Reducing Uncertainty Using Placement and Regrasp Planning on a Triangular Corner Fixture
Zhengtao Hu®, Weiwei Wan®, Senior Member, IEEE, Keisuke Koyama®, and Kensuke Harada®, Fellow, IEEE

Abstract—This paper presented a regrasp planning method to eliminate grasp uncertainty while considering the geometric constraints of a fixture. The method automatically finds the Stable Placement Poses (SPPs) of an object on a Triangular Corner Fixture (TCF), elevates the object from its SPPs to dropping poses and finds the Deterministic Dropping Poses (DDPs), builds regrasp graphs by using the SPP-DDP pairs and their associated grasp configurations, and searches the graph to find regrasp motion sequences for precise assembly. Since the SPPs and their associated regroasps are constrained by the TCF’s geometry and have high precision, the final object poses regroasped via it has low uncertainty and can be directly used for assembly by position control. In the experimental section, we study the performance of analytical and learning-based methods for estimating the DDPs of different objects and quantitatively examine the proposed method’s ability to suppress uncertainty using assembly tasks like peg-in-hole insertion and sheathing tubes, aligning holes, mounting bearing housings, etc. The results demonstrate the method’s robustness and efficacy.

Note to Practitioners—In production lines, robots interact with peripheral devices to improve efficiency and reduce uncertainty. In this work, we focus on a particular peripheral device—a Triangular Corner Fixture (TCF) made by three inclined and mutually perpendicular plates. We study using the TCF to improve manipulation precision. The inclined plates of the TCF form a gravity bucket that holds dropped objects in stable states under gravity. In a real scenario, a robot picks up an object and releases it above the TCF. The released object will reach a stable state on the TCF. Then, the robot regrasps and moves the stabilized object to the target pose with reduced uncertainty. Using the method proposed in this paper, a robot can automatically finish the above procedure by finding all the object’s stable states in the TCF, planning grasp configurations, invalidating infeasible states and grasps, building regrasp graphs and searching the graph to find a regrasp motion sequence that moves the object to a goal pose with high precision for assembly.

In industrial applications, the proposed method has the potential to improve the flexibility of robotic systems for high-precision tasks. In the research fields, it may promote the research on sensorless manipulation and extrinsic manipulation, and push forward the studies in robotic regrasp.

Index Terms—Fixture, precise assembly, regrasp planning.

I. INTRODUCTION

Uncertainty is a crucial problem in employing robotic manipulators for assembly tasks. Especially for autonomous manipulators that receive vision feedback and generate manipulation motion online, uncertainty is challenging to eliminate—They originate from a series of mutually coupled components like vision, control, contacts, etc. Overcoming them and achieving precise manipulation is tricky.

Contemporary literature tends to solve the uncertainty problem using multi-modal sensing and improved sensing algorithms. Related articles reported significant improvements in robotic perception [1]. However, despite the achievements, the improvements in sensing technology still fail to provide sufficient qualifications for autonomous manipulators, as sensing is not the only reason for uncertainty. On the other hand, researchers in the robotic planning and control community developed sophisticated integral motion planning and control policies to enable robots to correct object poses during manipulation. The policies include but are not limited to scanning search, spiral research, impedance control, hybrid force/position control, etc. [2], [3] [4], which need force sensors [5], tactile sensors [6], [7], or current sensors for feedback. Compliant mechanisms are hardware alternatives of the policies [8], [9]. They are effective and less expensive counterparts of the sensor-based implementation. The policy-based methods or the compliant mechanisms have advantages in regulated scenarios but tend to be influenced by environmental changes. Users need to adjust various parameters or key hardware components like springs for different applications.

Unlike methods that improve robotic perception and control or develop new compliant mechanisms, eliminating uncertainty through manipulation while considering geometric and physical constraints is more straightforward, robust, and cost-effective. The fundamental idea is deploying a fixture in the robot workspace. The pose of an object can be precisely aligned and determined by taking advantage of the geometric constraints induced by contacting the fixture and the physical constraints induced by gravity. The idea is not new. It is...
widely seen in factory automation for aligning randomly placed workpieces [10], [11], and has been practiced since the beginning of robotics. However, the object poses and the robot motion for dropping the objects in the previous systems were manually specified. Their stability relied on a system integration engineer's subjective adjustments and examination. After development, the systems were difficult to be altered or redeployed.

This paper reinspects the idea of employing a fixture to reduce uncertainty. Unlike conventional design and mechanical analysis, our focus is on planning. We assume object poses are unknown and detected online. There will be uncertainties in both position and rotation. We propose using a triangular corner fixture made of three mutually orthogonal planes to reduce uncertainty. We develop algorithms to compute an object’s stable poses on a fixture and employ these poses as intermediate states to build regrasp graphs and plan robotic manipulation sequences. Driven by gravity, an object dropped from above the tilted corner fixture may first contact the inner surfaces and then slide to the bottom under gravity and the guides of the mutually orthogonal planes. The algorithms could find the object’s stable intermediate poses at the bottom of a gravitational bucket formed by the fixture and use them to reduce the uncertainty in planned robotic manipulation sequences. The algorithms forced a robot to leverage an intermediate regrasping process to reduce uncertainty. Dropping an object to the triangular corner fixture and letting it converge to known stable placement poses help cancel the uncertainties in both position and rotation.

With the help of the fixture and the proposed algorithms, an object could be manipulated precisely to conduct difficult tasks like insertion. In detail, the method developed in this work automatically finds the Stable Placement Poses (SPPs) of an object on a Triangular Corner Fixture (TCF). Consequently, it enables auto-planned precise robotic regrasp and manipulation. It first computes the SPPs considering geometric contact constraints, physical feasibility, and static stability. Then, it elevates the object from its SPPs to dropping poses and finds the Deterministic Dropping Poses (DDPs) from them. When the object is released from the DDPs, it will rest at expected SPPs. Finally, the method computes the gripper configurations for grasping and regrasping the object considering the TCF, SPPs, and DDPs. The method outputs a pick-and-place sequence that manipulates the object under the TCF’s help with high precision. In the experiments, we studied the performance of different methods for estimating the DDPs of different objects and quantitatively examined the proposed method’s ability to eliminate uncertainty by inserting a peg into holes with different clearance. We also examined the method’s practical performance using real-world assembly tasks like peg-in-hole insertion, sheathing tubes, aligning holes, mounting housings, etc. The results verified that the method enabled a robot to finish assembly tasks without using sensors, compliant control, or complicated mechanism, making the robot system more robust and flexible.

Our novelties and contributions are as follows. First, we extended the functions of autonomous regrasp from the conventional object reorientation to uncertainty elimination. Second, we developed the learning-based estimators to screen the DDP-SPP pairs, which could overcome the limitations of the analytical methods. Third, we integrated planning SPPs, estimating DDP-SPP pairs, and regrasp planning to provide an automatic framework for a robot to use external fixtures.

The paper is organized as follows. Related work is presented in Section II. background information about the TCF and an overall workflow of the proposed method are introduced in Section III. Section IV-VI present the details of the SPP sub-planner, DDP estimator, and release/regrasp sub-planner, respectively. Experiments and analyses are shown in Section VII. Section VIII provides discussions. Conclusions and future work are presented in Section IX.

II. RELATED WORK

The research related to this study includes sensorless manipulation, placement estimation, and regrasp planning.

A. Sensorless Manipulation

Like its name, sensorless manipulation means manipulating objects without using sensors. It relies on the mechanic and geometric constraints of a task to pose objects and is simpler and more robust compared to sensor-based manipulation [12]. Sensorless manipulation is widely seen in automation lines to eliminate uncertainty. The exemplary mechanism used for sensorless manipulation includes chutes, hoppers, bowl feeders and feed tracks, etc [10]. For robotic applications, Mason initially discussed the basic concept of sensorless robotic manipulation in [13]. After that, a variety of sensorless robotic manipulation approaches were studied. For example, Brost et al. [14] proposed using combined pushing and squeezing and flat finger pads to grasp an object with uncertainty. Nie et al. [15] and Hirata et al. [16] designed special-shape finger pads to align uncertain objects. Ha et al. [17] developed an automatic designer that finds finger pad shapes for robustly grasping various objects. Goldberg et al. [18], and Zhou et al. [19] used a sequence of parallel grasp actions for orienting and positioning uncertain objects to a specific pose. Maeda et al. [20] and Varkonyi et al. [21] developed caging-based methods to achieve in-hand manipulation and parts feeding, respectively. Erdmann et al. [22] and Schmidt et al. [23] used the active actions of palms and boundary walls to manipulate objects. Berretty et al. [24] and Akella et al. [24] studied the usage of passive settings like fences. Grossman et al. [25], Erdmann et al. [12], and Mannam et al. [26] respectively used...
robotic manipulators to move a tray attached to its tool center point. As the robotic manipulator moves, an object in the tray will be slid into a trihedral corner and stopped by the tray’s walls. The final pose of the object can be determined by carefully planning and controlling the tray’s tilting motion. Attractive regions [27] also stand as an important method for sensorless manipulation. It allowed low-accuracy robotic systems to perform high-precision manipulation. The previously explored tasks by researchers working on attractive regions included assembly [28], object location [29], and grasping with alignment [30]. A drawback of the attractive region method was the curve of dimensionality. The method was well defined in $\mathbb{R}^3$ but was difficult to be extended for compensating rotational uncertainty. Su et al. [31] developed online estimation methods to overcome the difficulty. Moreover, Shome et al. [32] proposed to rearrange objects with combined suction and non-prehensile manipulation policies. It leveraged online visual observation to determine manipulation policies that were most robust to errors. Anders et al. [33] proposed leveraging the belief state space of objects (determined by environmental geometries) to plan rearranging actions. The method allowed robustly moving objects to desired goals without sensors.

