IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 29, 2021, accepted November 5, 2021, date of publication November 9, 2021,
date of current version December 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3126829

Pre-Grasp Manipulation Planning to Secure Space for Power Grasping

INHYUK BAEK[1], KYOOSIK SHIN[2], HYUNJUN KIM[3], SEUNGHOON HWANG[1], ERIC DEMEESTER[4], (Member, IEEE), AND MIN-SUNG KANG[5], (Member, IEEE)

[1]Department of Mechatronics Engineering, Hanyang University, Ansan 15588, South Korea
[2]Department of Robot Engineering, Hanyang University, Ansan 15588, South Korea
[3]Department of Interdisciplinary Robot Engineering Systems, Hanyang University, Ansan 15588, South Korea
[4]Department of Mechanical Engineering, KU Leuven, 3590 Diepenbeek, Belgium
[5]School of Smart Convergence Engineering, Hanyang University, Ansan 15588, South Korea

Corresponding author: Min-Sung Kang (wowmecha@hanyang.ac.kr)

This work was supported by the Ministry of Trade, Industry, and Energy (MOTIE), South Korea, under the Fostering Global Talents for Innovative Growth Program through the Korea Institute for Advancement of Technology (KIAT) under Grant P0008745.

ABSTRACT An object can be gripped firmly through power grasping, in which the gripper fingers and palm are wrapped around the object. However, it is difficult to power-grasp an object if it is placed on a support surface and the grasping point is near the support surface. Because there is no gap between the object and the support surface, the gripper fingers and the support surface will collide when the gripper attempts to power-grasp the object. To address this, we propose a pre-grasp manipulation planning method that uses two robot arms, whereby space can be secured for power grasping by rotating the object while being supported against the support surface. The objects considered in this study are appropriately shaped for a power grasp, but power grasping cannot be performed directly because the desired power-grasping location is close to the support surface on which the object is placed. First, to power-grasp the object, candidate rotation axes on the object and in contact with the support surface are derived based on a mesh model of the object. Then, for each such axis, the object pose that allows power grasping is obtained. Finally, according to the obtained object pose, the paths for rotating and power-grasping the object are planned. We evaluate the proposed approach through simulations and experiments using two UR5e robot arms with a 2F-85 gripper.

INDEX TERMS Manipulation planning, power grasp, reorientation, rotation axis.

I. INTRODUCTION

To increase the success rate of assembly tasks, it is necessary to grip objects firmly. This is because in assembly tasks, such as fastening a pin to a part, strong forces can be exerted, and slipping may more easily occur than in pick-and-place tasks. To prevent slipping, a force closure grasp, which can apply an arbitrary force and moment to an object using the fingers in contact with the object, can be considered [1]. Humans grip objects through a type of power grasping, which involves holding an object by wrapping the fingers and palm around it, with the fingers applying force against the palm [2]. In general, a power grasp is more powerfully than a precision grasp, which uses only the fingers [3]. For robots, a power grasp can be easily implemented using a gripper, which is an underactuated mechanism [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

However, even if an object is appropriately shaped, the gripper may fail to power grasp the object owing to the surrounding environment. Here, the term "appropriate shape" implies that the size or shape of the object at the location where power grasping is to be performed is such that the whole or part of the object can be placed between the gripper fingers (i.e., object size is smaller than the maximum distance between the gripper fingers). Assuming a free-floating object, power grasping can be performed on any part of the object as long as its shape is suitable. However, if the object is placed on a horizontal and flat support surface, such as a table, a power grasp may not be possible because there is no space between the object and the support surface, and thus the gripper fingers cannot be wrapped around the object. If an object is voluminous, such as a cuboid, it may be rotated and placed stably so that another side of the object is in contact with the support surface. Thereby, it is possible to secure space for power grasping at a desired point of the

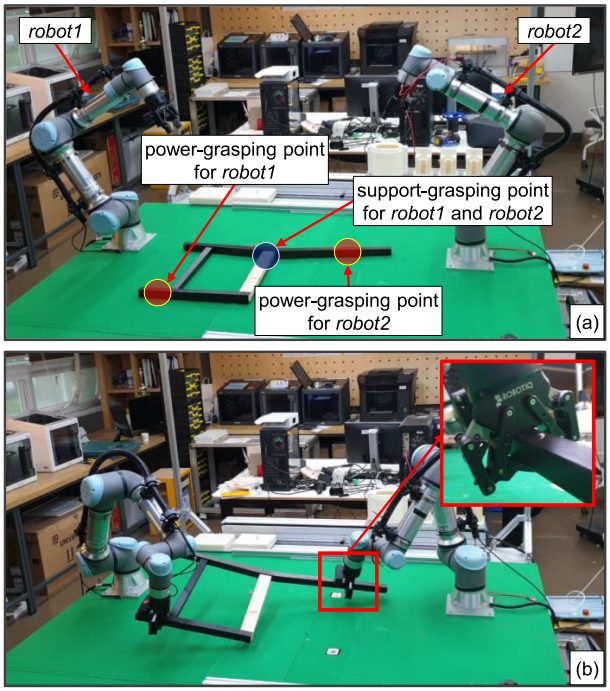

FIGURE 1. (a) Initial pose of the target object and power-grasping as well as support-grasping points; (b) power-grasping the object by two robot arms.

object. However, if an object is not voluminous, that is, one of its dimensions is significantly smaller than the others, it is difficult to create space for power grasping using this strategy. In another case, if an object is sufficiently small and light, the handover method can be used: one robot arm lifts the object to create space for grasping, and the other performs power grasping. However, this method cannot be used when the weight of the object exceeds the payload limit of the arm or when the grasping force is insufficient to lift the object stably.

In this paper, we propose pre-grasp manipulation planning so that space for power grasping may be secured (Figure 1). Pre-grasp manipulation is the process of determining preliminary manipulations for reorienting an object to an appropriate pose (position and orientation) so that a given task can be performed [5]. In the present study, it refers to the process of determining the appropriate rotated object pose so that space may be secured for power grasping, and the process of planning manipulations so that the object may be moved to the this pose, and power grasping may be finally performed. The conditions considered in this study are as follows:

(c1) The gripper has a mechanism capable of power and precision grasping.
(c2) Part of the object has a width that is smaller than the maximum distance between the gripper fingers. In addition, the height of the object is sufficient to be precision-grasped by the gripper when the object is placed stably on the support surface.
(c3) The object is assumed to be placed on a sufficiently wide horizontal and flat support surface. In addition, no parts of the surrounding environment, such as the edges of the support surface or the walls, can be used for manipulation planning, except for the support surface itself.
(c4) The object is not voluminous. This implies that one dimension of the object is assumed very small compared with the others. As shown in Figure 2, an object may be placed on the support surface in different poses, but it is not stable because it is not voluminous.
(c5) Because the gripping force of the precision grasp using only one gripper is not sufficient to lift the target object, grasping the object in an arbitrary position may cause slippage.

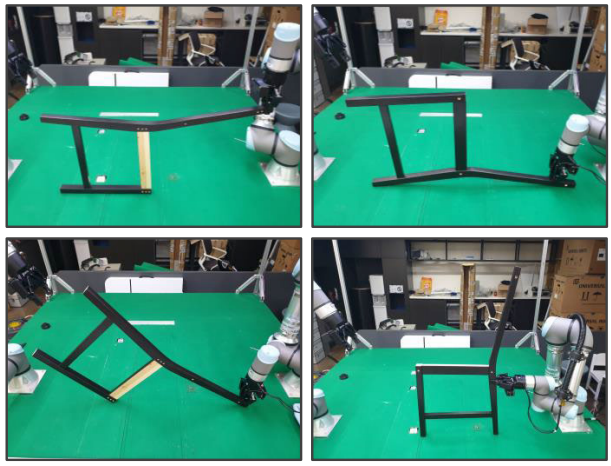

FIGURE 2. Examples of unstable object poses owing to a small contact area.

