."

E. Path Inference

The goal of the path inference is to take the model described in Eq. 1, a sequence of SDCs and a map of the environment with the location of some objects and infer a path through the environment. Before following directions, the system must first learn the probability distributions described in Eq. 2 from training data, which includes Flickr co-occurrence statistics and examples of spatial relations and verbs. Using these learned probability distributions, the robot uses a variant of the Viterbi algorithm [21] to find the path

> Go forward until you are able to make a left. Then move ahead until you reach the opposite wall, then make a right. Go straight, past one staircase and to the next staircase. At the top of this staircase you will find a fire alarm on your left at approximately 7ft up.

Fig. 5. An example set of directions from the informal corpus collected for MAVs.

that maximizes the joint distribution of the path and the text, as given in Eq. (1). The inputs to the algorithm include a starting viewpoint, a map of the environment with some labeled objects, and the sequence of SDCs extracted from the directions. The output of the direction understanding module is a series of viewpoints through the environment, along with the probability of this path.

VI. Experiments

To evaluate our system, we conducted a number of experiments in an environment resembling an indoor mall, consisting of a cafeteria, a small library, an exercise center, and several classrooms connected by a wide pedestrian corridor with scattered chairs and tables. The high ceilings and interior windows in this space are too high for most ladders to safely access, and require specialized lifting equipment for even simple inspections and maintenance.

A. *Corpus evaluation*

To assess the quality of the paths inferred by our system, we evaluated it on a corpus of directions collected from potential users in an informal study. Users were familiarized with the test environment and asked to instruct the pilot to take video of 7 objects in the environment, each from a different starting location. Objects to be inspected were difficult to see closely from the ground and included a wireless router mounted high on the wall, a hanging sculpture, and an elevated window. Subjects were told the vehicle's starting pose and asked to write down instructions for a human pilot to fly a MAV to the desired object and take video of that object. The corpus consists of forty-nine natural language commands generated in this way. Fig. 5 shows an example set of directions from this corpus. Subjects were engineering undergraduates unfamiliar with the system.

We report the proportion of inferred paths that terminate within a certain 2D distance of the target object. Fig. 6 shows this metric for each subject as the distance is increased. Performance varied widely for different subjects: the system successfully followed 71% of the directions from two subjects to within 10 meters of the object (subjects A and B), but none from another (subject G). For comparison, we also show the overall performance of an algorithm that randomly chooses a final destination.

Examining the differences in the types of language produced by the subjects shows that the system performed best when the subjects referred to distinctive landmarks in the environment. Some landmarks, such as the "butterfly sculpture" and the "police car", can be resolved unambiguously by the system because that type of landmark appears only once in the environment. Others, such as "a large circular pillar" or "the double doors" refer to relatively unambiguous landmarks for a human, but are not correctly identified by our system due to its inability to reason with adjectives. In some cases, subjects used language not well modeled by our system. For example, in every instance where the system failed on subjects A and B, the directions included phrases such as "Go… the full length of the building" or "Proceed all the way down the hallway."

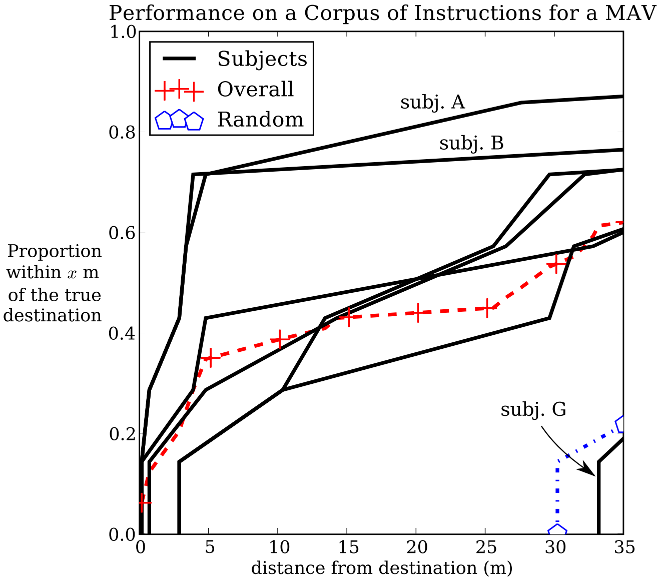

Fig. 6. System performance for each subject (black) and across all users (dashed red) on a corpus of natural language commands collected for the MAV. Each point represents the proportion of directions followed to within x meters of the true destination. Performance of a system that randomly chooses a destination is shown in blue for comparison.