Similar to the conventional sensorless manipulation systems, our proposed method uses geometric constraints to hold objects. The objects are supposed to be dropped by a robotic manipulator onto a TCF and trapped by the tilted TCF inner surfaces under gravity. We assume that visual recognition is used to locate an object’s initial pose, and allow recognition and other uncertainty. We develop algorithms to plan stable placement poses, estimate dropping poses, and plan grasp/regrasp poses to reduce uncertainty while taking advantage of the TCF’s geometric constraints. Our process is fully automatic and applies to various objects provided that their model information is given. The proposed method can be interpreted under the framework of attractive regions. Evaluating an object’s DDPs considering a TCF is similar to an attractive region problem. The corresponding SPPs form the bottom of the attractive regions.

### B. Placement Estimation

We consider a placement estimation as a two-step process. In the first step, we find a set of stable placement poses for an object. In the second step, we infer the dropping or releasing poses based on the placement poses. The related work in placement estimation is reviewed surrounding the two steps.

The most fundamental problem of placement estimation is finding a stable placement on a horizontal plane. In this case, the object’s stability can be determined by checking if its Center of Mass (CoM) projection passes through the convex supporting polygon [34], [35] [36]. As an extension to the fundamental problem, Wan et al. studied the placement planning on a tilted plane [37], a support pin [35], and arbitrary support structures [38]. Harada et al. [39] developed an algorithm to plan the stable object placement with non-flat contact considering the convexity of the paired contact surfaces. They assumed that the friction force was large enough to prevent sliding. The placement stability of rigid bodies and assemblies considering frictional contact was discussed in [40] and [41]. Contact Wrench Space (CWS) was widely used for stability estimation. The radius of a maximum inscribed sphere in the contact wrench cone indicates how much external wrench or inertial wrench a grasp can tolerate. It can be used to evaluate the grasp qualities and find optimum grasp configurations [42], [43] [44], [45], and can also be used to estimate the stability of structures [46], [47].

Dynamic dropping simulation is also widely used for placement estimation [38], [48]. However, to assure the reliability of the simulated results, various parameters need to be tuned, and repeated examinations must be performed, which makes the methods less credible and time-consuming. For this reason, many researchers studied fast alternatives for dynamic simulation. For example, Kriegman and David [49] proposed an algorithm that computed a maximal capture region of the desired stable pose in the configuration space where the object pose would converge into a desired one. Jørgensen et al. [50] presented to generate drop regions for stable poses and discussed two methods, the largest enclosing ellipsoid computation and the kernel density estimation, to determine optimal drop poses from them. Varkony [51] provided a statistics-based prediction method for estimating the resting poses of the dropped parts. Fekula et al. [52] used a similar method to perform the estimation, and based on the reasoned stable poses and the rendered top view images of them, they further positioned the objects using a vision-based method. Baumgartl et al. [53] developed a fast placement planner, which is capable of computing a stable position and orientation for a dropped object in complicated environments. In addition, learning-based methods also became popular for placement estimation and handling the uncertainties in manipulation processes. Lu et al. [54] proposed to train a probabilistic graphical model as a classifier to predict the appropriate grasp types (power grasp or precision grasp). Li et al. [55] developed a deep network that uses a single depth point cloud to estimate the pose of an articulated object. Cheng et al. [56] proposed a two-step learning framework to find an object’s stable placements and then used the learned placements for regrasp planning. Newbury et al. [57] used two Convolutional Neural Networks (CNNs) to estimate both the placement rotations and stabilities and obtain the human-preferred object placements and orientations. CNNs are also widely used to estimate the grasp configurations [58] and predict the grasp qualities [59] as well. Feng et al. [60] used a Support Vector Machine (SVM) and a Long Short-Term Memory (LSTM) model to analyze the features of tactile sensors to detect slip and unstable grasps. Additionally, learning methods improved the accuracy of multiple-view stereo based depth analysis [61], [62], which has the potential to be used for predicting the placement stability and the close-loop object placement with a visual servo.

In our proposed method, a TCF is used to hold the dropped object and constrain its final configuration. We first find the SPPs on the tray corner considering the geometric constraints at the contact. Then, we use analytical and learning methods to obtain the DDPs of the objects that lead to the found SPPs.
We compare the performances of the different estimation methods to understand the advantages and disadvantages.

C. Regrasp

Regrasp is a manipulation method for robots to reorient the grasped object. It refers to an intermediary fixture to help the robot overcome both robot kinematic constraints and environmental geometric interference. A robotic system, HANDEVY [63], was designed with the regrasp ability based on the algorithm presented in [64]. In some early work [65], [66] [67], the researchers decoupled the planning problem from the robot motion and concentrated on the regrasp sequences and online computation efficacy. More recent research paid attention to finding the optimal path, which combines the robot motion with regrasp [68], [69]. Wan et al. [70] presented to use a relational database to manage the grasp data, which enabled reusing more than 10,000 grasps and their relationship to search the regrasp path. In [71], the dynamic regrasp graph was presented, and the planning method considering regrasping was extended to assembly tasks. Additionally, researchers also explored employing end-to-end learning methods for regrasping. Balaguer et al. [72] proposed to achieve the dual-arm regrasp using a single stereo image as input. Gualtieri et al. [73] modeled the pick-and-place as a reinforcement learning problem and used it for regrasping objects with unknown geometries.

In the most up-to-date literature, researchers proposed learning handover while considering receiving grasping poses and affordance [74], [75]. The regrasps in these studies were from the robot or human side of human-robot interaction. The above end-to-end methods showed impressive flexibility. However, they focus less on precision, which limits their application in robotic assembly tasks.

In this work, we primarily focus on a tetrahedral intermediary fixture and use it to reduce uncertainty. Previously, Wan et al. [37] compared the efficacy of using a flat surface and a tilting surface in a work cell as the intermediary fixture [76]. Cao et al. [35] presented to leverage a support pin as the intermediary fixture to provide stable placement. Ma et al. [38] discussed regrasping objects using general structures. The comparison of using different intermediary fixtures was presented in [77]. The previous studies mainly concentrated on optimizing the motion sequence or improving the feasibility of orienting. Unlike them, we in this paper use regrasp to eliminate uncertainty and perform precise tasks. A TCF is used as the intermediary fixture considering that its crossing surfaces and tetrahedral shape can provide geometric constraints to align the dropped object. Together with the TCF, a dropping-regrasp process is implemented, and the estimation of robust dropping poses is studied.

III. PRELIMINARIES AND METHOD OVERVIEW

This section explains the background knowledge of a TCF, and presents the outline of the proposed method.

A. Background Knowledge of a TCF

A Triangular Corner Fixture (TCF) is made of three inclined plates (walls) intersecting at one bottom point, as shown in Fig. 2(a). The goal of the fixture is to hold an object like a cage [78] instead of immobilization [79], [80]. The three walls of the fixture form a gravitational basket [81] that holds a dropped object at a stable pose. The inclined walls can always pull a dropped object into configurations with minimal potential energy in the gravitational field.

Especially, the walls of the TCF used in this paper are three congruent isosceles right triangles and are installed in a mutually perpendicular configuration. The angles between the triangular walls and a horizontal plane are the same and equal to $54.74^\circ$. The reason we study this special TCF is that our goal is to assemble mechanical workpieces precisely. Although these workpieces have different shapes, they comprise geometric primitives like a cylinder, cuboid, ball, wedge, etc., and have three mutually perpendicular surfaces. We thus propose using a TCF made of three mutually perpendicular plates as an intermediary fixture to hold them. Figs. 2(b) shows a real-world fabrication. The three triangular walls are acrylic and detachable from the base. We use the notation $l_e$ to denote the isosceles length of the triangular walls. Since the walls are detachable, the $l_e$ can be changed and the TCF dimensions can be adapted for parts of different scales. We use an abbreviation, TCF-$l_e$, to denote a specific TCF. For instance, TCF-100 represents the TCF with 100.0 mm isosceles inner edge length. The TCF base is mounted on a 3-axes rotational platform for fine adjustment. The platform bottom has an adapter plate for connecting with other fixtures.\footnote{It should be noted that: (1) There are many other choices and our methods may or may not adapt to them; (2) Instead of the particular TCF, one may design an optimal fixture for a set of objects considering geometrical morphology [82]. Interested readers are encouraged to see parts C, D, and E of Section VIII for further discussions.}

B. Method Overview

We develop a planner that estimates the robust dropping poses and stable placements of an object in the mentioned TCF and hence finds the regrasp motion that leads to precise assembly. Fig. 3 shows the workflow of our proposed planner. It receives the meshed models of the target object, the robotic gripper, and the TCF as the input. Three sub-modules will use the input: the Stable Placement Pose (SPP) planning sub-module, the Deterministic Dropping Pose (DDP) estimation sub-module, and the grasp configuration planning sub-module, respectively, to find stable placement poses, estimate robust
dropping poses, and plan gripper configurations. Specifically, the SPP sub-module computes a set of stable placement candidates of a given object that satisfies geometry constraints and is statically stable in the TCF. The DDP estimation sub-module uses a classifier to predict if an object dropped from an elevation position can be aligned to the expected SPP and finds a set of SPP-DDP pairs. The grasp configuration planning sub-module computes the gripper configurations for releasing an object at the DDPs and regrasping the object at the SPPs in the found SPP-DDP pairs. These computed releasing and regrasping gripper configurations are used to build a regrasp graph and reason a robot motion sequence. The details of the three sub-modules will be explained in the following Sections IV-VI, respectively.