Because of (c3) and (c4), it is not possible to power-grasp the object directly if a grasp planning that does not allow collision with the surrounding environment is used. Therefore, it is necessary to determine an object pose that can secure space for two grippers to perform power grasping.

In this situation, because, initially, the target object can only be precision-grasped, and condition (c5) is assumed, it should be rotated while being supported against the surface so that its pose can be changed without slipping. To this end, we use a mesh model of the object to determine rotation-axis candidates based on the position of all vertices in the model. For each candidate axis, a target rotation angle is determined to secure space for two grippers to power-grasp the object. Subsequently, planning is performed to rotate the object using one robot arm, power-grasp it by the other arm, and finally regrasp it in a power-grasp manner by the robot arm used for the rotation. All grasping points for rotating and power grasping are heuristically predefined. Before the gripper moves to a grasping point, it first approaches a point slightly away from the grasping point and then moves to the grasping point, thus avoiding collisions (with the object) that might occur if the gripper moved directly to the grasping point.

The main contributions of this study are as follows: (i) The proposed method can be used to secure space for power grasping, even in the case of a heavy object that cannot be lifted stably using one gripper. (ii) The object is rotated to secure space while being supported against the support surface (e.g.,

a table). Therefore, except for the surface, no parts of the surrounding environment, such as the edges of the support surface or walls, are required.

The remainder of this paper is organized as follows: In Section II, we introduce related work. In Section III, we present an overview of the proposed approach. In Section IV, we propose a method for determining the candidate rotation axes. In Section V, we present an approach for obtaining object poses for power grasping. In Section VI, the integrated path is derived. The results of simulations and experiments are presented in Section VII. Finally, the paper is concluded in Section VIII.

II. RELATED WORK

Numerous studies have been concerned with modifying existing grippers or designing new grippers for effectively grasping objects. Unless a vacuum gripper is used, it is difficult to grasp a flat object placed on a support surface because it is difficult to find a valid grasp pose. To address this, a scooping grasp strategy was proposed for flat objects: sliding one finger of the gripper between the lower part of the object and the support surface, rotating the gripper to make contact with the upper part of the object, and finally gripping the object [6]. A thin nail made of smooth plastic and rubber was attached to each finger to increase the grasping success rate. In [7], a gripper was proposed that could grasp a flat object by attaching a crawler to one fingertip. If the force applied to the object while the gripper finger is closed exceeds a certain threshold, the crawler starts to move, and the object rotates in the direction of the gripper palm. Through this in-hand manipulation, the gripper can power-grasp flat objects.

Grippers have also been proposed for power-grasping general objects. In [8], a three-fingered gripper was presented, on one finger of which a small two-fingered gripper is mounted. It is thus possible to power-grasp an object by using a gripper selected according to the size of the target object, thereby separating the problem of maintaining force closure and manipulating the object. In another study, a plate was attached to each finger of an existing gripper for power grasping [9]. This plate plays the role of the palm during power grasping, and by analyzing the object model, grasp pose candidates are obtained, enabling a firm grip of the object using both the finger and the plate.

Effective power grasping can also be achieved by grasp planning. In [10], the power-grasping strategy was selected depending on the size of the object. Specifically, for small target objects, parallel fingers (index, middle, ring, and little) of the robot hand were used, but the thumb and palm were not used. For larger objects, the thumb and palm of the robot hand were also used. Another grasp planning method was based on hand–object geometric fitting [11]. Considering the surface of an object, target contacts suitable for the finger or palm of the robot hand were located for grasp planning. In the case of precision grasping, as many target contacts were located as the number of fingers. To power-grasp an object, both the finger and the palm of the robot hand were associated with the target contacts. However, the actual number of contacts was larger than that of the target contacts because the proximal links of the finger were encouraged to touch the object for more stable gripping. In another study, gripping was performed by reshaping the fitting weights to points on the finger and palm of the gripper [12]. To precision-grasp an object, more weight was applied to the fingertip area than elsewhere. In the case of a power grasp, the object was gripped by assigning more weight to the center of the finger and palm.

However, the gripper is not always possible to approach the grasp pose obtained by the aforementioned methods. For example, if an object is placed on a flat surface, or if there are obstacles, a collision may occur when the gripper moves to a grasp pose that would be valid if the surrounding environment were not considered. However, if pre-grasp manipulation is performed to relocate an object to an appropriate pose before final grasping, the gripper can grasp the object by attaining the obtained grasp pose. In [5], preparatory object rotation was presented, where a handled object was rotated about an axis perpendicular to the support surface so that the handle was in front of the robot before the object was lifted. This was motivated by the fact that a person usually rotates a handled object so that the person could hold the handle and lift the object conveniently. The authors in [13] also pre-rotated an object about an axis vertical to the support surface, but the target rotation angle was determined according to the manipulator payload calculated based on the load limit of each joint.

Another approach is to slide an object to render grasping possible. In [14] and [15], objects were classified into categories according to appearance, and each category had a pre-shape data structure comprising a robot hand, a set of starting poses, etc. When a target object was input, after the category in which the object belonged was determined, the robot hand slid and moved the object to the final region so that it could be grasped using the corresponding pre-shape. In [16], a combination of pre-grasp manipulation and transport task planning was proposed. The trajectory was optimized using the functional gradient optimization method. In addition, to increase the planning success rate, objects were released after pre-grasp manipulation, and were grasped again to perform the transport task. In [17], the configurations of the start and goal regions were sampled to determine the appropriate motion for grasping a thin object by pairing each configuration. In this strategy, the object was grasped after it was slid to the edge of the support surface. In [18], not only the edge of the support surface but also various parts of the surrounding environment (such as surface and walls) were used to grip an object. Specifically, a graph was constructed using environmental constraint exploitation and valid transitions, and manipulation planning was performed by regarding gripping motions as goal nodes.

Certain studies have been concerned with securing space for grasping by rotating an object around an axis parallel to the support surface. A method for rotating a cuboid object using hybrid force–velocity control was proposed in [19]. Grasp planning for a cuboid object in the corner of a box,

is difficult. However, using this method, the cuboid could be rotated so that space for grasping was secured. Two robot arms were used in [20] to grasp an object. One arm pushed and lifted the object using a half-sphere end-effector to secure space so that the other arm could grasp the object. To lift the object, a surface perpendicular to the table was used, and the amount of rotation of the object was determined according to the grasp location.

The regrasp method can be used to grasp an object in a desired pose. In [21], a pin perpendicular to the support surface was installed to increase the success rate of pick-and-place regrasp planning. Because an object could stably lean against the pin, the number of cases of intermediate placement of the object was increased, so that the number of regrasp candidates also increased. Accordingly, the possibility of gripping the object in the desired pose increased. Another strategy proposed in [22] was to reorient an object to an appropriate pose for performing the next task by decomposing the rotation matrix into roll, pitch, and yaw. The robot rotated the object around only one of these axes, and repeated this to move the object to the target pose. In [23], a regrasp method for gripping an object considering its usage, that is, a functional grasp, was presented. Based on a predefined functional grasping pose for each canonical model, a grasp pose enabling the use of the real object was derived. A handover motion was performed by a dual-robot arm so that the object could be grasped in this pose. In [24], pre-grasp planning was performed using multimodal motion planning. Using the regrasp graph, it was automatically determined whether to use only one robot arm or both arms to perform a given task, including pre-grasp manipulation. It was also determined whether to include handover motion if two robot arms were used.

