Zero Moment Two Edge Pushing of Novel Objects with Center of Mass Estimation

Ziyan Gao, Armagan Elibol, and Nak Young Chong

***Abstract*—Pushing is one of the fundamental nonprehensile manipulation skills to impart to an object changes in position and orientation. To exploit this skill to manipulate novel objects, explicit knowledge of their physical properties should be given *a priori*. In this work, we estimate the center of mass (CoM) of an object by narrowing down its probable location with a deep learning model and Mason's voting theorem. In addition, we propose the Zero Moment Two Edge Pushing (ZMTEP) method to translate a novel object without rotation to a goal pose. The proposed method enables a pusher to select the most suitable two-edge-contact configuration for a given object using the estimated CoM and the geometrical shape of the object. Notably, neither the friction between the object and its support plane nor the friction between the object and the pusher are assumed to be known. We evaluate the proposed CoM estimation and ZMTEP methods through a series of experiments in both simulation and real robotic pusher settings. The result shows that the CoM estimation method has good mean squared error properties and small standard deviation, and the ZMTEP method significantly outperforms competitive baseline methods.**

***Note to Practitioners*—This article aims to endow robotic arms with the capability of moving or aligning objects by pushing, which is much more simple and secure than pick-and-place or in-hand manipulations. Most in-demand manipulation skills require sophisticated hand design and control, which might not be affordable for industrial applications staying cost-competitive. In contrast, robot pushing can be implemented with different types of simple pushers and straightforwardly applied to pre-grasp manipulation. This article makes the estimation of an object's CoM location practical. Building upon the estimation method, a robust and noise-tolerant two-edge-contact pushing configuration selection method is presented to translate an arbitrarily shaped unknown object to its goal pose.**

***Index Terms*—Planar Pushing, Object CoM Estimation, Two Edge Pushing, Data-driven Automation**

I. Introduction

Moving an object from one location to another is a common task encountered in our daily life and industrial production lines. This task can be accomplished through pick-and-place, pushing, throwing, and similar others. Among these, pushing is an undemanding manipulation primitive [1] to locate and orient an object to a desired pose. Pushing neither needs form or force closure nor requires conforming to the object's geometrical shapes. Pushing is preferable to re-positioning of an object not directly graspable within the given pose. Therefore, pushing can be used to complement other tasks such as object placement [2], object singulation [3], [4], and grasping [5]–[8].

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

The pusher-slider system with a single point contact exhibits multiple contact modes (sticking, sliding, and separation), indeterminacy due to unknown pressure distribution, and underactuation. All of these introduce challenges on modeling, controlling, and planning. The pusher-slider system becomes stable by introducing extra contacts. Lynch and Mason [9] showed that there is a set of pushing directions along which the slider's pose relative to the pusher remained fixed given known physical properties. They proposed line pushing with multiple contacts on the same edge of a polygonal slider and stabilized its motion [9]–[11].

Supposing that a robot pushes an object on a flat surface, and there is nearly zero friction between the robot and the object, line pushing becomes extremely difficult as there is little friction that can be exploited. It is therefore advantageous to use multi-edge contact whose normal force direction positively spans the desired pushing direction. A robot equipped with an adjustable stroke gripper can make multiple contacts that are not necessarily on the object's same edge. It should be also noted that when a pusher maintains line (or point) contact with a slider to move it around, non-holonomic constraints need to be addressed. Zhou *et al.* [11] showed that Dubin's curve is the time optimal trajectory under the assumption of sticking contact.

In this work, instead of solving the trajectory planning problem in a complex way, we pose a new question: how can we select two contact points from two polygonal edges of an object with unknown physical and frictional properties, respectively, to slide it along a straight line to a goal pose? As the shortest distance between two points is a straight line, a solution to this problem yields a reasonably close optimal shortest path. There are two main issues to be addressed in this problem; first, the object's physical properties, such as the CoM and friction coefficient need to be identified. Secondly, the contact points are properly selected given noisy observations and error-prone estimates of physical properties. The following assumptions are made throughout this paper:

- Both the robot's end effector (pusher) and the object (slider) are rigid, and interact quasi-statically.
- Object geometry can be extracted by a vision sensor.
- The plane in which the object lies is flat and uniform.
- Coulomb's friction law applies.
- The pusher maintains contact with the slider during pushing.

In [12], we proposed a few shot learning model trained on our large-scale simulation dataset called **SimPush** that contains more than two million labeled pushing examples.

SimPush has proven useful in training a learning model and predicting object motions in real robot settings. We narrow down the probable location of the CoM of an object by combining our few shot learning and Mason's voting theorem [13]. The proposed method uses the spatial relationship between the pushing action and the predicted object rotation. Then, analyzing the force and moment conditions of pure translation, and the moment labeling representation of the contact force, we propose a double-contact pusher-slider interaction, called Zero Moment Two Edge Pushing (ZMTEP), to translate a novel object to a goal pose.

In real-world pushing, measurement noise and CoM estimation error are unavoidable, leading to an incorrect contact configuration. Therefore, we discuss the tolerance of ZMTEP to noisy observation and error-prone CoM estimates. We found that the tolerance to the CoM estimation error increases, as the distance between two contact positions increases. The tolerance to noisy observations increases up to a certain point and decreases as the contact point moves toward the endpoint of the object edge. A series of experiments are conducted to verify if the proposed CoM estimation and ZMTEP methods are valid and competitive compared to two baseline methods in simulation and real settings. The results revealed that the CoM estimation error was low overall that can be well tolerated by the two-edge-contact configuration selected by ZMTEP.

To summarize, the main contributions of this work are threefold:

- An integrated CoM estimation method combining the motion prediction model and the voting theorem.
- The most suitable two-edge-contact pushing configuration for translating a novel object without rotation.
- An analysis of tolerance to measurement noise and error-prone CoM estimates.

This paper is organized as follows. In Section II, we review the related work in object inertial parameter estimation and planar pushing. In Section III, we present our CoM estimation method. In Section IV, we conduct force and moment analysis and moment labeling representation for the two-edge-contact configuration with its tolerance capability. We then present the ZMTEP method for pure translation. From Section V to Section VII, we provide a detailed description of implementation and data we gathered in the CoM estimation and object translation experiments. Finally, Section VIII draws conclusions.

II. Related Work

A. Object Inertial Parameter Estimation

An object's inertial parameters can be identified by various techniques classified into three types: purely visual, exploratory, and fixed-object [14]. The fixed-object type requires a fixed connection between the robot and an object such as grasping, which is beyond the scope of this research.

The purely visual type directly estimates the inertial property using a vision sensor. Trevor *et al.* [15] proposed a deep learning model trained on a labeled dataset to estimate the volume and density of the object using a RGB-D image. Although this model has competitive performance on image-to-mass tasks compared with humans, the estimation result is highly affected by occlusion or light conditions.

The exploratory type requires the robot to interact with the object to measure the applied forces and object motion. Then, the inertial parameters are estimated by solving physics laws of motion. Yu *et al.* [16] used a two-finger pushing in a trial-and-error approach making the line of CoM pass between the fingers. This method may be inefficient when dealing with an arbitrarily shaped object. Mavrakis *et al.* [17] estimated inertial properties using a single-contact pushing and data-driven learning model. Kloss *et al.* [18] used an Extended Kalman Filter (EKF) to iteratively estimate object physical properties (such as the CoM, friction, and mass) based on the information extracted from the object mask. The EKF was implemented assuming an ellipsoid model [19] between the applied force and the resultant object motion. Song *et al.* [20] proposed to learn the coupled mass-friction parameters through minimizing the simulation-reality gap. This method required a set of hypothesized mass and friction models and the object was coarsely approximated by rigidly-connected 2D small grids. Allevato *et al.* [21], [22] used a neural network to tune the inertial parameters of the physics engine based on the difference in observation from the real object motion. However, it was limited to known objects. Instead of relying on an approximation model or physical simulator, data-driven models were used to learn the causality between the object motion and the inertial parameters in some studies. Li *et al.* [23] used recurrent neural networks to estimate the CoM of the object. Xu *et al.* [24] proposed a disentangled learning module to implicitly encode an object's physical properties through robot-object interaction. Kumar *et al.* [25] employed a policy network to interact with an articulated object, and a predictor network to predict the mass distribution of the object.

In this work, we propose an exploratory type CoM estimation method for an arbitrarily shaped object using a robot equipped with a vision sensor. In contrast to other exploratory type methods, only a few robot-object pushes are needed with no *a priori* assumption about friction.

B. Planar Pushing

Mason [13] analyzed the mechanics of quasi-static pushing and presented the voting theorem for determining the sense of rotation of a pushed object. Goyal *et al.* [26], [27] introduced the concept of limit surface to describe the relationship between the applied wrench and the object twist by assuming minimum energy dissipation. Lynch *et al.* [19] proposed the ellipsoidal approximation to the actual limit surface [9], [10], [19]. Zhou *et al.* [11] presented a physics-based data-driven model to approximate the limit surface. Kloss *et al.* [28] combined a deep learning model and an ellipsoid limit surface to improve both the generalization capability and accuracy. Lin *et al.* [29] integrated a recurrent neural network for pushing interaction modeling into a model predictive control framework.

