

Reducing Uncertainty Using Placement and Regrasp Planning on a Triangular Corner Fixture

Zhengtao Hu¹, Weiwei Wan^{1*}, Keisuke Koyama¹, and Kensuke Harada^{1,2}

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 regrasps are constrained by the TCF’s geometry and have high precision, the final object poses regrasped 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 special peripheral device – a Triangular Corner Fixture (TCF) made by three inclined and mutually perpendicular plates, and study using it to improve manipulation precision. The inclined plates of the TCF form a gravity bucket that holds a 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 presented 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 inspire 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 to employ robotic manipulators for assembly tasks. Especially for autonomous manipulators that receive vision feedback and generate manipulation motion online, uncertainty is challenging to eliminate

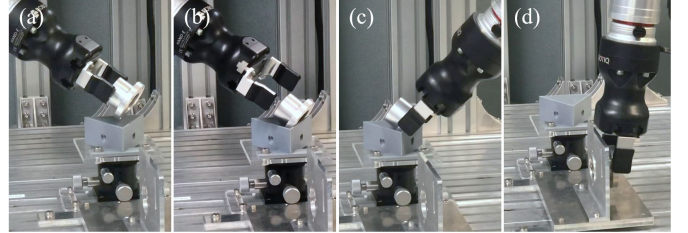


Fig. 1: Using the geometric constraints of a fixture to reduce uncertainty. (a-c) Drop and regrasp. (d) Insertion.

– 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 limit 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 a workpiece 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

This paper is based on results obtained from a project, JPNP20016, subsidized by the New Energy and Industrial Technology Development Organization (NEDO). ¹ Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, Japan. ² National Inst. of AIST.

* Contact: Weiwei Wan, wan@sys.es.osaka-u.ac.jp

robotics. This paper reinspects the idea of employing a fixture to reduce uncertainty. Different from the conventional design and mechanical analysis, our focus is on the planning aspect. We develop algorithms to compute an object's stable poses on a fixture and employ these poses as intermediate states to build manipulation graphs and plan robotic manipulation sequences. With the help of the stable intermediate poses on the fixture, the uncertainty in planned manipulation sequences can be reduced. A workpiece is manipulated precisely, and the manipulation results can be directly used to conduct difficult tasks like insertion. In detail, the fixture used in our study is a triangular corner fixture, which is comprised of three mutually orthogonal planes, as shown in Fig. 1. Driven by gravity, a workpiece dropped from above the tilted corner fixture may firstly contact the inner surfaces and then slide to the bottom under gravity and the guides of the mutually orthogonal planes. The workpiece will rest at a particular pose at the bottom of a gravitational bucket formed by the fixture.

Previously, dropping a workpiece onto a tilted corner for regrasp and reducing pose uncertainty was widely used in the automation industry. For practical purposes, the robot motion for a successful dropping was manually specified. Their stability relied on a system integration engineer's subjective adjustments and examination. The method developed in this work automatically finds the Stable Placement Poses (SPPs) of an object on a Triangular Corner Fixture (TCF) and consequently enables auto-planned precise robotic regrasp and manipulation. The method 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 DPPs, it will rest at expected SPPs. Finally, the method computes the gripper configurations for grasping and regrasping the object considering the TCF, SPPs, and DPPs. The method will output a pick-and-place sequence that manipulates the object with the help of the TCF by high precision. In the experiments, we study the performance of different methods for estimating the DPPs of different objects and quantitatively examine the proposed method's ability to eliminating 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, and mounting housings, etc. The results verify that the method enables a robot to finish assembly tasks without using sensors, compliant control, or complicated mechanism, making the robot system more robust and flexible.

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 SPP sub-planner, DDP estimator, and release/regrasp sub-planner, respectively. Experiments and analysis are shown in Section VII. Section VIII draws conclusions and discusses future work.

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 the 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 to 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.

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 a wide variety of objects given their model information.

B. Placement estimation

We consider a placement estimation as a two-part process. In the first part, we find a set of stable placement poses of the part. Then, based on the placement poses, we infer the dropping or releasing poses. The review of related work in placement estimation is carried out by inspecting the two parts.

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 [27] [28] [29]. As an extension to the fundamental problem, Wan et al. studied the placement

planning on a tilted plane [30], a support pin [28], and arbitrary support structures [31]. Harada et al. [32] 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 is large enough to prevent sliding. The placement stability of rigid bodies and assemblies considering frictional contact was discussed in [33] [34]. 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 [35] [36] [37] [38], and can also be used to estimate the stability of structures [39] [40].

Dynamic dropping simulation is also widely used for placement estimation [31] [41]. 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 [42] 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. Jorgensen et al. [43] 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 [44] provided a statistics-based prediction method for estimating the resting poses of the dropped parts. Fekula et al. [45] 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. [46] 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. [47] proposed to train a probabilistic graphical model as a classifier to predict the appropriate grasp types (power grasp or precision grasp). Li et al. [48] developed a deep network that uses a single depth point cloud to estimate the pose of an articulated object. Newbury et al. [49] 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 well used to estimate the grasp configurations [50] and predict the grasp qualities [51] as well. Feng et al. [52] 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.

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, HANDEY [53], was designed with the regrasp ability based on the algorithm presented in [54]. In some early work [55] [56] [57], the researchers decoupled the planning problem from the robot motion and concentrated on the regrasp sequences and online computation efficacy. More recent research pays more attention to finding the optimal path, which combines the robot motion with regrasp [58] [59]. Wan et al. [60] presented to use a relational database to manage the grasp data, which enabled reusing 10,000 of grasps and their relationship to search the regrasp path. In [61], the dynamic regrasp graph was presented, and the planning method considering regrasp is extended to assembly tasks.

We, in this work, especially focus on a tetrahedral intermediary fixture and using it to reduce uncertainty. Previously, Wan et al. [30] compared the efficacy of using a flat surface and a tilting surface in a work cell as the intermediary fixture [62]. Cao et al. [28] presented to leverage a support pin as the intermediary fixture to provide stable placement. Ma et al. [31] considered to regrasp objects using general structures. The comparison of using different intermediary fixtures was presented in [63]. The previous studies mainly concentrated on optimizing the motion sequence or improving the feasibility of orienting. Different from them, we in this paper use regrasp to eliminate uncertainty and perform precise tasks. A TCF is used as the intermediary fixture considering its crossing surfaces and tetrahedral shape can provide geometric constraints to align the dropped object. Together with the TCF, the place-regrasp process is treated as a release-regrasp way, 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 intersecting at one bottom point, as shown in Fig. 2(a). The three inclined flat plates form a gravitational basket [64] that holds a dropped object at a stable pose. Compared with a flat surface, the plates of a TCF can always pull a dropped object into configurations with minimal potential energy in the gravitational field.

Especially, the plates of the TCF used in this paper are mutually perpendicular. The reason we study this special case 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. Fig. 2(a) shows the structure of the mentioned TCF. Its three plates are fabricated as three congruent isosceles right triangles. The

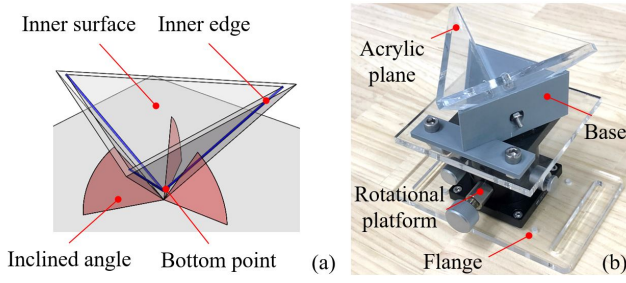


Fig. 2: Structure of a Triangular Corner Fixture (TCF).

angles between the triangular plates and a horizontal plane are the same (54.74°). Figs. 2(b) shows a real-world fabrication. The three triangular plates are made of acrylic boards and are detachable from the base. The isosceles length l_e of the triangular plates are equal. Since they are detachable, the l_e can be changed, and the TCF dimensions can be adapted for parts of different scales. 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.