In the case of subject G, the language used was ungrammatical, had no punctuation, and contained many typographical errors. For example, "you will a starcase to your right [*sic*]." In addition, the language lacked distinctive landmarks; when distinctive landmarks existed, they were often misspelled. (e.g., "Queston Mark [*sic*].") Most landmarks from this subject were distinguished by color and metrical distances such as "between the yellow and red wall" and "2 meters," which are not understood by the current system. Finally, the subject would say "right" when the correct turn direction was "left." Like humans, our system had trouble following directions with this type and frequency of ambiguity and errors.

B. *MAV Experiments*

To demonstrate the overall system, we developed user interfaces for issuing directions to the MAV using a speech recognition system, or by typing a string into a graphical application running on a laptop computer. Paths computed by the natural language direction understanding module were then executed autonomously by the MAV. Examples of successfully executed paths are shown in Fig. 7. The directions corresponding to these paths are:

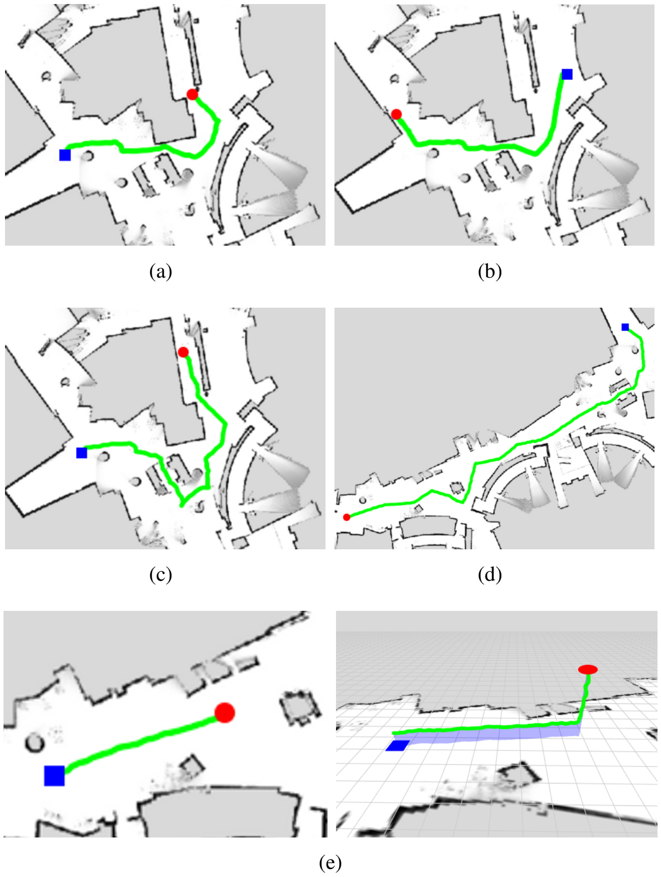

Fig. 7. Example paths (green) executed by the MAV superimposed on a map. Path start points are marked by blue squares, and end points are marked by red circles. In (a)-(d) the path is shown from an overhead viewpoint. In (e) both an overhead and a perspective view are given to illustrate the elevation change of the MAV. Natural language directions corresponding to each path are given in the text.

(a) Go past the library and tables till you see a cafe to the left. Fly past the cafe and there will be other eateries. Head into this area.

(b) Stand with your back to the exit doors. Pass the cafe on your right. Make a right directly after the cafe, and into a seating area. Go towards the big question mark.

(c) Go straight away from the door that says CSAIL, passing a room on your right with doors saying MIT Libraries. Turn left, going around the cafe and walk towards the cow.

(d) Turn right and fly past the libraries. Keep going straight and on the left near the end of the hallway there is a set of doors that say Children's technology center. You are at the destination.

(e) Fly to the windows and then go up.