Without focusing on the pusher-slider interaction models, some studies used data-driven methods to learn from historical

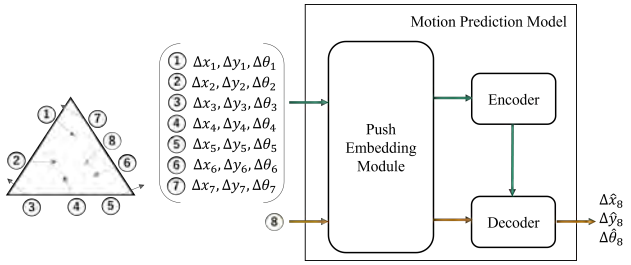

Fig. 1. Schematic of object motion inference using motion prediction model.

pushing interactions [30]–[32]. Hermans *et al.* [30] presented a kernel's method to learn the contact locations for pushing an unknown object. However, only the shape features were considered with a state feedback controller. In contrast, our ZMTEP method selects contact locations considering both the geometrical shape as well as the CoM of an object. Li and Payandeh [33] proposed a parametric formulation to find a two-point contact configuration of equilibrium push in which relative rotation between the pusher and the object will not happen employing a simplified model of a known object. Most notably, based on fewer assumptions, our method determines the most suitable two-edge-contact configuration, analyzing the mechanics of pure translation that exhibits high tolerance toward measurement and CoM estimation noises.

On the other hand, instead of relying on push interaction models, Lloyd and Lepora [34] employed tactile and proprioceptive feedback to push an object across planar and curved surfaces. Danielczuk *et al.* [6] presented and compared several pushing policies for object singulation in cluttered bin environments which did not consider pushing interactions.

III. CoM Estimation

We introduce a new CoM estimation method, combining our few-shot learning motion prediction model [12] and Mason's voting theorem (VT). Our learning model predicts the sense of rotation of a pushed object. The probable CoM region is selected through a decision process until it is narrowed down to the region whose centroid is considered the CoM of the object. In the following part, we review our few shot learning model and introduce the process of our CoM estimation method.

A. Prediction Model

Basically, the prediction model predicts object motion for a pushing primitive by integrating limited pushing priors based on the Residual Convolutional Network (ResNet) [35] and Attentive Neural Process (ANP) [36]. The ResNet is referred to as pushing embedding module. It takes the object mask and pushing action (and the resultant motion if available) as input and outputs a high-dimensional feature to represent the input. ANP with an encoder-decoder structure predicts the resultant motion for test data by integrating pushing priors. The encoders takes the pushing priors as input and outputs a representative features for each pushing prior and a permutation-invariant representation. The decoder incorporates the output of encoder to predict the corresponded object motion for test pushing data. The specific inference procedure is as follows:

- Collect m pushing priors $\{\mathbf{cp}_i, \mathbf{n}_i, \theta_{p_i}, \Delta O_i\}_{i=1,\dots,m}$, where $\mathbf{cp}_i$ is the contact location, $\mathbf{n}_i$ is the surface normal at $\mathbf{cp}_i$, θ_{p_i} is the pushing direction *w.r.t.* $\mathbf{n}_i$, and ΔO is the changes in object state represented by $\Delta x, \Delta y, \Delta\theta$.
- Feed the pushing priors to the pushing embedding model and encoder of few-shot learning model.
- Predict ΔO for test data.

An example is shown in Fig. 1, where a triangular object is subjected to pushing actions. There are seven pushes marked by the green arrows whose resultant object motions are known, and another push marked by the orange arrow whose resultant motion is unknown. Each push specifies a contact point $\mathbf{cp}$, a normal direction $\mathbf{n}$, and a push direction *w.r.t.* the normal direction θ_p. The pushes in green with its resultant motions form the pushing priors. The prediction model aims to predict the resultant motion for the push marked in the orange arrow by integrating the pushing priors. We denote the few-shot learning model by $\mathbf{F}$

$$\Delta\hat{x}, \Delta\hat{y}, \Delta\hat{\theta} = \mathbf{F}(\mathbf{cp}, \mathbf{n}, \theta_p, \mathbf{S}_{prior}) \tag{1}$$

where $\mathbf{S}_{prior}$ indicates pushing priors. As the few-shot learning model is used for the process of estimating the CoM, only $\Delta\hat{\theta}$ is used. Therefore, $\Delta\hat{x}$ and $\Delta\hat{y}$ are ignored in this work.

B. VT for CoM Estimation

VT relates the sense of rotation of a pushed object with the spatial relationship between several rays and the CoM of the object. VT states that three rays, R_L, R_R delimiting the left and right boundaries of the friction cone at the contact point and R_P indicating the line of pushing, vote for the sense of rotation. For instance, if any two of the three rays have a negative moment about the CoM, the object would rotate clockwise regardless of the third ray generated moment.

The probable location of the CoM can be determined by analyzing the relationship between the rays and the resultant object rotation. A pushing example is illustrated in Fig. 2(a). The red arrow is the pushing direction specifying the ray R_P, when R_L and R_R are assumed to be known. If the object rotates clockwise, at least two rays have a negative moment about the CoM. Therefore, R_L and R_R must have a negative moment and R_P has an indeterminate moment about the CoM. Then, the probable CoM location can be determined by R_L and R_R. As R_L has negative moment to area two, three, and four, while R_R has negative moment to area three and four, area three and four are the areas to which both R_L and R_R have negative moment. Therefore, CoM must be located in area three and four. The VT cannot be used to locate the CoM of a novel object as R_L and R_R are unknown.

C. Region Selection Rules

We describe our method for narrowing the probable location of the CoM. Notably, a friction cone is not given at the contact point. The method determines the CoM location using the pushing direction and the sense of rotation of a pushed object. In our method, the sign of moment that R_P generates about the CoM is always determined. Therefore, the probable CoM

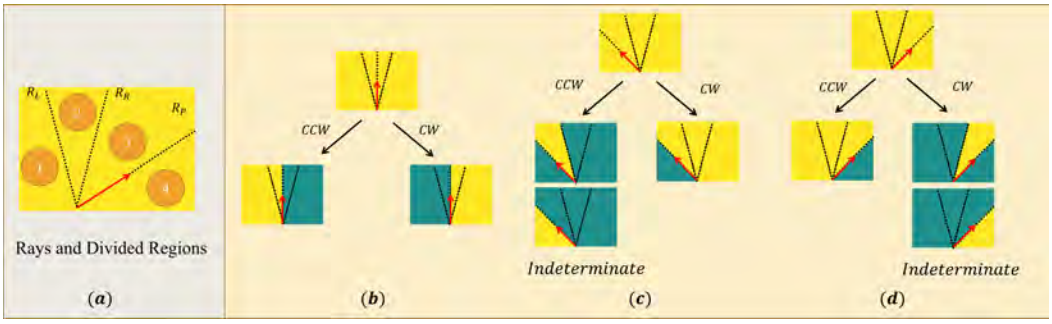

Fig. 2. Probable CoM location narrowing: Yellow region is considered the probable CoM location, while green region is excluded. Red arrow is the pushing direction. Black dash lines are friction cone limits at the contact point. Probable location is initialized to the convex hull of the object mask. When pushing direction is aligned with the contact normal, the location can be narrowed if the sense of rotation is known. When pushing direction is not aligned with the contact normal, there are two cases that we cannot narrow the location. In such cases, we use another push action to continue to narrow the location.

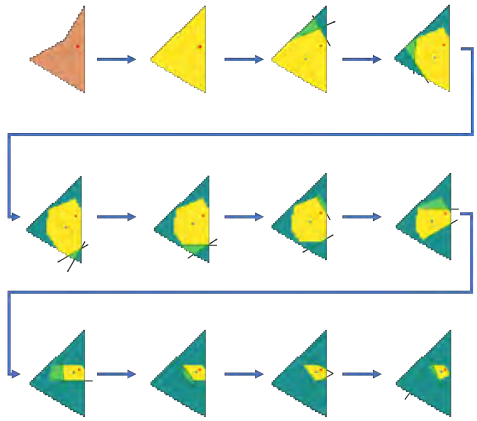

Fig. 3. CoM estimation process. The first image illustrates the object with ground truth CoM in red dot. The second image shows the convex hull of the object as the probable CoM location in yellow color. Other images show a sequence of narrowing locations until the size reaches a threshold or no pushing action helps narrow the location.

location can be narrowed using R_P and the resultant rotation. An illustrative example is shown in Fig. 2.