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 to reason a robot motion sequence. The details of the three sub-modules will be explained in Sections V-VII.

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 of different colors in Fig. 4. The first step includes the blue diamonds and blocks. In this step, the algorithm clusters the faces of the object's mesh model into facets and computes their

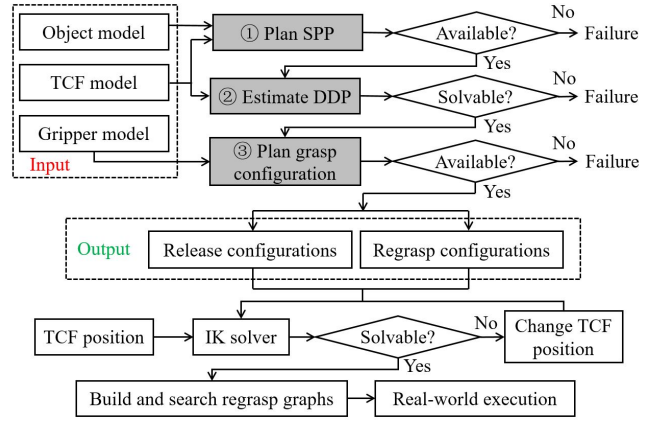


Fig. 3: Workflow of the proposed planner.

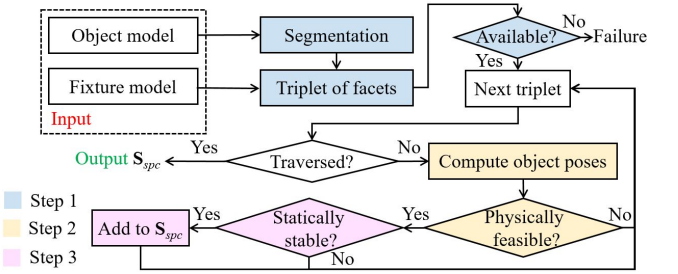


Fig. 4: Plan the SPPs. The diagram is a close-up view of the “Plan SPP” sub-module in Fig. 3.

triple combinations. The triplets of facets are the candidate contact faces with the inner surface of the TCF. The second step includes the yellow diamonds and blocks. In this step, the algorithm uses the triplets of facets to compute the object poses in the TCF. The triplets of facets must be in contact with the TCF's inner surface during the computation. Meanwhile, the object models are required to be not penetrating the TCF and the surrounding environments. The third step includes the pink diamonds and blocks. The algorithm in this step examines the static stability of the object poses computed in the second step and discards the unstable ones. The details of these three steps are presented below. We only consider face-to-face contact between the object and the TCF as effective contact candidates in the algorithm. Although a point-to-face or line-to-face contact can also stabilize an object, they are less reliable and ignored to simplify the planning algorithm.

A. Step 1: Facets and their Triplet Combinations

In this step, the algorithm clusters triangle faces of an object's mesh model \mathcal{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 [65]. Instead of the conventional methods, we use superimposed segmentation [66] 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 \mathcal{M}_o , we denote its superimposed facet set using $\mathbf{S}_o = \{s_i\} (i = 1, 2, \dots, m)$, where each s_i indicates a facet.

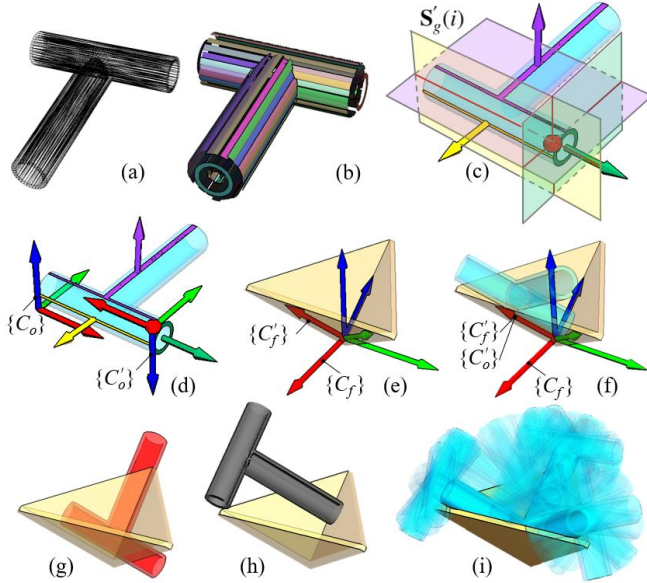


Fig. 5: (a) A raw meshed model. (b) Segmented surfaces. (c) A candidate triplet of facets. It has three mutual orthogonal surfaces. (d) The object's frame ($\{C_o\}$) and the object-to-TCF transformation coordinate described in it ($\{C'_o\}$). (e) The TCF's frame ($\{C_f\}$) and the object-to-TCF transformation coordinate described in it ($\{C'_f\}$). (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.

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 $\mathbf{S}_g = \{\mathbf{S}_g(j) = \{s_a, s_b, s_c | s_a, s_b, s_c \in \mathbf{S}_o\}, j = 1, 2, \dots, C_m^3\}$. Then, we examine the orthogonality of each triplet's facet normal. The one that has three mutual orthogonal normal is considered as a feasible candidate, as illustrated in Fig. 5(c). We use $\mathbf{S}'_g = \{\mathbf{S}'_g(i) = \{s_a, s_b, s_c | s_a \perp s_b, s_a \perp s_c, s_b \perp s_c\}, \mathbf{S}'_g(i) \in \mathbf{S}_g\}$ 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 $\{C'_o\}$ to denote the local frame of $\mathbf{S}'_g(i)$. The intersection point of $\mathbf{S}'_g(i)$'s three orthogonal facets are selected as $\{C'_o\}$'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 $\{C'_o\}$ defined considering the $\mathbf{S}'_g(i)$ shown in Fig. 5(c).

Next, we compute the placement pose of an object by transforming its $\{C'_o\}$ onto TCF. We define two coordinate systems for the TCF. One is $\{C_f\}$. 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 $\{C'_f\}$, which has the same origin as $\{C_f\}$ 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 $\{C'_o\}$ to $\{C'_f\}$, which means if we use a transformation matrix ${}^{C_f}_{C'_o}\mathbf{T}_i$ to denote the placement pose, it can be computed as ${}^{C_f}_{C'_o}\mathbf{T}_i = {}^{C_f}_{C_o}\mathbf{T}_i {}^{C_o}_{C'_o}\mathbf{T}_i = {}^{C_f}_{C'_f}\mathbf{T}_i {}^{C'_f}_{C'_o}\mathbf{T}_i$. An object may have many $\mathbf{S}'_g(i)$ and thus many ${}^{C_f}_{C'_o}\mathbf{T}_i$. We name ${}^{C_f}_{C'_o}\mathbf{T}_i$ s the Potential Placement Poses (PPPs). Fig. 5(f) illustrates one PPP of the T-junction object.