IV. PLAN STABLE PLACEMENT POSES

We define an SPP as follows: An object is at an SPP when it stays in the triangular corner fixture in a balanced static condition. We use the algorithm shown in Fig. 4 to plan the SPPs. The algorithm receives an object and a TCF model as input and returns all satisfying SPPs as output. It comprises three steps which are highlighted using diamonds and blocks. The workflow is as follows. With all segmented facets, we combine every three surfaces into a triplet and get a collection of triplets $S^3_p = \{S_p(j) = \{s_a, s_b, s_c | s_a, s_b, s_c \in S_p\}, j = 1, 2, \ldots, C^3_m\}$. Then, we examine the orthogonality of each triplet’s facet normal. The one that has three mutual orthogonal normals is considered as a feasible candidate, as illustrated in Fig. 5(c). We use $S'_p = \{S'_p(i) = \{s_a, s_b, s_c | (s_a \perp s_b, s_a \perp s_c, s_b \perp s_c)\}, S'_p(i) \in S'_p\}$ to denote the feasible candidate collection. The workflow can be accelerated using linear programming to avoid repeatedly examining the impossible combinations.

A. Step 1: Facets and Their Triplet Combinations

In this step, the algorithm clusters triangle faces of an object’s mesh model $M_o$ into facets and then uses the facets to find mutually perpendicular triplet combinations. Using conventional segmentation methods to cluster facets may lead to uneven area [83]. Instead of the conventional methods, we use superimposed segmentation [84] to generate uniform facets. The method especially has better performance when handling curved surfaces. Take the T-shape pipe junction object shown in Fig. 5 for example. The mesh model of the junction is shown in Fig. 5(a). The segmented superimposed facets are shown in Fig. 5(b). For a mesh model $M_o$, we denote its superimposed facet set using $S_o = \{s_i\} (i = 1, 2, \ldots, m)$, where each $s_i$ indicates a facet.

After getting the facets, we find the mutually perpendicular triplet facet combinations. Here, we assume to only consider the face-to-face contacts between an object and the TCF, and thus ignore the edge and vertex contact. The details of our workflow is as follows. With all segmented facets, we combine every three surfaces into a triplet and get a collection of triplets $S^3_p = \{S_p(j) = \{s_a, s_b, s_c | s_a, s_b, s_c \in S_p\}, j = 1, 2, \ldots, C^3_m\}$. Then, we examine the orthogonality of each triplet’s facet normal. The one that has three mutual orthogonal normals is considered as a feasible candidate, as illustrated in Fig. 5(c). We use $S'_p = \{S'_p(i) = \{s_a, s_b, s_c | (s_a \perp s_b, s_a \perp s_c, s_b \perp s_c)\}, S'_p(i) \in S'_p\}$ to denote the feasible candidate collection. The workflow can be accelerated using linear programming to avoid repeatedly examining the impossible combinations.

B. Step 2: Computing Transformations

In the second step, the algorithm computes the transformation that fits the triplet facets onto the inner surfaces of the TCF. We use $\{C_o\}$ and $\{C_f\}$ to respectively represent the object frame and the TCF frame, and use $\{C'_p\}$ to denote the
local frame of \( S_{f}'(i) \). The intersection point of \( S_{f}'(i) \)’s three orthogonal facets are selected as \( \{C_i\}_j \)’s origin. Its coordinate axes are determined considering the inverted normal directions of the facets (The exact x, y, and z choices are free, as long as they meet the right-hand rule). Fig. 5(d) illustrates a \( \{C_o\}_j \) defined considering the \( S_{f}'(i) \) shown in Fig. 5(c).

Next, we compute the placement pose of an object by transforming its \( \{C_o\}_j \) onto TCF. We define two coordinate systems for the TCF. One is \( \{C_f\}_j \). Its origin is at the bottom point of the TCF, and its orientation is the same as the world coordinate system. The other one is \( \{C_f\}_j \), which has the same origin as \( \{C_f\}_j \) but the x, y, z axes are along the intersection edges of the TCF’s perpendicular surfaces. The placement poses of the object can be computed by superposing \( \{C_o\}_j \) to \( \{C_f\}_j \), which means if we use a transformation matrix \( C_f/T_i \) to denote the placement pose, it can be computed as \( C_o/T_i = C_f/T_i C_o/C_f \). An object may have many \( S_{f}'(i) \) and thus many \( C_f/T_i \). We name \( C_f/T_i \)’s the Potential Placement Poses (PPPs). Fig. 5(f) illustrates one PPP of the T-junction object.

Note that the potential \( C_f/T_i \) may not be logically feasible since we did not check interference and contact. The object may penetrate the TCF, as shown in Fig. 5(g). It may also be floating in the air as the size of the TCF is limited and the contact is phantom (Fig. 5(h)). Thus, at the end of the second step, we screen the PPPs by detecting collisions and the existence of contact and removing the logically infeasible ones. We get a set of Logically Feasible Poses (LFPs) after the screening, as illustrated by Fig. 5(i).

C. Step 3: Examining the Static Stability

In the third step, we further use Contact Wrench Space (CWS) analysis to examine the static stability of the LFPs and obtain the SPPs. For a clear illustration, we use an L-shape object instead of the T-junction to exemplify this step. The workflow is as follows.

First, we extract the contact polygons between the object and the TCF’s inner surfaces. Each of the three TCF inner surfaces has a contact polygon set, which may have a single or multiple elements. We compute the convex hull of the contact polygons in each set to get three support polygons for the three inner surfaces. The SP1-3 in Fig. 6(a) illustrate the three support polygons of the L-shape object. Second, we consider the vertices of the three support polygons as the effective contact points that provide supporting forces for the object, compute a wrench cone formed by the wrenches exerted on them and the object’s gravity, and judge the stability of the object using the relation between the wrench cone and the origin of the wrench space. The yellow spheres in Fig. 6(a) illustrate the effective contact points. Assume there are in total \( k \) effective contact points \( p_i = \{x_i, y_i, z_i\} \) (i = 1, 2, ... , k). We build a local frame at each of the \( p_i \)s to describe the contact force. The x and y axes of the local frame compose a tangent plane on the contact point, and the z axis aligns with the normal of the TCF’s inner surfaces, as shown in Fig. 6(d). The contact force at \( p_i \) can be represented by the components along the three axes as \( f_i = \{f_{xi}, f_{yi}, f_{zi}\}^T \). The effect wrench exerted on \( p_i \) can be computed using \( w_i = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} f_i \end{bmatrix}_{\tau_i} \), where \( \tau_i \) indicates the exerted torque. Considering the frictional constraints at the contact point, \( f_i \) must be in a friction cone and \( f_{xi}, f_{yi}, \) and \( f_{zi} \) must meet \( \sqrt{f_{xi}^2 + f_{zi}^2} \leq \mu f_{yi} \), where \( \mu \) is the friction coefficient. Since \( f_i \) is inside a cone and is not deterministic, directly using the equation to compute the wrench cone is difficult. To overcome the difficulty, we approximate the friction cone with a pyramid [85], as shown in Fig. 6(b.2-3). The lateral edges of the pyramid represent the extreme \( f_i \) choices. They are named as \( f_i' \), where the granularity of the approximation determines \( j \). A linear combination of the \( f_i' \) could approximate a freely chosen \( f_i \) in the friction cone.

With this consideration, we represent \( w_i \) using multiple values \( w_i = \{w_i^j\} \) and use all \( w_i^j \) to compute the wrench cone. Each \( w_i^j \) will be the wrench from one \( f_i' \). Considering all of them for wrench cones is the same as considering a linear combination of the \( f_i' \). The wrench set \( W \) born by the object comprises the \( \{w_i^j\} \) at every \( p_i \) and the object’s center of mass. It can be expressed as \( W = \{w_i^j\} \cup \cdots \cup \{w_i^j\} \cup \{w_k\} \), where \( w_k \) denotes the gravitational wrench. The wrench cone \( W_i \) spanned by the wrench is essentially a convex hull of the elements in \( W \) [43]. The stability of LFPs is judged by examining the relationship between the origin of the wrench space and \( W_i \). In [43], the origin is inside the \( W_i \) of an LFP, the LFP is considered to be stable.