The main difference between the present study and the aforementioned studies is the target object, which is too heavy or too large to stably lift using only one gripper (c5). Therefore, handover or in-hand manipulation is not an appropriate method to secure space for grasping. To address this, when an object is rotated to secure gripping space, such as space for power grasping in this study, the support surface is used to sustain the weight of the object. In addition, because the target object is not voluminous (c4), it is difficult to stably bring another side of the object into contact with the support surface and secure grasping space at the desired location. Therefore, the proposed method uses two robot arms. One of them, the supportive arm, is used to rotate the object to secure space for power grasping by the two grippers. The rotated object is not stable if the supportive arm releases it, and thus it must be gripped by the supportive arm until the other robot arm, the master robot arm, power-grasps it. Then, the supportive arm finally regrasps the object in a power-grasp manner.

III. OVERVIEW OF PROPOSED APPROACH

Our task is to power-grasp an object in a situation where power grasping cannot be performed directly. The proposed method requires two robot arms: a supportive and a master robot arm. The role of the latter is to power-grasp an object after the former rotates the object. Then, after the supportive arm regrasps it in a power-grasp manner, the task is completed.

Algorithm 1 Determining Rotation-Axis Candidates

Input: P_{init}, O_{obj}, Δd_{s}, Δl, $\Delta\phi$, ϕ_{obj}
Output: Π

$C_{\text{obj}} \leftarrow$ `ConvexHull`(O_{obj})
$E_{\text{obj}} \leftarrow$ `Edges`(C_{obj})
$\Lambda, \Pi \leftarrow \emptyset$
foreach $e_{\text{obj}} \in E_{\text{obj}}$ **do**
  $d_{\text{e}} \leftarrow$ `DistEdgeSupport`(e_{obj})
  $l_{\text{e}} \leftarrow$ `EdgeLength`(e_{obj})
  if $d_{\text{e}} < \Delta d_{\text{s}}$ **and** $l_{\text{e}} > \Delta l$ **then**
    $\Lambda \leftarrow \Lambda \cup \{e_{\text{obj}}\}$
  end if
end foreach
foreach $\lambda \in \Lambda$ **do**
  foreach $r_{\text{dir}} \in \{r_{\text{cw}}, r_{\text{ccw}}\}$ **do**
    $T_{\Delta\phi} \leftarrow$ `TFRotObj`$(\lambda, r_{\text{dir}}, \Delta\phi)$
    $\theta_{\max} \leftarrow 0$
    $n_{\Delta\phi} \leftarrow 0$
    while $\theta_{\max} \leq \phi_{\text{obj}}$ **do**
      $n_{\Delta\phi} \leftarrow n_{\Delta\phi} + 1$
      $\theta_{\max} \leftarrow$ `RotObj`$(O_{\text{obj}}, T_{\Delta\phi}, n_{\Delta\phi})$
      if `CheckCollision`$(\cdot)$ **or** $\theta_{\max} == \phi_{\text{obj}}$ **then**
        $S_{\text{rot}} \leftarrow$ `ScoreRotAxis`$(\lambda, r_{\text{dir}}, \theta_{\max})$
        $\Pi \leftarrow \Pi \cup \{(\lambda, T_{\Delta\phi}, S_{\text{rot}})\}$
        break
      end if
    end while
    `PlaceObj`$(O_{\text{obj}}, P_{\text{init}})$
  end foreach
end foreach

For each object, a support-grasping pose and two power-grasping poses are heuristically predefined (Figure 1 (a)). The support grasping pose is associated with rotating the object, and the power grasping pose for each robot arm is the target pose for power grasping the object.

The pipeline of the proposed method is shown in Figure 3. First, in the simulation, the mesh model of the target object is loaded in the initial pose. To find cases in which the target object and any flat surface can be in an edge–face or two vertex–face contacts, the convex hull of the object is derived. Among the edges of the convex hull, those that are not short and are close to the support surface are designated as rotation-axis candidates that may be used to sustain the object while being rotated. For each candidate, the goal angle is determined for securing space so that power grasping may be performed using two grippers. When the target object is rotated by the goal angle around the candidate rotation axis, inverse kinematics solutions for reaching the support-grasping pose and power-grasping poses are calculated. Based on these solutions, integrated paths for rotating

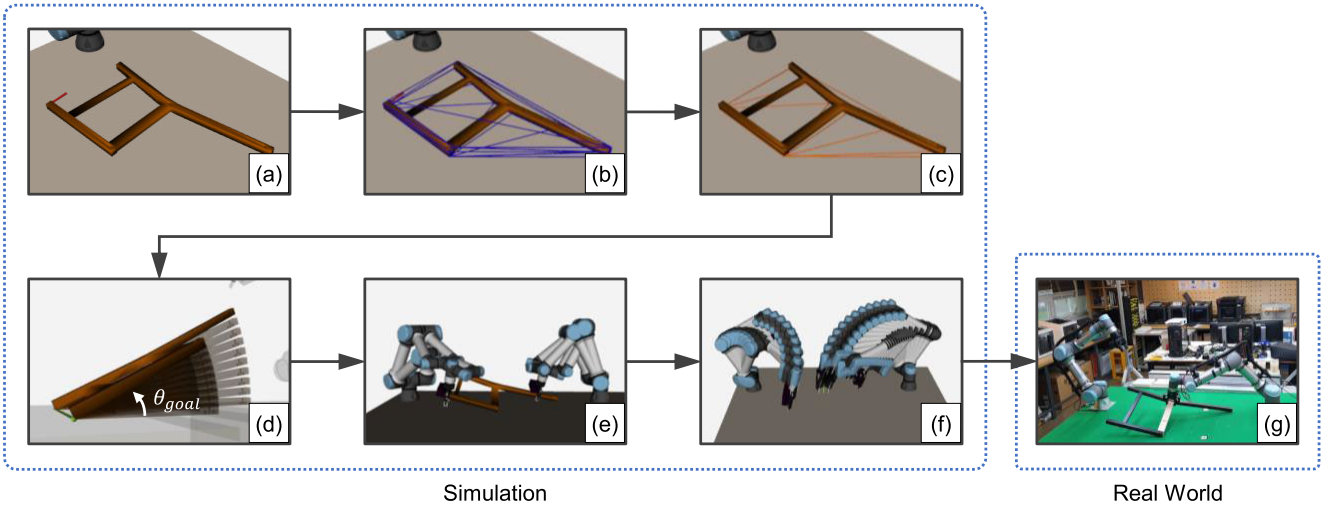

FIGURE 3. (a) Loading the mesh model of the target object in the initial pose; (b) deriving the edges (blue lines) of the convex hull; (c) finding rotation-axis candidates (orange lines); (d) calculating the goal angle θ_{goal} for each rotation-axis candidate (green arrow: one of the rotation-axis candidates used to find the goal angle); (e) finding inverse kinematics solutions of support-grasp and power-grasp pose for rotated object poses; (f) combining paths derived for rotating and power-grasping the object; (g) execution using real robot arms.

and finally power-grasping the object are planned. After the integrated paths are obtained, one of them is executed by the robot arms. In the next section, we provide a detailed explanation of the proposed method.

