# Predicting the Target in Human-Robot Manipulation Tasks

Mahmoud Hamandi[1], Emre Hatay[1], and Pooyan Fazli[2(✉)]

[1] Electrical Engineering and Computer Science Department, Cleveland State University, Cleveland, OH 44115, USA
{m.hamandi,e.hatay}@csuohio.edu
[2] Department of Computer Science, San Francisco State University, San Francisco, CA 94132, USA
pooyan@sfsu.edu

**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.

**Keywords:** Human-robot collaboration · Robot manipulation
Robot learning

## 1 Introduction

Automated systems have been increasingly used in factories for the past decade [1–3]. However, these machines are usually placed in isolation from any humans due to safety reasons [4] and their lack of understanding of human motion. Interest in human-robot coexistence is increasing in manufacturing environments [5]. 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 the robots to be aware of humans and plan their motions in such a way that avoids cluttering the shared workspace, blocking human path, or inflicting 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.

Early work presented by Jung and Park [6] learned the relation between the intended target of an operator and their pose through a neural network.


S. S. Ge et al. (Eds.): ICSR 2018, LNAI 11357, pp. 580–587, 2018.
https://doi.org/10.1007/978-3-030-05204-1_57

Their work connected the 3-dimensional target position with the pose of the human shoulder, elbow, and wrist, which they consider enough to understand the hand motion. Although their work is promising, it does not consider an online prediction phase, but rather understands the connection between the pose and the target positions to build an ergonomic product for the human operator.

Mainprice *et al.* [7] 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. While this approach provides promising results, it requires extensive data (25 trajectories) for each possible target location, rendering it difficult to generalize to continuous spaces.

Later, Perez and Shah [8] 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. Then during runtime, they warp each path to match the learned one using Dynamic Time Warping (DTW) [9] and classify the trajectory as reaching to the highest Gaussian probability density function. This approach is able to learn target specific trajectories with less data than the former method, however, it is not certain to generalize to unseen targets.

Similar to the above methods, Maeda *et al.* [10] 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.* [11] presented an interesting approach, where they predict the trajectory of the human using STOMP algorithm [12]. Their approach learns the human motion cost function from demonstrations using path integral inverse optimal control [13] and then predict the target and human motion online after applying the cost functions into STOMP. While this approach is very promising, it requires an accurate capture of the pose of multiple joints of the human arm while reaching for the target in real-time.

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 continuous environments solemnly from the trajectory of the tip of his hand. In addition, the network is trained to output its confidence of the estimate and update its belief as it receives more observations.

Figure 1 shows the offline and online phases of our system. In the offline stage, we build a dataset of hand trajectories reaching for one of the targets placed on a table and train the network to match the collected hand positions and targets. Then, in the online stage, we predict the target position intended by the human each time we receive a new hand position. The target position will be used later to help the robot plan its trajectory in a safe manner. We compare the performance of HINT with other methods from the literature to show the advantage of the proposed algorithm over the state of the art.