Fig. 5. (a) A raw meshed model. (b) Segmented surfaces. (c) A candidate triplet of facets. It has three mutual orthogonal surfaces. (d) The object’s frame \( \{C_o\}_j \) and the object-to-TCF transformation coordinate described in it \( \{C_f\}_j \). (e) The TCF’s frame \( \{C_f\}_j \) and the object-to-TCF transformation coordinate described in it \( \{C_r\}_j \). (f) A placement pose of the object on the TCF. (g) The first failure case: Penetration. (h) The second failure case: Phantom contact. (i) All planned LFPs.
ranges from LFPs and find a set of SPPs (and will be counted as an SPP). The planner will look over all the SPPs obtained in the last section have correspondent DDPs. We choose a random height in \( h \), as shown in Fig. 7. We elevate an SPP to \( h \) and choose \( h^+ \) to be outside \( d_f \) to take into account various possibilities.

V. ESTIMATE DETERMINISTIC DROPPING POSES

If an object released from a pose on top of the TCF has a deterministic and expectable SPP when it gets stabilized inside the TCF, we call the releasing pose a Deterministic Dropping Pose (DDP). This section presents methods to estimate if the TCF, we call the releasing pose a Deterministic Dropping Pose (DDP). This section presents methods to estimate if the object deterministically stabilizes at the SPP after being dropped from the releasing pose. If the algorithms suggest a positive predicted result, we save the SPP and the releasing pose as an SPP-DDP pair. All saved SPP-DDP pairs will be used for reasoning and planning the regrasp sequences to improve grasping precision.

Specifically, we propose two classes of methods, an analytical and a learning-based, for the estimation.

A. Analytical Methods

In the first class of methods, we predict the SPP by considering a static stability criterion and screen an SPP considering its capability of resisting external disturbance wrenches. The method is based on an assumption that an SPP with larger stability is more likely to have a DDP than a less stable one.

We have three candidate implementations for methods in this class. First, similar to the third step of planning SPPs, we use CWS to evaluate static stability. However, instead of directly generating the convex hull of contact wrenches in the wrench space, we construct the wrench cone by computing the convex hull of \( W \)’s Minkowski sum. We use the notation \( W_{mkw} \) to differ the wrench cone in this section from the \( W \) used before. Compared to \( W \), \( W_{mkw} \) can quantify the resistible external wrench, thus make it easier to decide an evaluation criteria [43]. In particular, we compute the shortest distance from the origin of the wrench space to the hyperplanes that constitute \( W_{mkw} \) and use the shortest distance as the stability index. Then, we the SPPs that have enough stability quality from the obtained SPP set and elevate them to get DDPs. The second implementation is a “projection method”. It uses the shortest distance from the projected CoM to the boundary of the support surfaces as the stability index. The third implementation is an optimization-based method published in [86]. It computes the minimal external disturbance that breaks the balance and uses the magnitude of the minimal disturbance as the stability index.

The analytical methods may find the DDPs with significant determinism from the SPPs of an object. However, the stability quality of different objects cannot be measured on a unified scale, making it difficult to set a unique threshold for a general estimation. Also, the DDPs may have uncertainty (positional and rotational noises and bouncing) in the real world, which is challenging to consider analytically. For these reasons, more advanced methods need to be explored.

B. Learning-Based Methods

In the second class of methods, we use machine learning to predict DDPs. Especially, we use a sim-to-real method [87] to obtain the training data and train different classifiers to judge if an SPP has a correspondent DDP.

1) Training Data: The training data comprises a data section and a label section. The data part comprises the contact polygons, the position of the object’s CoM, and the support surfaces of the TCF. They are projected onto a horizontal plane and formulated as a 2D grayscale image shown in Fig. 8 to simplify numerical computation. In detail, we assume a grayscale image with 224 \( \times \) 224 pixels. The background of the image is white (grayscale value: 255). The regions of the projected support polygons and the contact polygons are set to 220 (support surfaces) and 0 (contact polygons). The projected CoM is formulated as a circular patch. Its color is computed using \( v_{gray} = \phi \cdot h_{com} \), where \( \phi \) is the ratio between a real-world distance and the numbers of image pixels used to represent it, \( h_{com} \) indicates the vertical distance from the CoM to the bottom point of a TCF. The \( v_{gray} \) essentially normalizes \( h_{com} \) considering the dimensions of the TCF and the image.

The label section is collected by physical simulation. We place a work table and a TCF in simulation and generate the candidate releasing poses by randomly elevating an object from their SPPs, dropping the object from the candidate poses, and examining the finally stabilized poses. Unlike the analytical method, we add noises to the releasing poses to simulate uncertainty. The object falls from the releasing poses with random noises, and we compare the object’s stabilizing CoM with the CoM of the expected SPP when...
it gets stabilized. If the two CoMs coincide, a successful trial is recorded. Otherwise, a failure is recorded. Here we use the CoM as the reference to avoid misjudging symmetric objects with small support surfaces (e.g., balls and cylinders). The configurations of these symmetric objects are considered identical when moving around the symmetry center. By comparing the CoMs instead of the configurations, we may avoid misjudging the identical configurations. We run 100 trials for each releasing pose and compute a success rate. If the success rate is more significant than a given threshold, the releasing pose will be labeled as a positive sample.

2) Classifiers: Using the training data collected in the last section, we train classifiers to predict if the object dropped from a releasing pose can rest at an expected SPP. The classification is a simple binary one since there are only two labels. Various methods like Support Vector Machine (SVM), Fully Connected Network (FCN), and Convolutional Neural Network (CNN) can be used to model the classifier. Specifically, we implement and compare a linear SVM, a four-layer FCN, an AlexNet, and a ResNet50. The detailed results and discussions about the implementation and comparison will be presented in Section VII.A.

VI. PLAN GRASP CONFIGURATIONS AND REGRASP SEQUENCES

This section presents detailed releasing and regrasp planning algorithms for adjusting grasping precision. The algorithms are based on our previous work published in [84] and [88]. First, we plan grasps configurations for an object without considering any obstacles. The methods presented in [84] are directly applied. Then, based on the planned grasp configurations, we generate two sets of grasp configurations for the SPP and DDP in each SPP-DDP pair while considering different levels of collisions. Finally, we build a regrasp graph by reasoning and connecting the grasp sets associated with all SPP-DDP pairs, and search the graph to obtain regrasp sequences. We make a minor change to the original algorithm published in [88] when building the regrasp graph as the concepts of SPPs and DDPs were not available in the previous implementation: We additionally define SPP subgraphs and DDP subgraphs and connected the corresponding nodes and edges to the regrasp graph. The influences of target shapes, surrounding environments, and task goals are automatically considered, and the regrasp sequences obtained by searching the graph are optimized towards these constraints.

Fig. 9 and 10 exemplify the above workflow using the L-shape object. Fig. 9(a) shows the planned grasp configurations when there are no surrounding obstacles and the object pose is aligned with the global frame. Fig. 9(b-c) show an SPP and its associated grasp configuration set. These grasps in the set are transformed from (a) along with the object pose. The grasp configurations that collide with the TCF after the transformation are removed. Fig. 9(d-j) show the DDP paired with the SPP and the procedure for generating its associated grasp configuration set. Fig. 9(d) is the DDP. Fig. 9(e) is the grasps transformed from (a), with the ones in collision with the TCF removed. Fig. 9(f) shows the swept volume of the released object. The grasps in (e) are further examined considering the swept volume. If an opening hand collides with the swept volume, the released object will collide with the hand when it falls onto the TCF, leading to an unexpected resting pose. Thus, we further examine the collision between the grasp configurations in (e) and the swept volume, and remove the collided ones. Fig. 9(g.1-2) and 9(h.1-2) show a collision-free and a collided examples respectively. The grasp configuration in Fig. 9(g.1) does not collide with the swept volume after releasing in Fig. 9(g.2). Contrarily, the grasp configuration in Fig. 9(h.1) get collided in Fig. 9(h.2). Fig. 9(i) highlights all collided grasp configurations in (e) with red color. Fig. 9(j) shows the remaining collision-free grasps.

Fig. 10 shows the regrasp graph built using the two grasp configuration sets in Fig. 9(c) and (j). The black maximally connected graphs in Fig. 10(a) and (d) show the transit relations among the grasp configurations associated with the initial and goal object poses. The black maximally
connected graphs in Fig. 10(b) and (c) show the transit relations among the grasp configurations associated with the DDPs and their pairing SPPs. The blue edges among the maximally connected graphs show the transfer relations among the grasp configurations associated with the DDPs and their pairing SPPs. (a) Initial pose and its subgraph. Each circle indicates an object pose. The nodes inside the circle are the grasp configurations associated with the corresponding pose. First, the nodes in (a), (b), (c), and (d) are connected separately to represent transit relations. Second, the shared grasp configurations between (a) and (b), and between (c) and (d) are connected for transfer relations. Third, the nodes in (b) and (c) are connected to represent transit relations between DDPs and SPPs.

VII. EXPERIMENTS AND ANALYSIS

This section includes three parts. First, we compare the methods for estimating the DDP-SPP pairs. Second, we use the most satisfying method to perform regrasp and examine the regrasp precision. Third, we validate the benefits of the proposed method using real-world assembly tasks.