In addition to following individual sets of directions, the vehicle can accept directions interactively, while relaying on-board camera images and LIDAR measurements back to the user in real-time. In one such interaction, the vehicle was commanded to fly past a classroom, collect some video, and return to the user. The commands given were:

Fly past room 124 and then face the windows.
Go up.
Go back down.
Come back towards the tables and chairs.

As the MAV carries out the directions, on-board sensor data is continuously transmitted back to the user (Fig. 8). The accompanying video shows our platform following additional directions given by the user.[1]

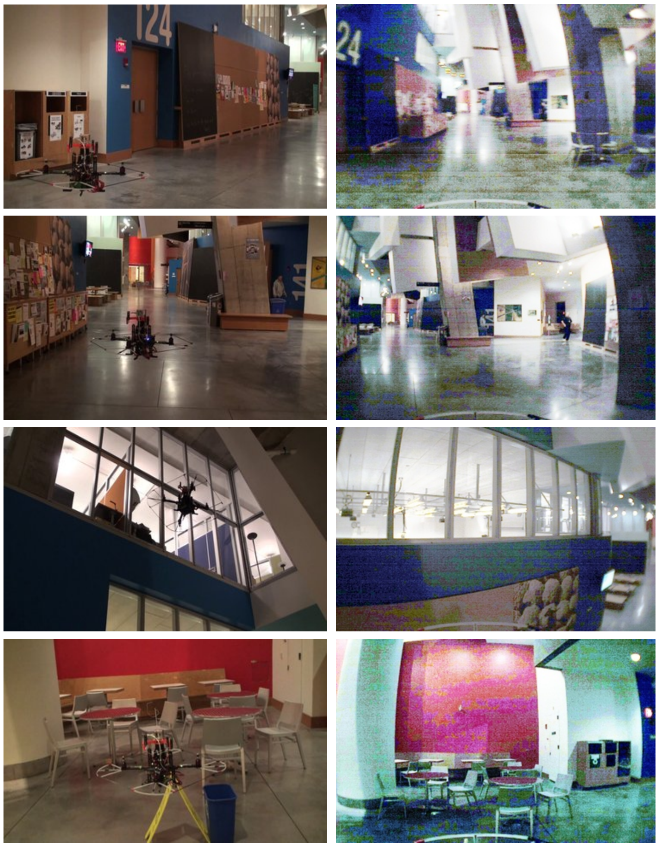

Fig. 8. (left) Photographs of the MAV executing an interactive series of instructions. (right) Imagery from the on-board camera, transmitted to the operator as the MAV flies. The commands issued to the MAV are given in the text.

VII. Discussion and Future Work

Our system operates in previously mapped environments, and requires a listing of known objects and their locations. As such, current applications are limited to situations such as maintenance and building inspection, where the vehicle visits previously explored areas. To enable its use in unknown environments, we are integrating our work in autonomous goal-directed exploration [22], direction understanding with partial information [1], and online object recognition.

Mapping, navigation, and control in 3D environments is still an active area of research. The current map does not model full 3D structure, and is not sufficient in areas with significant overhang or other environments where free space changes with vehicle height. While the vehicle's reactive obstacle avoidance system is typically sufficient to avoid nearby obstacles, robust and efficient 3D mapping and localization solutions are still required.

[1] Also available online at: http://groups.csail.mit.edu/rrg/video/10-iros-mav.mp4

Natural language is a compelling interface for specifying three-dimensional trajectories through space. Expanding the system's repertoire of verbs and spatial relations would make it more robust and flexible. For example, "over," "under," "above," and "through" require extracting features from a three-dimensional schematic model of the trajectory and landmark objects. Extending the system to understand more complicated referring expressions is also essential for naturalistic dialog. For example, it should be possible to say "Fly to the windows above room 124" and resolve "the windows above room 124" to the correct landmark. Finally, for truly interactive dialog, the system must understand deictic commands such as "Come back here" and be able to ask and answer questions about its state.

VIII. Conclusion

Human interaction with complex robots such as autonomous micro-air vehicles can be greatly simplified if the robots are capable of understanding natural language directions. The implementation of such a system raises a number of technical challenges, from parsing the raw input and estimating the desired path, to 3D navigation and control of the vehicle.