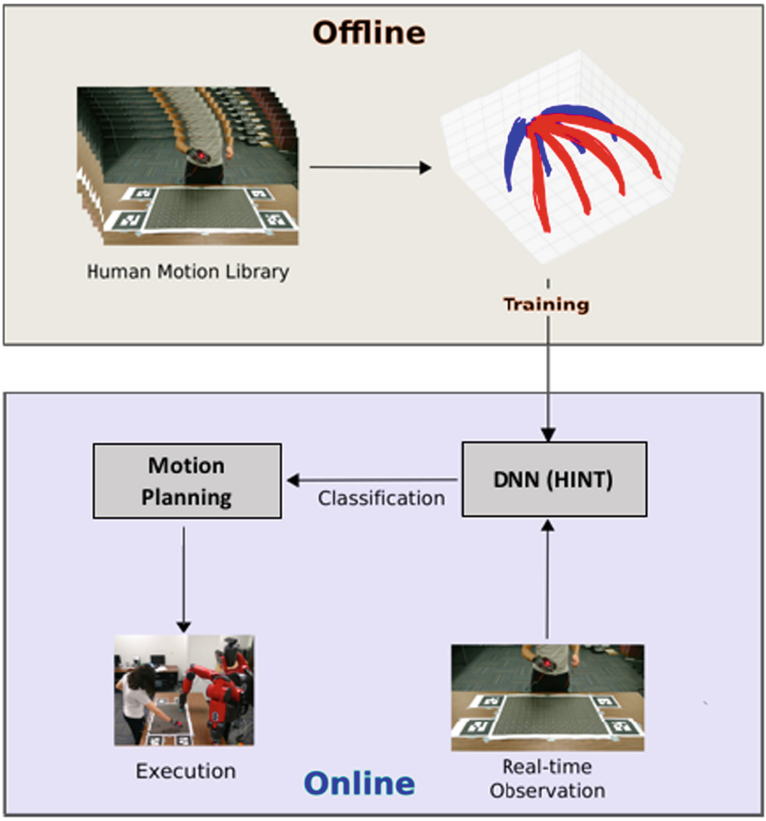



**Fig. 1.** In the offline phase, the network (HINT) is trained to match the collected hand trajectories and corresponding target positions. In the online phase, hand positions are detected in real time and corresponding target positions are predicted. Human intended target positions are used later to plan the robot motion in a safe manner.

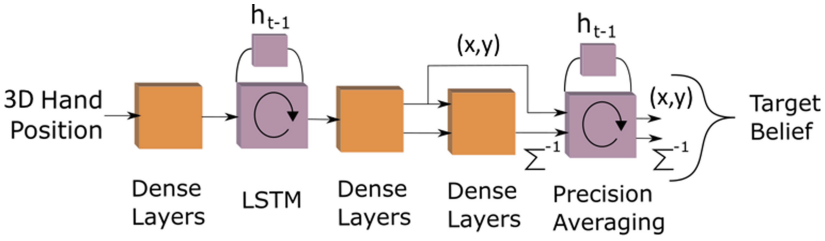



**Fig. 2.** HINT network architecture.

## 2 Technical Approach

In this paper, we design a recurrent neural network shown in Fig. 2 that predicts the reaching target and its confidence about the estimate for the given hand trajectory. The network reads the current hand position in the 3-dimensional space and processes the input with a series of four dense layers, each with 128 nodes followed by a *tanh* activation function. This first group of dense layers are followed by a recurrent *LSTM* layer with 128 hidden nodes. The recurrent layer is then followed by a group of three dense layers with 128, 64, and 2 nodes respectively, which output the $(x, y)$ position of the target. The first two of these layers are equipped with a *tanh* activation function, while the last one has no activation function.

The output $(x, y)$ position of the target is fed into another row of dense layers along with the output of the 64 node dense layer. This extra row of dense layers is used to predict the precision $\Sigma^{-1}$ of the $(x, y)$ estimate. As such, this row of dense layers is constructed out of three consecutive layers, with 32, 16, and 4 nodes respectively, where the first two have a *tanh* activation function, while the last has no activation function. The output 4 nodes represent the 2-dimensional precision matrix.

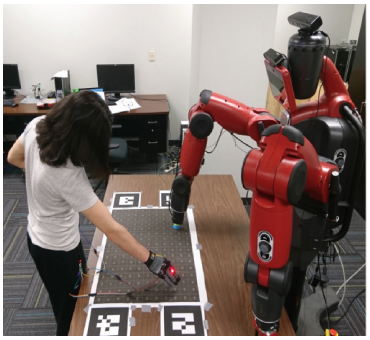
The output $(x, y)$ and precision $\Sigma^{-1}$ are then processed by a new recurrent layer, which we refer to as the the Precision Averaging layer, to provide the final target position and precision. This layer averages previously estimated positions based on their confidence and can be explained mathematically as:

$$(\mathbf{p}, \Sigma^{-1}) = (\frac{\Sigma_{new}^{-1}\mathbf{p}_{new} + \gamma\Sigma_{old}^{-1}\mathbf{p}_{old}}{\Sigma_{new}^{-1} + \gamma\Sigma_{old}^{-1}}, \Sigma_{new}^{-1} + \gamma\Sigma_{old}^{-1}), \tag{1}$$

