Estimating the Center of Mass of an Unknown Object for Nonprehensile Manipulation

Ziyan Gao, Armagan Elibol, and Nak Young Chong
School of Information Science
Japan Advanced Institute of Science and Technology
Nomi, Ishikawa 923-1292, Japan
{*s1920013, aelibol, nakyoung*}*@jaist.ac.jp*

***Abstract*—With the increasing prevalence of robot-led automation in many fields such as industry, health-care, agriculture, *etc.*, robot manipulator arms are often required to handle sophisticated manipulation tasks in which both the object and environmental physical parameters are unknown. In order to deal with these tasks, fast and accurate estimation of the inertial parameters of the object and frictional characteristics of flooring surfaces are of crucial importance toward developing intelligent and efficient object manipulation strategies. In this work, we propose an integrated framework for estimating the center of mass of an unknown planar object using a force sensor-less manipulator arm pushing the object on a horizontal plane. We evaluate two algorithmic solutions through extensive pusher-slider frictional interaction simulations. The result shows that the proposed framework can estimate the center of mass location efficiently and accurately only with a few pushing interactions.**

***Index Terms*—Planar Pushing, Frictional Interaction, Center of Mass Estimation, Few-Shot Learning, Voting Theorem**

I. INTRODUCTION

Currently, robots are expected to not only be able to execute human-programmed actions in environment-controlled industrial plants but also capable of perceiving object geometry and inertial parameters of the object for autonomous dexterous manipulation without human intervention. Probing the inertial parameters of the object is important for energy-efficient manipulation [1] as well as motion planning [2]–[4]. For instance, Mavrakis *et al.* [1] leverages the object's inertial parameters for grasp selection to minimize the joint torque. Song *et al.* [2] identifies the object mass distribution in order to efficiently re-arrange objects to facilitate grasping. As different locations of an object' center of mass (CoM) leads to a different response to the contact force in planar pushing, Li *et al.* [3] utilize a recurrent neural network simultaneously to learn a representative feature for forwarding modeling as well as to predict the CoM. Gao *et al.* [4] show that the CoM information can help improve the sampling efficiency for object re-arrangement.

In this work, we propose two methods for estimating an object's CoM. The first method uses a set of given actions and their resulting predicted motion via a few-shot motion prediction model proposed by Gao *et al* [5]. It accumulates the total amount of weighted predicted amount of rotation for the points in the object mask and aims to segment the area where this accumulation is minimum as it will be the region that is less likely to rotate. The second method estimates the CoM by combining the object motion prediction model with Mason's Voting Theorem (VT) [6] to refine the possible CoM region progressively with aiming to minimize the total number of pushes needed. This aim is modeled by using a proposed Average Equality Score function. We evaluated the proposed methods using 20 different objects with a total number of $6,000$ random sets of actions in a simulation environment.

II. RELATED WORK

Planar pushing has been a quite involved research topic studied in modeling, planning, control as well as physical property estimation perspective. Mason [6] pioneered the study of quasi-static dynamics of a pusher-slider system, relating the sense of object rotation to the spatial relationship between the object CoM and several rays specified by the friction cone at the contact point and pushing direction. Goyal [7] proposed the concept"limit surface" to relate the applied contact force with the object motion under the maximum energy dissipation and Coulomb friction law. This limit surface is approximated by an ellipsoid model for control by Lynch [8]. Recently, data driven models have been used for the same modeling problem by Zhou *et al.* [9] and Bauza *et al.* [10].

For estimating the physical properties of the object by planar pushing, many methods can be classified into the exploratory category in an overview by Mavrakis *et al.* [11]. Yu *et al.* [12] proposed a least-squares solution for finding a rectangular object's inertial parameters given a series of force and object acceleration measurements. On the other hand, Mavrakis *et al.* [13] proposed a data-driven method for inertial parameter estimation. Kloss *et al.* [14] employed an extended Kalman filter with an ellipsoid force-motion model to estimate these parameters without force measurement. However, the estimation error is prone to be large. Xu *et al.* proposed a disentangled learning framework to