In order to convey the idea clearly, we consider the following two cases. In the first case, the pushing direction is aligned with the contact normal as shown in Fig. 2(b), where R_P is inside the friction cone. In this case, if the sense of rotation is known, then the probable CoM region can be determined. Let us assume that R_P is aligned with the contact normal, and the object rotates clockwise. If the CoM of the object is outside the friction cone, both R_L and R_R should have a negative sign of moment about the CoM, and R_P must have the same sign of moment with R_L and R_R as R_P is inside the friction cone. Then the regions where R_P has negative sign of moment can be regarded as the probable CoM regions. If the CoM is between R_L and R_R, R_L and R_P must have a negative moment about the CoM, as there should be at least two rays that have a negative sign of moment about the CoM. In this case, the regions where R_P has a negative moment can be considered the probable CoM region.

The second case is that the pushing direction is different from the contact normal as shown in Fig. 2(c) and (d). This case introduces ambiguity as the sense of rotation might be dominated by R_L or R_R. Let us consider the left side of Fig. 2(c), where the object rotates counter-clockwise, and R_P can either have a positive or negative moment about the CoM. Therefore, the probable CoM region cannot be inferred by R_P. The same applies to the right side of (d) in Fig. 2. On the other hand, for the right side of Fig. 2(c), the object rotates clockwise, and R_P and R_L must have a negative sign of moment about the CoM. In this case, the CoM region can be narrowed by R_P and so is the left side of (d) in Fig. 2.

Algorithm 1: CoM Region Decision Process

Input: $\mathbf{F}, \mathbf{S}_{prior}, \mathbf{A}_{CH}, \{\mathbf{cp}_i\}^m, \{\mathbf{n}_i\}^m, \{\theta_p\}^a, \theta_T, a_T$

/* $\mathbf{F}$ is the prediction model, $\mathbf{S}_{prior}$ is the pushing priors, $\mathbf{A}_{CH}$, which consists of a set of pixel locations, represents the region inside the convex hull of the object. $\{\theta_p\}^a$ is a set of angles w.r.t. normal direction, $\{\mathbf{cp}_i\}^m$, $\{\mathbf{n}_i\}^m$ are sampled contour point and normal direction associated. θ_T is a threshold for predicted rotation. a_T is a threshold of the area of probable CoM region. */

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

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

$\mathbf{A}_{CoM} \leftarrow \mathbf{A}_{CH}$ // probable CoM region initialization

$\mathbf{S}_{push} \leftarrow \emptyset$ // initialize a cache

for θ_p *in* $\{\theta_p\}^a$ **do**

  for $\mathbf{cp}_i$ *in* $\{\mathbf{cp}_i\}_{i=1,\dots m}$ **do**

    $\hat{\theta}_i = \mathbf{F}(\mathbf{cp}_i, \mathbf{n}_i, \theta_p, \mathbf{S}_{prior})$ // rotation prediction

    if $\|\hat{\theta}_i\| > \theta_T$ **then**

      Append($\mathbf{S}_{push}$, $\{\mathbf{cp}_i, \mathbf{n}_i, \theta_p, \hat{\theta}_i\}$)

Sort $\mathbf{S}_{push}$ based on the amount of rotation in descending order

for $\{\mathbf{cp}_i, \mathbf{n}_i, \theta_p, \hat{\theta}_i\}$ *in* $\mathbf{S}_{push}$ **do**

  $\mathbf{A}_{CoM} \leftarrow$UPDATE($\mathbf{A}_{CoM}, \mathbf{cp}_i, \mathbf{n}_i, \theta_p, \hat{\theta}_i, a_T$)

Algorithm 2: Probable CoM Region Update

Input: $\mathbf{A}_{COM}, \mathbf{cp}_i, \mathbf{n}_i, \theta_p, \hat{\theta}_i, a_T$
Output: $\mathbf{A}_{CoM1}$

Function `Update(`$\mathbf{A}_{CoM}$, **cp**, **n**, θ_p, $\hat{\theta}$, a_T`)`**:**
  $\mathbf{A}_{CoM1} \leftarrow \mathbf{A}_{CoM}$ `// new probable CoM region initialization`
  Construct $\mathbf{R}_P$ based on **cp**, **n** and θ_p.
  if $\mathbf{A}_{CoM1}.size() >= a_T$ **then**
    return $\mathbf{A}_{CoM1}$
  else
    if $\mathbf{R}_P$.*intersect*($\mathbf{A}_{CoM1}$) *is True* **then**
      if $sign(\hat{\theta}) > 0$ **then**
        if $\theta_p \geq 0$ **then**
          for $\mathbf{p}_{pix}$ *in* $\mathbf{A}_{CoM1}$ **do**
            `/* cross operation */`
            if $\mathbf{R}_P \times \mathbf{p}_{pix} \leq 0$ **then**
              Delete($\mathbf{A}_{CoM1}$, $\mathbf{p}_{pix}$)
          return $\mathbf{A}_{CoM1}$
        else
          return $\mathbf{A}_{CoM1}$
      else
        if $\theta_p \leq 0$ **then**
          for $\mathbf{p}_{pix}$ *in* $\mathbf{A}_{CoM1}$ **do**
            if $\mathbf{R}_P \times \mathbf{p}_{pix} \geq 0$ **then**
              Delete($\mathbf{A}_{CoM1}$, $\mathbf{p}_{pix}$)
          return $\mathbf{A}_{CoM1}$
        else
          return $\mathbf{A}_{CoM1}$
    else
      return $\mathbf{A}_{CoM1}$

D. CoM Estimation Method

We leverage the predicted sense of rotation of a pushed object to narrow the probable CoM regions in an iterative manner using Algorithm 1 and Algorithm 2. First, we collect a series of pushing examples as pushing priors. These pushing priors are used to help infer resultant object rotations for other pushes. Then, we sample a series of contact points uniformly around the object perimeter. Incorporated with the pushing priors, for each contact point, we query the prediction model to predict the rotation for different pushing directions at that contact point. There is considerable empirical evidence that the deep learning model predicts accurately the sense of rotation of a pushed object if the pushing action causes a large amount of rotation, but it tends to become less accurate if the pushing action causes a small amount of rotation. Therefore, we remove the pushing actions predicted to cause an object to rotate by an angle smaller than θ_T, and sort the remaining pushing actions based on the amount rotation in descending order. After that, the sorted pushing actions with their predicted sense of rotation of an object are used to narrow down the probable CoM region until the ratio between the sizes of the probable CoM region and convex hull of the object becomes smaller than the area threshold a_T. Fig. 3 shows a sequence of snapshots of probable CoM regions being narrowed down.

IV. Two-Edge-Contact Pushing

We introduce our ZMTEP method for translating an object to a goal pose under the quasi-static assumption. First, the forces exerting on the object are analyzed when the object undergoes pure translation. We then revisit moment labeling representation for contact forces. We compare different contact configurations and analyze tolerance to noise on contact and estimated CoM positions. Finally, we present the ZMTEP method for two-edge contact configuration selection.

A. Quasi-Static Analysis of Pure Translation

Let us consider a pusher and an object lying on a flat surface with isotropic friction, where the CoM, the centroid of pressure distribution, and the centroid of friction are projected to the same point on the plane [37]. The magnitude of the contact force between the pusher and the object can be arbitrarily large due to the non-penetration constraint of a rigid body. In case the object moves in low speed, the inertia force would be in the order of the friction force. In translational motion, sliding friction is replaced by an equivalent resultant force passing through the CoM that opposes the motion of the object.

As the two-edge-contact pushing exploits the pusher's collaborative contact interactions with the object, the direction of the net force must be the same as the pushing direction to achieve translational motion. The pusher's contact force acts inward on the object, and two contact forces must be able to set the object in motion toward the direction of the net force additively. Also, the line of the net force must pass through the CoM of the object to generate zero moment about the CoM.

B. Zero Moment Two Edge Contact Pushing (ZMTEP)

We first define several explanatory lines of direction to convey the proposed idea of two-edge-contact configuration. We then analyze the tolerance range of the contact configuration, leading to the ZMTEP for pure translation.

1) Defined Lines: We define five lines: $l_d, l_n, l_{fl}, l_{fr}, l_c$. l_d is the line passing through the CoM of an object and is aligned with the pushing direction. l_n is perpendicular to the surface at the contact point between the pusher and the object. l_{fl} and l_{fr} are the left and right boundary of the friction cone at the contact point, respectively. l_c is the direction of the contact force bounded by l_{fr} and l_{fl}. When there is no friction between the pusher and the object, l_c coincides with l_n. Fig. 4 illustrates an example for pushing an object along l_d, where two contact points between the object and the pusher are denoted by the subscript 1 and 2, respectively. The shaded region is the moment representation for this contact configuration, which will be introduced latter.