Note that the potential ${}^{C_f}_{C'_o}\mathbf{T}_i$ 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 $\mathbf{p}_i = [x_i, y_i, z_i]$, ($i = 1, 2, \dots, k$). We build a local frame at each of the \mathbf{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 \mathbf{p}_i can be represented by the components along the three axes as $\mathbf{f}_i = [f_{xi}, f_{yi}, f_{zi}]^T$. The effect wrench exerted on \mathbf{p}_i can be computed using $\mathbf{w}_i = \begin{bmatrix} \mathbf{I} & 0 \\ [\mathbf{p}_i \times] & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \tau_i \end{bmatrix}$, where τ_i indicates the exerted torque. Considering the frictional constraints at the contact point, \mathbf{f}_i must be in a friction cone

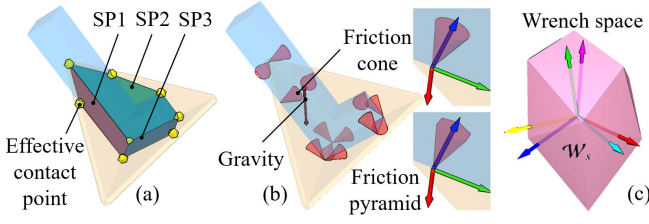


Fig. 6: (a) Support polygons and contact points formed by the contact between the object and the TCF. (b) Friction cones and gravity exerted on the object. The friction cone is simplified as a hexagonal pyramid. (c) Wrench cone \mathcal{W}_s in the wrench space.

and f_{xi} , f_{yi} , and f_{zi} must meet $\sqrt{f_{xi}^2 + f_{yi}^2} \leq \mu f_{zi}$, where μ is the friction coefficient. Since \mathbf{f}_i is inside a cone and is not deterministic, directly using the equation to compute wrench cone is difficult. To overcome the difficulty, we approximate the friction cone with a pyramid [67], as shown in Fig. 6(b.2-3). The lateral edges of the pyramid represent the extreme \mathbf{f}_i choices. They are named as \mathbf{f}_i^j , where the granularity of the approximation determines j . A linear combination of the \mathbf{f}_i^j could approximate a freely chosen \mathbf{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 \mathbf{f}_i^j . Considering all of them for wrench cones is the same as considering linear combination of the \mathbf{f}_i^j . The wrench set \mathbf{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 $\mathbf{W} = \{w_i^j\} \cup \dots \cup \{w_k^j\} \cup \{w_g\}$, where w_g denotes the gravitational wrench. The wrench cone \mathcal{W}_s spanned by the wrench is essentially a convex hull of the elements in \mathbf{W} [36]. The stability of LFPs is judged by examining the relationship between the origin of the wrench space and \mathcal{W}_s ¹. If the origin is inside the \mathcal{W}_s of an LFP, the LFP is considered to be stable and will be counted as an SPP. The planner will look over all LFPs and find a set of SPPs.

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 SPPs obtained in the last section have correspondent DDPs. The methods are based on the assumption that a candidate DDP is a pose linearly elevated from an SPP. The elevation height h_e ranges from h^- to h^+ , as shown in Fig. 7. We elevate an SPP to a random height in $[h^-, h^+]$ to get candidate releasing pose and use the methods presented in the following subsections 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 regrip sequences to improve grasping precision.

¹The magnitude of all elements in \mathbf{W} is set equally as 1 when computing the \mathcal{W}_s since we assume unknown masses.

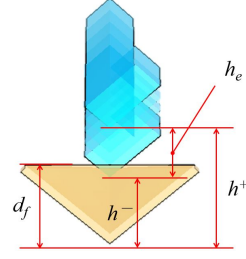


Fig. 7: Visualization of the elevation height h_e and its effective range $[h^-, h^+]$. The d_f is the depth of the TCF. It is not mentioned in the main text. We choose h^- to be a bit lower than d_f and choose h^+ to be outside d_f to take into account various possibilities.

Specifically, we propose two methods for the estimation. The first is an analytical method based on CWS, and the second is a learning-based method. Their details are as follows.

A. Analytical method

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

Similar to the third step of planning SPPs, the analytical method uses CWS to evaluate static stability. However, instead of directly generating convex hull of contact wrenches to span the wrench space, the method constructs the wrench cone by computing the convex hull of \mathbf{W} 's Minkowski sum. We use the notation \mathcal{W}_{mkv} to differ the wrench cone in this section from the \mathcal{W}_s used before. Compared to \mathcal{W}_s , \mathcal{W}_{mkv} can quantify the resistible external wrench, thus make it easier to decide an evaluation criteria [36]. In particular, the method computes the shortest distance from the origin of the wrench space to the hyperplanes that constitute \mathcal{W}_{mkv} and uses the shortest distance as the stability quality. Then, the method finds the SPPs that have enough stability quality from the obtained SPP set and elevate them to get DDPs.

The static stability criteria-based analysis may find the DDPs with large 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 criterion is based on intuition and is not fundamentally true. The DDPs may have uncertainty (positional and rotational noises, and also bouncing) in the real world, which are not considered by the method and may invalidate the intuition. For these reasons, more advanced methods need to be explored.

B. Learning-based method

In the second method, we use machine learning to predict DDPs. We use a sim-to-real method [68] 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

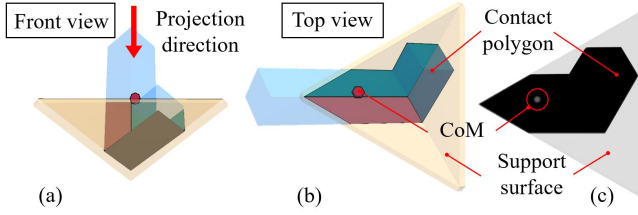


Fig. 8: An SPP shown in (a) is converted into a grayscale image in (c) by using the top view shown in (b).

grayscale image with 224×224 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 $v_{grey} = \phi \cdot h_{com}$, where ϕ 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_{grey} 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 take into account 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 to be 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, and an Alexnet-CNN. The detailed results and discussions about the implementation and comparison will be presented in Section.VIII-B.

VI. PLAN GRASP CONFIGURATIONS AND REGRAASP SEQUENCES

This section presents detailed releasing and regrasp planning algorithms for adjusting grasping precision. The algorithms are partially based on our previous work published in [66]

and [69]. First, we plan grasps configurations for an object without considering any obstacles using the methods presented in [66]. 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 [69] by reasoning and connecting the grasp sets associated with all SPP-DDP pairs, and search the graph to obtain regrasp sequences.

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 different object states. The yellow edges represent strong connections between the black maximally connected graphs in Fig. 10(b) and (c). They indicate that a grasp configuration associated with a DDP can transit to another grasp configuration associated with its pairing SPP. To precisely regrasp the L-shape object, our planner will search a path on the graph by starting from one node in (a) (the initial grasp) and ending at another node in (b) (the final grasp). The path essentially maps to a sequence where the object will be grasped from the initial pose (a node in (a)), moved to a DDP (a node in (b)), dropped down onto the TCF and regrasped (a node in (c)), and finally moved to the goal (a node in (d)).