IV. DETERMINING ROTATION-AXIS CANDIDATES

Under condition (c5), the target object should be rotated while being sustained on the support surface so that space may secured for power grasping by the grippers. For this reason, the rotation axis must be on the object surface and in contact with the support surface while the object is rotated. Algorithm 1 is proposed to determine candidate rotation axes satisfying this requirement.

A. ROTATION-AXIS CANDIDATES

Various types of contact can occur between two objects [25]. In order that an object be rotated around an axis while being supported by the support surface, an edge–face contact must occur. In this study, the edge in this contact is the rotation axis that lies on the object surface. In addition, the support surface corresponds to the face of the contact. Rotation is possible even when an object is supported by a single vertex–face contact. However, the axis of rotation may be distorted because the part supporting the object is small. Therefore, in this study, rotation is assumed to occur only in the case of an edge–face or two vertex–face contacts. The reason that two vertex–face contacts are included is that two vertices can form an edge that may be used as a rotation-axis candidate.

To determine the rotation axis that induces an edge–face or two vertex–face contacts, the convex hull C_{obj} of the object O_{obj} is calculated, and the edges E_{obj} of C_{obj} are obtained using the `ConvexHull(·)` and `Edges(·)` functions, respectively (Figure 3 (b)). Among these edges, one that is not short and is close to the support surface is designated as a rotation axis candidate (Figure 3 (c)). To find the candidate rotation axes, in the simulation, we use the position of the endpoints for each edge $e_{\text{obj}} \in E_{\text{obj}}$ and that of the support surface parallel to the xy-plane in the world frame. These points can be derived from the mesh model of the object and the predefined height of the support surface in the simulation. Based on these positions, the `DistEdgeSupport(·)` and `EdgeLength(·)` functions return the maximum distance d_{e} between the endpoints of the edge e_{obj} and the support surface, and the length l_{e} of the edge e_{obj}, respectively. If d_{e} is within a certain value Δd_{s}, and l_{e} is larger than a threshold Δl, the corresponding edge e_{obj} becomes a candidate rotation axis. Because a very short edge e_{obj} is similar to a vertex–face contact, the threshold Δl is used to exclude it. In addition, the value Δd_{s} is used to find edges that can make contact with the support surface. In fact, the rotation-axis candidates obtained in the simulation should not be in contact with the support surface, but only close to it. If the rotation-axis candidates are in contact, then the object already collides with the support surface because the candidates are on the object surface. Therefore, the maximum rotation θ_{max} cannot be determined by checking the collisions between the object and the support surface. To resolve this, in the simulation, the object is slightly separated from the support surface (Figure 4 (a)). The detailed derivation of the maximum rotation θ_{max} is described in Section IV-B.

B. MAXIMUM ROTATION

To determine the maximum rotation θ_{max} for each rotation-axis candidate λ and rotation direction r_{dir}, the object is rotated repeatedly by $\Delta\phi$ until a collision occurs (Figure 4 (b)). The rotation direction r_{dir} for each rotation-axis candidate may be clockwise (r_{cw}) or counterclockwise (r_{ccw}). Based on λ, r_{dir}, and $\Delta\phi$, the `TFRotObj(·)` function returns the transformation matrix $T_{\Delta\phi}$ for rotating the object by $\Delta\phi$ with respect to the rotation

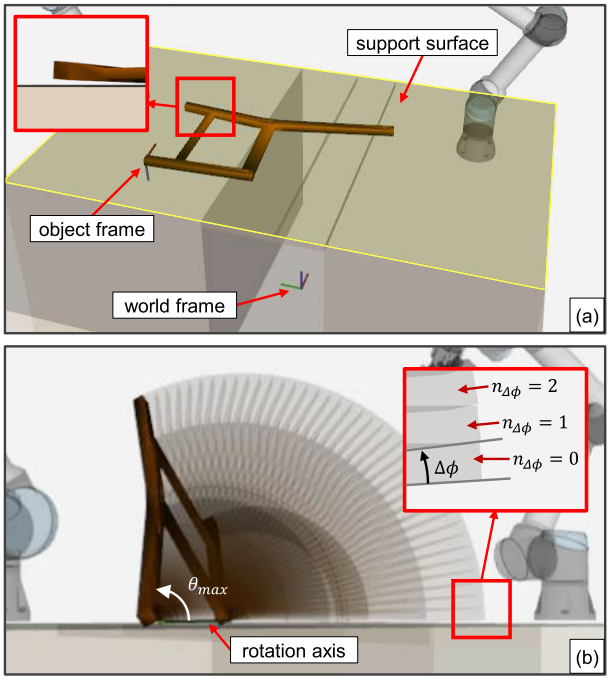

FIGURE 4. (a) Target object is placed in the initial pose P_{init}, which is slightly separated from the support surface; (b) maximum rotation θ_{max} for the rotation axis (green arrow) – $n_{\Delta\phi}$ indicates the number of rotations by $\Delta\phi$.

axis candidate λ and in the direction r_{dir}. If the object collides with the support surface before it is rotated to the limit angle ϕ_{obj}, that is, the `CheckCollision(·)` function returns `true` and θ_{max} is less than ϕ_{obj}, the current rotation angle returned from the `RotObj(·)` function is the maximum rotation θ_{max}; otherwise, the limit angle ϕ_{obj} is the maximum rotation θ_{max}.

The suitability of the rotation axis is determined based on the length λ_{length} of the rotation axis and the maximum rotation θ_{max}. If the length of the rotation axis that supports the object becomes longer, the possibility of distortion of the rotation axis is reduced; hence, a high score is assigned to long rotation-axis candidates. Furthermore, a larger maximum rotation indicates that the object is less likely to collide with the surrounding environment during rotation; therefore, higher scores are given in proportion to the maximum rotation θ_{max}. Thereby, the rotation suitability evaluation S_{rot} returned from the `ScoreRotAxis(·)` function is defined as follows for each rotation axis candidate and its rotation direction:

$$S_{\text{rot}} = \omega_{\text{l}}\lambda_{\text{length}} + \omega_{\text{m}}\theta_{\text{max}}, \tag{1}$$

where ω_{l} and ω_{m} are empirically selected weighting factors. λ, $T_{\Delta\phi}$, and S_{rot} are coupled and added to the set Π so that they can be used in Algorithm 2.

V. OBTAINING OBJECT POSES FOR POWER GRASPING

In the previous section, the rotation-axis candidates and their suitability scores were obtained. Herein, we describe the determination of rotation angles and object poses so that space may obtained for power grasping (Algorithm 2).