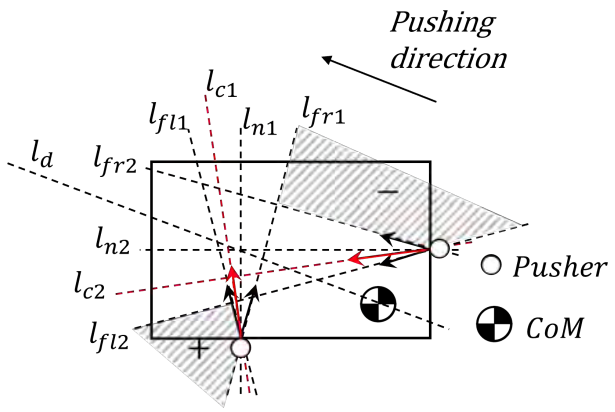

Fig. 4. Illustration of the lines defined in IV-B1.

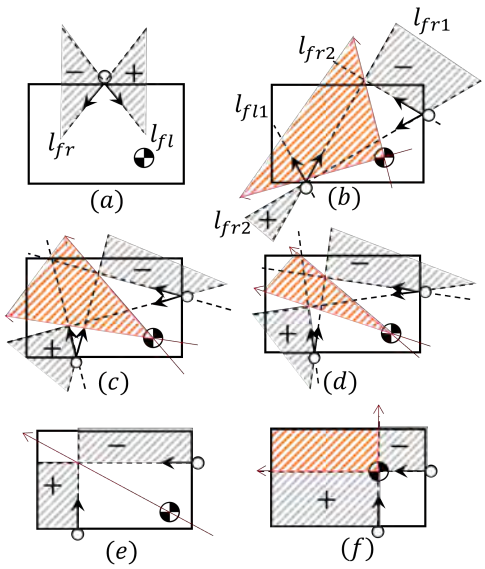

Fig. 5. Moment labeling representation of pusher-slider system. As in Fig. 4, pushers are denoted by small circles which make contact with the slider represented by a rectangle. The CoM has an offset from the centroid. Black arrows delimit the friction cone at the contact. Red regions show the pushing directions along which the object can be translated using the contact configuration employed.

2) Two-Edge Contact Configuration: Within the scope of two-edge-contact pushing between the pusher and the object, the contact configuration can be described by $\mathbf{C}_{tec}$

$$\mathbf{C}_{tec} = (\mathbf{cp}_1, \mathbf{n}_1, \mathbf{cp}_2, \mathbf{n}_2), \tag{2}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are specified by the definition of l_n. In this work, we only consider the case that $\mathbf{n}_1$ and $\mathbf{n}_2$ are not parallel. In other words, l_{n1} intersects with l_{n2}. If the friction coefficient is known, $l_{fl1}, l_{fr1}, l_{fl2}.l_{fr2}$ can be also specified.

3) Moment Labels for Two-Edge-Contact Configuration: Moment labeling gives a graphical representation of the composite wrench cone. Each contact has a friction cone associated with the contact point whose boundary is determined by the friction coefficient between the pusher and the object. Fig. 5(a) shows a moment labeling of a composite wrench cone for a single-point contact. The gray shaded regions represent the sign of moment that the resultant force would generate about the region. This can be interpreted as all of the forces that have a different sign of moment about the shaded region or the forces that pass through the shaded region cannot be generated.

Figs. 5(b)-(f) show moment labeling representations for two-edge-contact configurations. The moment labels for each contact can be combined to represent the composite wrench cone. The red shaded regions show the composite wrench cone passing through the CoM with two-edge-contact configuration, where any resultant wrench inside the composite wrench cone satisfies the force and moment conditions. Therefore, the composite wrench cone also specifies the directions along which the object can be translated.

Figs. 5(b)-(e) show the same $\mathbf{C}_{tec}$ with gradually reduced friction cones, and the red shaded regions become narrower correspondingly. Fig. 5(e) is a special case that the friction coefficient between the pusher and the object is equal to zero. In this case, shaded regions converge to a single point. The resultant forces must pass through this intersection point. Therefore, all of the resultant forces that also pass through the CoM have the same direction, which is from the CoM to the intersection of the shaded regions. This set of forces is always a subset of the set of forces generated in Fig. 5(b)-(d). In other words, this direction can be always achieved by the contact configuration regardless of friction cones. When two contact normal forces pass through the CoM as shown in Fig. 5(f), the resultant forces are positively spanned by these two normal forces even though there is no friction between the pusher and the object.

Theorem 1: Given l_d and a pushing direction, if there exists a two-edge-contact configuration with which the corresponding contact normal forces positively span the pushing direction, also l_{n1}, l_{n2} and l_d intersect at a single point, then the object can be translated along the pushing direction using the two-edge-contact configuration.

Based on the moment labeling analysis of Fig. 5(e), given a known object with $\mathbf{C}_{tec}$, we can always find a pushing direction along which the object is translated.

4) Contact Position Tolerance Analysis: Robots may not precisely achieve a desired contact configuration due to various types of image noise. Therefore, it is of importance to investigate under what circumstances the contact noise can be tolerated. As shown in Fig. 6, given an object with known CoM and its pushing direction, we can draw l_d. For a $\mathbf{C}_{tec}$, the set of composite wrenches passing through the CoM can be illustrated by the red region. A $\mathbf{C}_{tec}$ is valid, if l_d lies inside the composite wrench cone. Let us examine two-edge-contact configurations by the orange and green pusher in Fig. 6(a)-(b), where the blue arrow is the pushing direction and l_d is coincident with the boundary of the wrench cone. Keeping the position of the orange pusher on the horizontal edge fixed, we change the position of the green pusher. Without loss of generality, any combination of two contact forces can positively span the pushing direction. Fig. 6(a) shows the lower limit position of the green pusher found by drawing a line that passes through $l_d \cap l_{fr1}$ parallel to l_{fr2}. Similarly, Fig. 6(b) shows the upper limit position of the green pusher found by drawing a line that passes through $l_d \cap l_{fl1}$ parallel to l_{fl2}. If we draw a line segment whose endpoints are the lower

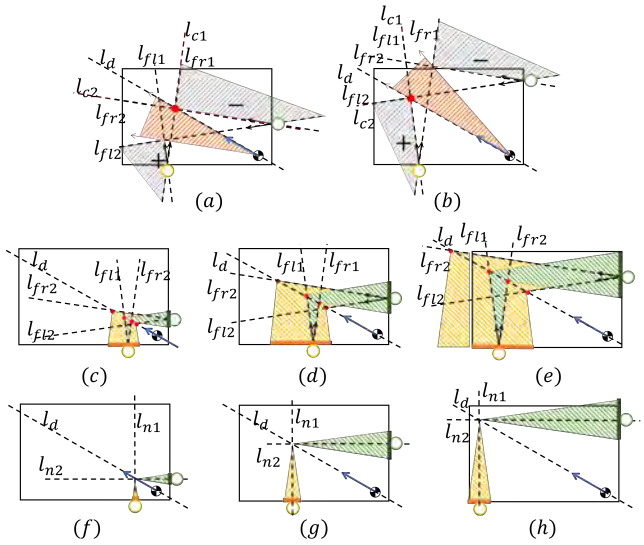

Fig. 6. Contact position tolerance range. (a) and (b) show the position limits of green pusher when orange pusher remains fixed. (c)-(e) show the tolerance range of each pusher position when the other pusher remains fixed. (f)-(h) shows common tolerance ranges of both pushers' positions within which the object can be purely translated.

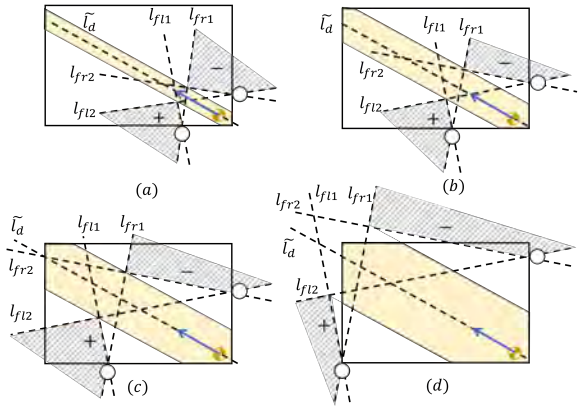

Fig. 7. CoM tolerance range given the two-edge-contact configurations.

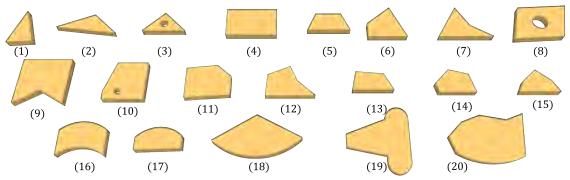

Fig. 8. The twenty object shapes used in the simulation experiments.

and upper limit positions of the green pusher, we can obtain the tolerance range of the green pusher position. Similarly, keeping the position of the green pusher fixed, we can find the tolerance range of the orange pusher. In some cases (see Fig. 6(e)), the endpoint may be out of the object boundary. In such cases, the line segment ends on the boundary.