A. Comparison of the DDP Estimation Methods

1) Simulation Results With Random Objects: We proposed three analytical and four learning-based methods in Section V) for estimating the DDP. In this subsection, we compare their performance using physical simulation. For the learning-based methods, we used the 14 primitives shown in Fig. 11(a) to obtain the training data. The primitives were scaled from 50% to 150% with 10% granularity, as shown in Fig. 11(b), to obtain 154 objects. The primitives and their scaled shapes were designed considering our expected target categories. Using these objects, we got 4464 SPPs. We collected training data using these SPPs in a PyBullet-based physical simulator. According to the real-world model, the friction coefficient and the bounce rate between the object and the TCF in the simulator were set to 0.3 and 0.2, respectively. We assumed a TCF with a 50.0 mm inner edge length (TCF-50) and a 28.0 mm depth ($d_f = 28.0$ mm), and collected the training data by repeatedly elevating the objects from the SPPs to random start positions between $h^- = 0.8d_f$ and $h^+ = 1.5d_f$ with maximally 3.0 mm positional and 3.0° rotational noises, and dropping them from the start position. When the objects got stabilized, we compare their CoMs with that of the source SPPs and label the results. Through the physical simulation, we collected 1770 positive samples and 2674 negative samples. During training, we randomly selected 80% of the data for learning the estimators and 20% of the data for cross-validation [89]. The datasets were restrained to the given primitives.

The results using the different methods are shown in Fig. 12. Here, we used a linear kernel for the SVM. The performance of SVM relies on selecting an appropriate kernel. It is not easy to make the selection general for all objects, and we made the most primary choice: a linear kernel. We used the cross-entropy loss function, the SGD optimizer, and a learning rate of $10^{-4}$ to train the FCN and AlexNet. Each training batch size was 16. The whole training process was carried out for 50 epochs. To train the Resnet50, we used the cross-entropy loss function and the Adam optimizer. We kept the batch size to 16 but reduced the number of epochs to 20 to avoid over-fitting. The results imply that the learning methods had better performance. The ResNet exhibited the highest success rate (90.7%). The AlexNet also exhibited a competitive performance (90.1%). The CWS had poor performance because “finding the SPPs with enough stability quality” needed a threshold. For practical purposes, we only used the most stable configuration, which easily leads to ignored DDPs. Note that even if one configuration has the most stable stability, there is no guarantee that its elevated counterpart is a DDP. The DDPs found by the method are not necessarily robust.

2) Real-World Results: Besides the simulated data, we also validated the various methods using four real objects shown in Fig. 13. They included: (a) an L-shape object; (b) a T-junction; (c) a bracket; (d) a bearing housing. We used TCF-50 for the L-shape object and bearing housing, and used TCF-71 for the bracket and the T-junction. The reason we used TCFs with different sizes and how we determined them could be found...
in the discussion section (Section VIII.E). For the learning-based methods, we used the classifiers trained above to judge DDP-SPP pairs. For the analytical methods, we used the most stable configuration. The results are shown in Table I. The table’s ground truth values (Denominators of the “Unpaired” and “DDP-SPP” columns) were obtained by repeated physical simulation. There were 54 SPPs for the L-shape object. 6 of them had DDP-SPP pairs, as shown by the denominators of the L-shape object’s “DDP-SPP” column. The remaining 48 did not have counterpart DDps, as shown by the denominators of the L-shape object’s “Unpaired” column. The numerators of the “Unpaired” column show the actual number of SPPs that did not have a DDP. The numerators of the “DDP-SPP” column show the actual number of SPPs that had a DDP.

The results show that the three analytical methods had similar performance. They worked effectively for the L-shape object and the bracket, but provided poor estimation for the T-junction object and the bearing housing. Especially, although they exhibited high success rates in estimating the unpaired candidates, they performed poorly in finding DDP-SPP pairs. The results are the same as our expectations. The learning-based methods are better on average but may have shortages for specific objects (i.e., the bracket). The AlexNet method has the best performance on the L-shape object and T-junction. The ResNet50 shows advantages in the Bracket and Bearing housing. The results are consistent with Fig. 12.

We also carried out experiments to particularly examine individual estimation performance. First, we placed the object with a selected SPP on the TCF. Then, the robot grasped the object, elevated it to a DDP with random offset noises, and opened the gripper to release the object. We observed the dropping process, checked if the object gets stabilized at the selected SPP, and recorded the results. The process was repeated 30 times for each SPP to get a statistical view. Fig. 14 shows the results. It is impossible to show all SPPs due to page limits, and we only present some representative DDP-SPP results for readers’ convenience. The results show that the estimation mostly accords with the real world.

Since the analytical method had an extremely bad performance on the T-junction object and the bearing housing, we further analyzed the detailed contact between these objects and the TCF surfaces to understand the reason. We found that

| Method       | L-shape object | T-junction | Bracket | Bearing housing |
|--------------|----------------|------------|---------|----------------|
|              | Unpaired       | DDP-SPP    | Unpaired | DDP-SPP        |
| CWS          | 48/48          | 6/6        | 60/60   | 3/12           |
| Projection   | 48/48          | 6/6        | 60/60   | 3/12           |
| Chen et al.  | 48/48          | 4/6        | 60/60   | 1/12           |
| SVM          | 48/48          | 6/6        | 30/60   | 6/6            |
| FCN          | 48/48          | 6/6        | 30/60   | 6/6            |
| AlexNet      | 48/48          | 6/6        | 30/60   | 6/6            |
| ResNet50     | 48/48          | 6/6        | 30/60   | 6/6            |

*The Unpaired and DDP-SPP columns are presented in a fraction style. The denominator values indicate the ground truth obtained using repeated physical simulation. The numerator values indicate the estimated results.*
Fig. 14. The results of testing the estimator using different objects. (a) L-shape object. (b) T-junction. (c) bracket. (d) bearing housing. The SPPs of each object are selectively shown. The left column shows the real-world photos, the middle column shows the placements in the simulator, and the right column illustrates the projected images.

the DDP-SPP pairs with the best SPP stability were like the ones shown in Fig. 12(b.3) and (d.2). These SPPs had high static stability qualities, but their contact areas were distributed around the objects’ CoMs (as shown by the 2D grayscale images in the figure). A large section of an object was not in contact with the inner surface of the TCF. The object thus had a low chance to stably “stand” on the distributed contact when being dropped. It was easy to get stuck by the edges of the TCF.

B. Performance on Eliminating Uncertainty

In this subsection, we analyze the uncertainty sources and carry out two experiments to study the performance of our method for eliminating uncertainty.

An assembly sequence found based on our method comprises three steps: Object pose recognition, grasp and motion planning, and real-world execution. Uncertainty may happen in each of the steps as follows. In the first step, a depth sensor captures the point clouds of the target object and the surrounding environment. The recognition algorithm clusters the point clouds, transforms them to the robot frame, and registers the object pose by matching its model with the processed point clouds. The uncertainty in this step depends on sensor measuring qualities, robot-sensor calibration algorithms, and pose recognition algorithms. For our system, we used the Phoxi 3D Scanner M (Photoneo S.R.O.) depth sensor for capturing point clouds. The sensor has a 0.1 mm measuring uncertainty. Its calibration uncertainty with the robot coordinate system is affected by multiple factors like the calibration board’s fabrication precision, board installation precision, etc. The calibration uncertainty is much larger than the measuring quality. Its value is coupled with pose recognition uncertainty to influence final pose estimation results. After discussing the pose recognition uncertainty, we will show their total maximal values. We used DBSCAN and Iterative Closest Point (ICP) algorithms for object pose recognition. The selected convergence criteria of these algorithms determine the algorithm precision and thus recognition uncertainty. There could be a total ±3.25 mm uncertainty at maximum (experimentally measured) coupled with calibration. In the second step, the grasp planner finds grasp candidates for holding an object at the initial and goal poses, and the motion planner generates trajectories for moving the held object. The uncertainty in this part originates from the modeling error in the simulation, which is about ±0.72 mm (experimentally measured). In the third step, the robot moves following the planned trajectories and gripper actions. The robot control accuracy determines the uncertainty in this step. The UR3e robot arm used in our work has a ±0.03 mm control accuracy. Thus, the uncertainty in the third step is also ±0.03 mm.
In summary, there are five uncertainty sources in our study: (i) Sensor, (ii) sensor-robot calibration, (iii) object recognition, (iv) modeling in simulation, and (v) robot control. The factors related to the vision system are the most problematic. The total uncertainty could reach as much as ±4.0 mm.

We conducted two experiments to analyze the proposed method’s performance in eliminating the above uncertainty. In the first experiment, we recorded the goal position of an object moved by the robot, and observed the deviations of the recorded results to understand the reducible uncertainty. In the second experiment, we performed robotic peg-in-hole insertion tasks, and observed the affordable minimal shaft-to-hole clearance to understand the method’s usefulness.

In either experiment, we considered a conventional method and two varied implementations of the proposed method. They are shown in Fig. 15. All of them assume position control. The conventional method directly plans to move a picked object to a goal pose, as shown in Fig. 15(a). It is abbreviated as DPM in the following context. For our proposed method, we build and search a regrasp graph to find a regrasp sequence. We have two particular implementations for it: Regrasp with All Grasps (RAG) and Regrasp with a Prescribed Grasp (RPG). In the RAG implementation, all grasp configurations for the goal pose and SPP are considered when building the regrasp graph, as shown in Fig. 15(b). The built graph is exactly the same as Fig. 10. In the RPG implementation, a goal SPP and a goal grasp configuration are prescribed manually, as shown in Fig. 15(c). The motion between the prescribed goal SPP and the goal pose using the prescribed grasp configuration is taught instead of planned. The built graph is different from Fig. 10 due to the prescribed data: The connections in Fig. 10(c) and Fig. 10(d) are replaced with a given path and the goal pose subgraph collapses to a single node. Our expectations for them are as follows: The DPM implementation will be influenced by all uncertainties, as the TCF is not used to reduce uncertainty; The RAG implementation will reduce the uncertainties from sensors, sensor-robot calibration, and object recognition. The RPG implementation will further reduce modeling uncertainty.