VII. EXPERIMENTS AND ANALYSIS

This section includes three parts. First, we compare the methods for estimating the DDP-SPP pairs. Second, we use

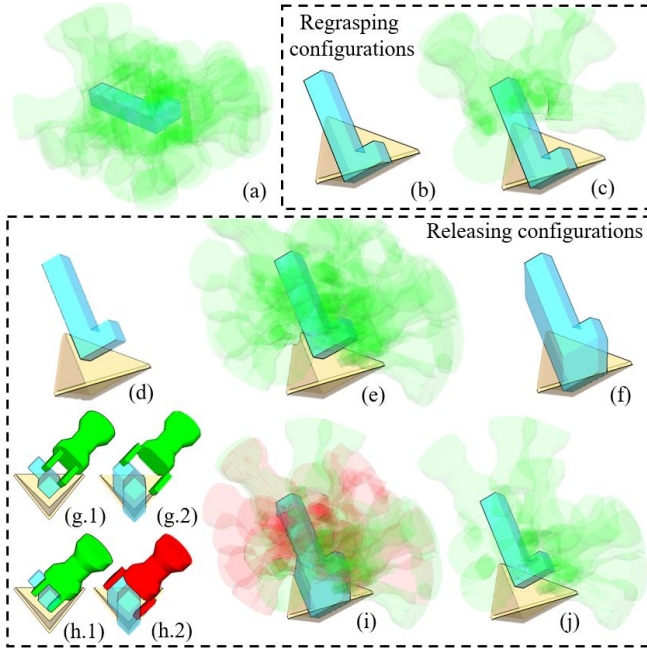


Fig. 9: (a) Grasp configurations planned without considering any surrounding obstacles or pose changes. (b) An SPP. (c) The grasp configuration set associated with the SPP in (b). (d) A DDP. (e) Transformed collision-free grasps from (a), with the ones in collision with the TCF removed. (f) Swept volume of the dropped object. (g-h) Releasing hands may collided with the swept volume. (i) Grasp configurations that collide with the swept volume are highlighted in red and will be removed. (j) Grasp configurations associated with the DDP.

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

We proposed one analytical method and three learning-based methods in Section V-B-2) for estimating the DDPs. In this subsection, we compare their performance using physical simulation. Especially for the learning-based methods, we use the 14 primitives shown in Fig. 11(a) to obtain the training data. The primitives are scaled from 50% to 150% with 10% granularity, as shown in Fig. 11(b), to obtain 154 objects. Using these objects, we get 4464 SPPs. We collect 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 are set to 0.3 and 0.2. The inner edge length of the TCF is set to 50.0 mm. Its d_f is set to 28.0 mm. We collect 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 get 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

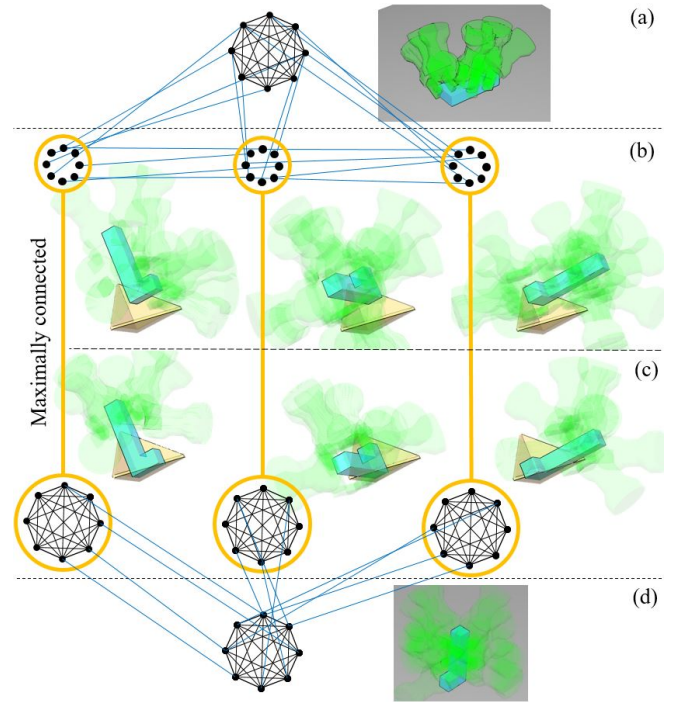


Fig. 10: A regrasp graph. (a) Initial pose and its subgraph. (b) DDP and its subgraph. (c) SPP and its subgraph. (d) Goal pose and its subgraph. Each black node in the graph indicates one grasp pose. Each circle indicates an object pose. The nodes inside the circle are the grasp configurations associated 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.

2674 negative samples. 80% of the data is used to train the estimators with cross-validation used to verify the results. The remaining 20% is used for the test.

The results using different methods (including the analytical one) are shown in Fig. 12. The learning methods have better performance, of which the AlexNet shows the highest success rate (90.1%). The analytical method has poor performance because “finding the SPPs with enough stability quality” needs a threshold. For practical purposes, we only used the most stable configuration, which easily leads to ignored DDPs. Meanwhile, 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 may thus be unconvincing.

Besides the simulated data, we also validate the various methods using four real objects shown in Fig. 13. They include: (a) an L-shape object; (b) a T-junction; (c) a bracket; (d) a bearing housing. According to the object’s size, we chose a 50.0 mm-TCF for the L-shape object and bearing housing, and a 70.7 mm-TCF for the bracket and the T-junction. For the learning-based method, we used the classifiers trained above to judge DDP-SPP pairs. For the analytical method,

TABLE I: Comparing the success rate of different estimation methods.

Method	L-shape object			T-junction			Bracket			Bearing housing		
	Unpaired	DDP-SPP	Total(%)	Unpaired	DDP-SPP	Total(%)	Unpaired	DDP-SPP	Total(%)	Unpaired	DDP-SPP	Total(%)
Analytical	48/48	6/6	100	0/48	0/24	0	42/42	12/12	100	72/72	6/72	54
SVM	48/48	6/6	100	38/48	14/24	72	33/42	12/12	83	72/72	45/72	81
FCN	48/48	6/6	100	37/48	14/24	71	32/42	9/12	76	72/72	50/72	85
AlexNet	48/48	6/6	100	41/48	18/24	82	38/42	10/12	89	72/72	60/72	92

* 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.

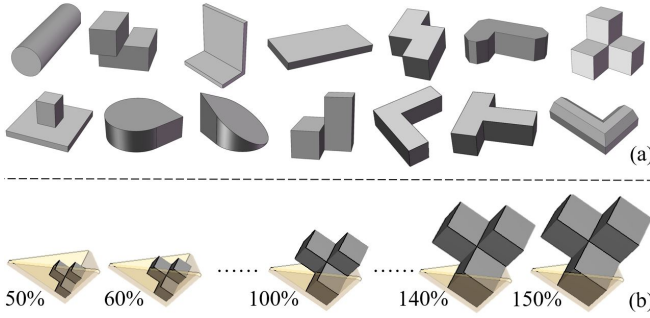


Fig. 11: Objects used for obtaining the training data. By scaling the 14 primitive objects in (a) using the rules shown in (b) (resize the object from 50% to 150% at every 10%), in total 154 objects are prepared.

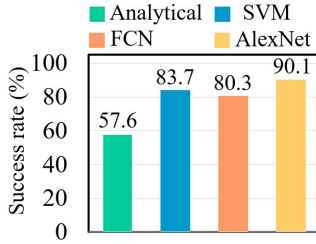


Fig. 12: Comparison of the different estimation methods. The learning-based methods have higher estimation success rate. Especially, the AlexNet method is the most effective one.

we use 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) are obtained by repeated physical simulation. There are 54 SPPs for the L-shape object. 6 of them have DDP-SPP pairs, as shown by the denominators of the L-shape object's "DDP-SPP" column. The remaining 48 does 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 do not have a DDP. The numerators of the "DDP-SPP" column show the actual number of SPPs that have a DDP. The T-junction has 72 SPPs, where 24 of them have DDP-SPP pairs, and the remaining 48 do not have DDP counterparts. The bracket has 12 DDP-SPP pairs and 42 unpaired SSPs. The bearing housing has 72 DDP-SPP pairs and 72 unpaired ones. The results show that the analytical method works effectively for the L-shape object and the bracket, but performs poorly for the T-junction object and the bearing housing. The learning-based methods are better on average but may have shortages for specific objects (i.e. the bracket). The AlexNet method is the best of all learning-based

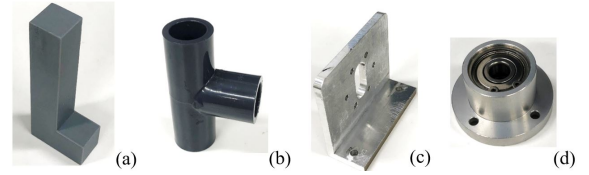


Fig. 13: The objects used for testing the planner. (a) L-shape object. (b) T-junction. (c) Bracket. (d) Bearing housing.

methods, which is consistent with the results shown in Fig. 12.