Fig. 6(c)-(e) show the contact position tolerance ranges represented by colored line segments. We found that, as the distance between the contact points increases, the tolerance range also increases up to a certain point and then decreases as the contact point approaches the object boundary. Also, the tolerance range increases if the friction cone becomes larger.

In practice, the positions of both pushers may be perturbed at the same time. Fig. 6(f)-(h) show the common tolerance range of two pushers. In other words, if two pushers select the contact position within the common tolerance range, the net force can be generated for translating the object. This common tolerance range is specified by the position limits of both pushers found as follows: first, we assume that both pushers make contact with the object by Theorem1, and l_d, l_{n1}, and l_{n2} intersect at a single point. Then, we let the orange pusher move toward left (or right) on its edge until l_{fr1} (or l_{fl1}) passes through the intersection point $l_d \cap l_{n2}$. The limits for the green pusher position can be found in a similar way. The common tolerance range depends on the distance between the contact points in the same way as Fig. 6(c) to (e).

5) CoM Estimates Tolerance Analysis: Error-prone CoM estimates also cause incorrect contact configuration. Therefore, we need to analyze the tolerance range of the CoM. Given an object with the estimated CoM and a pushing direction, we can draw $\tilde{l}_d$. If the ground truth CoM is on $\tilde{l}_d$, there is no estimation error introduced, because we select $\mathbf{C}_{tec}$ based on Theorem 1. Fig. 7 shows four two-edge-contact configurations for pushing an object along the blue arrow with an estimated CoM. Based on the friction cone, we can obtain the CoM tolerance range by the following procedure:

- Find the intersection points $l_{fl1} \cap l_{fl2}$ and $l_{fr1} \cap f_{fr2}$.
- Draw two boundary lines passing through one intersection point parallel to the pushing direction.
- The CoM tolerance range is wrapped by the boundary lines and the object boundary.

If the ground truth CoM is inside the yellow shaded region, the estimated CoM can be tolerated. In other words, the yellow shaded region denotes the tolerance range of the CoM estimates. With the distance between the pushers becomes larger, the tolerance range enlarges. The tolerance range will also be larger for the same two-edge-contact configuration if the friction cone becomes larger. It can be observed that the two-edge-contact configuration with a large distance between two contact points shows high tolerance toward inaccurate estimates.

We now propose our ZMTEP method in order to find two-edge-contact configurations for pushing a novel object as shown in Algorithm 3. Given a number of sampled contour points with associated contact normals, the CoM position, and the pushing direction, this algorithm outputs the most suitable two-edge-contact configuration in which the distance between two contact points is maximized. As the tolerance range becomes smaller due to near-corner contact selection, we remove the contour points near object corners. When dealing with a novel object, ZMTEP estimates the CoM using the method proposed in Section III.

V. EXPERIMENTAL SETTINGS

We carried out experiments both in simulation and real robot settings. We denote the surface friction coefficient as μ_s and

Algorithm 3: Zero Moment Two-Edge Pushing

Input: $\{\mathbf{cp}_i\}^m, \{\mathbf{n}_i\}^m, \mathbf{P}_{CoM}, d_p$

/* $\{\mathbf{cp}_i\}^m$, $\{\mathbf{n}_i\}^m$ are the sampled contour points and the associated normal direction. $\mathbf{P}_{CoM}$ is the position in image frame. d_p is the pushing direction. */

Output: $\mathbf{C}^*_{tec}$

/* $\mathbf{C}^*_{tec}$ refers to a two-edge contact configuration */

$\mathbf{C}^*_{tec} \leftarrow \emptyset$ // initialize $\mathbf{C}^*_{tec}$

$D \leftarrow 0$ // initialize a variable representing the distance between two contact points

for $\mathbf{cp}_i$, $\mathbf{n}_i$ *in* $\{\mathbf{cp}_i\}^m, \{\mathbf{n}_i\}^m$, i *is from 1 to* m **do**

  Get l_{n1} based on $\mathbf{cp}_i$ and $\mathbf{n}_i$

  Get l_d based on $\mathbf{P}_{CoM}$ and d_p

  Get $l_{n1} \cap l_d$

  for $\mathbf{cp}_j$, $\mathbf{n}_j$ *in* $\{\mathbf{cp}_j\}^m$, $\{\mathbf{n}_j\}^m$, j *is from* i *to* m **do**

    Get l_{c2} based on $\mathbf{cp}_j$ and $\mathbf{n}_j$

    /* $\|a,b\|$ represents the Euclidian distance between a and b. ε is a small value */

    if $\|l_{n2} \cap d_p, l_{n1} \cap l_d\| < \varepsilon$ **then**

      if $\|\mathbf{cp}_i, \mathbf{cp}_j\| > D$ **then**

        $\mathbf{C}^*_{tec} \leftarrow (\mathbf{cp}_i, \mathbf{n}_i, \mathbf{cp}_j, \mathbf{n}_j)$

        $D \leftarrow \|\mathbf{cp}_i, \mathbf{cp}_j\|$

return $\mathbf{C}^*_{tec}$

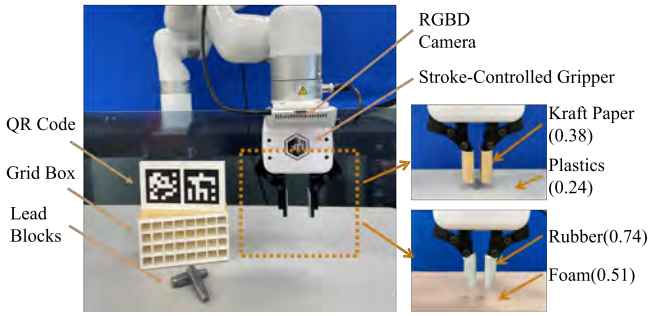

Fig. 9. Real robotic pusher experimental platform. The left figure shows the CoM estimation and object translation experiment setting. The two right figures show the low and high frictional settings with the measured frictional coefficients in parentheses under the assumption of Coulomb's friction law.

the pusher friction coefficient as μ_p. We created a simulation environment composed of a plane and two spherical pushers. Coulomb's law of friction applies to the push-slider-supporting plane interaction and μ_s and μ_p can be adjusted. We used twenty different objects with convex and concave shapes in Fig. 8. The size of objects varies from $27cm^2$ to $150.5cm^2$.

A real experimental setup is shown in Fig. 9. A 5-DoF manipulator equipped with a variable opening stroke parallel-jaw gripper pushes a CoM changeable grid box on a flat surface. The stroke ranges from $0cm$ to $8.6cm$ and the size of the grid box is $14cm \times 8cm$. We wrap the gripper with two different materials: a low friction kraft paper and a high friction rubber. As for the plane surface, we test the table top surface itself and the foam surface. The grid box is wrapped in kraft paper. As the box has uniform density, the CoM is the same as its centroid. We put two lead blocks into different cells to change the CoM. The box weighs around 200 grams and each lead box has around 80 grams. The position and orientation of the box are monitored using ArUco Markers [38] attached to the top cover of the box.

VI. CoM Estimation Experiment

A. Simulation Experiment

We evaluate the accuracy of our CoM estimation method in the following three aspects: (1) objects of different shapes and sizes, (2) different frictional settings, and (3) impact of the proposed deep learning model. For the first purpose, we use twenty different objects depicted in Fig. 8. For each object, we assign ten different CoM locations. For the second purpose, we assign 0.1 and 1.0 for a low frictional setting and a high frictional setting, respectively, to both μ_s and μ_p. For the third purpose, as the prediction accuracy of the deep learning model highly depends on the pushing priors, we use different single contact pushing priors collected as follows. Given a specific object, we randomly sample several contact points. The robot pushes the object at each contact point $3cm$ forward along any of the $\{\theta_p\}^a$ directions. The contact points and pushing directions, as well as the resultant object motion are recorded as the pushing priors.

The pipeline is as follows: The pushing priors are collected for each object with a specified CoM location in each frictional setting. During pushing prior collection, We collect a total of twelve pushing priors for each estimation procedure. Then, the pushing priors are fed into the prediction model. As we set the total number of sampled contact points $\{\mathbf{cp}_i\}^m$ to fifty in Algorithm 1, and there is five direction pushing directions ($\{\theta_p\}^a$ is set to $\{-60°, -30°, 0°, 30°, 60°\}$.*w.r.t.* the normal direction at each contact point), the prediction model predicted resultant object motion for two hundred and fifty different pushing actions. All of the $\mathbf{R}_P$'s specified by the five pushing directions at each sampled contact point can divide the convex hull of the object into small enough regions to ensure that the estimation error will be bounded by the area threshold a_T to achieve the estimation accuracy requirement. The angle threshold θ_T is set to 2°, that is, if the predicted amount of rotation is less than 2°, we will not update the potential CoM region based on this pushing action. We set the area threshold a_T to 0.1, which means that the CoM estimation process will be terminated if the ratio between the sizes of the probable CoM region and convex hull of the object becomes smaller than 0.1. Finally, we run Algorithm 1 to find the potential CoM region and use its centroid as the estimated CoM. This CoM estimation pipeline is repeated three hundred times for each object with a specific CoM in each frictional setting. In each estimation procedure, different pushing priors are used. As we assign ten different CoMs and conduct CoM estimation in two friction settings, the CoM estimation was conducted six thousand times for each object.