Algorithm 2 Obtaining Object Poses for Power Grasping

Input: $O_{\text{obj}}, \Pi, P_{\text{init}}, G_{\text{p1}}, G_{\text{p2}}, G_{\text{s}}, n_{\text{ext}}, \Delta d_{\text{a}}$
Output: Ω

$\Omega \leftarrow \emptyset$
foreach $(\lambda, T_{\Delta\phi}, S_{\text{rot}}) \in \Pi$ **do**
  $n_{\text{rot}} \leftarrow 0$
  $d_{\text{g1}}, d_{\text{g2}} \leftarrow$ `DistRotPow`$(\lambda, G_{\text{p1}}, G_{\text{p2}})$
  if $d_{\text{g1}} > \Delta d_{\text{a}}$ **and** $d_{\text{g2}} > \Delta d_{\text{a}}$ **then**
    while `CheckCollision1(·)` == `false` **do**
      `RotObjDummy`$(O_{\text{obj}}, G_{\text{p1}}, G_{\text{p2}}, T_{\Delta\phi})$
      if `CheckCollision2(·)` **then**
        $n_{\text{rot}} \leftarrow n_{\text{rot}} + 1$
        if $n_{\text{rot}} > n_{\text{ext}}$ **then**
          $P_{\text{obj}}, \theta_{\text{goal}} \leftarrow$ `ObjPoseAngle(·)`
          $C^{\text{p}}_{\text{IK}} \leftarrow$ `IKSolPow`$(G_{\text{p1}}, G_{\text{p2}})$
          $C^{\text{s}}_{\text{IK}}, R_{\text{mas}}, R_{\text{sup}} \leftarrow$ `IKSolSup`$(\lambda, d_{\text{g1}}, d_{\text{g2}}, G_{\text{s}})$
          if $C^{\text{p}}_{\text{IK}} \neq \emptyset$ **and** $C^{\text{s}}_{\text{IK}} \neq \emptyset$ **then**
            $\Omega \leftarrow \Omega \cup \{(T_{\Delta\phi}, S_{\text{rot}}, P_{\text{obj}}, C^{\text{p}}_{\text{IK}}, C^{\text{s}}_{\text{IK}}, R_{\text{mas}}, R_{\text{sup}})\}$
          else
            break
          end if
        end if
      end if
    end while
  end if
  `PlaceObj`$(O_{\text{obj}}, P_{\text{init}})$
end foreach

When one robot arm attempts to rotate the target object, the robot precision-grasps the object in the support-grasping pose G_{s}. Then, two robot arms power-grasp the object in the power-grasping poses $G_{\text{p1}}, G_{\text{p2}}$. As noted earlier, $G_{\text{s}}, G_{\text{p1}}$, and G_{p2} are heuristically predefined with respect to the object; G_{p1} corresponds to robot1, G_{p2} to robot2, and G_{s} to both robot1 and robot2 (Figure 1 (a)).

A. SUITABLE OBJECT POSES FOR POWER GRASPING

If a rotation-axis candidate is close to any of the power-grasping points for the two robot arms, a search for an object pose securing power grasping is not performed. This is because if the power-grasping point is within a certain distance Δd_{a} from the rotation axis, the gripper is highly likely to collide with the support surface even if the object continues to rotate to the maximum rotation θ_{max}. Because the endpoints of λ and the location of G_{p1} and G_{p2} are known, the `DistRotPow(·)` function returns the distance d_{g1} between λ and G_{p1}, and the distance d_{g2} between λ and G_{p2}.

When a rotation-axis candidate is not near the power-grasping points, gripper dummies are created at these points (Figure 5 (b)). The dummies are the same as the gripper attached to the robot arm, except that they can freely move in space without restrictions. When the object is not in collision with the support surface, the object and the gripper dummies

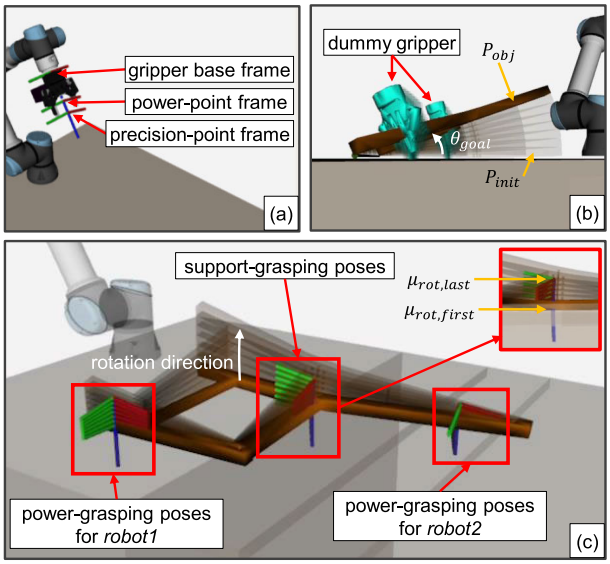

FIGURE 5. **(a) Power-point and precision-point frames are defined with respect to the gripper base frame; (b) Dummy grippers are used to check collisions between the grippers and the support surface so that P_{obj} and θ_{goal} may be determined—the dummy grippers are placed in the power-grasping poses whenever the object is rotated by $\Delta\phi$; (c) Power-grasping and support-grasping poses whenever the object is rotated by $\Delta\phi$—when the gripper grasps the object for rotating, the precision-point frame should coincide with the support-grasping pose, whereas if the object is power-grasped by the gripper, the power-grasping pose must match the power-point frame.**

are rotated together around the axis of rotation as long as a collision between the dummies and the support surface does not occur. In this case, both `CheckCollision1(·)` and `CheckCollision2(·)` functions return `false`. For stability, they are rotated several more times n_{ext} if no collision has occurred. If the dummies and the object do not collide with the support surface during the additional rotation, the `ObjPoseAngle(·)` function returns the current object pose $P_{\text{obj}} \in SE(3)$, and the goal angle θ_{goal} at which the object has rotated (Figure 5 (b)). That is, this pose represents the case in which space is secured so that two grippers can power-grasp the object in G_{p1} and G_{p2}, when the object is rotated around λ.

B. INVERSE KINEMATICS SOLUTIONS

Although the previously obtained object pose secures space for the grippers to power-grasp the object, it does not ensure that the gripper attached to each robot arm can approach G_{p1} or G_{p2}. To determine whether this is feasible, the `IKSolPow(·)` function yields inverse kinematics solutions C^{p}_{IK} for the two robot arms reaching G_{p1} and G_{p2} when the object is in P_{obj}. Specifically, the solutions induce G_{p1} and G_{p2} to coincide with the power-point frame that is defined with respect to each gripper base (Figure 5 (a) and (c)). The gripper has a mechanism that enables power-grasping the object when the power-grasping pose $G_{pi}(i = 1, 2)$ coincides with the power-point frame (c1).

Because of (c2), to rotate the object from the initial pose to P_{obj}, the gripper should precision-grap the object. Therefore, it is necessary to determine whether the gripper can reach the

Algorithm 3 Planning Integrated Paths

Input: $O_{\text{obj}}, \Omega, P_{\text{init}}, G_{\text{p1}}, G_{\text{p2}}, G_{\text{s}}$
Output: Z