1) Goal Deviation Measurements: In this experiment, we spread a grid paper on the work table to record the final position of a shaft (diameter: 10.0 mm) and asked the robot to grasp the shaft from different initial poses, move it to a defined goal pose above the grid paper with different methods and implementations, and lower it down to touch the paper. The setups of our experiments are shown in Fig. 16(a.1). When the shaft touched the paper, we drew the contact profile of the shaft on the grid paper to record the result. With multiple initial shaft poses, we could obtain a collection of goal profiles to understand the uncertainty range. The results with the three different methods/implementations mentioned in the last part are shown in Fig. 16(b). Here, the manually recorded goal positions are redrawn using computer graphics software for better visual effects. The yellow circles show all the recorded goal positions. The red circles are the minimum circumscription. Fig. 16(b.3) is the result of the RPG implementation. Its uncertainty was minimal and was not observable using the proposed method (less than ±0.03 mm, the robot control accuracy). Considering the maximum uncertainty was ±4.0 mm, the RAG and RPG implementations could reduce the uncertainty by ±3.25 mm and ±3.97 mm, respectively. The results were the same as our expectations about the methods/implementations.

2) Peg-in-Hole Insertion: In this experiment, we prepared a series of holes by cutting an acrylic board with a laser cutter. The holes had different clearance, as shown in Fig. 17(a). The diameters of the holes ranged from 10.1 mm to 18.0 mm. The length and diameter of the peg were respectively 75.0 mm and 4.0 mm. The results were ±3.97 mm, the robot control accuracy). Considering the maximum uncertainty was ±4.0 mm, the RAG and RPG implementations could reduce the uncertainty by ±3.25 mm and ±3.97 mm, respectively. The results were the same as our expectations about the methods/implementations.

In summary, there are five uncertainty sources in our study: (i) Sensor, (ii) sensor-robot calibration, (iii) object recognition, (iv) modeling in simulation, and (v) robot control. The factors related to the vision system are the most problematic. The total uncertainty could reach as much as ±4.0 mm.

We conducted two experiments to analyze the proposed method’s performance in eliminating the above uncertainty. In the first experiment, we recorded the goal position of an object moved by the robot, and observed the deviations of the recorded results to understand the reducible uncertainty. In the second experiment, we performed robotic peg-in-hole insertion tasks, and observed the affordable minimal shaft-to-hole clearance to understand the method’s usefulness.

In either experiment, we considered a conventional method and two varied implementations of the proposed method. They are shown in Fig. 15. All of them assume position control. The conventional method directly plans to move a picked object to a goal pose, as shown in Fig. 15(a). It is abbreviated as DPM in the following context. For our proposed method, we build and search a regrasp graph to find a regrasp sequence. We have two particular implementations for it: Regrasp with All Grasps (RAG) and Regrasp with a Prescribed Grasp (RPG). In the RAG implementation, all grasp configurations for the goal pose and SPP are considered when building the regrasp graph, as shown in Fig. 15(b). The built graph is exactly the same as Fig. 10. In the RPG implementation, a goal SPP and a goal grasp configuration are prescribed manually, as shown in Fig. 15(c). The motion between the prescribed goal SPP and the goal pose using the prescribed grasp configuration is taught instead of planned. The built graph is different from Fig. 10 due to the prescribed data: The connections in Fig. 10(c) and Fig. 10(d) are replaced with a given path and the goal pose subgraph collapses to a single node. Our expectations for them are as follows: The DPM implementation will be influenced by all uncertainties, as the TCF is not used to reduce uncertainty; The RAG implementation will reduce the uncertainties from sensors, sensor-robot calibration, and object recognition. The RPG implementation will further reduce modeling uncertainty.

1) Goal Deviation Measurements: In this experiment, we spread a grid paper on the work table to record the final position of a shaft (diameter: 10.0 mm) and asked the robot to grasp the shaft from different initial poses, move it to a defined goal pose above the grid paper with different methods and implementations, and lower it down to touch the paper. The setups of our experiments are shown in Fig. 16(a.1). When the shaft touched the paper, we drew the contact profile of the shaft on the grid paper to record the result. With multiple initial shaft poses, we could obtain a collection of goal profiles to understand the uncertainty range. The results with the three different methods/implementations mentioned in the last part are shown in Fig. 16(b). Here, the manually recorded goal positions are redrawn using computer graphics software for better visual effects. The yellow circles show all the recorded goal positions. The red circles are the minimum circumscription. Fig. 16(b.3) is the result of the RPG implementation. Its uncertainty was minimal and was not observable using the proposed method (less than ±0.03 mm, the robot control accuracy). Considering the maximum uncertainty was ±4.0 mm, the RAG and RPG implementations could reduce the uncertainty by ±3.25 mm and ±3.97 mm, respectively. The results were the same as our expectations about the methods/implementations.

2) Peg-in-Hole Insertion: In this experiment, we prepared a series of holes by cutting an acrylic board with a laser cutter. The holes had different clearance, as shown in Fig. 17(a). The diameters of the holes ranged from 10.1 mm to 18.0 mm. The length and diameter of the peg were respectively 75.0 mm and 4.0 mm. The results were ±3.97 mm, the robot control accuracy). Considering the maximum uncertainty was ±4.0 mm, the RAG and RPG implementations could reduce the uncertainty by ±3.25 mm and ±3.97 mm, respectively. The results were the same as our expectations about the methods/implementations.
10.0 mm. The clearance of the peg-in-hole tasks (difference between the peg and hole diameters) thus ranged from 0.1 mm to 8.0 mm. We ran the insertion for each hole by 15 repetitions with each of the three methods/implementations to obtain average success rates and got the methods’ performance by observing the smallest clearance with 100% success rate. In each repetition, we placed the object in a random initial position on a table. A robot detected it using the depth sensor and moved it to a common goal pose with or without regrasp using the TCF. At the goal pose, the robot inserted the peg by moving along a straight line with position control. The results are summarized in Fig. 17 using a bar chart. The horizontal axis of the chart is the different hole diameters. The vertical axis is the average success rate. The results showed that the minimally affordable clearance of the DPM, RAG, and RPG methods were 7.0 mm, 1.0 mm, and 0.1 mm, respectively. Compared with the goal deviation measurements in the first experiment, the robot could successfully insert the shaft even if the expected deviation was larger than the hole clearance. The reason was that the chamfer of the shaft helped to partially afford the uncertainty.

C. Time Costs

Our search for DDPs and SPPs was carried out in a brute force style. The time complexity was \( O(n^2) \) where \( n \) denoted the number of planar segments on a mesh surface. Despite the cubic asymptotic time complexity, the method met our requirements because we assumed industrial workpieces as the target objects. These workpieces tended to have a small number of planar surface segments and could be solved in a practically limited time. This subsection demonstrates the claimed practicality by analyzing the time costs of the method applied to the four objects used in our real-world experiments and discusses possible improvements based on the results. Table II shows the time costs. Each value in the table is the average success rates and got the methods’ performance by observing the smallest clearance with 100% success rate. In each repetition, we placed the object in a random initial position on a table. A robot detected it using the depth sensor and moved it to a common goal pose with or without regrasp using the TCF. At the goal pose, the robot inserted the peg by moving along a straight line with position control. The results are summarized in Fig. 17 using a bar chart. The horizontal axis of the chart is the different hole diameters. The vertical axis is the average success rate. The results showed that the minimally affordable clearance of the DPM, RAG, and RPG methods were 7.0 mm, 1.0 mm, and 0.1 mm, respectively. Compared with the goal deviation measurements in the first experiment, the robot could successfully insert the shaft even if the expected deviation was larger than the hole clearance. The reason was that the chamfer of the shaft helped to partially afford the uncertainty.

D. Performance in Practical Real-World Tasks

Finally, we test the proposed method using four real-world assembly tasks: (1) Inserting the L-shape object into a rectangular groove; (2) Sheathing the T-junction with a tube; (3) Aligning the holes of the bracket and a base plate; (4) Mounting the bearing housing on a bracket. These tasks are frequently seen at industrial manufacturing sites.

1) Inserting the L-Shape Object: In this task, we fixed an acrylic board with a rectangular groove on a table, and asked the robot to insert the L-shape object into the rectangular groove. Fig. 18(a.1) shows the sizes of the object and the groove. The clearance between them was 2.0 mm.

2) Sheathing the T-Junction: The goal of this task was to sheathe a tube into the T-junction. The tube was vertically fixed on the table, and the robot was asked to manipulate the T-junction to perform the sheathing action. Fig. 18(b.1) shows the sizes of the T-junction and tube. The maximum clearance between the inner circle of the T-junction and the outer circle of the tube was 0.3 mm.