The obtained results are presented in Fig. 10. The statistical results are computed over ten different CoM locations assigned

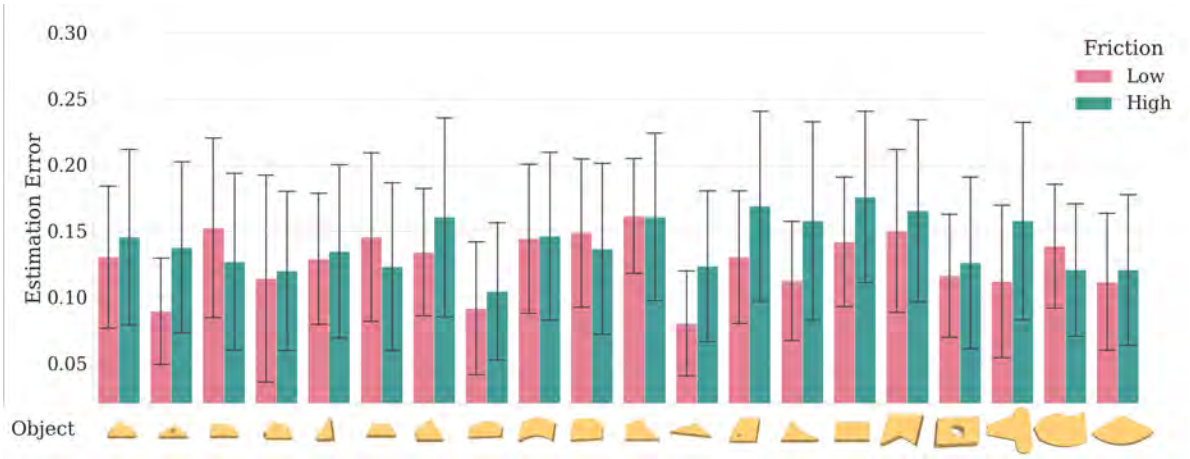

Fig. 10. CoM Estimation Results in Simulation Environments. Objects are sorted by their representative size. The height of bars shows the mean estimation error in percentage and the intervals represent standard deviations. Red and green bars show the estimation errors for low friction and high friction, respectively.

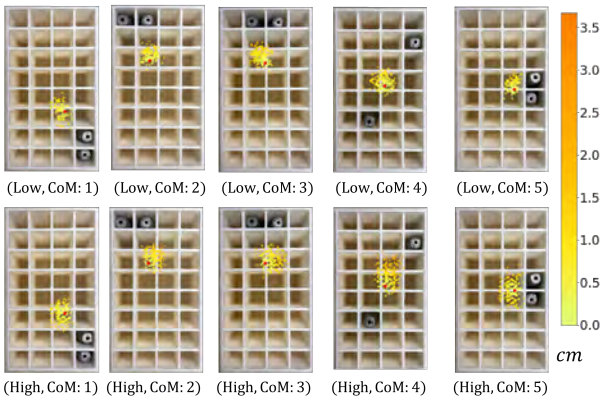

Fig. 11. CoM estimation results in low and high friction settings. Red dot represents the ground truth CoM. Other dots are estimated CoMs. The brighter the color, the closer to the ground truth.

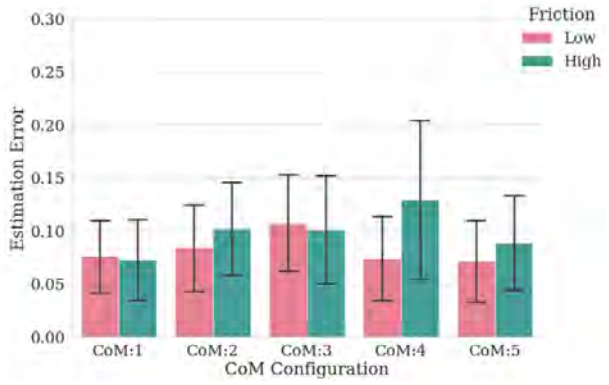

Fig. 12. CoM Estimation Result in Real Experiment. Red and green bars show the estimation errors in percentage for low friction and high friction, respectively.

to each object. The objects are sorted by the representative size (RS), defined by the distance between the object's centroid to the farthest point on its perimeter, and the estimation errors are multiplied by the reciprocal of RS. The main idea behind this is that if an object's physical properties are not observable, one can assume that the CoM is located at the centroid of that object. This would be same with the initial step in our algorithm before we apply any pushing action. The maximum error of the estimation occurs when the CoM ground truth is located on the object perimeter farthest from its centroid. Given that RS is the maximum estimation error, we provide the relative error computed separately for each object. As shown in Fig. 10, the mean estimation error is around 0.15. After investigating the effects of different frictional settings, we reached a conclusion that our CoM estimation method performs more accurately overall in a low frictional environment.

B. Experiments in Real Platform

In real experiments, we insert two lead blocks into the box to change the CoM, $\mathbf{P}_{CoM}$, that can be computed by Eq. 3.

$$\mathbf{P}_{CoM} = \frac{1}{M}\Sigma_i^n m_i \mathbf{r}_i, \tag{3}$$

where M is the total mass equal to the sum of masses of the box and two lead blocks, denoted by m_i, respectively, and $\mathbf{r}_i$ represents the position of m_i in a reference coordinate frame in the plane. We measured the masses of the empty box and each lead block using an electric scale with precision 0.1 gram. The $\mathbf{r}_i$ coordinate of the empty box was regarded as the centroid of the box as it has uniform density, and the $\mathbf{r}_i$ value of each lead block approximated to the center of the compartment in which each block was inserted.

We follow the previous pushing prior collection procedure in two different settings: a foam surface and the rubber wrapped gripper referred to as high frictional setting and a plastic surface and the kraft paper wrapped gripper referred to as low frictional setting. For each frictional setting, we select five different lead block cells as shown in Fig. 11, resulting in five different CoMs. We sample twelve contact points

uniformly across the box perimeter, and the robot pushes each contact point along five directions. The grid box was pushed sixty times for each CoM configuration and frictional setting. The CoM estimation pipeline is the same as the simulation. The estimation procedure was repeated six hundred times, for each of which, twelve pushing priors are sampled from sixty pushing examples.

We use the same method to represent the estimation error in the real experiments of CoM estimation, which is shown in Fig. 12. It can be observed from the figure that our CoM estimation method can generalize to real settings, comparable to the results of simulation experiments. Overall, the estimation uncertainty is smaller in low frictional setting. As the result for the fourth CoM configuration, the estimation error became large. One reason is that, compared with other CoM configurations, the pressure distribution is more decentralized. Such a configuration causes an object to rotate less, leading to inaccurate CoM estimates.

VII. Object Pushing Experiments

A. Simulation Experiment

We conduct three simulation experiments to

- find feasible sets of two-edge contact configuration.
- evaluate the effect of the surface and pusher friction.
- evaluate the performance of ZMTEP for different objects.

In the first simulation, we find the feasible two-edge-contact configurations that translate the object (11) in Fig. 8 along a pushing direction. Here both μ_s and μ_p are set to 0.1. The two contact points are selected on different edges whose normal directions are not parallel. There are ten different choices to select two edges. We uniformly sample the contact points from each of the two edges and combine each pair of contact points as a two-edge-contact configuration. A push is considered a success if the object is translated to a goal pose within $0.5cm$ and 5°, respectively.

