JAIST Repository
https://dspace.jaist.ac.jp/

Title	Estimating the Center of Mass of an Unknown Object via Dynamic Pushing
Author(s)	Gao, Ziyan; Elibol, Armagan; Chong, Nak Young
Citation	IEEE 18th International Conference on Automation Science and Engineering (IEEE CASE2022)
Issue Date	2022-08-21
Type	Conference Paper
Text version	author
URL	http://hdl.handle.net/10119/18455
Rights	Copyright (c) 2022 Authors. Ziyan Gao, Armagan Elibol, Nak Young Chong, IEEE 18th International Conference on Automation Science and Engineering (IEEE CASE2022), 2022.
Description	IEEE 18th International Conference on Automation Science and Engineering (IEEE CASE2022), Mexico City, August 20-24, 2022

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

Estimating the Center of Mass of an Unknown Object via Dynamic Pushing

Ziyan Gao, Armagan Elibol, and Nak Young Chong

***Abstract*— An object's inertial parameters, such as the mass, the center of mass (CoM), and the moment of inertia, affect the response to the external forces exerted on it. It is important to estimate these parameters in order to facilitate robot-led automation including grasping and manipulation. Traditionally, the estimation is conducted by a specific hardware in a controlled environment, which may not be always available for a robotic system. We propose an efficient novel method for estimating the CoM of an object via force sensor-less dynamic pushing and a vision sensor detecting the change in the object's pose. The simulation results showed that the proposed method achieved an accurate and stable estimation under both the unknown isotropic and anisotropic floor friction conditions.**

I. INTRODUCTION

In robotic grasping and manipulation, it is widely assumed that an object's inertial parameters are *a priori* known. The parameters can be estimated using a special hardware in a controlled environment, which may not be suitable for commonly used industrial robots. In this work, we aim to estimate the CoM of a novel object via planar pushing using only a position-controlled robot arm and a vision sensor available commercially off-the-shelf. Following the taxonomy in [1] the estimation methods for inertial parameters can be categorized into three categories: purely visual, exploratory, and fixed-object. The purely visual category employs vision sensors to measure the volume of an object, assuming the object density. The exploratory category requires the robot to interact with the object to measure the object motion and applied force, then estimate the parameters by solving physical laws. The fixed-object category requires the object firmly attached to the robot's end-effector.

Our proposed method with planar pushing falls into the exploratory category. A close work was done by Yu *et al.* [2], utilizing a two-fingered robot arm equipped with a force sensor to slide a rectangular object across a isotropic floor. Different from [2], our method neither assumes a isotropic floor nor utilizes the force sensor to measure the applied force. Xu *et al.* [3] showed that pushing the object at high speed (referred to as dynamic pushing) makes the object's physical properties more distinguishable than pushing at low speed (referred to as quasi-static pushing). In quasi-static pushing interaction, the inertia term can be ignored, and the object motion are equally affected by the applied wrenches of both the pusher and the floor. In dynamic interaction, the wrench of the pusher is dominant compared to the floor frictional force. We propose to use dynamic pushing to interact with the object.

All authors are with the School of Information Science, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 923-1292, Japan {s1920013, aelibol, nakyoung}@jaist.ac.jp

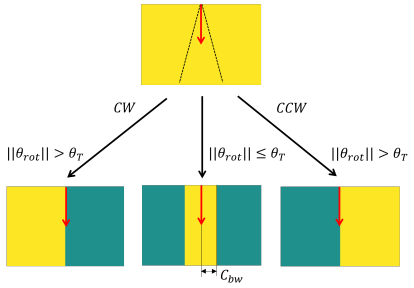

Fig. 1. Estimating the CoM region: The red arrow in the direction of the contact normal represents the pushing action. The black dash lines delimit the friction cone. CW and CCW refer to clockwise and counter-clockwise rotations, respectively. The yellow regions represent the CoM region and the dark green regions do the non-CoM region.

In this work, we claim that dynamic pushing can improve the accuracy and robustness of the CoM estimation of an unknown object to unknown inertial and friction parameters. Specifically, we propose a novel CoM estimation method using only a position-controlled robot arm and a vision sensor. We conduct a series of simulation experiments to compare the impact of pushing speed and (an)isotropic flooring surface frictions on the estimation accuracy.