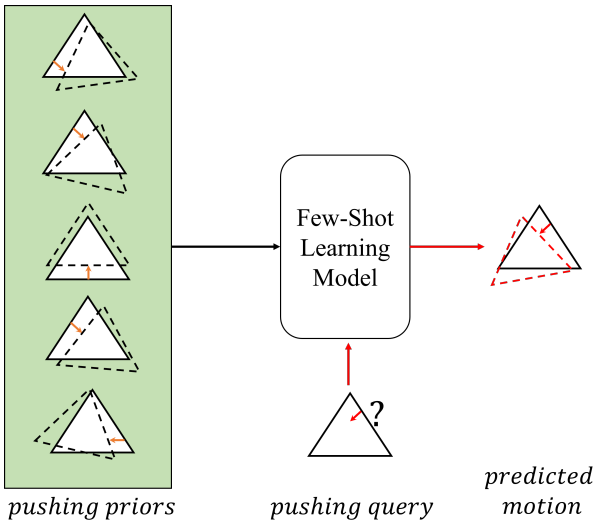

Fig. 1. Few-shot learning model for object motion prediction

distinguish different objects' physical properties by iterative dynamic pushing and colliding. Kumar *et al.* [15] proposed a reinforcement learning method to interact with the articulated object for predicting the mass distribution of the object. In this work, we aim to estimate the CoM of an unknown object pushed by a force sensor-less manipulator arm on a horizontal surface. Instead of directly predicting the CoM location, we first employ a few-shot learning model to predict the resultant object rotation, which gives clues to the location of CoM. We then estimate a region containing the CoM based on the spatial relationship between the applied pushing action and the resultant object rotation.

III. Center of Mass Estimation Methods using Motion Prediction Model

A. few-shot learning model

In this work, we make use of a few-shot learning motion prediction model illustrated in Fig. 1. This model is developed based on attentive neural process (ANP) [16], leveraging the pushing priors whose resultant object motions are known to help predict the resultant motions for the pushing query. The benefit of ANP is that, unlike Gaussian Process (GP) whose time complexity is in $\mathbf{O}(n^3)$ which might be intractable with the increase of pushing priors, the relationship between the computational cost and the number of pushing priors is linear.

Specifically, this model predicts the object motion for a given action defined by the starting and ending point coordinates in the image frame. Each pushing prior is composed of an object mask, a pushing action, and the resultant object motion. Each pushing query consists of the same object mask and pushing action only. The model employs a residual convolutional neural network [17] mapping pushing priors and pushing query into high dimensional feature vectors. Then the model utilizes the Transformer [18] to predict the resultant object motion for pushing query based on these feature vectors. In the training phase, the model is trained to predict the resultant object motion for pushing actions with a magnitude of $3cm$ under the quasi-static interactions assumption that the object will stop sliding/rotating immediately if the pusher breaks contact with it. After training, the model leverages as few as twelve pushing priors to predict the resultant object motion for arbitrary pushing actions. This model is originally used to re-arrange the object to the target pose by pushing, but in this work, we use it to estimate the CoM by combining it with the VT.

B. Voting Theorem

In the VT, if the floor surface on which the object slides is flat and uniform and the quasi-static assumption holds, then the relationship between the contact force and the object rotation caused can be determined by the spatial relationship between the object's CoM and several rays. The rays are illustrated in Fig. 2. R_L and R_R delimit the left and right boundaries of the friction cone at the contact position, respectively, and the R_P is the pushing direction at the contact position. R_L, R_R and R_P vote on the sense of rotation. The vote is performed by examining what sign of moment the ray has to the CoM of the object. If R_L and R_R vote for clock-wise rotation and R_P votes for counter clock-wise rotation (illustrated on the left of Fig. 2), then the object will rotate clock-wise as two rays have voted for clock-wise. On the other hand, if R_L votes for clock-wise rotation but R_R and R_P vote for counter clock-wise (illustrated on the right of Fig. 2), then the object will rotate counter clock-wise correspondingly.