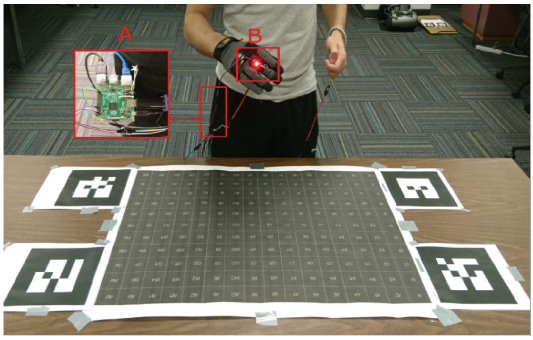
where $\mathbf{p}_{old}$ and $\Sigma_{old}^{-1}$ are the predicted target position $(x, y)$ and precision for the previous step of the trajectory, $\mathbf{p}_{new}$ and $\Sigma_{new}^{-1}$ are the predicted target position $(x, y)$ and precision up to the Precision Averaging layer after seeing the last step of the trajectory, $(\mathbf{p}, \Sigma^{-1})$ are the resultant target prediction $(x, y)$ and precision calculated by the layer, and $\gamma$ 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. Since the predicted $(x, y)$ position is a precision-weighted average of all previous predictions, the network learns to 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 in a continuous space 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.



(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. 3.** Human-robot collaboration setup.

## 3 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 Fig. 3. The targets were chosen randomly out of the 176 possible targets presented by the $16 \times 11$ grid fixed on the table, with each target being represented by a $5 \times 5\,\mathrm{cm}^2$. 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 targets changed, the human was allowed to move freely and as such multiple start positions exist for each trajectory. 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 positions, 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 positions 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 Fig. 3, which we tracked in the image and point cloud to record the 3-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.

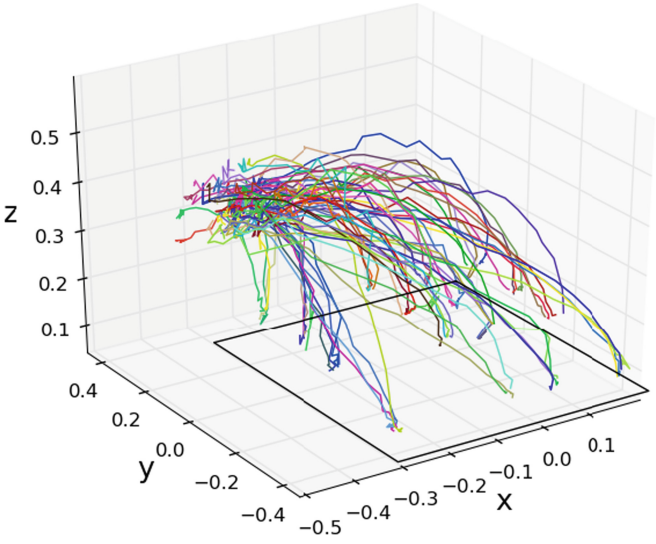



**Fig. 4.** Sample hand trajectories reaching 16 targets in the grid.

Figure 4 shows the collected trajectories for 16 targets in the grid, demonstrating that the trajectories are difficult to separate especially near the starting position.

## 4 Results

We compare our algorithm with one other method from the literature [8] where they suggest to build a motion library presenting a probability flow tube (PFT) [14] 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) [9] to match the trajectories with those in the motion library. Finally, they classify the target location as the one with the smallest Mahalanobis distance [15] 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 positions without multi-threading, while our dataset presents 176 target positions.