II. METHODS

Following the right-hand rule for rotation, z-axis points perpendicular to the plane of the paper. Here, counter-clockwise rotations are considered positive. We refer to a region that contains the CoM as the CoM region, and otherwise the non-CoM region. We model the CoM estimation as a progressive decision process based on the Voting Theorem (VT) by Mason [4]. After a few interactions, the CoM location can be narrowed down to a small region whose centroid will be considered the most probable CoM estimate.

VT states that three rays, R_L and R_R representing the left and right boundaries of the friction cone as well as R_P representing the pushing direction at the contact point, vote on the sense of rotation. The vote is conducted by examining which sign of moment (positive or negative) the ray has to the CoM of an object. If two or more rays vote on positive rotation, then the object will rotate positively. Based on VT, for a known object, the CoM region can be narrowed down by applying a series of pushes. For each push and the corresponding object rotation, a CoM region without any ambiguity can be found by using the ray in the middle as the boundary between the CoM region and the non-CoM region. For unknown objects, however, this method cannot be applied, as R_L and R_R at the contact point are unknown. Therefore, R_p cannot be determined.

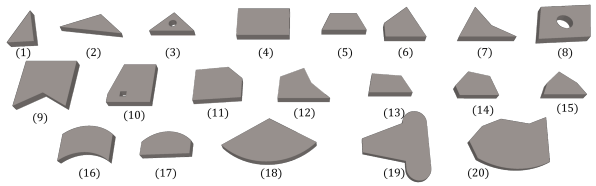

Fig. 2. Simulation objects for CoM estimation.

Our main idea is to constrain the pushing direction to the contact normal to ensure that R_P lies in the middle of the two boundaries of the friction cone. By doing so, the CoM region can be uniquely determined by the spatial relationship of pushing direction as well as the CoM. In practice, there are some circumstances that the object rotates so small that this amount cannot be easily detected by a vision sensor due to sensor limitations (*e.g.*, resolution or distance between the object and the sensor). Fortunately, in such cases, the distance between CoM and R_P is small. Therefore, we intuitively set a threshold θ_T for object rotation and a confidence bandwidth C_{bw} for possible CoM region selection. If the resultant object rotation is less than the threshold, the region whose inner pixel locations to R_P is less than the selected confidence bandwidth is regarded as the CoM region. Fig. 1 illustrates the selection rules for the proposed method.

The pseudocode of the proposed method is illustrated in **Algorithm 1** and **Algorithm 2**. The CoM region is initialized in **Algorithm 1** with the region inside the convex hull of the object mask. For each push, we use Principal Component Analysis of the current CoM region to find its centroid $\mathbf{c}$ and principal components $\mathbf{V}$. Then, we check if the corresponding line of each pushing action passes through the CoM region or not.

We compute the distance vector $\mathbf{d}$ between $\mathbf{c}$ and each line of pushing specified by the contact normal $\mathbf{N}_{ct}$. We then use a linear cost function to score each pushing action given by

$$\mathbf{s} = \mathbf{w}^\top \begin{pmatrix} \mathbf{d}^\top \\ (\mathbf{N}_{ct}\mathbf{V}_1)^\top \end{pmatrix}, \tag{1}$$

where $\mathbf{w}$ is a two-dimensional weight vector and $\mathbf{V}_1$ is the main principal vector. As both $\mathbf{N}_{ct}$ and $\mathbf{V}_1$ are normalized, $(\mathbf{N}_{ct}\mathbf{V}_1)^\top$ represents the cosine similarity between $\mathbf{N}_{ct}$ and $\mathbf{V}_1$. We aim to find the pushing action that has a small cosine similarity to $\mathbf{V}_1$ to avoid a prolate-shaped CoM region. Using this cost function, the pushing action can be selected that has a close distance to $\mathbf{c}$ and small cosine similarity with $\mathbf{V}_1$. Following resultant object rotation, **Algorithm 2** updates the CoM region.