In the scenario of CoM estimation, we can use VT to determine a region which must contains the CoM for some pushing action and resultant rotation pairs. When the R_P is along with the contact normal, then the region can be determined without knowing the coefficient of friction at the contact point. If R_P disagrees with the contact normal, there are some cases where the region can be determined without knowing the coefficient of friction. For instance, given the contact normal referred to as $\mathbf{n}$, if $R_P \times \mathbf{n}$ is positive and the resultant object rotation is counter-clockwise, then the region can still be determined based on VT.

C. CoM detection methods

Inspired by the fact that if the contact force passes through the CoM, then the object will be purely translated, otherwise, the object will rotate clock-wise or counter clock-wise based on VT, we propose an algorithm summarized in Alg. 1. It assumes given a set of actions defined by their start and end-points and their predicted motions on the object. Our idea is based on accumulating rotations for each point (pixel) in the object mask image weighted accordingly to

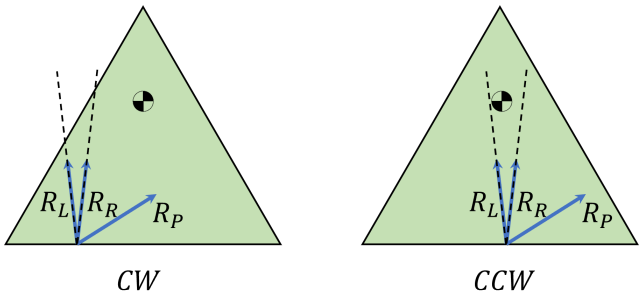

Fig. 2. Illustration of rotation prediction by Voting Theorem

their distance to lines defined by the start and end-point actions and identifying the region that has the lowest amount of accumulated rotation as a potential region where the CoM is possibly located. This can be also interpreted as the region where actions generating a small amount of rotations pass. For each action line (calculated via its start and end-points), we compute the distance from pixels to the line and use this distance as a weight for the predicted amount of rotation generated by this action. This way of the computed amount is accumulated for each pixel used for each action. We then apply a threshold to segment the potential region. Due to the selected threshold, there might be some isolated locations. In order to filter them out, we make use of the connected component method to find the largest connected region within the segmented pixels. This largest connected region is regarded as the potential CoM region and its center is computed as the estimated CoM position.

Algorithm 1: CoM Region Detection Process

Input: Object Mask $\mathbf{P} = (x_i, y_i) \quad i = 1, 2, \cdots, m$,
List of actions $\mathbf{A} = (sx_i, sy_i, ex_i, ey_i)$ and predicted motion $(\Theta_i, tx_i, ty_i) \quad i = 1, 2, \cdots, n$.
Area segmentation threshold $s_t \in [0, 1]$
Output: Predicted location of CoM, $CoM = (x, y)$

```
/* Create an auxiliary matrix,
   S_θ(x,y), for accumulating distance
   weighted rotation amount          */
```

$\mathbf{S}_\theta(\mathbf{x}, \mathbf{y}) \leftarrow 0$
foreach *action a in* $\mathbf{A}_{keep}$ **do**
  $\mathbf{l} \leftarrow$ Compute the pushing line using a
  foreach *point p in* $\mathbf{P}$ **do**
    $d \leftarrow$ Compute distance between p and $\mathbf{l}$
    $\mathbf{S}_\theta(p) \leftarrow \mathbf{S}_\theta(p) + e^{(-d)} \cdot |\Theta_a|$
$\mathbf{S}_\theta \leftarrow$ normalize $\mathbf{S}_\theta$ to the interval $[0, 1]$
$\mathbf{P_c} \leftarrow S_\theta < s_t$ /* Create a segmented region of object mask as $\mathbf{P_c}$ */
$\mathbf{P_c} \leftarrow$ The largest connected region in $\mathbf{P_c}$
$CoM \leftarrow$ Compute mean x and y of points in $\mathbf{P_c}$

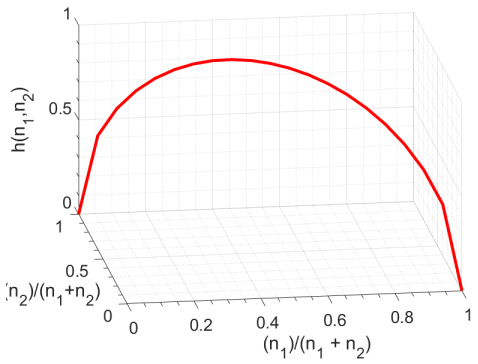

Fig. 3. Plot of an Action Equality Score function formulated in Eq. 1. It reaches its maximum value when divided regions are equal to each other.