We further performed real-world dropping tests using the four objects. The process is as follows. First, we place the object with a selected SPP on the TCF. Then, the robot will grasp the object, elevate it to a DDP with random offset noises, and open the gripper to release the object. Finally, we observe the dropping process, check if the object gets stabilized at the selected SPP, and record the results. The process is repeated 30 times for each SPP to get a statistical view. Fig. 14 shows results. Due to page limits, it is impossible to show all SPPs and we only present some representative DDP-SPP cases for readers' convenience. The results indicate that the estimation mostly accords with the real-world results.

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 the DDP-SPP pair that has the best SPP stability is like Fig. 12(b.3) and (d.2). These SPPs have high static stability qualities, but their contact areas are distributed around the objects' CoMs (as shown by the 2D grayscale images of the figure). A large section of an object is not in contact with the inner surface of the TCF. The object will have a low chance to stably "stand" on the distributed contact when being dropped. It may get stuck by the edges of the TCF.

B. Performance on eliminating uncertainty

In the experiments of this subsection, we use robotic peg-in-hole insertion tasks to evaluate the performance of the proposed method on eliminating uncertainty and compare it with the conventional method that does not use TCF regrasp.

Fig. 15 shows the difference of the methods used for comparison. The first method is the conventional one which directly plans to move the picked object to the goal pose, as shown in Fig. 15(a). The method is abbreviated as DPM in the following context. The second method is our proposed

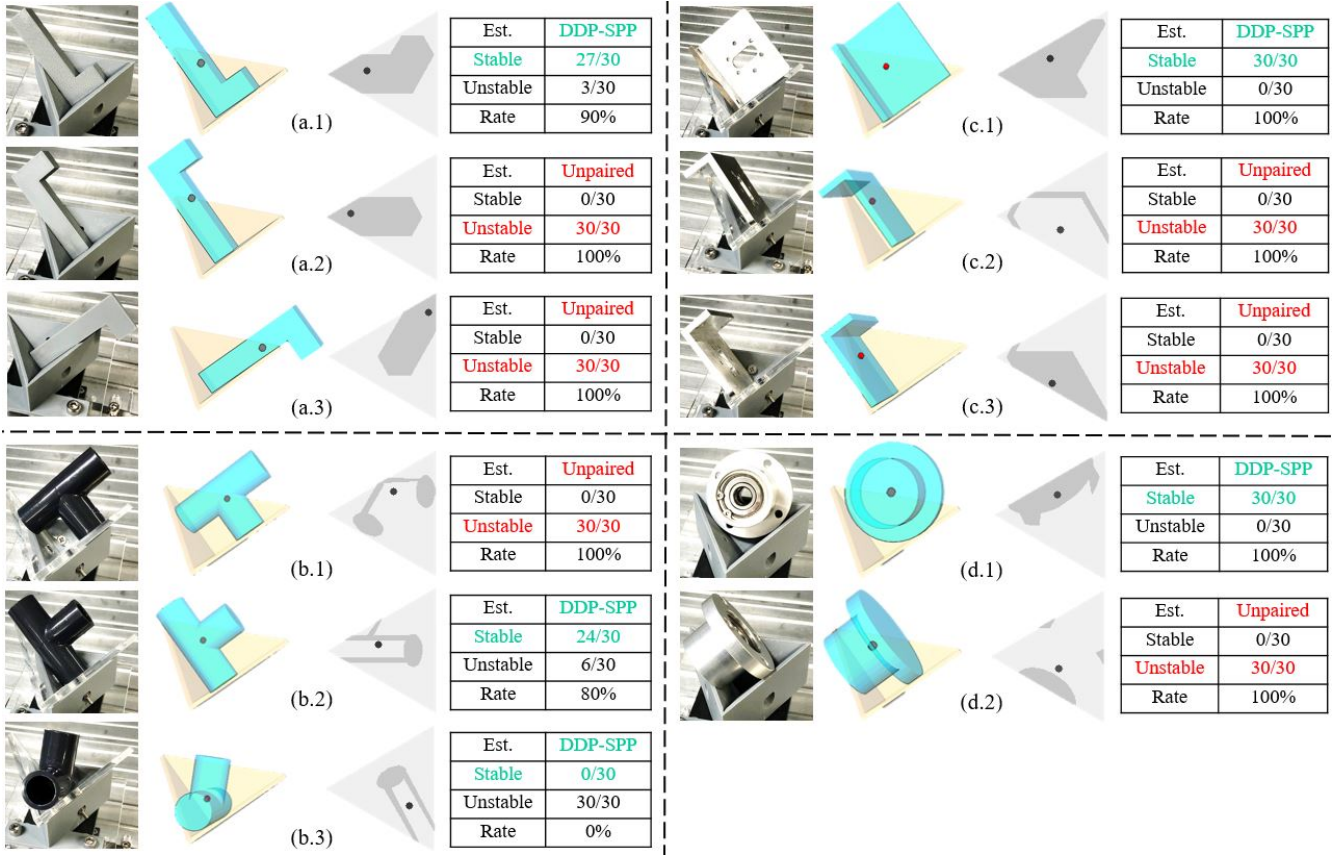


Fig. 14: The results of testing the estimator using different objects. (a) L-shape object. (b) T-junction. (c) bracket. (d) bearing housing. The SPPs of each object are selectively shown. The left column shows the real-world photos, the middle column shows the placements in the simulator, and the right column illustrates the projected images.

method, in which we build and search a regrasp graph to find a regrasp sequence. Especially, we propose two implementations of the method: Regrasp with All Grasps (RAG) and Regrasp with a Prescribed Grasp (RPG). In the RAG implementation, all the grasp configurations for the goal pose and SPP are considered to build the regrasp graph, as shown in Fig. 15(b). The implementation exactly follows the graph shown in 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 prescribed goal pose using the prescribed grasp configuration is taught instead of planned. From the viewpoint of Fig. 10, the connections in Fig. 10(c) and Fig. 10(d) are replaced with a given path. The regrasp sequence planner plans to (c) and uses the given path to reach (d). Both the RAG and RPG methods can take advantages of the TCF fixture to reduce the uncertainty of the yellow object poses. However, the RAG implementation's performance relies a lot on a robot's absolute precision. The robot action is online generated until the last step. In contrast, the RPG method leverages taught motion to move the object from the TCF to the final goal. Its performance is dominated by a robot's repeatability precision. We compare all the DPM, RAG, and RPG methods (or implementations) in the experiments. The holes of the insertion tasks in the experiments have different clearance, as shown in Fig. 16(a).