Algorithm 1: CoM Region Decision Process

Input: $agent, \mathbf{P}_{ch}, \mathbf{P}_{ct}, \mathbf{N}_{ct}, \mathbf{w}, \theta_T, C_{bw}, n_{push}$

/* $\mathbf{P}_{ch}$, which consists of a set of pixel locations, represents the region inside the convex hull of the object. $\mathbf{P}_{ct}$, $\mathbf{N}_{ct}$ are the sampled contour points and normal directions associated. $\mathbf{P}_{ch}$, $\mathbf{P}_{ct}$, $\mathbf{N}_{ct}$ are all $(*\times 2)$ matrices, where $*$ is the number of sampled contour points or the size of region inside the convex hull of the object. n_{push} is the number of pushes that we set. */

Output: $\mathbf{P}_{CoM}$

/* $\mathbf{P}_{CoM}$ is probable CoM Region which consists of a set of pixel coordinates. */

$\mathbf{P}_{CoM} \leftarrow \mathbf{P}_{ch}$ // probable CoM region initialization

for $i = 1$ *to* n_{push} **do**

  /* opencv library */

  $\mathbf{c}, \mathbf{V}, \mathbf{x} \leftarrow PCA(\mathbf{P}_{CoM})$

  /* shapely library */

  $\mathbf{P}_{ct}, \mathbf{N}_{ct} \leftarrow valid(\mathbf{P}_{ct}, \mathbf{N}_{ct}, \mathbf{P}_{CoM})$

  /* calculate distances between $\mathbf{c}$ to the lines specified by $\mathbf{P}_{ct}$ and $\mathbf{N}_{ct}$ */

  $\mathbf{d} \leftarrow \frac{\mathbf{c}-\mathbf{P}_{ct}}{\|\mathbf{c}-\mathbf{P}_{ct}\|} \times \mathbf{N}_{ct}$

  /* Calculate scores for each contacts. */

  $\mathbf{s} = \mathbf{w}^\top \begin{pmatrix} \mathbf{d}^\top \\ (\mathbf{N}_{ct}\mathbf{V}_1)^\top \end{pmatrix}$

  j = argmin $\mathbf{s}$

  $\theta_{rot} \leftarrow agent.execute(\mathbf{P}_{ct_j}, \mathbf{N}_{ct_j})$

  $\mathbf{P}_{CoM} \leftarrow$UPDATE($\mathbf{P}_{CoM}$, $\mathbf{P}_{ct_j}$,$\mathbf{N}_{ct_j}$, θ_{rot}, c_{bw},θ_T)

III. SIMULATION EXPERIMENTS

eWe use CoppeliaSim and Vortex physics engine for the simulation experiment under the conditions detailed in Table I. We set two different friction coefficients for the floor and pusher, respectively, leading to 4 different combinations of frictional conditions. We set the slider friction coefficient to 0.5. Thre are 20 objects with different sizes and shapes as shown in Fig. 2. For each object, we assign 10 different CoM locations randomly. We set the mass of all objects to $0.2kg$. To examine the impact of the mass moment of inertia of an object about the z-axis, denoted by I_{zz}, we scale the default value of I_zz by a factor of 1 and 10. We use two different pushing speeds: one $(40cm/s)$ is considered dynamic pushing, and the other $(4cm/s)$ quasi-static pushing. In each run of CoM estimation, we execute 5 pushes, resulting in $6,400$ CoM estimations in total. Furthermore, to evaluate the performance on the anisotropic floor, we create a surface plane with two different frictional coefficients (0.1 and 1, respectively) along two principal axes, and set the pusher friction coefficient to 0.5.

The object mask is obtained by a depth camera of 224×224 pixels aligned to perpendicular to the floor. The floor is a flat square with an area of $0.36m^2$. We obtain the surface normal vector at the contact point from the simulator having inherent errors due to the scale of the surface triangulation. For the CoM estimation of an object, we follow **Algorithm 1** and set $\mathbf{w}$ to $[1, 0.5]^\top$. We set rotation threshold θ_T and confidence bandwidth C_{bw} to 1° and 10 pixels. The pushing length is set to $3cm$.