Combining the prediction model with the VT makes it possible to narrow down the possible CoM locations. Thus, we propose a progressive algorithm given in Alg. 2. It aims to narrow down progressively the potential area where the CoM is located as well as reduce the total number of pushes, unlike the previous algorithm. Based on the VT and the resulting object rotation (clockwise or counter-clockwise) of an applied action, the potential area for CoM can be restricted. In order to minimize the total number of actions used, we propose to select the action whose line divides the potential region into approximately equal two pieces. In order to measure efficiently the equality of the divided regions in terms of number of points (or pixels), we devise to use the following function named as Action Equality Score;

$$h(n_1, n_2) = \frac{2\sqrt{n_1 \times n_2}}{n_1 + n_2} \tag{1}$$

For a given action a, n_1 and n_2 are the total number of points in two regions divided by the action line in the object mask image. This function reaches its maximum value (which is 1) when n_1 is equal to n_2 and its plot is illustrated in Fig. 3. We also remove actions that divides subregions extremely unequal (*e.g.*, $(\%10, \%90)$) using a threshold value on their equality score.

IV. Numerical Results

In order to evaluate the performance of the proposed methods, we created a simulation environment using CoppeliaSim and Vortex Physics Engine. We tested our algorithms on 20 objects designed in different shapes and sizes. Fig. 4 shows the objects used in computational experiments.

An example of a normalized heat map representation for accumulated distance weighted amount of rotation (represented by $\mathbf{S}_\theta$, which is computed in Alg. 1) is depicted in Fig. 5. Following this example, by applying a segmentation threshold and connected component algorithm, a potential CoM region can be determined. Fig. 6 denotes applying these steps to the example case.

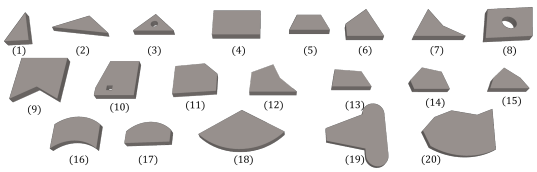

Fig. 4. Tested objects used in computational experiments.

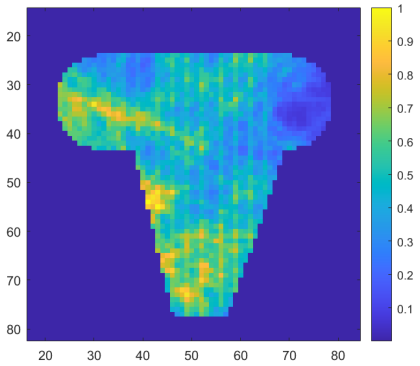

Fig. 5. An example of normalized heat map representation for accumulated distance weighted amount of rotation using Object19.

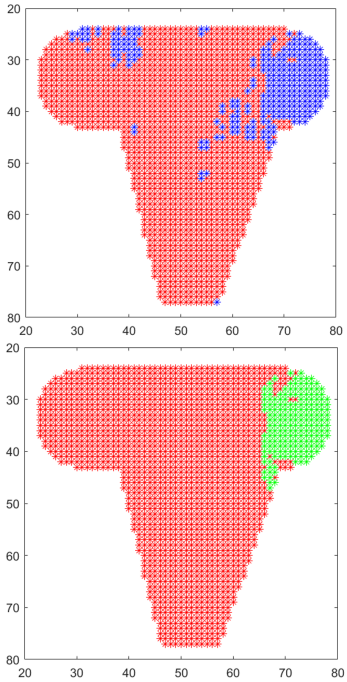

Fig. 6. While the figure on top denotes segmented points, the figure bottom shows the largest connected component extracted from the segmented points. The center of this region is the estimated CoM and for this specific example, the distance between the estimated CoM and the real one is 2.02 pixels equivalent to 0.54cm.