We show four examples in Fig. 13, where each two-edge-contact configuration $\mathbf{C}_{tec}$ forms an intersection point $l_{n1} \cap l_n 2$. We observed that almost all of the feasible two-edge-contact configurations represented by the blue circle marks are found by combining the edges whose contact normals can positively span the pushing direction. In this figure, the blue intersection points are distributed symmetrically along the blue arrow passing through CoM aligned in the pushing direction denoted as l_d in Section IV-B1. The closer the intersection point to l_d, the less the change in object orientation during pushing. The two-edge-contact configurations whose intersection points are on the blue arrow satisfy Theorem1 and therefore result in pure translational motion. Also, the number of blue circle marks increased in the direction of l_d arrowhead including those marks located away from l_d, leading to an increase in the number of suitable configurations available. This phenomenon can be explained by the CoM estimates tolerance analysis. For those contact configurations whose normal directions positively span the pushing direction, each contact configuration obeys Theorem1 but the CoM estimates may not be accurate. When moving closer to the arrowhead, the distances between two pushers increases so

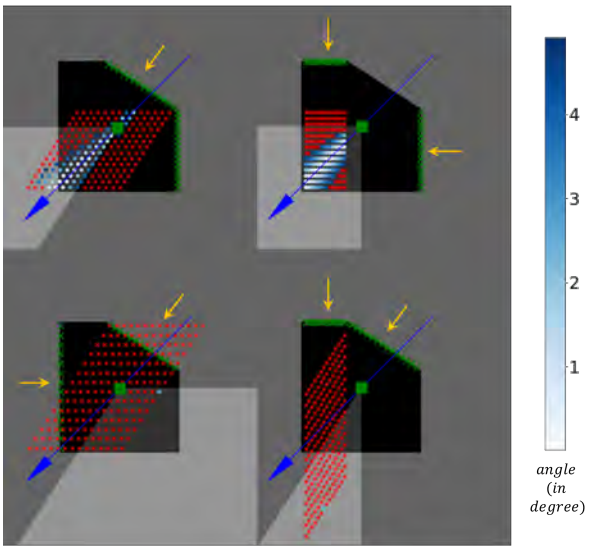

Fig. 13. Results of object translation simulation. Blue arrow shows the pushing direction and green square mark shows the CoM. Green cross marks on object edges represent the sampled contact points. Yellow arrows show the contact normals that can positively span bright gray regions. A pair of contact points are selected from the sampled contact points on each of two edges. Red and blue cross marks are intersection points formed by two-edge-contact configurations. Red marks are fail and blue marks are success.

that the configuration shows high tolerance to the error of CoM estimates. Observing the distribution of the red cross marks, the failure can be a result of the following three reasons: (1) the corresponding normal directions cannot positively span the pushing direction, such as the second row of Fig. 13 (2) the two-edge-contact configuration whose corresponding intersection point are too far away from l_d, and (3) the distance between the contact points is too narrow so that even a small error in contact points results in failure.

In the second simulation, we use nine different friction settings by permuting three different friction coefficients $(0.1, 0.5, 1.0)$. In each of the settings, we push the triangular object (1) shown in Fig. 8 along random directions. We select two edges whose normal directions can positively span the pushing direction. We then combine each contact point on each edge to find feasible two-edge push configurations. A push is considered a success using the same criterion as the first simulation experiment. From the result of the second simulation experiment, we found that with the increase of the pusher friction or surface friction, the number of successful pushing has also increased. We conclude that both the friction between the pusher and the object and the friction between the surface and the object mutually contribute to enlarging the contact tolerance range. Similar to the first pushing simulation experiment, the number of intersection points increased in the direction of l_d arrowhead for each of the frictional setting, which leads to the same conclusion that the contact configuration having larger distance between two contact points exhibits high CoM estimates tolerance. In addition, as each row or column of intersection points along the contact normals can be seen as a set of contact configurations that one pusher position gets perturbed when the other remains fixed. Along

TABLE I
OBJECT TRANSLATION SIMULATION EXPERIMENT

Object	Low Friction				High Friction			
	Proposed1	**Proposed2**	**Baseline1**	**Baseline2**	**Proposed1**	**Proposed2**	**Baseline1**	**Baseline2**
1	**1.000**	0.985	0.277	0.362	**1.000**	**1.000**	0.823	0.769
2	**0.942**	0.925	0.050	0.175	**1.000**	**1.000**	0.350	0.500
3	**1.000**	0.975	0.117	0.242	**1.000**	0.983	0.608	0.567
4	**0.975**	0.900	0.267	0.492	**1.000**	**1.000**	0.992	0.733
5	**0.984**	0.952	0.234	0.419	**0.960**	**0.960**	0.863	0.677
6	**1.000**	0.921	0.414	0.443	0.979	**0.993**	0.957	0.786
7	**0.938**	0.906	0.312	0.328	**1.000**	0.969	0.744	0.721
8	**0.981**	0.915	0.321	0.481	**1.000**	**1.000**	0.849	0.660
9	**0.905**	0.781	0.248	0.489	**0.978**	0.942	0.964	0.761
10	**0.985**	0.916	0.298	0.511	**1.000**	**1.000**	0.955	0.674
11	**0.992**	0.961	0.403	0.597	0.992	**1.000**	0.891	0.845
12	**0.949**	0.934	0.301	0.368	**1.000**	0.993	0.853	0.757
13	**0.967**	0.919	0.366	0.528	0.992	0.992	**1.000**	0.815
14	**1.000**	0.835	0.339	0.543	**1.000**	**1.000**	0.808	0.752
15	**1.000**	0.976	0.317	0.357	**1.000**	**1.000**	0.762	0.683

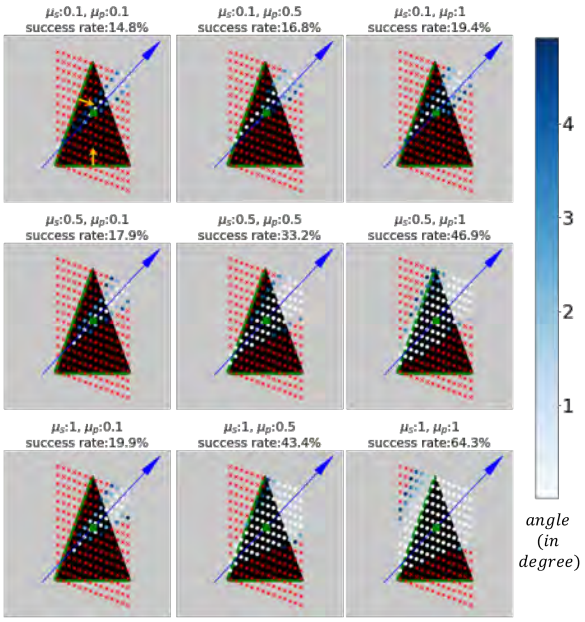

Fig. 14. Nine simulation environments with different surface and pusher frictions. Success rate is computed by calculating the ratio between the total number of success pushes and the total number of pushes.

the pushing direction, when counting the number of success pushes in rows or columns, the number of success pushes increases up to a certain point and then deceases, which is supported by the contact position tolerance analysis.

In the third simulation, we verify the ZMTEP using the objects in the first two rows in Fig. 8 in a low friction setting (μ_s and μ_p are 0.1) and a high friction setting (μ_s and μ_p are 1.0). Each object assigned to ten different CoMs is translated 25cm along the fifteen uniformly sampled directions. Henceforth, a push is considered a success if the slider is within 1cm and 10° of a goal pose.

For comparison purpose, we used following methods:

- **Proposed1** selects two contact points by ZMTEP, given the CoM.
- **Proposed2** selects two contact points by ZMTEP, estimating the CoM.
- **Baseline1** uses two contact points that have the maximum distance from the slider's centroid along the direction perpendicular to the pushing direction. Increasing the distance between two contact points, errors of contact location and CoM estimation can be better tolerated.
- **Baseline2** uses two contact points that are the largest possible equal distance away from the given CoM along the direction perpendicular to the pushing direction. This method relies on the CoM ground truth and ensures that the CoM is in the middle between two contact points along the pushing direction. The distance between two contact points tends to be smaller than **Baseline1**.

The obtained results in the form of mean ratio of successful pushes are presented in Table I. From the result, in a low friction environment, **Proposed1** performed best and **Proposed2** is the second best. The baseline methods performed far less accurately than our proposed methods. **Baseline1** performs worse than **Baseline2**, as the tolerance capability is restricted by the limited friction. **Baseline2** reaps the benefits of knowing the CoM ground truth. In high friction environment, the performance of **Proposed1** and **Proposed2** improved further. Compared with the baseline methods in a low friction environment, even though the two-edge contact configurations in high friction setting are the same as the ones in low friction setting, their performances are greatly improved. **Baseline1** performed better than **Baseline2** as the tolerance capability gets enlarged due to the increase in friction. This result shows that high friction helps expand the wrench cone. Therefore, the performance of baseline methods highly rely on the friction. On the other hand, the proposed methods work well irrespective of frictional conditions. Compared with **Proposed1**, even though **Proposed2** use the estimated CoM which may introduces undesired contact configuration, the performance of **Proposed2** drops only slightly. The reason is that we select a two-edge-contact configuration with a tolerance range.

