Learning to Generate Backup Paths in Cooperative Transportation of Human-Robot Teams

Dung Nguyen[1] and Tsz-Chiu Au[1]

***Abstract*— Many interactions between humans and robots are cooperative, but humans can easily make mistakes that cause failure. A human-robot system is *fail-safe* if the robots can steer the system to a safe state after a human error occur. However, the success of this emergency maneuver depends on how quickly the system generates a backup path (i.e., a motion plan) for the robots. In this paper, we describe a new neural network model that can be used to speed up the generation of backup paths for robots in emergency situations. To test our idea, we consider a cooperative transportation task in which a human and a robot move a rectangular object from one location to another. We collected training data from a simulation in which human deviates from his intended path. Then we train our machine learning model using the backup motion plans generated by RRT in these emergency situations. The model can be utilized by a modified RRT algorithm to speed up the generation of backup paths for new emergency situations. Our experimental results shows that our approach can speed up the generation of backup paths by an order of magnitude.**

I. INTRODUCTION

A human-robot system is *fail-safe* if the robots can find ways to steer the system to a safe state and avoid catastrophic consequences when human errors occur. However, if the emergency maneuver cannot be found in a short time, the human-robot system cannot guarantee fail-safe. In this paper, we propose a machine learning model for speeding up the process of computing a *backup path*—an emergency maneuver in form of a motion plan—in a cooperative transportation task in which a human and a robot move a table from one location to another together in an office environment. More specifically, we propose (1) a new deep neural network model called GuidanceNet that learns to generate backup paths from local maps and derived paths of a human, and (2) a modified RRT algorithm that adapts a backup path generated by the GuidanceNet model to generate a new backup path that is more robust when a human derived from its intended path. Our experiment showed that our approach can speed up the search for backup paths by an order magnitude when compared with generating backup paths from scratch.

II. RELATED WORKS

There have been many works on using machine learning to enhance sampling-based motion planners by biasing the sampling process. Zucker and Paliwal [6] uses reinforcement learning to bias the sampling distribution of RRTs in workspace. Ichter et al. [3] took advantage of a conditional variational autoencoder (CVAE) to learn a sampling distribution from demonstrations, and then used this distribution to bias sampling. Recently, researchers start to consider using deep learning to guide the sampling process in motion planning. Recurrent neural networks (RNNs) has been immensely successful in many tasks that relate to sequence processing including trajectory prediction [1]. Gao et al. [2] introduces Intention-Net, which simultaneously uses two CNNs to extract features from several types of input data including camera images, local paths, and intention to generate a sequence of actions for mobile robots during replanning. Intention-Net uses fully-connected layers to infer the actions from the extracted features. By contrast, we adopt a type of RNNs called bidirectional recurrent neural networks (Bi-RNNs), which take the past and future of sequence into account [5]. More specifically, we opt for bidirectional long short term memory (Bi-LSTM) to generate a guidance sequence of actions for our robot to follow a human trajectory while moving a table.

[1]School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology, UNIST-gil 50, Eonyang-eup, Ulju-gun, Ulsan, Republic of Korea. {dung, chiu}@unist.ac.kr

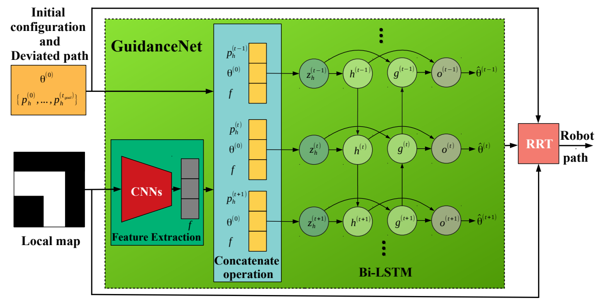

Fig. 1: The architecture of the GuidanceNet model.

III. PROBLEM DEFINITION

In a 2D workspace $\mathcal{W}$ with a set of obstacles $\mathcal{B}_i$ ($i = 1, ..., m$), a human and a robot cooperatively transport a rigid rectangular object by holding the object at two fixed grasp points, also known as pivot points [4]. The robot and human positions are $\boldsymbol{p}_r = (x_r, y_r) \in \mathbb{R}^2$ and $\boldsymbol{p}_h = (x_h, y_h) \in \mathbb{R}^2$, respectively. The object, together with the human and the robot, forms a *human-object-robot system* $\mathcal{M}$. The configuration space $\mathcal{C}$ for the human-object-robot system is defined by a triple $\boldsymbol{q} = (x_h, y_h, \theta)$, where (x_h, y_h) is the position of the center of the system and θ is the orientation of the system. We can construct a function $g : \mathbb{R}^4 \mapsto \mathbb{R}^3$ which maps a tuple (x_h, y_h, x_r, y_r) to a configuration (x_h, y_h, θ). A joint motion plan for $\mathcal{M}$ is a pair of motion plans, one for the human and the other for the robot, that moves the human-object-robot system from an initial configuration to a goal configuration without colliding with any obstacle. During the transportation, a human may deviate from the given joint motion plan and follow a deviated path $\{\boldsymbol{p}_h^{(0)}, \ldots, \boldsymbol{p}_h^{(T)}\}$ instead, for a period of time T. Our goal is to generate a backup plan $\{\boldsymbol{p}_r^{(0)}, \ldots, \boldsymbol{p}_r^{(T)}\}$ for the robot in response to the given deviated path such that no collision occurs.

