I Picture You:
Imaginative localization on remote teammates

Taeyeong Choi, Sehyeok Kang, and Theodore P. Pavlic

***Abstract*— We propose *IPY*, as an extension of [1], to tackle the Remote Teammate Localization (RTL) problem where a robot embedded in a multi-robot team is to predict positions of all other teammates when each robot has a limited sensor radius and a relatively simple motion rule with dependency on the position of its nearest neighbors. The previous work presented feasibility of a scalable machine learning approach for the predicting robot to proactively assist a team behavior in caging scenario. In this work, *IPY* is designed to improve scalability by taking into account physical constraints on the team formation as well as producing probabilistic predictions in the form of image through deep learning pipeline. Furthermore, while we previously relied only on computer simulations, all experiments in this work are performed in a physical robotic platform, *Thymio*, to demonstrate the performance gain in more realistic environments.**

I. Introduction

In multi-robot systems including swarms, every robot is usually allowed to observe only a subset of its team members and interact with them to determine the next action according to relatively simple motion rules. Such a property enables the entire system to perform in a distributed manner, and also the behavior of it can be driven easily by a few leader robots to eventually achieve the goal [x]. This implies that if a robot has an ability to recognize useful property, e.g.) formation, of the whole team in real time using its local sensors, it could present adjustive actions accordingly to better promote a collective behavior for the sake of the team.

In [1], we suggested a machine learning method to solve the RTL problem where a robot embedded in a multi-robot team is to predict positions of all other teammates using only local observations about a single nearby teammate. Since each robot has a limited sensor radius and a relatively simple motion rule with dependency on the position of its nearest neighbors in line formation, the predicting robot has to be able to learn the regularity of the observed motions. Through simulated experiments, we showed the feasibility of the idea especially in caging scenario in which the predicting robot could recognize the team behavior and use a proactive maneuver accordingly.

RTL problem provides some unique characteristics compared to general localization problems. In RTL, the robot that executes predictions does not try to localize itself but its teammates using accessible observations. Moreover, the robot is not allowed to communicate with other members during prediction phase to consider communication-free scenarios, which differs from works on cooperative localization. Hence, RTL usually assumes that robots behave with correlations with their neighbors so that the resulting behaviors contain the state information of their neighbors. In this sense, state observers in networked robotic system could be a more similar configuration to RTL, but RTL emphasizes possible applications to variable sizes of robotic system and also a general framework that would not impose any constraint on dynamics of the system.

*This work was not supported by any organization

T. Choi, S. Kang, and T. P. Pavlic are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA {tchoi4, skang66, tpavlic}@asu.edu

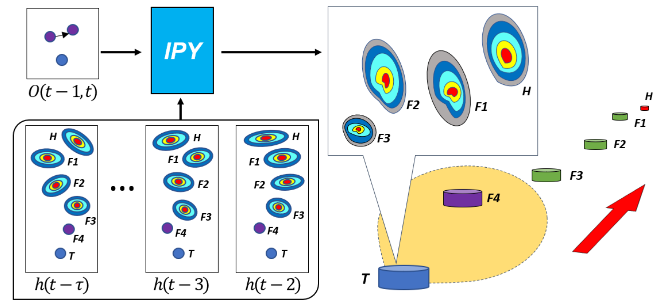

Fig. 1: Example of *IPY* performing RTL on 6-robot team. While the *Head* (red) is driving the team in line formation, the *Tail* can observe the motions of *Follower 1* (purple) within its sensor radius (yellow). The observation, O, and the historical positions of all robots, h, that the *Tail* has known so far become an input to *IPY* to pro All positions of robots in the local view of *Tail* are visualized in image as probabilistic representation. Note here that since *Follower 4* can be observed directly, the image representation does not illustrate any uncertainty for it.

In this work, we propose *IPY* to significantly reduce prediction errors solving the RTL problem, which would allow to gain stable performance even with more robots added. *IPY* runs a deep neural network as a backbone, first to consider physical constraints on the sequential change of team formation and, second to use a probabilistic representation in the form of image for obtained observations and new predictions. Specifically, every observed or predicted position is encoded as an image in which the summation of all values must be 1, which can be viewed as a probability distribution of position of the robot. Hence, *IPY* learns how to synthesize inputs with some uncertainty to produce a probabilistic prediction outcome in end-to-end fashion. Furthermore, a recurrent layer is deployed in *IPY*, which enables the model to take into account possible evolution of team formation based on recent history and reduce the searching space for predicted solution. With all combined, historical probabilistic (soft) formations are used as input per prediction providing more flexibility to the predictor in selecting salient features and finally leading to more robust

	Duration	Num. of Samples	Num. of Instances
3 robots	120 minutes	$2,000$	500
5 robots	60 minutes	$1,000$	250

TABLE I: Description of data collected from executions of 3-robot and 5-robot teams.

performance even when the input is noisy to some level.

We conduct all experiments on a physical robotic platform, *Thymio* [2], to demonstrate the improvement in more realistic environments in contrast to the previous work that relied only on computer simulations. All codes are available online[1], and supplementary videos are submitted as well.

This paper is organized as follow. In Section II, we explore related literature and the distinction of our work. Section III explains more details about our setting of RTL problem. Then, we introduce our method, *IPY*, in Section IV, and Section V explains details about experiments performed on real robots, including data collection, hyperparameters used for learning, and the results. Lastly, we summarize our research and discuss future directions in Section VI.