Algorithm 2: Progressive CoM Region Detection Algorithm

Input: Object Mask $\mathbf{P} = (x_i, y_i) \quad i = 1, 2, \cdots, m$,
List of actions $\mathbf{A} = (sx_i, sy_i, ex_i, ey_i)$ and predicted motion $(\Theta_i, tx_i, ty_i) \quad i = 1, 2, \cdots, n$,
Line-to-point distance threshold d_t
Area ratio $s_t \in [0,1]$, t_θ small rotation threshold, and t_h action score threshold
/* We used $s_t = 0.1$, $d_t = 0.5pix.$, */
/* $t_\theta = 5^\circ$, and $t_h = 0.6$ */
Output: Predicted location of CoM, $CoM = (x, y)$

$\mathbf{A}_{keep} \leftarrow$ Remove actions $|\Theta_i| < \mathbf{t}_\theta$
$\mathbf{P_c} \leftarrow \mathbf{P}$
while *size of* $\mathbf{P_c} > s_t \times$ *size of* $\mathbf{P}$ *or* $\mathbf{A}_{keep} \neq \varnothing$ **do**
  foreach *action a in* $\mathbf{A}_{keep}$ **do**
    Compute the number of points (n_1, n_2) in two subregions of $\mathbf{P_c}$ divided by the action a line
    Compute Equality Score, h_a via the function defined in Eq. 1.
  Choose action a_s in $\mathbf{A}_{keep}$ with the highest h score dividing $\mathbf{P_c}$ approximately equal two pieces $\mathbf{P_{c1}}$ and $\mathbf{P_{c2}}$.
  $\mathbf{P_c} \leftarrow \mathbf{P_{c1}}$ or $\mathbf{P_{c2}}$ accordingly to the predicted rotation Θ_a following VT Theory
  $\mathbf{A}_{keep} \leftarrow$ Remove actions $h_{a_i} < t_h$ and a_s
$\mathbf{P_c} \leftarrow$ The largest connected region in $\mathbf{P_c}$
$CoM \leftarrow$ Compute mean x and y of points in $\mathbf{P_c}$

Computational experiments were carried out on 300 different simulations for each object obtained using different parameters in object properties. Table I presents the statistical summaries (in the form of the mean and standard deviation) as well as the minimum and maximum distance between the real and the estimated CoM coordinates using Alg. 1. These distance measurements are given in cm and we also included the object sizes in cm^2.

Experimental results are detailed in Table II for the method described in Alg. 2. Since this algorithm works in a progressive manner and also aims to reduce the total number of pushing actions, we present the number of pushing actions used in order to estimate the CoM position. In order to verify the efficiency of using the Action Equality Score function, we tested our algorithm without using this score function-based elimination. Instead, we simply sort actions with the total absolute (discarding clock-wise or counter clock-wise) amount of their predicted rotation in ascending order, since a pushing action generating a small amount of rotation is likely that its line passes near a CoM location. Fig. 7 denotes the comparison for the total number of pushes done. It can be observed that filtering and selecting actions via their equality score provided a notable reduction in the