IV. GUIDANCENET

As shown in Fig. 1, GuidanceNet has three inputs: (1) an initial configuration $\boldsymbol{q}^{di} = (x_h^{(0)}, y_h^{(0)}, \theta^{(0)})$ of the human-object-robot system; (2) a sequence of human's positions in the deviated path $\{\boldsymbol{p}_h^{(0)}, ..., \boldsymbol{p}_h^{(T)}\}$; and (3) a gray scale image of a 2D local floor map. The size of this local map should be large enough to contain the given human path, but it can be larger. Given these inputs, GuidanceNet generates the orientation of the system at each time step: $\boldsymbol{o}^{(t)} = \theta^{(t)} \in [0, 2\pi]$, for $t \in (0, T]$. The CNNs in GuidanceNet, which consists of two convolutional layers and a pooling layer, extracts a feature vector f of length 2034 from a 2D image of a local map. The first and the second convolutional layers used 32 filters and 64 filters, respectively, whose size is 3×3 with stride of 1. The feature vector f, the initial orientation $\theta^{(0)}$, and the position of the human $\boldsymbol{p}_h^{(t)}$ at time t in a deviated path are concatenated to become an input $z^{(t)}$ of the Bi-LSTM model. The hidden recurrence layer $\boldsymbol{h}$ propagates information forward, while the hidden recurrence layer $\boldsymbol{g}$ propagates information backward in time. The output is a linear combination of $\boldsymbol{h}$ and $\boldsymbol{g}$. The loss function for training is the mean square error of output $\theta^{(t)}$.

We cannot directly use the sequence $\{\hat{\theta}^{(0)}, \ldots, \hat{\theta}^{(T)}\}$ generated by GuidanceNet as a backup path since the sequence can easily lead to collision. But this sequence can be used to speed up the search process in RRT. To this end, we developed a modified RRT algorithm to generate a motion plan for robots using the actual local map at the point of deviation of the intended path. Our modified RRT algorithm uses the sequence generated by GuidanceNet as a sampling guidance in order to reduce the search time. The idea is to give a higher priority to sample configurations that are close to the ones derived from the sequence in the sampling step in RRT.

V. EXPERIMENTAL RESULTS

We compared our approach with the one that does not use a machine learning model to guide the sampling process in RRT. To train our model, we used three 2-D floor plans of actual office environments of size $150 \times 150 m^2$. We randomly cropped 60 local maps of size $30 \times 30 m^2$ from two of the three maps then sampled 500 initial configurations from each local map. For each initial configuration, 10 deviated paths of human were obtained by drawing 10 line segments with length $l \in \{2, 3, 4, 5, 6\}$ and the human moved 1 unit of length per 1 time step along these line segments. For each deviated path, we used RRT to find a backup path for the robot. Then a sample in our dataset consisted of a local map of size 30×30, an initial configuration of the system, and a derived path. The label of a sample is the backup path.

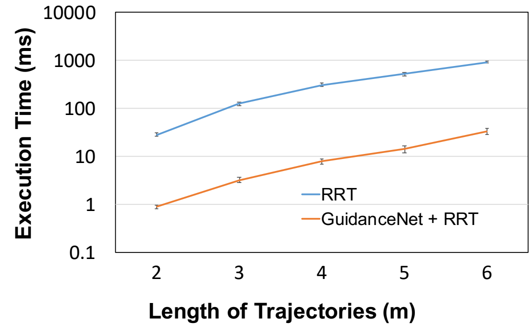

Fig. 2: The length of the deviated path versus the execution time of the RRT algorithms.

GuidanceNet was implemented in Keras with Tensorflow backend, running on a computer with a Intel Core i7 4GHz CPU and GeForce GTX 1080 Ti graphics card. GuidanceNet was trained with 50 epochs and the batch of size 25 for each map. Adam optimizer was used with a 0.001 learning rate, whereas $\beta_1 = 0.9$ and $\beta_2 = 0.999$. In the test phase, we cropped 30 local maps of size 30×30 from the third 2-D map which was not used in the training process. Once again, we generated 30 derived paths for 30 randomly chosen initial configurations. Then we used our modified RRT algorithm to find a backup path for the robot, and the execution time of the algorithm was recorded. Fig. 2 shows that our approach outperforms RRT in terms of the execution time by an order of magnitude.

VI. SUMMARY AND FUTURE WORK

This paper presents a machine learning model called GuidanceNet for speeding up the generation of backup paths in cooperative transportation of human-robot teams. In the future, we intend to extend our model to generate recovery paths that can reach the goal configuration in addition to collision avoidance.

Acknowledgments: This work has taken place in the ART Lab at UNIST. ART research is supported by NRF (2.180186.01 and 2.170511.01).

REFERENCES

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Fei-Fei Li, and Silvio Savarese. Social LSTM: human trajectory prediction in crowded spaces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 961–971, 2016.

[2] Wei Gao, David Hsu, Wee Sun Lee, Shengmei Shen, and Karthikk Subramanian. Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, pages 185–194, 2017.

[3] Brian Ichter, James Harrison, and Marco Pavone. Learning sampling distributions for robot motion planning. *CoRR*, abs/1709.05448, 2017.

[4] Antoine Rioux, Claudia Esteves, Jean-Bernard Hayet, and Wael Suleiman. Cooperative vision-based object transportation by two humanoid robots in a cluttered environment. *I. J. Humanoid Robotics*, 14(3):1–30, 2017.

[5] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45(11):2673–2681, 1997.

[6] Matthew Zucker, James Kuffner, and James A. Bagnell. Adaptive workspace biasing for sampling-based planners. In *2008 IEEE International Conference on Robotics and Automation, ICRA 2008, May 19-23, 2008, Pasadena, California, USA*, pages 3757–3762, 2008.