II. Related work

III. Problem Description

IV. Method

V. Experiments

To demonstrate the effectiveness of our method, we employ a physical robotic platform, *Thymio* [2], which allows to execute a team of small two-wheeled mobile robots. We use a central computer connected with a overhead camera to simulate better proximity sensors, a more powerful computing power, and a GPS system that would easily run on each robot in real scenarios. The central system is set up to detect the locations of robots in real time and communicate with a *Raspberry Pi* board [3] on each robot to send the next command relying on its neighbors. Although the experiment configuration involves such external computations and sensors due to limited capability of *Thymio*, most of realistic assumptions hold, and the robots are still influenced by physical constraints and disturbance in motion. For better understanding of readers, we submit a supplementary video.

The location detection is performed at 4 frames per second at each of which a new command is received by each robot. Also, all coordinates and orientations at 2 frames per second are collected, which is not necessarily synchronized with the command timing.

Two different sizes of robot team are deployed throughout following experiments: 3 robots and 5 robots in which the *Head* robot continues to receive a command to move to an arbitrary destination point nearby. Table **??** shows details about the data collected from each configuration where a sample refers to a set of coordinates and orientations recorded at an instant time step, and an instance is a sequence of samples for 13 seconds. The instances are clustered to have at least 7-second time gap to another so that the motions between separate ones have only little dependency.

70% data from the 3-robot team is used to train *IPY* by feeding each sample, and the rest 20% is used for test. The trained model is also tested with the 5-robot team, in which the relay prediction is initialized at the beginning of each instance. The length of history is set to 5 seconds to accept inputs for past 10 time steps, and the predictions occur for the next 8 seconds.

IPY is implemented in *Tensorflow Python* library [2] to realize the entire pipeline. The evaluation is reported in Euclidean distance between the predicted position and the truth, using the model that achieves the best performance during 100 epochs, by which the loss mostly converges to zero.

A. Results

1) Qualitative analysis:

VI. Discussion & Future work

We have shown *IPY*, a powerful approach to tackle the remote localization problem where a robot with limited sensors is to predict positions of all others only using observations about its nearest neighbor. *IPY* follow the fundamental spirit of [1] to apply a predictor trained for a small robot team to a chain of them in a larger team, since it can provide a scalable feature without re-training. The distinction in *IPY* is , however, to use a sequence of historical knowledge and make predictions based on imagination of robot states including position and orientation. We expected learning with history would help regulate the model on physical constraints on the team formation, and also the imaginative representation could contain certainty about the prediction, which would bring about more robust performance.

The experiment results were obtained using a realistic robotic platform to prove that our method outperforms not only the regressor introduced in [1] but also a general state-of-the-art neural network model that is able to use historical data. Especially, our model can present stable performance even when the input contains some errors which have occurred in previous prediction.

In future work,

References

[1] T. Choi, T. P. Pavlic, and A. W. Richa, "Automated synthesis of scalable algorithms for inferring non-local properties to assist in multi-robot teaming," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2017, pp. 1522–1527.

[2] J. Shin, R. Siegwart, and S. Magnenat, "Visual programming language for thymio ii robot," in *Conference on Interaction Design and Children (IDC'14)*. ETH Zürich, 2014.

[3] E. Upton and G. Halfacree, *Raspberry Pi user guide*. John Wiley & Sons, 2014.

[1] http://www.github.com/ctyeong

[2] https://www.tensorflow.org/

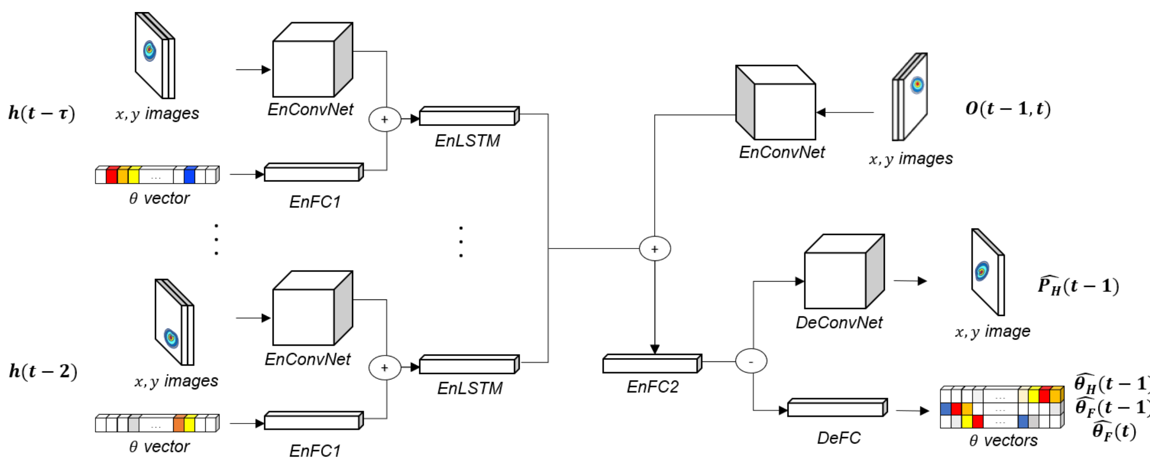

Fig. 2: Structure of *IPY*. This is an snapshot example when applied to a focused module of 3 robots, called *Tail, Follower, Head* within it, at time t. The Encoder-Decoder structure encodes 1) historical positions and orientations of *Follower* and *Head* until $t-2$ and 2) the observed positions of the *Follower* at $t-1$ and t. The decoder part learns to estimate 1) the position of *Head* at $t-1$ and 2) orientations of *Follower* at $t-1$ and t and *Head* at $t-1$.