3) Aligning the Holes: In this task, a base plate with thread holes was fixed on a table. The two through-holes on the short side of the bracket were required to be aligned with the thread holes on the base plate. If a screw bolt could be fastened in the thread holes across the through-holes, we judge the alignment to be successful. Fig. 18(c.1) shows the sizes of the bracket and

---

3A 100% average success rate means the method could always suppress the peg’s uncertainty within a range indicated by the clearance value.

4The two angles were named \( \theta_{plan} \) and \( \theta_{fcl} \) respectively in [84]. They were set to 0.99 and 0.975 radians, respectively, in this work.
the thread holes. The difference between the inner thread-hole diameter and through-hole diameter is the task’s clearance. Its value was 1.7 mm.

4) Mounting the Bearing Housing: This task required the robot to mount the bearing housing on a fixed bracket. The sizes of the bearing housing and mounting hole are shown in Fig. 18(d.1). The clearance between them was less than 0.1 mm.

Like the previous experiments, the environment model, object models, and the configuration of the TCF were pre-given and pre-calibrated. Also, the goal poses of them in the assembly tasks were known. The initial poses of the objects were random. The conventional method (DPM) and the two implementations of our method (RAG and RPG) were tested. For each of the above tasks, we ran ten times of experiments using different methods. Table III shows the experiment results. Using the DPM method, only two successful attempts were observed in inserting the L-shape object. All other tasks failed. Using the RAG method, all attempts to insert the L-shape object and align the holes succeeded, but no success was observed in the tasks of sheathing a T-junction and mounting a bearing housing. All tasks were successfully performed when the RPG method was used. The results showed that the proposed method could provide reliable and robust performance for these tasks, especially when the RPG method was used.

Fig. 18(a.2-d.2) show execution pictures of some successful results in Table III. Readers may also refer to the video supplementary attached to this manuscript to observe the detailed robotic actions.

### VIII. DISCUSSIONS

A. Weighted Regrasp Graph Instead of Thresholding

We used thresholds to judge placement poses and corresponding stability of dropping poses, and then used a non-weighted regrasp graph to plan a regrasp path considering a small number of edges. Besides the threshold-based method, one may consider building a weighted and directed regrasp graph to plan the path. Instead of thresholding, the weighted and directed regrasp graph will include all dropping poses and placements. Placements with high SPP scores or superior learning results will be associated with high weights. The planner will discover a path with large total weights to assure stability and action efficiency. The idea is advisable. In this article, however, we did not practice it since designing such a graph can be tricky and will inevitably introduce an unclear interplay between weight values and the number of edges along the path (the number of regraps). When the number of regrasps is the priority, the current non-weighted implementation stands as a handy choice.
on sensors and delicate control algorithms. Since there is a belief state [90]. Manipulation compliance may either be advantage of manipulation compliance to guide the object to the influence of gravity. An alternative solution could be taking onto the TCF, and the object would reach a stable pose under C. Using Compliant Poking Instead of Throwing

The robot regressed twice to finish the task.

The results show that our planner successfully solved 73 of the 400 tests, where 69 of them can be done using a single regasp. The remaining four required two regrasp. None of the 400 tests, where 69 of them can be done using a single regasp. The remaining four required two regrasp. None of the tasks required more than three regasp.

The proposed regasp graph and searching method could have flexible results, i.e., a long regrasp sequence that repeats the release-regrasp actions multiple times. The number of repetitions depends on task settings. It is influenced by many factors like the initial object poses, the goal object poses, the obstacles in the environment, the pose of the TCF, etc. For a simple task where the objects and TCF have shared grasp configurations, only one time of regrasp is needed. Contrarily, if the grasp configurations are highly constrained, i.e. the initial or goal object poses are near the boundary of the robot workspace, the planner may require more regrasps. We statistically measured the required number of regrasps by asking a robot to move the L-shape objects from four initial poses to one hundred random goal poses (400 tests in total). The results show that our planner successfully solved 73 of the 400 tests, where 69 of them can be done using a single regasp. The remaining four required two regrasp. None of the tests required more than three regasp.

Fig. 19 shows two examples. In Fig. 19(a), the initial and goal poses have shared grasp configurations, and the robot used a single regasp to precisely repose it. In Fig. 19(b), the initial and goal poses are near the workspace boundary, and the robot regressed twice to finish the task.

C. Using Compliant Poking Instead of Throwing

In the proposed method, we considered throwing an object onto the TCF, and the object would reach a stable pose under the influence of gravity. An alternative solution could be taking advantage of manipulation compliance to guide the object to a belief state [90]. Manipulation compliance may either be implemented actively or passively. Active compliance relies on sensors and delicate control algorithms. Since there is rich contact between the TCF and objects, using sensor data to actively judge contact states and stability is challenging and is beyond our scope. Passive compliance takes advantage of gravity and spring-dumper mechanisms or soft materials.

For our robot hand, we may attach soft tips to the fingers to implement passive compliance and align objects to the TCF by poking them using the soft tips. Then, it would be interesting to explore designs and methods to overcome the following problems that may arise: (1) Modifying the grippers considering various constraints; (2) Determining which point to poke when the object has an uncertain pose and an arbitrary mesh shape; (3) Predicting bounce and uncertainty caused by deformation and energy storage, and thus improve the releasing and regrasping accuracy in the TCF.

D. Limitations of the Proposed TCF

We studied this particular TCF because our goal was to stably hold objects for industrial assembly tasks. Although these objects have different shapes, they comprise basic geometric elements like a cylinder, a cuboid, a ball, a wedge, etc., and have three mutually perpendicular surface segments. Such properties inspired us to use a TCF made of three mutually perpendicular plates as an intermediary fixture to hold them. Meanwhile, from theoretical robotic grasping studies, we understood that at least three contact points with unparalleled contact normals were needed to form a form closure together with gravity [91], [92] and thus assure stable poses. The condition also motivated us to choose three mutually intersecting

Despite the deliberate selection, the proposed TCF has a fatal problem that it cannot eliminate the rotational uncertainty of centrosymmetric objects. Fig. 20 exemplifies such an object. The fixture and the proposed method cannot differentiate rotation angles. Also, we assumed rigid objects, and the proposed method cannot deal with deformation.

E. Other Fixtures

The proposed method can be adapted to other types of fixtures, given that the fixture comprises three supporting walls. Fig. 21 exemplifies several such fixtures. To adapt the current method to them, we need the following adjustments: (1) While planning the SPPs, our method finds the candidate poses that can match the triplet contact of an object with the three walls of the fixture. The proposed TCF’s walls are mutually perpendicular. Thus the matching condition was defined as that the normals of the triplet contact were orthogonal. For other types of three-wall fixtures, we need to define different matching conditions following the changes in the walls. After finding the candidate poses, our method will screen the found candidate poses, considering stability and collision. The screening results change with the pose of the fixture. For the proposed TCF, the screening was carried out considering a horizontal pose, as explained in Section V of the main manuscript. When a fixture is installed with
The proposed method generates collision-free grasp poses. The estimator should be retrained. (3) While planning grasp poses, the DDPs for different fixtures or installation poses, and the estimator should be retrained. (2) While planning the DDPs, the proposed method estimates the DDP-SPP pairs. The estimator is trained using simulation data. The simulation data should be recollected for different fixtures or installation poses, and the estimator should be retrained. (3) While planning grasp poses, the proposed method generates collision-free grasp poses. The same algorithms can be reused for different fixtures, given that the CAD models of the objects and fixtures are known.

On the other hand, the proposed method cannot handle fixtures with arbitrary numbers of supporting walls or other supporting elements. For example, adapting to the V-shape, pyramid, and pentagonal pyramid fixtures shown in Fig. 22, or fixtures with point and line contacts [35], [38], is beyond the scope of the current method: The V-shape fixture shown in Fig. 22(a) is widely used for alignment. However, it relies on friction to stop sliding along the intersection axis; Fixtures with more than three supporting walls have redundant contacts and lower flexibility (Fig. 22(b)). They are usually designed as special-purpose fixtures and require careful placement design; Lines and points contacts are helpful to improve the flexibility of placements. They are however ignored as balancing with edge and point contacts are either difficult to be planned and predicted.

**F. Handling Objects With Large Size Variations**

A single fixture is not enough to handle objects with large size variations as the fixture’s dimension may obstruct small objects from regrasp or may not provide enough support for large objects. One solution is provide a set of fixtures to account for different contact surfaces and object sizes. In this part, we present a fixture screening algorithm to find an optimal set of fixtures for handling a set of objects with large size variations. We represent the set of target objects using $\{obj_i\}$ and a set of TCFs with varying sizes but the same configuration as our particular one using $S_i = \{tcf_i\}$. For each object $obj_i$, we traverse all fixtures $tcf_i$ in $S_i$ to find a set of qualified candidates $S_{qf}(obj_i)$. The optimal set of fixtures for $S_{obj}$ is a minimum set $S_{ot}$ that has intersection with all $S_{qf}(obj_i)$. Fig. 23 shows an example. The row header and the column header respectively illustrate the candidate set of fixtures $S_i$ and target objects $S_{obj}$. Each cell in the table shows the representative SPP and the regrasp hands. The cells with green regrasp hands indicate the corresponding TCFs are the qualified elements of $S_{qf}(obj_i)$. The minimal $S_{ot}$ that has intersection with all $S_{qf}(obj_i)$ is (TCF-100, TCF-71, TCF-50). Consequently, we may select them for handling the four objects. Note that this algorithm is a brute-force one. It could be very consumptive for traversing a large $S_i$ set. Also, it could be challenging to take into account other varied three-wall fixtures or fixtures comprising other elements (lines [35], points [38], etc.). These problems deserve further exploration.