The diameters of the holes range from 10.1 mm to 18.0 mm. The length and diameter of the peg are 75.0 mm and 10.0 mm, respectively. The clearance thus ranges from 0.1 mm to 8.0 mm. We run the insertion for each hole using one of the methods repeatedly by 15 times to obtain an average success rate, and get the methods' performance on eliminating uncertainty by considering the smallest clearance with 100% success rate.² In each repetition, we place the object in a random initial position on a table. A robot will detect it using a PhotoneoPhoXi 3D Scanner M depth sensor and move it to a pre-given goal pose with or without regrasp at the TCF. At the goal pose, the robot will insert the peg by moving a straight line with position control.

The results are shown in Fig. 16 as a bar chart, where the horizontal axis is the different hole diameters, and the vertical axis is the average success rate. The results tell that the smallest clearance of the DPM, RAG, and RPG methods are 7.0 mm, 1.0 mm, and 0.1 mm, respectively. The methods share the same uncertainty origins, including visual recognition, fabrication, robotic control, etc., but they eliminate the uncertainty to different ranges. The RAG clearance is larger than RPG, which confirms that the robot has low absolute precision compared to repeatability precision.

²A 100% average success rate means the method can always suppress the peg's uncertainty within a range indicated by the clearance value.

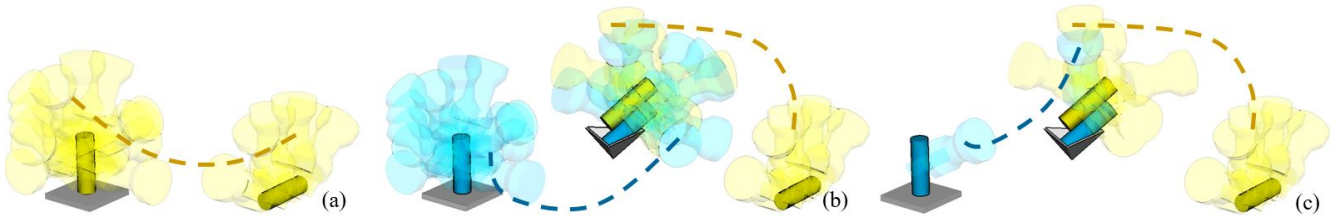


Fig. 15: Illustrations of the task process using different methods. The initial pose, the intermediate DDP, and their associated grasps are illustrated in a yellow color. The intermediate SPP, the goal pose, and their associated grasp poses are illustrated in a cyan color. (a) DPM: Directly plan to move the object to the goal pose. (b) RAG: An implementation of the proposed method that plans regrasp using all grasp configurations for the goal pose and the regrasp. (c) RPG: Another implementation that plans regrasp using a prescribed grasp configuration for the goal pose and the regrasp. The cyan object poses and grasp configurations (a single one for each object pose) indicate the prescribed items.

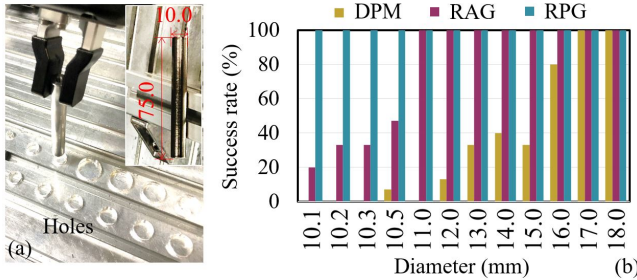


Fig. 16: The success rates (%) of inserting a peg into holes with different diameters using different methods.

C. Performance in practical real-world tasks

Finally, we test the proposed method using four practical 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 fix an acrylic board with a rectangular groove on a table, and ask the robot to insert the L-shape object into the rectangular groove. Fig. 17(a.1) shows the sizes of the object and the groove. The clearance between them is 2.0 mm.

2) *Sheathing the T-junction*: The goal of this task is to sheathe a tube into the T-junction. The tube is vertically fixed on the table, and the robot is asked to manipulate the T-junction to perform the sheathing action. Fig. 17(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 is 0.3 mm.

3) *Aligning the holes*: In this task, a base plate with thread holes is fixed on a table. The two through-holes on the short side of the bracket are required to be aligned with the thread holes on the base plate. If a screw bolt can be fastened in the thread holes across the through-holes, we judge the alignment to be successful. Fig. 17(c.1) shows the sizes of the bracket and the thread holes. The difference between the inner thread-hole diameter and through-hole diameter is the task's clearance. Its value is 1.7 mm.

TABLE II: Results of various methods in the practical tasks.

	Clearance	DPM	RAG	RPG
Insert L-shape	2.0 mm	2/10	10/10	10/10
Sheath T-junction	0.3 mm	0/10	0/10	10/10
Align holes	1.7 mm	0/10	10/10	10/10
Mount bearing housing	<0.1 mm	0/10	0/10	10/10

4) *Mounting the bearing housing*: This task requires the robot to mount the bearing housing on a fixed bracket. The sizes of the bearing housing and mounting hole are shown in Fig. 17(d.1). The clearance between them is less than 0.1 mm.

Like the previous experiments, the environment model, object models, and the configuration of the TCF are pre-given and pre-calibrated. Also, the goal poses of them in the assembly tasks are known. The initial poses of the objects are random. The conventional method (DPM) and the two implementations of our method (RAG and RPG) are tested. For each of the above tasks, we run ten times of experiments using different methods. Table II 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 show that the proposed method can provide reliable and robust performance for these tasks, especially when the RPG method is used.

Fig. 17(a.2-d.2) shows execution pictures of some successful results in Table II. Readers may also refer to the video supplementary attached to this manuscript to observe the detailed robotic actions.

VIII. CONCLUSIONS AND FUTURE WORK

This paper presented a regrasp planning method to eliminate grasp uncertainty. The proposed method first computes all SPPs on a TCF, then estimates the DDP to find all DDP-SPP pairs, and finally generates the grasp configurations for releasing and regrasping the object. In particular, an analytical