TABLE I
COMPUTATIONAL RESULTS OBTAINED USING ALG. 1

Objects	Object Size	Low Friction				High Friction			
		mean	std	min	max	mean	std	min	max
Obj1	32.9	0.742	0.251	0.056	1.691	0.717	0.323	0.020	1.785
Obj2	35.6	0.696	0.317	0.056	1.693	0.697	0.327	0.020	1.950
Obj3	27.0	0.458	0.235	0.009	1.491	0.671	0.335	0.007	2.473
Obj4	103.3	1.004	0.381	0.019	2.076	1.311	0.515	0.005	3.213
Obj5	42.5	0.786	0.349	0.034	1.846	0.835	0.390	0.010	2.193
Obj6	58.0	0.856	0.335	0.008	2.293	1.034	0.397	0.021	2.412
Obj7	53.4	0.849	0.362	0.045	1.991	1.127	0.502	0.023	3.095
Obj8	110.4	1.096	0.407	0.047	2.471	1.223	0.510	0.065	3.084
Obj9	119.4	1.240	0.470	0.043	2.662	1.320	0.587	0.026	4.118
Obj10	91.0	0.906	0.427	0.013	2.030	1.462	0.536	0.034	2.786
Obj11	83.0	1.132	0.441	0.058	2.324	1.208	0.603	0.013	2.738
Obj12	58.3	1.100	0.361	0.050	2.463	1.075	0.461	0.022	3.643
Obj13	38.2	0.673	0.255	0.025	1.395	0.791	0.313	0.022	2.407
Obj14	43.0	0.859	0.440	0.066	2.185	0.749	0.344	0.015	2.153
Obj15	35.2	0.633	0.317	0.017	1.606	0.751	0.328	0.026	2.154
Obj16	67.8	0.939	0.323	0.079	3.673	0.884	0.378	0.016	2.352
Obj17	51.6	0.563	0.270	0.007	1.355	0.578	0.278	0.002	1.814
Obj18	120.4	1.029	0.445	0.043	2.801	1.170	0.591	0.012	4.210
Obj19	126.3	1.025	0.648	0.013	2.947	1.377	0.737	0.021	4.279
Obj20	150.5	1.233	0.485	0.173	3.329	1.317	0.561	0.101	4.133

TABLE II
COMPUTATIONAL RESULTS OBTAINED USING ALG. 2

Objects	Object Size	Low Friction Settings								High Friction Settings							
		Errors in cm				Number of Pushing Actions				Errors in cm				Number of Pushing Actions			
		mean	std	min	max	mean	std	min	max	mean	std	min	max	mean	std	min	max
Obj1	32.9	0.576	0.250	0.024	1.690	2.688	0.665	1	5	0.741	0.344	0.038	2.158	3.519	0.754	2	6
Obj2	35.6	0.764	0.361	0.026	2.163	3.146	0.627	2	5	0.817	0.447	0.013	2.461	3.976	0.503	2	6
Obj3	27.0	0.537	0.300	0.024	1.506	2.808	0.593	2	5	0.597	0.306	0.011	2.168	3.575	0.678	2	5
Obj4	103.3	1.120	0.527	0.092	2.353	3.198	0.939	1	6	1.342	0.618	0.062	3.799	4.160	0.606	2	6
Obj5	42.5	0.812	0.387	0.051	1.801	2.858	0.709	1	5	0.846	0.420	0.032	3.111	4.049	0.281	3	6
Obj6	58.0	0.729	0.382	0.035	2.009	3.473	0.638	2	5	1.017	0.486	0.014	3.005	3.830	0.600	2	6
Obj7	53.4	0.892	0.413	0.049	1.991	3.050	0.656	2	5	1.235	0.629	0.029	3.427	3.445	0.708	2	6
Obj8	110.4	1.089	0.494	0.055	2.676	2.885	0.765	2	5	1.247	0.571	0.060	3.125	3.876	0.653	2	7
Obj9	119.4	1.346	0.609	0.131	2.802	3.228	0.646	2	6	1.246	0.591	0.031	4.806	3.743	0.656	2	7
Obj10	91.0	1.241	0.408	0.120	2.536	3.398	0.807	2	5	1.099	0.499	0.042	3.103	3.795	0.718	2	6
Obj11	83.0	1.101	0.381	0.162	2.780	3.013	0.685	1	5	0.941	0.454	0.036	2.882	4.093	0.635	2	6
Obj12	58.3	1.103	0.462	0.069	2.629	3.120	0.752	2	5	1.138	0.492	0.009	3.743	3.855	0.627	2	6
Obj13	38.2	0.735	0.289	0.039	1.802	2.878	0.783	1	5	0.726	0.382	0.059	2.472	4.175	0.444	2	6
Obj14	43.0	0.865	0.497	0.017	2.253	3.222	0.615	2	6	0.759	0.374	0.036	2.556	4.236	0.615	2	6
Obj15	35.2	0.614	0.376	0.031	1.716	3.156	0.755	2	6	0.626	0.348	0.049	2.141	4.027	0.654	2	6
Obj16	67.8	1.087	0.496	0.151	2.432	3.171	0.764	2	6	0.935	0.476	0.044	2.935	4.139	0.661	2	6
Obj17	51.6	0.631	0.382	0.031	1.705	3.154	0.778	2	6	0.684	0.357	0.035	2.124	4.171	0.430	3	6
Obj18	120.4	1.607	0.575	0.272	3.098	3.387	0.785	2	6	1.322	0.660	0.085	4.635	3.914	0.567	2	6
Obj19	126.3	1.094	0.594	0.029	2.656	3.633	0.674	2	6	1.424	0.814	0.104	4.541	3.965	0.499	2	6
Obj20	150.5	1.263	0.489	0.250	3.443	3.802	0.672	2	6	1.236	0.555	0.062	3.215	4.292	0.531	2	6