$M_{\text{rot}}, Z \leftarrow \emptyset$
foreach $(T_{\Delta\phi}, S_{\text{rot}}, P_{\text{obj}}, C^{\text{p}}_{\text{IK}}, C^{\text{s}}_{\text{IK}}, R_{\text{mas}}, R_{\text{sup}}) \in \Omega$ **do**
  foreach $c^{\text{s}}_{\text{IK}} \in C^{\text{s}}_{\text{IK}}$ **do**
    $\mu_{\text{rot}} \leftarrow \texttt{RotPath}(c^{\text{s}}_{\text{IK}}, T_{\Delta\phi}, P_{\text{obj}}, P_{\text{init}}, G_{\text{s}}, R_{\text{sup}})$
    if $\mu_{\text{rot}} \neq \emptyset$ **then**
      $M_{\text{rot}} \leftarrow M_{\text{rot}} \cup \{(\mu_{\text{sup}}, S_{\text{rot}}, P_{\text{obj}}, C^{\text{p}}_{\text{IK}}, R_{\text{mas}}, R_{\text{sup}})\}$
    end if
  end foreach
end foreach
foreach $(\mu_{\text{rot}}, S_{\text{rot}}, P_{\text{obj}}, C^{\text{p}}_{\text{IK}}, R_{\text{mas}}, R_{\text{sup}}) \in M_{\text{sup}}$ **do**
  $\mu_{\text{rot,first}} \leftarrow \texttt{FirstPath}(\mu_{\text{sup}})$
  $\mu_{\text{rot,last}} \leftarrow \texttt{LastPath}(\mu_{\text{sup}})$
  $\mu_{\text{sup1}} \leftarrow \texttt{ReachPath1}(R_{\text{sup}}, \mu_{\text{rot,first}})$
  foreach $c^{\text{p}}_{\text{IK}} \in C^{\text{p}}_{\text{IK}}$ **do**
    $\mu_{\text{mas1}} \leftarrow \texttt{ReachPath2}(R_{\text{mas}}, c^{\text{p}}_{\text{IK}})$
    $\mu_{\text{sup2}} \leftarrow \texttt{ReachPath3}(R_{\text{sup}}, c^{\text{p}}_{\text{IK}}, \mu_{\text{rot,last}})$
    if $\mu_{\text{sup1}} \neq \emptyset$ **and** $\mu_{\text{mas1}} \neq \emptyset$ **and** $\mu_{\text{sup2}} \neq \emptyset$ **then**
      $Z \leftarrow Z \cup \{(\mu_{\text{sup1}}, \mu_{\text{rot}}, \mu_{\text{mas1}}, \mu_{\text{sup2}}, S_{\text{rot}})\}$
    end if
  end foreach
end foreach

support-grasping pose G_{s} when the object is in P_{obj}. First, one of the two robot arms is selected to rotate the object. If $d_{\text{g1}} > d_{\text{g2}}$, then robot1 and robot2 become the master robot arm R_{mas} and the supportive robot arm R_{sup}, respectively; otherwise, R_{mas} and R_{sup} are assigned reversely. Subsequently, inverse kinematics solutions C^{s}_{IK} are calculated for the supportive robot arm R_{sup} reaching G_{s} when the rotated object is located in P_{obj}. That is, we obtain the solutions whereby G_{s} coincides with the precision point frame defined with respect to the gripper base of the supportive robot arm R_{sup}. The `IKSolSup(·)` function performs the above process and returns C^{s}_{IK}, R_{mas}, and R_{sup}. If both C^{p}_{IK} and C^{s}_{IK} are valid, the corresponding $T_{\Delta\phi}$, S_{rot}, P_{obj}, C^{p}_{IK}, C^{s}_{IK}, R_{mas}, and R_{sup} are coupled and added to the set Ω so that they can be used in the next algorithm.

VI. PLANNING INTEGRATED PATHS

In the previous section, P_{obj} was obtained for each rotation axis candidate, and inverse kinematics solutions were derived so that the two robot arms could reach the power-grasping points. Based on the set Ω, herein, the integrated paths for pre-grasp manipulation are planned (Algorithm 3). When the robot arm attempts to reach the gripping location, namely, G_{p1}, G_{p2}, or G_{s}, it initially approaches a position slightly away from the gripping location, and then the robot reaches the gripping location. Thereby, collisions between the gripper and the object, which could occur if the gripper moved directly to the gripping location, are avoided. If valid

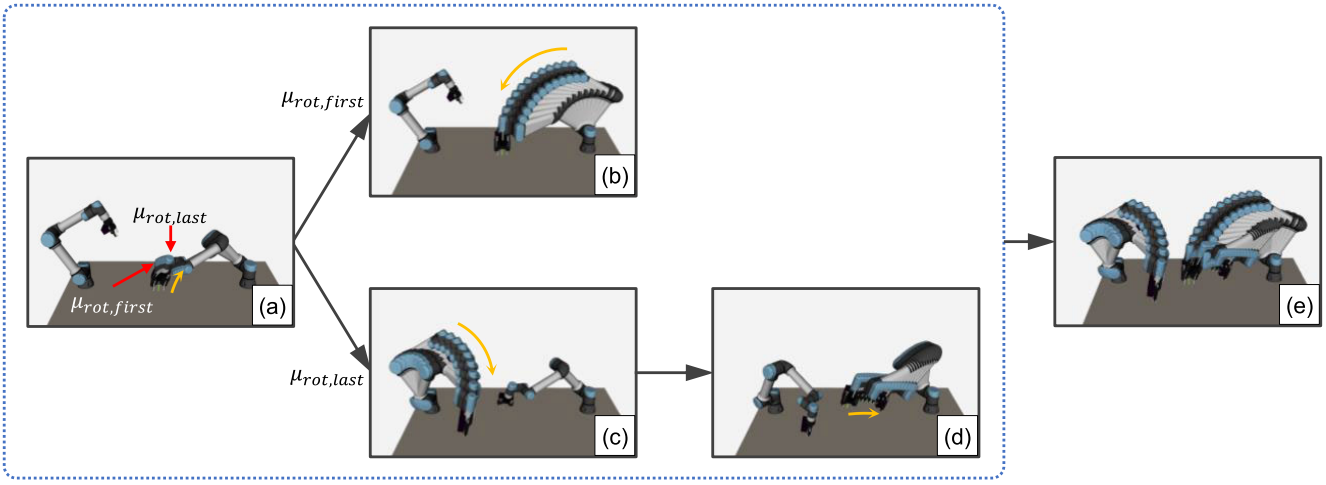

FIGURE 6. (a) Motion of the supportive robot arm μ_{rot} to rotate an object; (b) motion of the supportive robot arm μ_{sup1} from initial configuration to $\mu_{rot,first}$; (c) motion of the master robot arm μ_{mas1} from initial configuration to c^{p}_{IK} of the master robot arm; (d) motion of the supportive robot arm μ_{sup2} from (c) to c^{p}_{IK} of the supportive robot arm. (e) motion of the two robot arms for pre-grasp manipulation.

integrated paths are obtained, pre-grasp manipulation can be performed using real robots.

A. PATH FOR ROTATING THE OBJECT

The `RotPath(·)` function returns the path μ_{rot} for the supportive robot arm R_{sup} to rotate the object as follows: The obtained $c^{s}_{IK} \in C^{s}_{IK}$ is one of the solutions for reaching G_s when the object is rotated to P_{obj}. To find a path for rotating the object based on c^{s}_{IK}, we should first obtain the path for rotating the object from P_{obj} to P_{init}. In fact, as the supportive robot arm R_{sup} should rotate the object from P_{init} to P_{obj}, the order of the points in the obtained path should be reversed. In Figures 5 (b) and (c), the points of the path μ_{rot} corresponding to P_{obj} and P_{init} are $\mu_{rot,last}$ and $\mu_{rot,first}$, respectively.

To determine the path μ_{rot}, inverse kinematics solutions corresponding to all the support-grasping poses shown in Figure 5 (c) should be obtained. These poses are specified whenever the object is rotated by $\Delta\phi$. To check whether the obtained path μ_{rot} is valid, it is necessary to observe the change for each joint of the supportive robot arm R_{sup} with respect to adjacent points on the path μ_{rot}. If the change in all joint values is sufficiently small, μ_{rot} is valid. An example of path μ_{rot} is shown in Figure 6 (a).

