<table>
<thead>
<tr>
<th>Title</th>
<th>Estimating the Center of Mass of an Unknown Object for Nonprehensile Manipulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Gao, Ziyan; Elibol, Armagan; Nak-Young, Chong</td>
</tr>
<tr>
<td>Citation</td>
<td>2022 IEEE International Conference on Mechatronics and Automation (ICMA): 1755-1760</td>
</tr>
<tr>
<td>Issue Date</td>
<td>2022-08</td>
</tr>
<tr>
<td>Type</td>
<td>Conference Paper</td>
</tr>
<tr>
<td>Text version</td>
<td>author</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://hdl.handle.net/10119/18153">http://hdl.handle.net/10119/18153</a></td>
</tr>
</tbody>
</table>

This is the author's version of the work. Copyright (C)2022 IEEE. 2022 IEEE International Conference on Mechatronics and Automation (ICMA), 2022, pp.1755-1760. DOI:10.1109/ICMA54519.2022.9855972. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

| Date of Conference | 07-10 August 2022 |
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’s 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
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 $O(n^3)$ 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 rotation (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 $n$, if $R_P \times n$ 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...
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 rotation generated by this action. This way of the computed rotation, 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.

Algorithm 1: CoM Region Detection Process

```
1 S_θ(x,y) ← 0
2 foreach action a in A_{keep} do
3   l ← Compute the pushing line using a
4   foreach point p in P do
5     d ← Compute distance between p and l
6     S_θ(p) ← S_θ(p) + e^{-d} \cdot |θ| where θ is computed in Alg. 1
7   S_θ ← normalize S_θ to the interval [0,1]
8 P_c ← S_θ < s_θ /* Create a segmented region of object mask as P_c */
9 CoM ← Compute mean x and y of points in P_c
```

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:

\[ h(n_1, n_2) = \frac{2 \sqrt{n_1 n_2}}{n_1 + n_2} \] (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 \( S_θ \), 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.
Algorithm 2: Progressive CoM Region Detection Algorithm

**Input:** Object Mask \( P = (x_i,y_i) \quad i = 1,2,\cdots,m \),
List of actions \( A = (sx_i, sy_i, ex_i, ey_i) \) and predicted motion \( (\Theta_i, tx_i, ty_i) \quad i = 1,2,\cdots,n \),
Line-to-point distance threshold \( d_t \)
Area ratio \( s_t \in [0,1] \), \( t_\theta \) small rotation threshold, and \( t_h \) action score threshold

\[
/* \text{We used } s_t = 0.1, \quad d_t = 0.5 \text{pix}. \quad */
/* \text{We used } t_\theta = 5^\circ, \quad and \quad t_h = 0.6 \quad */

**Output:** Predicted location of CoM, \( \text{CoM} = (x, y) \)

1. \( A_{\text{keep}} \leftarrow \text{Remove actions } |\Theta_i| < t_\theta \)
2. \( P_c \leftarrow P \)
3. while size of \( P_c \) > \( s_t \times \text{size of } P \) or \( A_{\text{keep}} \neq \emptyset \) do
4.     foreach action \( a \) in \( A_{\text{keep}} \) do
5.         Compute the number of points \((n_1,n_2)\) in two subregions of \( P_c \) divided by the action \( a \) line
6.         Compute Equality Score, \( h_a \) via the function defined in Eq. 1.
7.     Choose action \( a_s \) in \( A_{\text{keep}} \) with the highest \( h \) score dividing \( P_c \) approximately equal two pieces \( P_{c1} \) and \( P_{c2} \).
8.     \( P_c \leftarrow P_{c1} \) or \( P_{c2} \) accordingly to the predicted rotation \( \Theta_a \) following VT Theory
9.     \( A_{\text{keep}} \leftarrow \text{Remove actions } h_a < t_h \) and \( a_s \)
10. \( P_c \leftarrow \text{The largest connected region in } P_c \)
11. \( \text{CoM} \leftarrow \text{Compute mean } x \text{ and } y \text{ of points in } P_c \)

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
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