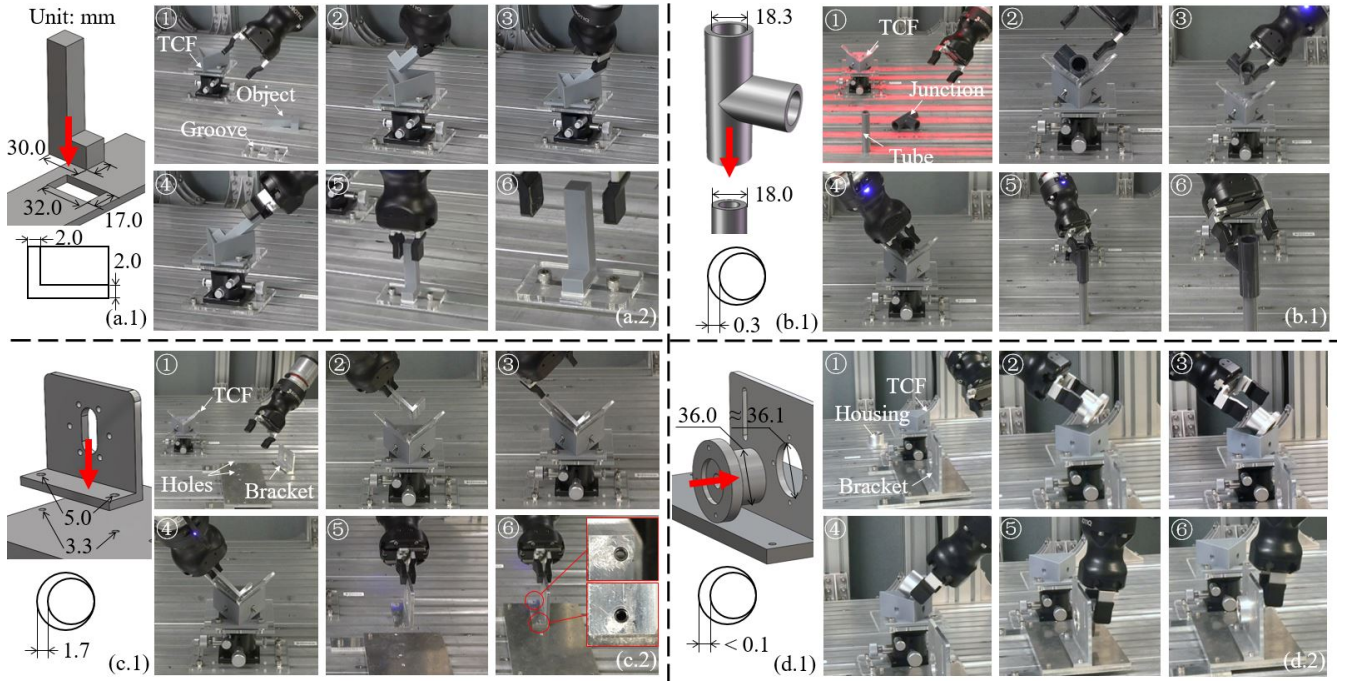


Fig. 17: (a) Inserting the L-shape object. (b) Sheathing the T-junction. (c) Aligning the holes. (d) Mounting the bearing housing. The left column shows the main boundary dimensions of the components and also points the assembly direction. The right column shows the demonstration of the successful task.

and a learning-based method are proposed for the DDP estimation. Experimental results verified that the learning-based method is more reliable than the analytical one. The regrasp sequence planned by the proposed method is demonstrated to reduce uncertainty to less than 0.1 mm using an RPG implementation, which is way more robust than a conventional regrasp sequence that does not take into account a TCF. Several real-world applications are also presented to show the proposed method's promising usage in assembly tasks.

Note that we ignored the influence of different materials in our work, and we assumed uniform density, fixed friction coefficient (0.3), and bounce rate (0.2). We also ignored the rotation around an object's symmetric axis (i.e., rotation of the bearing housing). It is thus impossible to stabilize this axis using the assumed TCF. 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.

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 Transactions on Mechatronics*, vol. 24, no. 6, pp. 2718–2732, 2019.
- [2] H. Chen, J. Li, W. Wan, Z. Huang, and K. Harada, "Integrating combined task and motion planning with compliant control," *International Journal of Intelligent Robotics and Applications*, vol. 4, no. 2, pp. 149–163, 2020.
- [3] K. Nie, F. Von Drigalski, J. C. Triyonoputro, C. Nakashima, Y. Shibata, Y. Konishi, Y. Ijiri, T. Yoshioka, Y. Domae, T. Ueshiba *et al.*, "Team o2as' approach for the task-board task of the world robot challenge 2018," *Advanced Robotics*, vol. 34, no. 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," *Assembly Automation*, 2018.
- [5] S. Liu, Y.-F. Li, and D. Xing, "Sensing and control for simultaneous precision peg-in-hole assembly of multiple objects," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 310–324, 2019.
- [6] X. Li, W. Li, Y. Zheng, K. Althoefer, and P. Qi, "Criminisi algorithm applied to a gelsight fingertip sensor for multi-modality perception," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2020, pp. 190–195.
- [7] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable manipulation with a tactile-reactive gripper," in *Robotics: Science and Systems*, 2020.
- [8] R. K. Jain, S. Majumder, and A. Dutta, "Scara based peg-in-hole assembly using compliant ipmc micro gripper," *Robotics and Autonomous Systems*, vol. 61, no. 3, pp. 297–311, 2013.
- [9] D. Shetty, "Compliant gripper for precision robotic assembly," in *IEEE International Conference on Systems Engineering*, 1990, pp. 335–338.
- [10] A. Gupta and S. K. Arora, *Industrial automation and robotics*. Laxmi publications, 2009.
- [11] Y. Hou, Z. Jia, and M. T. Mason, "Fast planning for 3d any-pose-reorienting using pivoting," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 1631–1638.
- [12] M. A. Erdmann and M. T. Mason, "An exploration of sensorless manipulation," *IEEE Journal on Robotics and Automation*, vol. 4, no. 4, pp. 369–379, 1988.
- [13] M. Mason, "The mechanics of manipulation," in *IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 544–548.
- [14] R. C. Brost, "Automatic grasp planning in the presence of uncertainty," *The International Journal of Robotics Research*, vol. 7, no. 1, pp. 3–17, 1988.
- [15] K. Nie, W. Wan, and K. Harada, "An adaptive robotic gripper with l-shape fingers for peg-in-hole tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 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 *IEEE International Conference on Robotics and Automation*, 2011, pp. 4370–4377.
- [17] H. Ha, S. Agrawal, and S. Song, "Fit2Form: 3D generative model for robot gripper form design," in *Conference on Robotic Learning*, 2020.

