Combining Computer Vision and Real Time Motion Planning for Human-Robot Interaction

Jae Sung Park, Dinesh Manocha
http://gamma.cs.unc.edu/SafeMP/NLP/ (full video)

***Abstract*— We present an algorithm for combining computer vision, natural language processing (NLP) and realtime robot motion planning to enable human-robot interaction and automatically generate safe robot movements. Our approach uses Gaussian Process-based offline learning of human actions along with temporal coherence to predict the human actions. We use dynamic constraint mapping to generate robot actions from NLP commands. Our formulation transforms the complex, attribute-based natural language instructions into appropriate cost functions and parametric constraints for optimization-based motion planning.**

I. Introduction

Motion planning algorithms are used to compute collision-free paths for robots among obstacles. In the field of human-robot interaction (HRI), it is important to develop an interface to communicate a human's intent to a robot [1], [2], [3] and use reliable planning algorithms that can safely handle dynamic and unpredictable human motion.

As robots are increasingly used in complex scenarios and applications, it is important to develop a new generation of motion planning and robot movement techniques that can respond appropriately to diverse, attribute-based NLP instructions for HRI, e.g., those containing negation based phrases or references to position, velocity, and distance constraints. Furthermore, we need efficient techniques to automatically map the NLP instructions to such motion planners. As we generate robot movements in response to these instructions, we need to use vision-based techniques to predict human action and generate collision-free and robot movements. As the humans move, it is important for the robots to predict the human actions and motions from sensor data and to compute appropriate trajectories. In some scenarios, it is possible to infer high-level human intent using learning-based algorithms [4], and thereby perform a better motion planning that generates safer robot trajectories.

At a high level, natural language instructions can be decomposed into task description and attributes. Task descriptions are usually verb or noun phrases that describe the underlying task performed by a robot. The attributes include various adjectives, adverbs, or prepositional phrases are used to specify additional conditions the robot must (or must not) satisfy. For example, these conditions may specify some information related to the speed, orientation, physical space characteristic, or the distances. Therefore, it is important to design motion planners that take into account these robotic task descriptions and robot motion constraints.

We use *Dynamic Grounding Graphs* (DGG) to parse and interpret the commands and generate the constraints. Moreover, our formulation includes the latent parameters in the grounding process and that allows us to model many continuous variables in our grounding graph [5]. Furthermore, we use a new dynamic constraint mapping that takes DCG as the input and computes different constraints and parameters of the motion planner. The appropriate motion parameters correspond to the speed, orientation, position, smoothness, repulsion, and avoidance. The final trajectory of the robot is computed using a realtime constraint optimization solver [6]. Overall, our approach can automatically handle complex natural language instructions corresponding to spatial and temporal adjectives, adverbs, superlative and comparative degrees, negations, etc.

We address the problem of planning safe and reliable motions for a robot that is working in environments with humans. As the humans move, it is important for the robots to predict the human actions and motions from sensor data and to compute appropriate trajectories. We developed a high-DOF motion planning approach to compute collision-free trajectories for robots operating in a workspace with human-robot cooperating scenarios. We track the positions of the human using depth cameras and present a new method for human action prediction using combination of classification and regression methods. Given the sensor noises and prediction errors, our online motion planner uses probabilistic collision checking [7] to compute a high dimensional robot trajectory that tends to compute safe motion in the presence of uncertain human motion.

We highlight the performance of our algorithms in a simulated environment as well as on a 7-DOF Fetch robot operating next to a human in a safe manner. Our approach can handle a rich set of natural language commands and can generate appropriate paths in realtime. These include complex commands such as picking (e.g., *"pick up a red object near you"*), correcting the motion (e.g., *"don't pick up that one"*), and negation (e.g., *"don't put it on the book"*).

II. Implementation and Results

We have implemented our algorithm and evaluated its performance in a simulated environment and on a 7-DOF Fetch robot. All the timings are generated on a multi-core PC.with Intel i7-4790 8-core 3.60GHz CPU and a 16GB RAM.