B. PRE-GRASP MANIPULATION PATH

If the path μ_{rot} is valid, the first and last configurations of μ_{rot} should be used so that a path can be obtained, whereby the master robot arm R_{mas} can power-grasp the object, and the supportive robot arm R_{sup} can regrasp the object in a power-grasp manner. First, the `FirstPath(·)` function is used to return $\mu_{rot,first}$, which is the first configuration of the path μ_{rot}. That is, the supportive robot arm R_{sup} precision-grasps the object in P_{init}. Using $\mu_{rot,first}$, the `ReachPath1(·)` function returns the path μ_{sup1} planned to connect the initial configuration and $\mu_{rot,first}$ of R_{sup} (Figure 6 (b)). Next, to power-grasp the object using the master robot arm R_{mas}, the `ReachPath2(·)` function with $c^{p}_{IK} \in C^{p}_{IK}$ is used to obtain the path μ_{mas1} of R_{mas} that connects the initial configuration and c^{p}_{IK} of R_{mas} (Figure 6 (c)). Then, the path μ_{sup2} of the supportive robot arm R_{sup} is obtained from the `ReachPath3(·)` function using c^{p}_{IK} and $\mu_{rot,last}$ (Figure 6 (d)); $\mu_{rot,last}$ returned from the `LastPath(·)` function is the last configuration of the path μ_{rot}, that is, the supportive robot arm R_{sup} precision-grasps the object located in P_{obj}.

Based on the valid μ_{sup1}, μ_{rot}, μ_{mas1}, and μ_{sup2}, the integrated path for pre-grasp manipulation can be obtained, and finally, power-grasping can be performed by securing space for the grippers (Figure 6 (e)). The integrated path and the corresponding rotation suitability evaluation S_{rot} are coupled and added to the set Z so that pre-grasp manipulation can be performed by selecting the rotation axis candidate considering the S_{rot} value. The set Z is used to execute the actual robot arm movements.

VII. SIMULATION AND EXPERIMENT

A. RESULTS OF SIMULATIONS AND EXPERIMENTS

Two UR5e robot arms from Universal Robots, with six degrees of freedom, were installed on a table. A 2F-85 gripper from Robotiq, which is capable of precision and power grasping, was attached to each robot arm. To test the proposed method, we conducted experiments using three objects: a side part of a chair, of a shelf, and of a stool satisfying (c2) and (c4). The weight of these objects was smaller than the payload limit of the robot arm, but slipping would occur if the object was lifted after being gripped in a precision-grasping manner by only one gripper. This was because, as mentioned in (c5), the gripping force of the precision grasp is not sufficient to lift the object stably. In addition, for each object, the power-grasping and the precision-grasping pose

FIGURE 7. Examples of applying algorithms to objects in the simulation: a side part of a chair, of a shelf, and of a stool. The rows present the process of the algorithms for each object, and the columns show snapshots of applying each algorithm to the object.

were predefined. For the precision grasp, the same pose was assigned to both robot arms.

In the simulation, the proposed algorithms were implemented using MoveIt! [26], and real robot arms were used to demonstrate the proposed pre-grasp manipulation. Figure 7 shows snapshots from the simulations. The rows show the process of all algorithms for each object, and the columns represent the results of applying each algorithm to each object. A triangular mesh model (in the form of an STL file) of each object was manually constructed by measuring the actual object. The edges of the convex hull (blue lines), rotation-axis candidates (orange lines), and the process of determining the maximum rotation $\theta_{\max}$ are presented in the snapshots showing Algorithm 1 applied to each object in Figure 7. Open3D [27] was used to determine the convex hull of the mesh model. The snapshots for Algorithm 2 in Figure 7 show

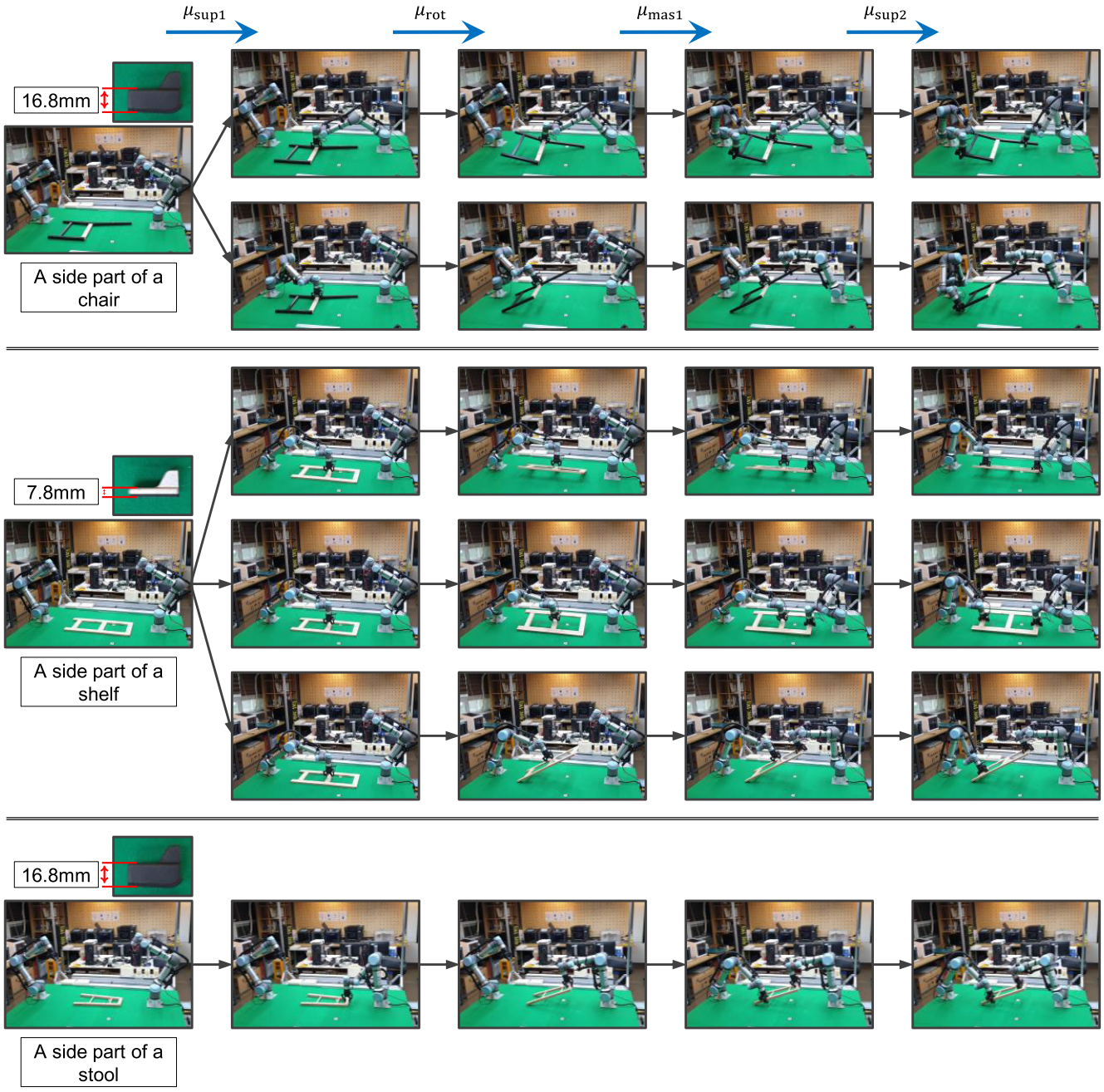

FIGURE 8. Examples of pre-grasp manipulation with real robot arms. To prevent the movement of the object after power grasping, gripper tips were selected according to the volume at the power-grasping locations of the object. The gripper tips used for a side part of a chair and of a stool were thicker than the those used for a side part of a shelf.

the pose of the rotated object P_{obj}, and inverse kinematics solutions C^{p}_{IK} and C^{s}_{IK}. The process for obtaining μ_{rot}, μ_{sup1}, μ_{mas1}, μ_{sup2}, and the integrated path are presented in the snapshots for Algorithm 3 in Figure 7.

