Human Target Prediction in Cooperative Manipulation Tasks

Mahmoud Hamandi, Emre Hatay, and Pooyan Fazli

***Abstract*—We present a novel approach for fast prediction of human reaching motion in the context of human-robot collaboration in manipulation tasks. The method trains a recurrent neural network to process the three-dimensional hand trajectory and predict the intended target along with its certainty about the position. The network then updates its estimate as it receives more observations while advantaging the positions it is more certain about. To assess the proposed algorithm, we build a library of human hand trajectories reaching targets on a fine grid. Our experiments show the advantage of our algorithm over the state of the art in terms of classification accuracy.**

I. Introduction

Automated systems have been increasingly used in factories for the past decade, with many factories being operated completely by robots. However, these machines are usually placed in isolation from any humans due to safety reasons and their lack of understanding of human motion. Interest in human-robot coexistence is increasing in manufacturing environments. Significant productivity can be achieved if humans and robots can share the same workspace and work in close proximity. Robots can reduce the task completion time by concurrently working with humans.

With the increased proximity between humans and robots, it becomes imperative for robots to be aware of humans and eventually plan their motions as not to clutter the shared workspace, block human path, or inflict injuries. Inspired by these requirements, this work investigates the early classification of human arm motion, where the robot has to predict the target position the human is reaching for and plan its actions accordingly.

Mainprice *et al.* [1] provided a solution for the problem, where they learn a Gaussian Mixture Model representation for each possible target location and classify the arm trajectories online as reaching to the most probable one. The target classification is followed by a voxel occupancy calculation to know the safe area for the robot to reach. Later, Perez and Shah [2] presented another method, where they learn a motion library consisting of a Gaussian distribution for each target location and classify each trajectory to the most probable target after adding task level priors.

Similarly, Maeda *et al.* [3] presented a framework where they find the most likely sequence from the pool of sample trajectories stored in a lookup table and assume the human will follow a similar trajectory to the target. While these approaches can classify a small number of discrete targets accurately, it is desirable to learn a continuous space representation that allows targets to be placed anywhere in the workspace.

Mainprice *et al.* [4] presented an interesting approach, where they predict the trajectory of the human using STOMP algorithm [5] after learning the associated cost function from human demonstrations.

In this paper, we propose a network architecture, which we refer to as Human-INtended Target (HINT), that predicts the target position the human is reaching for. In addition, the network is trained to output its confidence of the estimate and update its belief as it receives more observations. We build a dataset that covers a large number of targets placed on a fine grid. We compare the performance of HINT with other methods from the literature. Our experiments show the advantage of the proposed algorithm over the state of the art.

II. Technical Approach

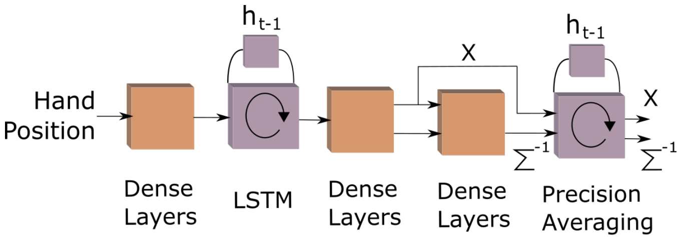
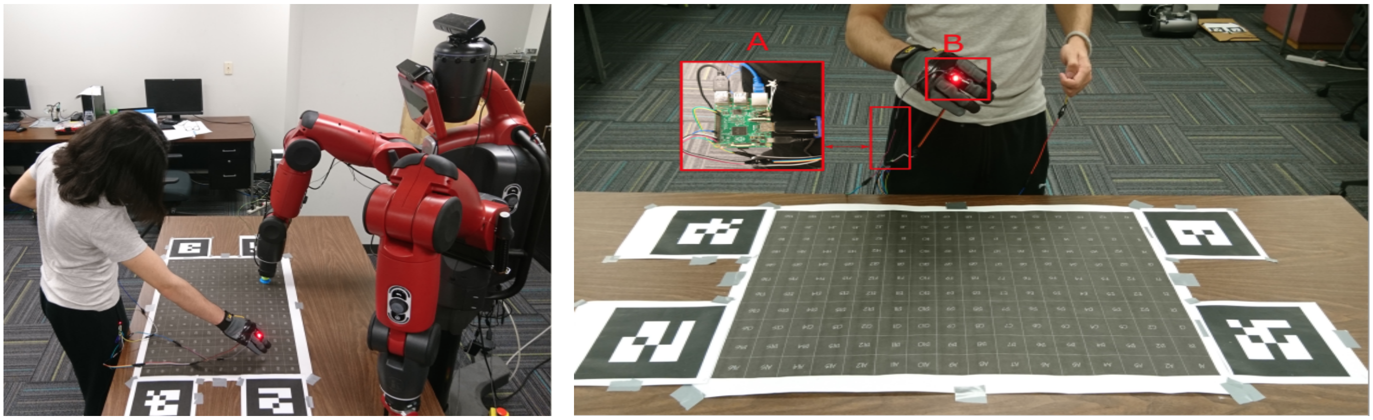


Fig. 1: HINT network architecture.