A. Training DGGs for Demonstrations

The training dataset for NLP is created with enough number of training samples, up to $100,000$ samples in

(a) (b) (c)

Fig. 1. Initially the user gives the "pick and place" command. However, when the robot gets closer to the book, the person says *"don't put it there"* (i.e. negation) and the robot avoids the book using our dynamic constraint mapping functions and optimization-based planning. Our approach can generate appropriate motion plans for such attributes.

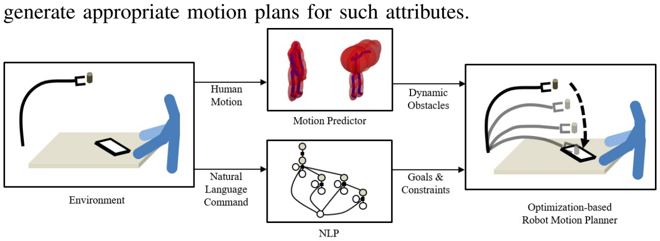


Fig. 2. The simulated Fetch robot arm is reaching towards one of the two red objects . (a) When a command *"pick up one of the red objects"* is issued, the robot moves to the right red object because of DGG algorithm. (b) If the user doesn't want the robot to pick up the right object, he/she uses a command *"don't pick up that one"*. Our DGG algorithm dynamically changes the cost function parameters. (c) The robot approaches the right object and stops.

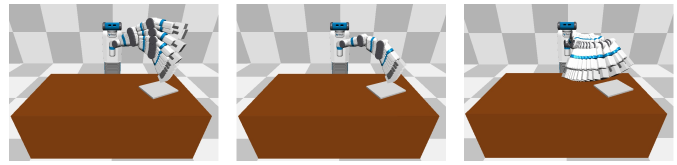
our experiments. For each demonstrations, we write tens of different sentences that specifying goals of tasks and constraints of motion plans, with different nouns, pronouns, adjectives, verbs, adverbs, preposition, etc.. Sentences that specify constraints may start with *"Don't"*. For each sentence, we generate random robotic environment and initial state of robot. Some objects in the environment, such as tables, are in fixed locations, while other objects such as small boxes or cans to pick up are randomly placed on the table.

B. Simulations and Real Robot Demonstrations

We have evaluated the performance in complex environments composed of multiple objects and local minima in the optimization problems. Based on the NLP commands, the robot decides to pick an appropriate object or is steered towards the goal position in a complex scene. In particular, the user gives NLP commands such as *"move right"*, *"move up"*, *"move left"* or *"move down"* to guide the robot. For each such command, we compute the appropriate cost functions.

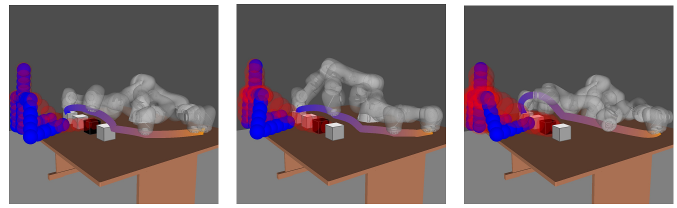
We also integrated our NLP-based planner with ROS and evaluated its performance on the 7-DOF Fetch robot. In a real-world setting, we tested its performance on different tasks corresponding to: (1) moving a soda can on the table from one position to an other; (2) not moving the soda can over the book. More details are given in the technical report [5]. The robot recomputes the cost functions and avoids the region around the book.

We observe reliable human motion prediction results and smoothly planned trajectories, as shown in Fig. II-B. This is because the robot changes its path in advance before the



(a) (b) (c)

Fig. 3. The human asks the robot to *"put the cube on table"* (a). As it approaches the laptop (b), the human uses a negation NLP command *"don't put it there,"* so the robot places it at a different location (c).