We have presented a system that accepts natural language input and executes the maximum likelihood trajectory estimated by a direction understanding module. We have extended our previous work in understanding natural language directions to accommodate words such as "up" and "down" that imply a desired change in vehicle height. Finally, we have demonstrated these capabilities interactively on an autonomous micro-air vehicle operating in confined spaces.

IX. Acknowledgments

Albert Huang, Abraham Bachrach, Stefanie Tellex, Deb Roy and Nicholas Roy were supported by the Office of Naval Research under MURI N00014-07-1-0749 and under MURI N00014-09-1-1052. We also thank the Spoken Language Systems Group at CSAIL for the use of the SUMMIT speech recognizer.

References

[1] T. Kollar, S. Tellex, D. Roy, and N. Roy, "Toward understanding natural language directions," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, 2010.

[2] A. Bachrach, R. He, and N. Roy, "Autonomous flight in unstructured and unknown indoor environments," in *Proceedings of EMAV*, 2009.

[3] R. He, S. Prentice, and N. Roy, "Planning in information space for a quadrotor helicopter in a GPS-denied environments," in *Proc. ICRA*, Los Angeles, CA, 2008, pp. 1814–1820.

[4] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *IEEE International Conference on Robotics and Automation*, May 2009, pp. 2878–2883.

[5] M. Montemerlo, N. Roy, and S. Thrun, "Perspectives on standardization in mobile robot programming: The Carnegie Mellon Navigation (CARMEN) Toolkit," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 3, October 2003, pp. 2436–2441.

[6] Y. Wei, E. Brunskill, T. Kollar, and N. Roy, "Where to go: Interpreting natural directions using global inference," in *IEEE International Conference on Robotics and Automation*, 2009.

[7] P. Rybski, J. Stolarz, K. Yoon, and M. Veloso, "Using dialog and human observations to dictate tasks to a learning robot assistant," *Intelligent Service Robotics*, vol. 1, no. 2, pp. 159–167, 2008.

[8] M. Levit and D. Roy, "Interpretation of spatial language in a map navigation task," *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, vol. 37, no. 3, pp. 667–679, 2007.

[9] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," *Proceedings of the National Conference on Artificial Intelligence*, pp. 1475—1482, 2006.

[10] G. Look, B. Kottahachchi, R. Laddaga, and H. Shrobe, "A location representation for generating descriptive walking directions," in *International Conference on Intelligent User Interfaces*, 2005, pp. 122–129.

[11] M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock, "Spatial language for human-robot dialogs," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 2, pp. 154–167, 2004.

[12] G. Bugmann, E. Klein, S. Lauria, and T. Kyriacou, "Corpus-based robotics: A route instruction example," *Proceedings of Intelligent Autonomous Systems*, pp. 96—103, 2004.

[13] B. Landau and R. Jackendoff, ""What" and "where" in spatial language and spatial cognition," *Behavioral and Brain Sciences*, vol. 16, pp. 217–265, 1993.

[14] L. Talmy, "The fundamental system of spatial schemas in language," in *From Perception to Meaning: Image Schemas in Cognitive Linguistics*, B. Hamp, Ed. Mouton de Gruyter, 2005.

[15] S. Ahrens, D. Levine, G. Andrews, and J. How, "Vision-based guidance and control of a hovering vehicle in unknown, GPS-denied environments," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 2643–2648.

[16] S. Hrabar and G. Sukhatme, "Vision-based navigation through urban canyons," *J. Field Robot.*, vol. 26, no. 5, pp. 431–452, 2009.

[17] T. Kudo, "CRF++: Yet another CRF toolkit," http://crfpp.sourceforge.net, 2009.

[18] E. Brunskill, T. Kollar, and N. Roy, "Topological mapping using spectral clustering and classification," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[19] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things." in *IEEE International Conference on Robotics and Automation*, 2009.

[20] S. Tellex and D. Roy, "Grounding spatial prepositions for video search," in *Proceedings of the International Conference on Multimodal Interfaces*, 2009.

[21] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *Information Theory, IEEE Transactions on*, vol. 13, no. 2, pp. 260–269, 1967.

[22] A. Bachrach, R. He, S. Prentice, and N. Roy, "RANGE-robust autonomous navigation in gps-denied environments," in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, May 2009.