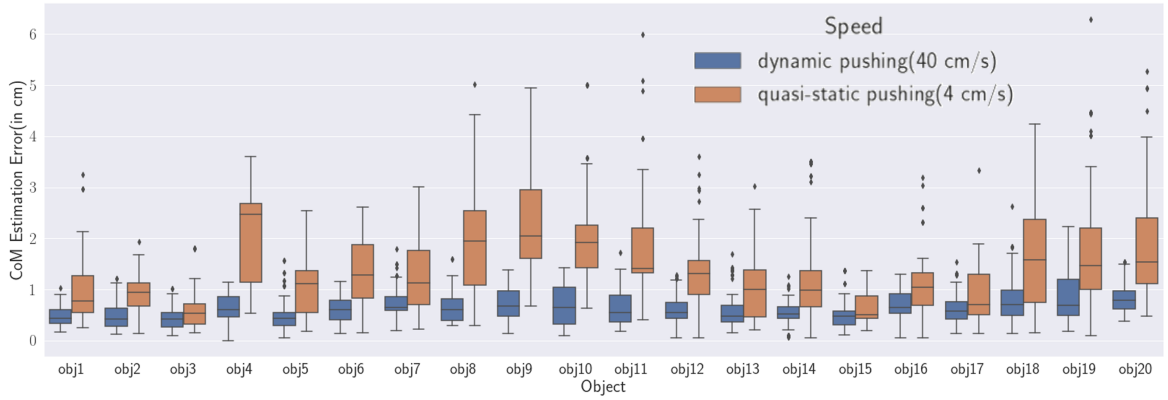

Fig. 3. Experimental results on isotropic frictional floor.

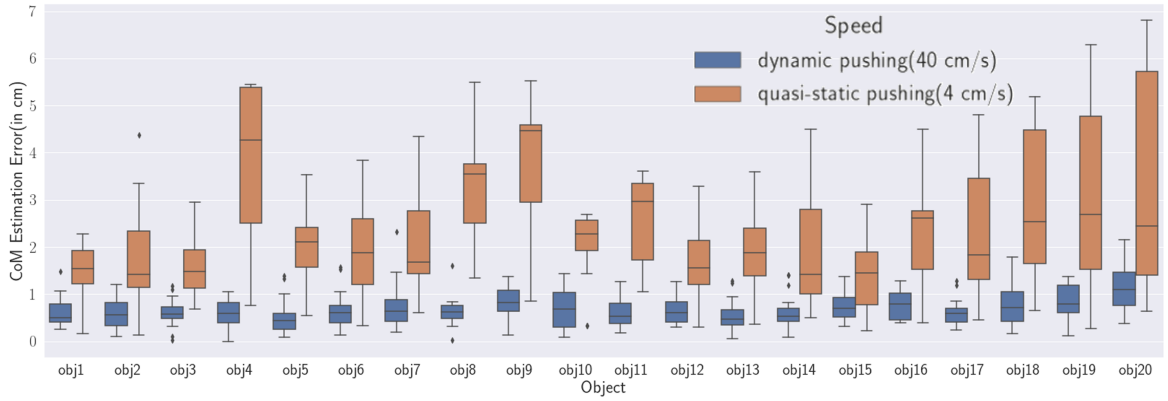

Fig. 4. Experimental results on anisotropic frictional floor.

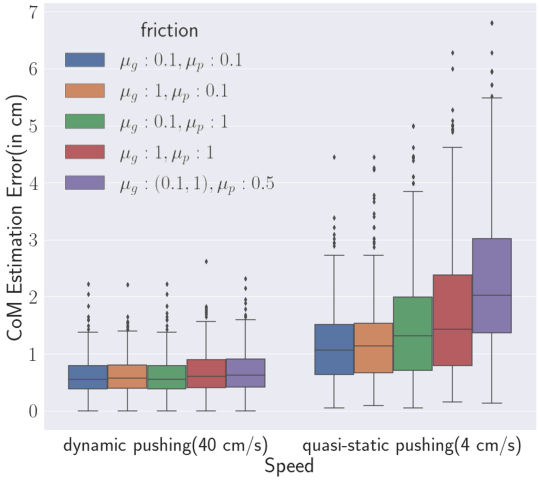

Fig. 5. Robustness comparison between dynamic and quasi-static pushing.

The simulation results of CoM estimation on isotropic frictional floor are detailed in Fig. 3 and Table III. The result shows that the CoM estimation error with dynamic pushing is smaller than quasi-static pushing. We observed one notable case with quasi-static pushing that the estimation error tends to be large when an object has a large I_{zz}, or pushed across a high frictional floor, which results in a small amount of rotation misleading CoM region selection. On the other hand, in the case of dynamic pushing, as the wrench exerted by the floor on the object no longer dominates the object motion, the object motion follows the VT and the resultant rotation is larger than that of quasi-static pushing.