(a) (b) (c)

Fig. 4. The human is moving arm forward to blocks on the table. The human's current pose is colored in blue, and predicted future motions are colored in red. The motion planner generates safe robot trajectories avoiding human.

human obstacle actually blocks the robot's shortest path if human motion prediction is used.

C. Analysis

We evaluated the performance based on the following metrics:

- *Success Rate:* The ratio of successful task completion among all trials.
- *Trajectory Duration:* The duration between the time the first NLP command is given and the robot's successful completion of the task after trajectory computation.
- *Trajectory Smoothness Cost:* A cost based on evaluating the trajectory smoothness based on standard metrics and dividing it by the trajectory duration. A lower cost implies a smoother and more stable trajectory.

Table I shows the results on our benchmarks with varying numbers of training data samples on the simulation environment shown in Fig. 2. When the number of training data samples increases, the success rate also increases, and the trajectory duration and the trajectory smoothness cost decrease.

We quantitatively measure prediction time, smoothness, jerkiness of robot trajectories and distance from the robot to obstacles. Table II shows the performance of our I-Planner algorithm with a real robot near a human.

References

[1] T. Kollar, S. Tellex, M. R. Walter, A. Huang, A. Bachrach, S. Hemachandra, E. Brunskill, A. Banerjee, D. Roy, S. Teller, *et al.*, "Generalized grounding graphs: A probabilistic framework for understanding grounded language," *JAIR*, 2013.

[2] T. M. Howard, S. Tellex, and N. Roy, "A natural language planner interface for mobile manipulators," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6652–6659.

[3] C. Matuszek, D. Fox, and K. Koscher, "Following directions using statistical machine translation," in *HRI*. IEEE, 2010.

# Training Data	Success Rate	Duration	Smoothness Cost
1,000	5/10	23.46s (5.86s)	8.72 (5.56)
3,000	9/10	16.02s (3.28s)	2.56 (0.64)
10,000	10/10	13.16s (1.24s)	1.21 (0.32)
30,000	10/10	12.81s (0.99s)	0.78 (0.12)
100,000	10/10	12.57s (0.97s)	0.72 (0.10)

TABLE I

PLANNING PERFORMANCES WITH VARYING SIZES OF TRAINING DATA FOR THE SCENARIO IN FIG. 2.

Scenarios	Model	Prediction Time	Smooth-ness	Jerkiness	Distance
Waving Arms	ITOMP	N/A	N/a	6.23	2.1 cm
	I-Planner	23.5 ms	6.1 cm	1.25	10.5 cm
Moving Cans	ITOMP	N/A	N/A	7.83	3.9 cm
	I-Planner	50.0 ms	8.8 cm	1.32	13.5 cm

TABLE II

PERFORMANCE OF THE PLANNER WITH A REAL ROBOT RUNNING ON THE 7-DOF FETCH ROBOT NEXT TO DYNAMIC HUMAN OBSTACLES.WE OBSERVE ALMOST 5X IMPROVEMENT IN THE SMOOTHNESS OF THE TRAJECTORY DUE TO OUR PREDICTION ALGORITHM.

[4] J. S. Park, C. Park, and D. Manocha, “Intention-aware motion planning using learning based human motion prediction,” *arXiv preprint arXiv:1608.04837*, 2016.

[5] J. S. Park, B. Jia, M. Bansal, and D. Manocha, “Generating realtime motion plans from complex natural language commands using dynamic grounding graphs,” *CoRR*, vol. abs/1707.02387, 2017. [Online]. Available: http://arxiv.org/abs/1707.02387

[6] C. Park, J. Pan, and D. Manocha, “ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments,” in *Proceedings of International Conference on Automated Planning and Scheduling*, 2012.

[7] J. S. Park, C. Park, and D. Manocha, “Efficient probabilistic collision detection for non-convex shapes,” in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1944–1951.