total number of pushes needed to estimate the CoM location while maintaining similar accuracy.

V. CONCLUSIONS

Robotic manipulator arms have been in use for a variety of tasks mostly in the automation of repetitive tasks in industry settings. Lately, rather than a single/special purpose robot manipulator, enhanced multiple manipulation capabilities have been demanded for flexible robot-led automation. Endowing robots with such capabilities requires more intensive sensing and identification of the physical properties of objects being manipulated. The CoM is one of the most important physical properties of an object and in order to translate an object without rotating it, the force needs to be applied along the line passing as close as possible to the CoM location. In this paper, we presented two novel methods to estimate the CoM locations of a novel object using different actions and predicted motions of actions, which has proven to be an accurate and fast approach in realistic simulation settings. We will validate the effectiveness of the proposed algorithms in real robotic pusher-slider settings and further investigate the impact of accuracy of estimated CoM locations on various nonprehensile manipulation tasks.

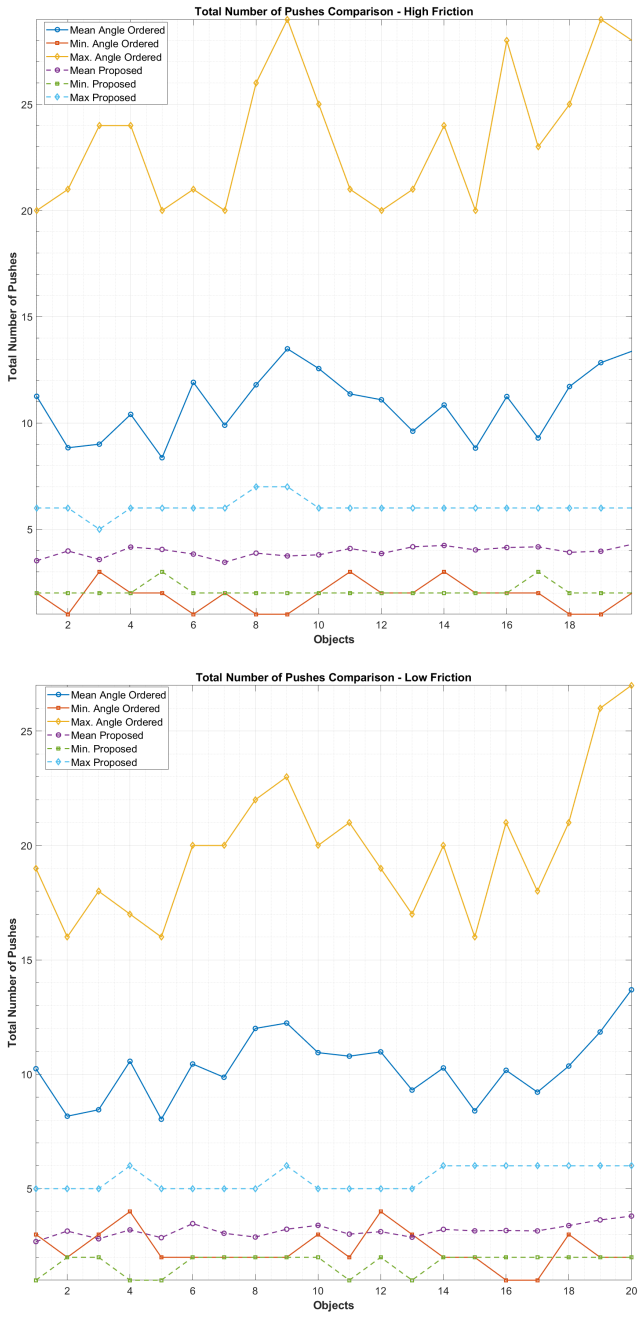

Fig. 7. Comparison of the total number of pushes. The mean, minimum and maximum number of pushes for test objects are represented as curves. While dashed lines report the results of using the Equality Score Function (named as 'proposed'), solid lines show the result without using the proposed function.

REFERENCES

[1] N. Mavrakis, A. M. Ghalamzan E., R. Stolkin, L. Baronti, M. Kopicki, and M. Castellani, "Analysis of the inertia and dynamics of grasped objects, for choosing optimal grasps to enable torque-efficient post-grasp manipulations," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 171–178, 2016.

[2] C. Song and A. Boularias, "A probabilistic model for planar sliding of objects with unknown material properties: Identification and robust planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5311–5318, 2020.

[3] J. K. Li, W. S. Lee, and D. Hsu, "Push-net: Deep planar pushing for objects with unknown physical properties," in *Robotics: Science and Systems*, 2018.

[4] Z. Gao, A. Elibol, and N. Y. Chong, "A 2-stage framework for learning to push unknown objects," in *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics*, pp. 1–7, 2020.