In this paper, we design a recurrent neural network shown in Figure 1 that predicts the reaching target and its confidence about the estimate for the given hand trajectory. The predicted target and confidence are then used in a new layer, which we refer to as the Precision Averaging layer. This layer averages previously estimated positions based on their confidence and can be explained mathematically as:

$$(\mathbf{x}_k, \Sigma_k^{-1}) = (\frac{\Sigma_{new}^{-1}\mathbf{x}_{new} + \gamma\Sigma_{old}^{-1}\mathbf{x}_{old}}{\Sigma_{new}^{-1} + \gamma\Sigma_{old}^{-1}}, \Sigma_{new}^{-1} + \gamma\Sigma_{old}^{-1}*), \quad (1)$$

where $\mathbf{x}_{old}$ and Σ_{old}^{-1} are the target prediction and precision for the previous step of the trajectory, $\mathbf{x}_{new}$ and Σ_{new}^{-1} are the target prediction and precision up to th*e Precision Averaging layer after seeing the last step of the trajectory, $(\mathbf{x}_k, \Sigma_k^{-1})$ are the resultant target prediction and precision calculated by the layer, and γ is a discount factor chosen to balance between the effect of the old prediction and the new one. The network is trained to minimize the squared error between the predicted and the actual target position $\mathbf{x}$. Since $\mathbf{x}$ is a precision weighted average of all previous predictions, the network learns to



(a) Human and robot sharing the same workspace in a collaborative manipulation task.

(b) The setup from the robot's point of view. (A) shows the microcontroller processing the IMU data, and (B) shows the red LED detected by the depth camera.

Fig. 2: Human-robot collaboration setup.

output a precision for each prediction relative to its certainty about the target position so that the final estimate would be dominated by the values it is more certain about.

To use the network for classification, we choose the class with the minimum distance to the predicted target position. However, due to the network's ability to output (x, y) positions instead of choosing a class, it is able to generalize to locations that it has not seen during training, while if it was trained for classification it would have been restricted to the training classes.

III. Experiments

To assess the performance of our algorithm, we collected a dataset of hand trajectories reaching for one of the cells of the grid shown in Figure 2. The targets were chosen randomly out of the 176 possible targets presented by the 16×11 grid fixed on the table with each target being represented by a $5 \times 5\ cm^2$. The aim of the grid is to discretize the workspace for a human to visually find a randomly assigned target, while being fine enough for the network to learn a continuous probability distribution of the target.

The human subject was instructed to start from the position where they feel most comfortable at and reach for the target being displayed on a screen in front of him. As the human reached for the target, we collected the hand position using a depth camera as well as readings from a 6 DOF IMU fixed to the hand, RGB images from the camera, and the point cloud generated by the camera. In total, we collected 704 trajectories distributed equally over the 176 target locations, with the order being chosen at random to assure the independence between one trajectory and another. These trajectories were split equally between training and testing, with some target locations being represented in only one of the two sets. Each target is represented by zero to four trajectories in the training set, and the remaining of the four collected trajectories are placed in the test set. To detect and track the hand position in the image and the point cloud, we fixed a bright LED light on the tip of the hand as shown in Figure 2, which we tracked in the image and point cloud to record the three-dimensional position of the hand. In addition, the image and the point cloud allow the detection of the grid location inferred by the four AR markers placed near the grid edges.

We intend to open-source the above described setup containing the drawings of the grid and AR markers, the software used for data collection, hardware description as well as the full collected dataset.

Figure 3 shows the collected trajectories for 16 targets in the grid, which shows that the trajectories are difficult to separate especially near the starting position.