B. Real Experiment

Changing the friction properties of the pusher and the plane as well as the CoM of a grid box slider, we find a set of feasible

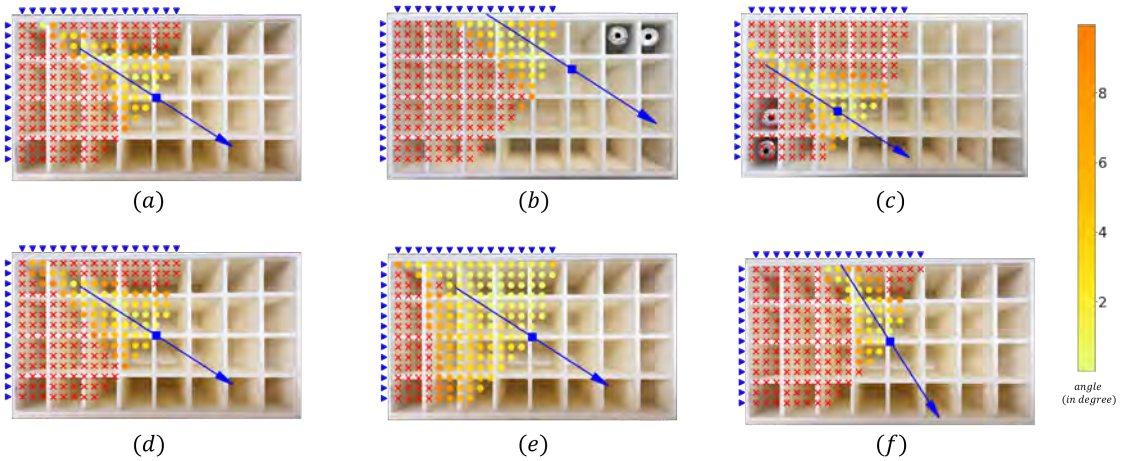

Fig. 15. Real robot pushing of a box with a variable stroke gripper along the blue arrow direction. Blue triangle markers are sampled contact points. Each pair of contact points form an intersection point denoted by red cross markers and orange dot markers. Blue squares are the CoMs of the grid box. The CoMs in (b) and (c) are biased by two lead blocks near the corner.

two-edge-contact configurations. The robot uses its parallel-jaw gripper to translate the box slider 25cm along a desired direction. Given a pushing direction, we sample 14 contact points on the short edge and 16 contact points on the long edge, yielding a set of 224 paired configurations. As we set the gripper stroke limit to 8.1cm, some of the paired configurations cannot be achieved, leading to 183 pushes available.

We conduct a total of six experiments as illustrated in Fig. 15(a)-(f), where the robot pushes the box in the direction of the blue arrow. In (b) and (c), the position of CoM is biased arising from the addition of lead blocks. In all experiments, the object is placed on a plastic plane pushed by the kraft paper wrapped gripper except (d) and (e). (d) uses a high friction foam surface and (e) uses a rubber wrapped gripper.

From the result of the real object translation experiment shown in Fig. 15(a), the closer to the line of pushing the intersection points are, the more stable the pushes are. The intersection points of feasible two-edge-contact configuration form a cone. With the stroke of the parallel-jaw gripper increases, the contact configurations tolerate more. In Fig. 15(b) and (c), the feasible set of intersection points is biased the same amount as the CoM. In Fig. 15(d), as the surface friction is larger than others, the number of successful pushes increases correspondingly. In Fig. 15(e), we changed the friction between the pusher and the object, the number of successful pushes greatly increased. Fig. 15(d) and (e) state that both surface and pusher frictions contribute to enlarging the tolerance range of the contact. In Fig. 15(f), we changed the pushing direction, the feasible set is also changed correspondingly. In summary, we verified Theorem 1 on a real platform under various conditions. The feasible two-edge-contact configurations depend on the CoM location as well as the pushing direction. Increasing the friction of supporting surface or the pusher enlarges the contact tolerance.

VIII. Conclusion

We addressed the problem of planar multi-contact pushing of novel objects undergoing pure translational motion. First, we presented a novel CoM estimation method by combining Mason's voting theorem and a deep learning motion prediction model trained on our simulation dataset **SimPush**. We demonstrated the process of how the probable CoM location can be narrowed down with no *a priori* assumption about friction between the pusher and object. Secondly, we proposed the two-edge-contact pushing interaction called ZMTEP to translate an object to a goal pose. Using moment labeling representation, we showed that the pushing configuration stated in Theorem 1 satisfies the condition of pure translation. Thirdly, toward real-world applications suffering from image sensor noise and error-prone CoM estimates, we analyzed the tolerance region of contact configuration to both contact noise and inaccurate CoM estimates. We showed that as the distance between two contact locations increases, the proposed ZMTEP exhibits high tolerance to both issues.

We demonstrated through extensive experiments both in simulation and real robotic pusher-slider settings that the proposed CoM estimation method has good mean squared error properties and small standard deviation and ZMTEP outperformed the baseline methods. To the best of our knowledge, this research is the first attempt to provide a thorough analysis and empirical evidence of multi-edge-contact pushing based on fewer assumptions to achieve a reasonably close optimal shortest path of translational motion. One of the future directions would be investigating multi-edge-contact configuration selection for re-orienting an object before and after an action executes. In addition, pushing an object at a high speed, beyond the quasi-static approximation, would be also of importance to further reduce the time needed for translating an object to a goal pose.

References

[1] M. T. Mason, "Progress in nonprehensile manipulation," *The International Journal of Robotics Research*, vol. 18, no. 11, pp. 1129–1141, 1999.

[2] A. Cosgun, T. Hermans, V. Emeli, and M. Stilman, "Push planning for object placement on cluttered table surfaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4627–4632, 2011.

[3] F. Ebert, C. Finn, A. X. Lee, and S. Levine, "Self-supervised visual planning with temporal skip connections," in *1st Annual Conference on Robot Learning*, vol. 78 of *Proceedings of Machine Learning Research*, pp. 344–356, 2017.

[4] Z. Dong, S. Krishnan, S. Dolasia, A. Balakrishna, M. Danielczuk, and K. Goldberg, "Automating planar object singulation by linear pushing with single-point and multi-point contacts," in *IEEE International Conference on Automation Science and Engineering*, pp. 1429–1436, 2019.

[5] L. Chang, J. Smith, and D. Fox, "Interactive singulation of objects from a pile," *IEEE International Conference on Robotics and Automation*, pp. 3875–3882, 2012.

[6] M. Danielczuk, J. Mahler, C. Correa, and K. Goldberg, "Linear push policies to increase grasp access for robot bin picking," in *IEEE International Conference on Automation Science and Engineering*, pp. 1249–1256, 2018.

[7] M. Dogar and S. Srinivasa, "A framework for push-grasping in clutter," *Robotics: Science and Systems*, vol. 1, 2011.

[8] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4238–4245, 2018.

[9] K. M. Lynch and M. T. Mason, "Stable pushing: Mechanics, controllability, and planning," *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 533–556, 1996.

[10] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 755–773, 2020.

[11] 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.

[12] 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.

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

[14] 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.

[15] T. Standley, O. Sener, D. Chen, and S. Savarese, "image2mass: Estimating the mass of an object from its image," in *Proceedings of the 1st Annual Conference on Robot Learning*, vol. 78 of *Proceedings of Machine Learning Research*, pp. 324–333, 2017.

[16] 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.

[17] 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.

[18] 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.

[19] K. 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*, vol. 1, pp. 416–421, 1992.

[20] 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.

[21] A. Allevato, E. S. Short, M. Pryor, and A. Thomaz, "Tunenet: One-shot residual tuning for system identification and sim-to-real robot task transfer," in *Proceedings of the Conference on Robot Learning*, vol. 100, pp. 445–455, 2020.

[22] A. Allevato, M. Pryor, and A. Thomaz, "Multi-parameter real-world system identification using iterative residual tuning," in *Proceedings of the ASME International Design and Technical Conference*, 2020.

[23] 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.

[24] 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.

[25] 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.

[26] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction part 1. limit surface and moment function," *Wear*, vol. 143, no. 2, pp. 307–330, 1991.

[27] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction part 2. dynamics of motion," *Wear*, vol. 143, no. 2, pp. 331–352, 1991.

[28] A. Kloss, S. Schaal, and J. Bohg, "Combining learned and analytical models for predicting action effects," *CoRR*, vol. abs/1710.04102, 2017.

[29] C. Lin, M. Grner, P. Ruppel, H. Liang, N. Hendrich, and J. Zhang, "Self-adapting recurrent models for object pushing from learning in simulation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5304–5310, 2020.

[30] T. Hermans, F. Li, J. M. Rehg, and A. F. Bobick, "Learning contact locations for pushing and orienting unknown objects," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 435–442, 2013.

[31] S. Krivic and J. Piater, "Online adaptation of robot pushing control to object properties," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4614–4621, 2018.

[32] C.-Y. Chai, W.-H. Peng, and S.-L. Tsao, "Adaptive unknown object rearrangement using low-cost tabletop robot," in *IEEE International Conference on Robotics and Automation*, pp. 2372–2378, 2020.

[33] Q. Li and S. Payandeh, "Manipulation of convex objects via two-agent point-contact push," *The International Journal of Robotics Research*, vol. 26, no. 4, pp. 377–403, 2007.

[34] J. Lloyd and N. F. Lepora, "Goal-driven robotic pushing using tactile and proprioceptive feedback," *IEEE Transactions on Robotics*, pp. 1–12, 2021.

[35] 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.

[36] 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.

[37] M. T. Mason, *Mechanics of Robotic Manipulation*. Cambridge, MA, USA: MIT Press, 2001.

[38] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.