[5] Z. Gao, A. Elibol, and N. Y. Chong, "Planar pushing of unknown objects using a large-scale simulation dataset and few-shot learning," in *IEEE International Conference on Automation Science and Engineering*, pp. 341–347, 2021.

[6] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986.

[7] "Planar sliding with dry friction part 1. limit surface and moment function," *Wear*, vol. 143, no. 2, pp. 307–330, 1991.

[8] K. M. Lynch, H. Maekawa, and K. Tanie, "Manipulation and active sensing by pushing using tactile feedback.," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 416–421, 1992.

[9] J. Zhou, M. T. Mason, R. Paolini, and D. Bagnell, "A convex polynomial model for planar sliding mechanics: theory, application, and experimental validation," *The International Journal of Robotics Research*, vol. 37, no. 2-3, pp. 249–265, 2018.

[10] M. Bauza and A. Rodriguez, "A probabilistic data-driven model for planar pushing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3008–3015, 2017.

[11] N. Mavrakis and R. Stolkin, "Estimation and exploitation of objects ' inertial parameters in robotic grasping and manipulation: A survey," *Robotics and Autonomous Systems*, vol. 124, 2020.

[12] Y. Yu, T. Arima, and S. Tsujio, "Estimation of object inertia parameters on robot pushing operation," in *IEEE International Conference on Robotics and Automation*, pp. 1657–1662, 2005.

[13] N. Mavrakis, A. M. Ghalamzan E., and R. Stolkin, "Estimating an object's inertial parameters by robotic pushing: A data-driven approach," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 9537–9544, 2020.

[14] A. Kloss, M. Bauza, J. Wu, J. B. Tenenbaum, A. Rodriguez, and J. Bohg, "Accurate vision-based manipulation through contact reasoning," in *IEEE International Conference on Robotics and Automation*, pp. 6738–6744, 2020.

[15] K. N. Kumar, I. Essa, S. Ha, and C. K. Liu, "Estimating mass distribution of articulated objects using non-prehensile manipulation," *arXiv preprint arXiv:1907.03964*, 2019.

[16] H. Kim, A. Mnih, J. Schwarz, M. Garnelo, A. Eslami, D. Rosenbaum, O. Vinyals, and Y. W. Teh, "Attentive neural processes," *arXiv preprint arXiv:1901.05761*, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *International Conference on Neural Information Processing Systems*, (Red Hook, NY, USA), p. 6000–6010, Curran Associates Inc., 2017.