IV. Results

We compare our algorithm with one other method from the literature [2], where they suggest to build a motion library presenting a probability flow tube (PFT) for each target position (i.e., the mean and variance of each position along the trajectory). During online classification, they warp the test trajectory using Dynamic Time Warping (DTW) [6] to match the trajectories with those in the motion library. Finally, they classify the target location as the one with the smallest Mahalanobis distance based on the matched means and variances from the motion library. It should be noted that this algorithm is not real time for more than 3 target locations without multi-threading, while our dataset presents 176 target locations. Figure 4 shows the average classification accuracy of our method and the benchmark algorithm. As the graph shows, our method has a higher accuracy throughout the trajectory. In addition, PFT has near zero accuracy in the beginning where the trajectories are non-separable, while our algorithm has near 30% accuracy. It is also observed that near the end of the trajectories HINT reaches a plateau of 80% while PFT reaches only 30%. This can be related to two reasons: first, PFT is designed to be trained on roughly 20 demonstrations of each

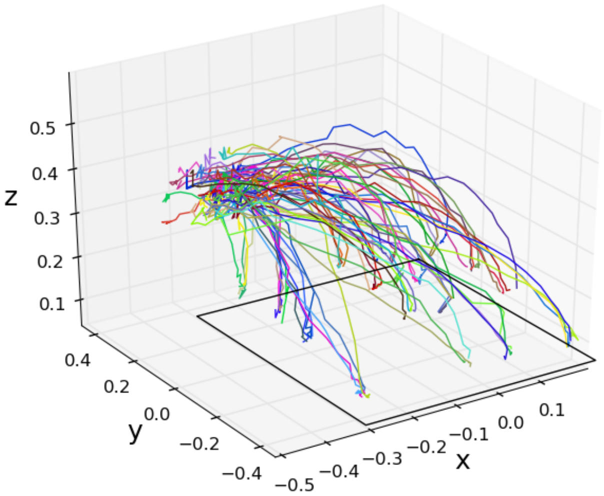


Fig. 3: Sample hand trajectories reaching 16 targets on the grid.

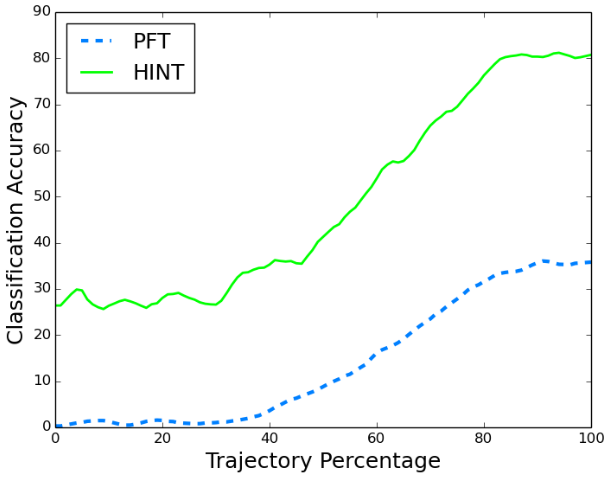


Fig. 4: Classification accuracy comparison between our method and the benchmark method.

class to build an understanding of the shape of the trajectories and the possible variance of each, while the dataset presented here has zero to four examples of each in the training set. Second, since HINT learns to output an (x, y) position instead of a class, it is able to generalize to targets not shown in the training dataset.

V. Conclusion and Future Work

We presented HINT, a network designed to predict human target location after processing one's hand trajectory. Our experiments showed HINT's advantage over the PFT approach from the literature [2]. In the next phase, we intend to compare our method against the Gaussian Mixture Model (GMM) approach presented in [1], where they learn a GMM for each target location from the corresponding sample trajectories. In comparison with PFT, this method should require more trajectories for each target position to learn a GMM instead of just one Gaussian model; however, it might be able to perform better on our data as it can handle the variance of the starting position, shown in Figure 3, better than PFT.

Finally, we intend to collect more trajectories corresponding to the same 176 target locations to provide a more thorough comparison against both PFT and GMM, as both require more training examples than our method, which can train on a smaller number of demonstrations.

References

[1] J. Mainprice and D. Berenson, "Human-robot collaborative manipulation planning using early prediction of human motion," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, 2013, pp. 299–306.

[2] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *Proceedings of the International Conference on Robotics and Automation, ICRA*, 2015, pp. 6175–6182.

[3] G. Maeda, A. Maloo, M. Ewerton, R. Lioutikov, and J. Peters, "Anticipative interaction primitives for human-robot collaboration," in *AAAI Fall Symposium Series*, 2016.

[4] J. Mainprice, R. Hayne, and D. Berenson, "Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2015, pp. 885–892.

[5] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2011, pp. 4569–4574.

[6] J. Kruskal and M. Liberman, *The Symmetric Time-Warping Problem: From Continuous to Discrete*, Addison-Wesley, 1983.