- [18] K. Y. Goldberg, "Orienting polygonal parts without sensors," *Algorithmica*, vol. 10, no. 2-4, pp. 201–225, 1993.
- [19] J. Zhou, R. Paolini, A. M. Johnson, J. A. Bagnell, and M. T. Mason, "A probabilistic planning framework for planar grasping under uncertainty," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2111–2118, 2017.
- [20] Y. Maeda and T. Asamura, "Sensorless in-hand caging manipulation," in *IEEE/RSJ International Conference on Intelligent Autonomous Systems*, 2016, pp. 255–267.
- [21] P. L. Várkonyi, "Sensorless part feeding with round cages in two and three dimensions," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 724–731, 2016.
- [22] M. Erdmann, "An exploration of nonprehensile two-palm manipulation," *The International Journal of Robotics Research*, vol. 17, no. 5, pp. 485–503, 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 Robotics and Automation Letters*, vol. 5, no. 2, pp. 985–992, 2020.
- [24] R.-P. Berretty, K. Goldberg, M. H. Overmars, and A. F. van der Stappen, "Computing fence designs for orienting parts," *Computational Geometry*, vol. 10, no. 4, pp. 249–262, 1998.
- [25] D. D. Grossman and M. W. Blasgen, "Orienting mechanical parts by computer-controlled manipulator," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 5, pp. 561–565, 1975.
- [26] P. Mannam, A. Volkov, R. Paolini, G. Chirikjian, and M. T. Mason, "Sensorless pose determination using randomized action sequences," *Entropy*, vol. 21, no. 2, p. 154, 2019.
- [27] W. Wan and K. Harada, "Reorientating objects with a gripping hand and a table surface," in *IEEE-RAS International Conference on Humanoid Robots*, 2015, pp. 101–106.
- [28] C. Cao, W. Wan, J. Pan, and K. Harada, "Analyzing the utility of a support pin in sequential robotic manipulation," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 5499–5504.
- [29] D. Sánchez, W. Wan, K. Harada, and F. Kanchiro, "Regrasp planning considering bipedal stability constraints," in *IEEE-RAS 18th International Conference on Humanoid Robots*, 2018, pp. 271–277.
- [30] W. Wan, M. T. Mason, R. Fukui, and Y. Kuniyoshi, "Improving regrasp algorithms to analyze the utility of work surfaces in a workcell," in *IEEE International Conference on Robotics and Automation*, 2015, pp. 4326–4333.
- [31] J. Ma, W. Wan, K. Harada, Q. Zhu, and H. Liu, "Regrasp planning using stable object poses supported by complex structures," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 11, no. 2, pp. 257–269, 2018.
- [32] K. Harada, T. Tsuji, K. Nagata, N. Yamanobe, H. Onda, T. Yoshimi, and Y. Kawai, "Object placement planner for robotic pick and place tasks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 980–985.
- [33] H. Mosemann, F. Rohrdanz, and F. M. Wahl, "Stability analysis of assemblies considering friction," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 805–813, 1997.
- [34] Y. Liu, M. Saboia, V. Thangavelu, and N. Napp, "Approximate stability analysis for drystack structures," in *International Conference on Robotics and Automation*, 2019, pp. 8819–8824.
- [35] K. Harada, T. Tsuji, S. Uto, N. Yamanobe, K. Nagata, and K. Kitagaki, "Stability of soft-finger grasp under gravity," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 883–888.
- [36] T. Tsuji, K. Harada, and K. Kaneko, "Easy and fast evaluation of grasp stability by using ellipsoidal approximation of friction cone," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 1830–1837.
- [37] H. Pham and Q.-C. Pham, "Critically fast pick-and-place with suction cups," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 3045–3051.
- [38] M. Honarpardaz, J. Ölvander, and M. Tarkian, "Fast finger design automation for industrial robots," *Robotics and Autonomous Systems*, vol. 113, pp. 120–131, 2019.
- [39] F. Furrer, M. Wermelinger, H. Yoshida, F. Gramazio, M. Kohler, R. Siegwart, and M. Hutter, "Autonomous robotic stone stacking with online next best object target pose planning," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 2350–2356.
- [40] V. Thangavelu, Y. Liu, M. Saboia, and N. Napp, "Dry stacking for automated construction with irregular objects," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2018, pp. 1–9.
- [41] P. Song, J. C. Trinkle, V. Kumar, and J.-S. Pang, "Design of part feeding and assembly processes with dynamics," in *IEEE International Conference on Robotics and Automation*, vol. 1, 2004, pp. 39–44.
- [42] D. J. Kriegman, "Let them fall where they may: Capture regions of curved objects and polyhedra," *The International Journal of Robotics Research*, vol. 16, no. 4, pp. 448–472, 1997.
- [43] J. A. Jorgensen, L.-P. Ellekilde, and H. G. Petersen, "Handling uncertainties in object placement using drop regions," in *ROBOTIK 2012; 7th German Conference on Robotics*, 2012, pp. 1–6.
- [44] P. L. Várkonyi, "Estimating part pose statistics with application to industrial parts feeding and shape design: New metrics, algorithms, simulation experiments and datasets," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 658–667, 2014.
- [45] M. Fekula and G. Horváth, "Determining stable equilibria of spatial objects and validating the results with drop simulation," *Procedia CIRP*, vol. 81, pp. 316–321, 2019.
- [46] J. Baumgartl, T. Werner, P. Kaminsky, and D. Henrich, "A fast, gpu-based geometrical placement planner for unknown sensor-modelled objects and placement areas," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 1552–1559.
- [47] Q. Lu and T. Hermans, "Modeling grasp type improves learning-based grasp planning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 784–791, 2019.
- [48] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3706–3715.
- [49] R. Newbury, K. He, A. Cosgun, and T. Drummond, "Learning to place objects onto flat surfaces in human-preferred orientations," *arXiv preprint arXiv:2004.00249*, 2020.
- [50] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning object grasping for soft robot hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2370–2377, 2018.
- [51] E. Dessalene, Y. H. Ong, J. Morrow, R. Balasubramanian, and C. Grimm, "Using geometric features to represent near-contact behavior in robotic grasping," in *IEEE International Conference on Robotics and Automation*, 2019, pp. 2772–2777.
- [52] Q. Feng, Z. Chen, J. Deng, C. Gao, J. Zhang, and A. Knoll, "Center-of-mass-based robust grasp planning for unknown objects using tactile-visual sensors," *arXiv preprint arXiv:2006.00906*, 2020.
- [53] T. Lozano-Pérez, J. Jones, E. Mazer, P. O'Donnell, W. Grimson, P. Tournassoud, and A. Lanusse, "Handey: A robot system that recognizes, plans, and manipulates," in *IEEE international conference on robotics and automation*, vol. 4, 1987, pp. 843–849.
- [54] P. Tournassoud, T. Lozano-Pérez, and E. Mazer, "Regrasping," in *IEEE international conference on robotics and automation*, vol. 4, 1987, pp. 1924–1928.
- [55] F. Rohrdanz and F. M. Wahl, "Generating and evaluating regrasp operations," in *International Conference on Robotics and Automation*, vol. 3, 1997, pp. 2013–2018.
- [56] K. Cho, M. Kim, and J.-B. Song, "Complete and rapid regrasp planning with look-up table," *Journal of Intelligent and Robotic Systems*, vol. 36, no. 4, pp. 371–387, 2003.
- [57] H. Terasaki and T. Hasegawa, "Motion planning of intelligent manipulation by a parallel two-fingered gripper equipped with a simple rotating mechanism," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp. 207–219, 1998.
- [58] T. Siméon, J.-P. Laumond, J. Cortés, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 729–746, 2004.
- [59] E. Yoshida, C. Esteves, O. Kanoun, M. Poirier, A. Mallet, J.-P. Laumond, and K. Yokoi, "Planning whole-body humanoid locomotion, reaching, and manipulation," in *Motion planning for humanoid robots*, 2010, pp. 99–128.
- [60] W. Wan and K. Harada, "Regrasp planning using 10,000 s of grasps," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 1929–1936.
- [61] —, "Integrated single-arm assembly and manipulation planning using dynamic regrasp graphs," in *IEEE International Conference on Real-time Computing and Robotics*, 2016, pp. 174–179.
- [62] W. Wan, E. C. Cheung, J. Pan, and K. Harada, "Optimizing the parameters of tilting surfaces in robotic workcells," in *IEEE International Conference on Automation Science and Engineering*, 2015, pp. 593–599.
- [63] C. Cao, W. Wan, J. Pan, and K. Harada, "An empirical comparison among the effect of different supports in sequential robotic manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 2548–2553.
- [64] E. D. Rimon, F. T. Pokorny, and W. Wan, "Geometric characterization of two-finger basket grasps of 2-d objects: Contact space formulation,"

- in *IEEE International Conference on Robotics and Automation*, 2020, pp. 1560–1566.
- [65] A. D. Kalvin and R. H. Taylor, “Superfaces: polygonal mesh simplification with bounded error,” *IEEE Computer Graphics and Applications*, vol. 16, no. 3, pp. P.64–77, 1996.
 - [66] W. Wan, K. Harada, and F. Kanehiro, “Planning grasps with suction cups and parallel grippers using superimposed segmentation of object meshes,” *IEEE Transactions on Robotics*, vol. 37, no. 1, pp. 166–184, 2020.
 - [67] J. C. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo, “On dynamic multi-rigid-body contact problems with coulomb friction,” *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 77, no. 4, pp. 267–279, 1997.
 - [68] J. C. Triyonoputro, W. Wan, and K. Harada, “Quickly inserting pegs into uncertain holes using multi-view images and deep network trained on synthetic data,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 5792–5799.
 - [69] W. Wan, K. Harada, and F. Kanehiro, “Preparatory Manipulation Planning Using Automatically Determined Single and Dual Arm,” *IEEE Transactions on Industrial Informatics*, vol. 16, pp. 442–453, 2020.