**IX. Conclusion and Future Work**

This paper presented a regrasp planning method to eliminate grasp uncertainty. The proposed method first computed SPPs on a TCF, then estimated DDPs to find DDP-SPP pairs, and finally generated grasp configurations for releasing and regrasping an object and eliminating uncertainty. In particular, analytical and learning-based algorithms were proposed for the DDP estimation. Experimental results verified that the learning-based algorithm is more reliable than the analytical one. The regrasp sequences planned by the proposed method was demonstrated to be able to reduce uncertainty to robot
control using an RPG implementation, which was way more robust than a conventional regrasp sequence without taking into account a TCF. Several real-world applications were also presented to show the proposed method’s praeclality in assembly tasks.

In the future, we are interested in building a large deep neural network that generalizes to many common materials and developing flexible features and planners to consider more complicated object shapes.

### APPENDIX

See Table IV.

### REFERENCES

[1] R. Li and H. Qiao, “A survey of methods and strategies for high-precision robotic grasping and assembly tasks—Some new trends,” IEEE/ASME Trans. Mechatronics, vol. 24, no. 6, pp. 2718–2732, Dec. 2019.
[2] H. Chen, J. Li, W. Wan, Z. Huang, and K. Harada, “Integrating combined task and motion planning with compliant control,” Int. J. Intel. Robot. Appl., vol. 4, no. 2, pp. 149–163, Jun. 2020.
[3] K. Nie et al., “Team O2AS’ approach for the task-board task of the world robot challenge 2018,” Adv. Robot., vol. 34, nos. 7–8, pp. 477–498, 2020.
[4] Z. Hou, M. Philipp, K. Zhang, Y. Guan, K. Chen, and J. Xu, “The learning-based optimization algorithm for robotic dual peg-in-hole assembly,” Assem. Automat., vol. 38, no. 4, pp. 369–375, 2018.
[5] R. K. Jain, S. Majumder, and A. Dutta, “SCARA based peg-in-hole assembly using compliant IPMC micro gripper,” Robot. Auto. Syst., vol. 61, no. 3, pp. 297–311, Mar. 2013.
[6] M. A. Erdmann and M. T. Mason, “An exploration of sensorless manipulation,” IEEE J. Robot. Autom., vol. 4, no. 4, pp. 369–379, Aug. 1988.
[7] M. A. Erdmann and M. T. Mason, “A probabilistic planning framework for planar grasping under uncertainty,” IEEE Int. Conf. Robot. Autom., May 2011, pp. 4370–4377.
[8] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[9] Y. Maeda and T. Asamura, “Sensorless in-hand caging manipulation,” in Proc. Int. Conf. Robot. Autom., May 2011, pp. 3–17, Feb. 1988.
[10] H. Ha, S. Agrawal, and S. Song, “Fit2Form: 3D generative model for robot gripper form design,” in Proc. IEEE/RSJ Int. Conf. Robot. Autom., Sep. 2016, pp. 153–160.
[11] Y. Maeda and T. Asamura, “Sensorless in-hand caging manipulation,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[12] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[13] Y. Maeda and T. Asamura, “Sensorless in-hand caging manipulation,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[14] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[15] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[16] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[17] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[18] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[19] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[20] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[21] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[22] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[23] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[24] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[25] Y. Hirata, A. Kaisumi, K. Yamaguchi, and K. Kosuge, “Design of handling device for caging and aligning circular objects,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS), Oct. 2018, pp. 4022–4028.
[22] M. Erdmann, “An exploration of nonprehensile two-palm manipulation,” Int. J. Robot. Res., vol. 17, no. 5, pp. 485–503, May 1998.

[23] A. Schmidt, V. M. Baez, A. T. Becker, and S. P. Fekete, “Coordinated particle relocation using finite static friction with boundary walls,” IEEE Robot. Autom. Lett., vol. 5, no. 2, pp. 985–992, Apr. 2020.

[24] R.-P. Berretty, K. Goldberg, M. H. Overmars, and A. F. van der Stappen, “Computing fence designs for orienting parts,” Comput. Geometry, vol. 10, no. 4, pp. 249–262, Jul. 1998.

[25] D. D. Grossman and M. W. Blasgen, “Orienting mechanical parts by computer-controlled manipulator,” IEEE Trans. Syst., Man, Cybern., vol. SMC-5, no. 5, pp. 561–565, Sep. 1975.

[26] P. Mannam, A. Volkov, R. Paolini, G. Chirikjian, and M. T. Mason, “Computational fence designs for orienting parts,” IEEE/ASME Trans. Mechatronics, vol. 21, no. 2, p. 154, Feb. 2019.

[27] H. Qiao, M. Wang, J. Su, S. Jia, and R. Li, “The concept of ‘attractive footholds’ and its application in high-precision tasks with low-precision systems,” IEEE/ASME Trans. Mechatronics, vol. 20, no. 5, pp. 2311–2327, Oct. 2015.

[28] R. Li and H. Qiao, “Condition and strategy analysis for assembly based on attractive region in environment,” IEEE/ASME Trans. Mechatronics, vol. 22, no. 5, pp. 2218–2228, Oct. 2017.

[29] C. Liu, H. Qiao, and D. Zhang, “Stable sensorless localization of 3-D objects,” IEEE Trans. Syst., Man, Cybern., C, Appl. Rev., vol. 41, no. 6, pp. 923–941, Nov. 2011.

[30] C. Liu, H. Qiao, J. Su, and P. Zhang, “Vision-based 3-D grasping of 3-D objects with a simple 2-D gripper,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 44, no. 5, pp. 605–620, May 2014.

[31] J. Su, H. Qiao, Z. Ou, and Y. Zhang, “Sensor-less insertion strategy for high-precision assembly in a hole of the crankshaft and bearing assembly,” Asien. Autom., vol. 32, no. 1, pp. 86–99, Feb. 2012.

[32] R. Shome et al., “Towards robust product packing with a minimalistic end-effector,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 9007–9013.

[33] A. A. Sharone, “Reliably arranging objects: A conformant planning approach to robot manipulation,” Ph.D. dissertation, Massachusetts Inst. of Technol., Cambridge, MA, USA, 2019.

[34] W. Wan and K. Harada, “Reorientating objects with a gripping hand and a table surface,” in Proc. IEEE-RAS 15th Int. Conf. Hum. Robots (Humanoids), Nov. 2015, pp. 101–106.

[35] C. Cao, W. Wan, J. Pan, and K. Harada, “Analyzing the utility of a support pin in sequential robotic manipulation,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2016, pp. 5499–5504.

[36] D. Sánchez, W. Wan, K. Harada, and F. Kanchiro, “Regrasp planning considering bipedal stability constraints,” in Proc. IEEE-RAS 18th Int. Conf. Hum. Robots (Humanoids), Nov. 2018, pp. 271–277.

[37] W. Wan, M. T. Mason, R. Fukui, and Y. Kuniyoshi, “Improving regrasp algorithms to analyze the utility of work surfaces in a workcell,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2015, pp. 4326–4333.

[38] J. Ma, W. Wan, K. Harada, Q. Zhu, and H. Liu, “Regrasp planning using stable object poses supported by complex structures,” IEEE Trans. Cogn. Dev. Syst., vol. 11, no. 2, pp. 257–269, Jun. 2019.

[39] K. Harada et al., “Object placement planner for robotic pick and place tasks,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Oct. 2012, pp. 980–985.

[40] H. Mosemann, F. Rohrdanz, and F. M. Wahl, “Stability analysis of assemblies considering friction,” IEEE Trans. Robot. Autom., vol. 13, no. 6, pp. 805–813, Dec. 1997.

[41] Y. Liu, M. S. Sabo, V. Thangavelu, and N. Napp, “Approximate stability analysis for drystacked structures,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 8819–8824.

[42] K. Harada, T. Tsuji, S. Uto, N. Yamanobe, K. Nagata, and K. Kitagaki, “Stability of soft-finger grasp under gravity,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2014, pp. 883–888.

[43] T. Tsuji, K. Harada, and K. Kaneko, “Easy and fast evaluation of grasp stability by using ellipsoidal approximation of friction cone,” in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Oct. 2009, pp. 1830–1837.

[44] H. Pham and Q.-C. Pham, “Critically fast pick-and-place with suction cups,” in Proc. Int. Conf. Robot. Autom. (ICRA), May 2019, pp. 3045–3051.

[45] M. Honarpardaz, J. Ölvander, and M. Tarkian, “Fast finger design automation for industrial robots,” Robot. Auto. Syst., vol. 113, pp. 121–131, Mar. 2019.

[46] F. Furrer et al., “Autonomous robotic stone stacking with online next best object target pose planning,” in Proc. IEEE Int. Conf. Robot. Autom. (ICRA), May 2017, pp. 2350–2356.