Figure 5 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 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

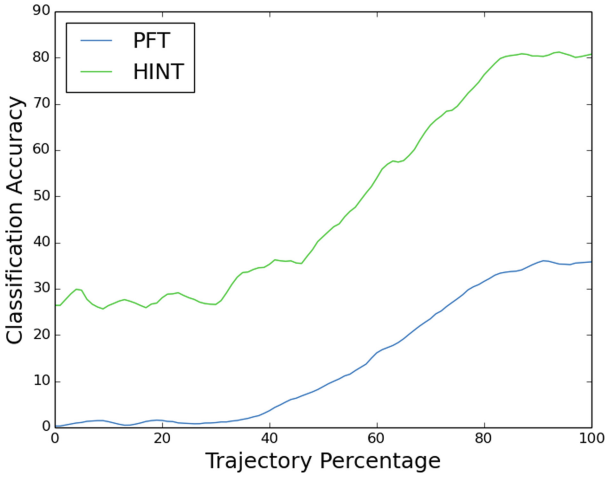



**Fig. 5.** Classification accuracy comparison between our method, HINT, and the benchmark method, PFT.

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 set.

## 5 Conclusion and Future Work

We presented HINT, a novel method for early prediction of human intended target. Our deep learning based method reads 3-dimensional hand positions and predicts the 2-dimensional target position and its confidence about the prediction. We built an extensive dataset of hand trajectories reaching for known targets on a table. Our dataset contains multiple trials for each target position, with each trial starting from a randomly chosen position. HINT outperformed the benchmark algorithm PFT by 50% in classification accuracy when trained and tested on the created dataset. In the future, we intend to collect our data with a motion capture system to reduce the noise in the recorded positions.

**Acknowledgments.** The authors would like to thank Nuo Zhou for her assistance in developing the software and collecting the data.

## References

1. Lasota, P.A., Fong, T., Shah, J.A., et al.: A survey of methods for safe human-robot interaction. Found. Trends Robot. **5**(4), 261–349 (2017)
2. Broquere, X., Sidobre, D., Herrera-Aguilar, I.: Soft motion trajectory planner for service manipulator robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 2808–2813 (2008)

3. Calinon, S., Sardellitti, I., Caldwell, D.G.: Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 249–254 (2010)
4. Dhillon, B., Fashandi, A., Liu, K.: Robot systems reliability and safety: a review. J. Qual. Maint. Eng. **8**(3), 170–212 (2002)
5. Shi, J., Jimmerson, G., Pearson, T., Menassa, R.: Levels of human and robot collaboration for automotive manufacturing. In: Proceedings of the Workshop on Performance Metrics for Intelligent Systems, PerMI, pp. 95–100 (2012)
6. Jung, E.S., Park, S.: Prediction of human reach posture using a neural network for ergonomic man models. Comput. Ind. Eng. **27**(1–4), 369–372 (1994)
7. Mainprice, J., Berenson, D.: 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, pp. 299–306 (2013)
8. Pérez-D'Arpino, C., Shah, J.A.: 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, pp. 6175–6182 (2015)
9. Kruskal, J., Liberman, M.: The Symmetric Time-Warping Problem: From Continuous to Discrete. Addison-Wesley, Reading (1983)
10. Maeda, G., Maloo, A., Ewerton, M., Lioutikov, R., Peters, J.: Anticipative interaction primitives for human-robot collaboration. In: Proceedings of the AAAI Fall Symposium Series, pp. 325–330 (2016)
11. Mainprice, J., Hayne, R., Berenson, D.: 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, pp. 885–892 (2015)
12. Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S.: STOMP: stochastic trajectory optimization for motion planning. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pp. 4569–4574 (2011)
13. Kalakrishnan, M., Pastor, P., Righetti, L., Schaal, S.: Learning objective functions for manipulation. In: Proceedings of the IEEE International Conference on Robotics and Automation, ICRA, pp. 1331–1336 (2013)
14. Dong, S., Williams, B.: Learning and recognition of hybrid manipulation motions in variable environments using probabilistic flow tubes. Int. J. Soc. Robot. **4**(4), 357–368 (2012)
15. De Maesschalck, R., Jouan-Rimbaud, D., Massart, D.L.: The mahalanobis distance. Chemom. Intell. Lab. Syst. **50**(1), 1–18 (2000)