The simulation result of CoM estimation on anisotropic frictional floor is shown in Fig. 4 and Table III. It can be observed that dynamic pushing still exhibits good performance even on the anistropic frictional floor. Compared with the result on the isotropic frictional floor, the estimation error does not increase much. However, the estimation error with quasi-static pushing increases significantly. Fig. 5 shows the influence of friction on estimation error with dynamic and quasi-static pushing, respectively. Encouragingly, compared with the quasi-static pushing case, friction has a limited influence on estimation accuracy with dynamic pushing. The anisotropic frictional setting affects estimation accuracy small with dynamic pushing but deteriorates estimation accuracy significantly with quasi-static pushing.

IV. CONCLUSION

In this work, we proposed an efficient method for predicting the center of mass of an unknown object. Unlike other works, our method requires only a position-controlled robot arm and a commercial off-the-shelf vision sensor. We showed that dynamic pushing can improve the accuracy and

Algorithm 2: Probable CoM Region Update

Input: $\mathbf{P}_{CoM}$, $\mathbf{P}_{ct_j}$,$\mathbf{N}_{ct_j}$, θ_{rot}, c_{bw}, θ_T
Output: $\mathbf{P}_{CoM}$

Function `Update` ($\mathbf{P}_{CoM}$, $\mathbf{P}_{ct_j}$,$\mathbf{N}_{ct_j}$, θ_{rot}, c_{bw}, θ_T) **:**
  if $\|\theta_{rot}\| > \theta_T$ **then**
    if $sign(\theta_{rot} > 0)$ **then**
      for $\mathbf{P}_{CoM_i}$ *in* $\mathbf{P}_{CoM}$ **do**
        if $\mathbf{N}_{ct_j} \times (\mathbf{P}_{CoM_i} - \mathbf{P}_{ct_j}) \leq 0$ **then**
          Delete($\mathbf{P}_{CoM}$, $\mathbf{P}_{CoM_i}$)
      return $\mathbf{P}_{CoM}$
    else
      for $\mathbf{P}_{CoM_i}$ *in* $\mathbf{P}_{CoM}$ **do**
        if $\mathbf{N}_{ct_j} \times (\mathbf{P}_{CoM_i} - \mathbf{P}_{ct_j}) \geq 0$ **then**
          Delete($\mathbf{P}_{CoM}$, $\mathbf{P}_{CoM_i}$)
      return $\mathbf{P}_{CoM}$
  else
    for $\mathbf{P}_{CoM_i}$ *in* $\mathbf{P}_{CoM}$ **do**
      $d \leftarrow (\mathbf{P}_{CoM_i} - \mathbf{P}_{ct_j}) \times \mathbf{N}_{ct_j}$
      if $d \geq C_{bw}$ **then**
        Delete($\mathbf{P}_{CoM}$, $\mathbf{P}_{CoM_i}$)
    return $\mathbf{P}_{CoM}$

TABLE I
SIMULATION EXPERIMENTAL SETTING

flooring surface friction coefficient μ_g	0.1, 1
pusher friction coefficient μ_p	0.1, 1
slider friction coefficient	0.5
pushing speed	40 cm/s, 4 cm/s
number of objects	20
number of CoM locations per object	10
mass	0.2 kg
scale factor for I_{zz}	1, 10
pushing speed	40 cm/s, 4 cm/s
maximum number of pushes per object	5

TABLE II
RESULTS OF COM ESTIMATION ON ISOTROPIC FLOOR

Object	High Speed Pushing				Low Speed Pushing			
	mean	std	max	min	mean	std	max	min
obj1	0.489	0.206	1.034	0.166	0.949	0.657	3.251	0.255
obj2	0.476	0.278	1.208	0.123	0.915	0.364	1.932	0.137
obj3	0.423	0.205	1.008	0.091	0.591	0.362	1.804	0.15
obj4	0.631	0.29	1.136	0.0	2.06	0.844	3.604	0.536
obj5	0.481	0.351	1.568	0.05	1.071	0.553	2.537	0.176
obj6	0.605	0.274	1.154	0.134	1.297	0.599	2.611	0.149
obj7	0.713	0.315	1.789	0.196	1.3	0.673	3.015	0.217
obj8	0.651	0.303	1.597	0.299	1.936	1.174	5.018	0.299
obj9	0.742	0.32	1.379	0.134	2.2	1.177	4.939	0.67
obj10	0.66	0.389	1.43	0.092	2.067	0.963	4.999	0.625
obj11	0.619	0.342	1.723	0.174	1.805	1.031	5.999	0.402
obj12	0.619	0.291	1.285	0.054	1.423	0.7	3.599	0.054
obj13	0.598	0.365	1.695	0.153	1.004	0.602	3.027	0.203
obj14	0.53	0.258	1.252	0.064	1.253	0.901	3.502	0.052
obj15	0.521	0.28	1.367	0.11	0.678	0.356	1.368	0.187
obj16	0.689	0.274	1.292	0.054	1.103	0.603	3.187	0.054
obj17	0.645	0.309	1.533	0.136	0.919	0.567	3.332	0.136
obj18	0.8	0.472	2.624	0.143	1.735	1.132	4.238	0.155
obj19	0.87	0.532	2.227	0.176	1.811	1.17	6.288	0.088
obj20	0.812	0.282	1.542	0.377	1.862	1.079	5.276	0.476

TABLE III
RESULTS OF COM ESTIMATION ON ANISOTROPIC FLOOR

	High Speed Pushing				Low Speed Pushing			
	mean	std	max	min	mean	std	max	min
obj1	0.597	0.296	1.478	0.255	1.408	0.667	2.283	0.166
obj2	0.595	0.351	1.208	0.097	1.744	1.014	4.382	0.137
obj3	0.607	0.284	1.173	0.019	1.602	0.644	2.953	0.688
obj4	0.598	0.295	1.046	0.0	3.651	1.836	5.439	0.758
obj5	0.51	0.365	1.384	0.086	2.039	0.818	3.536	0.552
obj6	0.649	0.38	1.576	0.134	2.006	0.93	3.835	0.34
obj7	0.742	0.483	2.323	0.196	2.098	1.115	4.347	0.615
obj8	0.631	0.303	1.597	0.027	3.22	1.184	5.488	1.349
obj9	0.832	0.324	1.379	0.134	3.789	1.413	5.517	0.858
obj10	0.667	0.407	1.43	0.092	2.073	0.676	2.697	0.334
obj11	0.606	0.32	1.266	0.174	2.513	0.937	3.608	1.058
obj12	0.655	0.288	1.26	0.299	1.76	0.929	3.294	0.308
obj13	0.556	0.328	1.28	0.05	1.938	0.881	3.601	0.364
obj14	0.596	0.363	1.399	0.089	1.83	1.174	4.493	0.502
obj15	0.771	0.321	1.368	0.31	1.347	0.68	2.898	0.232
obj16	0.764	0.296	1.286	0.388	2.525	1.126	4.502	0.388
obj17	0.599	0.272	1.288	0.233	2.428	1.479	4.809	0.45
obj18	0.832	0.456	1.787	0.164	2.863	1.532	5.179	0.649
obj19	0.822	0.381	1.376	0.12	3.026	1.756	6.287	0.267
obj20	1.15	0.491	2.148	0.383	3.319	2.189	6.807	0.641

robustness of estimation, supported by a series of simulation experiments. As a result, the proposed CoM estimation method with dynamic pushing achieved impressive performance on CoM estimation for novel objects both in isotropic and anisopropic frictional floors. As we only evaluate the proposed method in simulation, one straightforward strategy to improve the estimation accuracy would be increasing the number of pushing steps or the image resolution. In the future, we will carry out experiments on a real robotic platform. In addition, we will attempt to exploit the CoM estimates toward facilitating robotic grasping or manipulation.

REFERENCES

[1] 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.

[2] 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.

[3] Z. Xu, J. Wu, A. Zeng, J. B. Tenenbaum, and S. Song, "Densephysnet: Learning dense physical object representations via multi-step dynamic interactions," in *Robotics: Science and Systems*, 2019.

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