Based on the results of the simulations, pre-grasp manipulation with real robot arms was performed, and examples are presented in Figure 8, where each column shows the final state of each path (μ_{rot}, μ_{sup1}, μ_{mas1}, μ_{sup2}) applied to each object. Depending on the volume of the object at the gripping point, the power grasp may fail to hold the object firmly. That is, if the volume of the object at the power-grasping location is not sufficiently large for the gripper fingers to apply force to the palm and restrain the object, the object cannot be gripped firmly. Among the test objects, the side part of the chair and of the stool did not have sufficiently large volume at the power grasping locations for a firm power grasp. To power-grasp these objects firmly, we used thick gripper tips so that sufficient force may be applied against the palm to hold the object even if the volume is small (Figure 8).

B. LIMITATIONS

The proposed method secures space so that two grippers may power-grasp an object in such a way that it is rotated while touching the support surface, which sustains the weight of the object. The rotation axis was not distorted in the simulation; however, in experiments with real robot arms, there are cases in which the rotation axis is distorted as the object is rotated. That is, the contact between the support surface and part of the rotation axis is broken while the object is rotated. This is because the mesh model of the object is manually constructed, and thus the shape of the real object is not accurately represented. In addition, the real object must be placed in the same pose as in the simulation so that it can be rotated using the path derived in the simulation. However, because the actual object is placed manually, an error in the pose may occur. In addition, when the real robot arm power-grasps the object, it may move arbitrarily, and thus distortion of the rotation axis can also occur. These factors cause inconsistencies between the rotation axis in the simulation and that in the real world, so that the derived path may not be appropriate for manipulating the object as planned.

To address this, the gap between the simulation and the real world should be reduced. To this end, A vision system and the non-rigid registration and pose refinement methods in [23] can be used. In addition, the in-hand pose estimation used in [23] can be applied when the object moves arbitrarily while the robot arm power-grasps the object. Based on this estimation, the actual pose of the object can be calculated such that the modified pose for power grasping can be determined. Therefore, the success rate of power-grasping can be improved.

VIII. CONCLUSION AND FUTURE WORK

We proposed pre-grasp manipulation planning involving two robot arms to secure space for power grasping. One of the conditions is that the gripping force of the precision grasp using only one gripper is insufficient to lift the object stably. Therefore, to obtain space for power grasping, the object was rotated while being supported against the support surface. To this end, rotation-axis candidates that allow the object to rotate while touching the support surface were derived from a mesh model of the object, and the rotation required for the gripper to perform power grasping was calculated for each rotation axis. The robot arm for rotating the object was determined based on the distance between the rotation axis and the power-grasping point of each arm. After the object pose allowing power grasp by two grippers was obtained, the path for rotating the object in this pose using one robot arm was determined. Finally, the path for power-grasping the object using two robot arms was derived. The proposed method was verified through simulations and experiments with two UR5e robot arms attached to a 2F-85 gripper.

We set up the conditions for the target object and environment so that existing pre-grasp manipulation methods are not suitable for obtaining space for power grasping. However, as the proposed method is valid under these conditions, it can be concluded that this paper presented another example of pre-grasp manipulation planning. Currently, the object is only rotated for power grasping, but the proposed method can be applied to any task in which the robot arm cannot grip an object at a given location.

In the future, we intend to automatically, rather than manually, determine power-grasping and support-grasping points. Thereby, it is expected that pre-grasp manipulation planning may be more effective, and the proposed method can be easily applied to other objects.

REFERENCES

[1] V. D. Nguyen, "Constructing force-closure grasps," *Int. J. Robot. Res.*, vol. 7, no. 3, pp. 3–16, 1988.

[2] J. R. Napier, "The prehensile movements of the human hand," *J. Bone Joint Surg.*, vol. 38, no. 4, pp. 902–913, 1956.

[3] M. R. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 269–279, Jun. 1989.

[4] R. Ozawa and K. Tahara, "Grasp and dexterous manipulation of multi-fingered robotic hands: A review from a control view point," *Adv. Robot.*, vol. 31, nos. 19–20, pp. 1030–1050, Aug. 2017.

[5] L. Y. Chang, G. J. Zeglin, and N. S. Pollard, "Preparatory object rotation as a human-inspired grasping strategy," in *Proc. 8th IEEE-RAS Int. Conf. Humanoid Robots (Humanoids)*, Dec. 2008, pp. 527–534.

[6] F. Lévesque, B. Sauvet, P. Cardou, and C. Gosselin, "A model-based scooping grasp for the autonomous picking of unknown objects with a two-fingered gripper," *Robot. Auton. Syst.*, vol. 106, pp. 14–25, Aug. 2018.

[7] T. Ko, "A tendon-driven robot gripper with passively switchable underactuated surface and its physics simulation based parameter optimization," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5002–5009, Oct. 2020.

[8] L. U. Odhner, C. Walker, and A. M. Dollar, "Simplifying robot hands using recursively scaled power grasps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2909–2914.

[9] S. Park, J. Baek, S. Kim, and J. Park, "Rigid grasp candidate generation for assembly tasks," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2020, pp. 589–594.

[10] M. A. Roa, M. J. Argus, D. Leidner, C. Borst, and G. Hirzinger, "Power grasp planning for anthropomorphic robot hands," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 563–569.

[11] P. Song, Z. Fu, and L. Liu, "Grasp planning via hand-object geometric fitting," *Vis. Comput.*, vol. 34, no. 2, pp. 257–270, Feb. 2018.

[12] Y. Fan and M. Tomizuka, "Efficient grasp planning and execution with multifingered hands by surface fitting," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3995–4002, Oct. 2019.

[13] L. Y. Chang, S. S. Srinivasa, and N. S. Pollard, "Planning pre-grasp manipulation for transport tasks," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2010, pp. 2697–2704.

[14] D. Kappler, L. Chang, M. Przybylski, N. Pollard, T. Asfour, and R. Dillmann, "Representation of pre-grasp strategies for object manipulation," in *Proc. 10th IEEE-RAS Int. Conf. Humanoid Robots*, Dec. 2010, pp. 617–624.

[15] D. Kappler, L. Y. Chang, N. S. Pollard, T. Asfour, and R. Dillmann, "Templates for pre-grasp sliding interactions," *Robot. Auton. Syst.*, vol. 60, no. 3, pp. 411–423, Mar. 2012.

[16] J. E. King, M. Klingensmith, C. M. Dellin, M. R. Dogar, P. Velagapudi, N. S. Pollard, and S. S. Srinivasa, "Pregrasp manipulation as trajectory optimization," in *Proc. Robot., Sci. Syst.*, Berlin, Germany, 2013.

[17] K. Hang, A. S. Morgan, and A. M. Dollar, "Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 662–669, Apr. 2019.

[18] C. Eppner and O. Brock, "Planning grasp strategies that exploit environmental constraints," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2015, pp. 4947–4952.

[19] Y. Hou and M. T. Mason, "Robust execution of contact-rich motion plans by hybrid force-velocity control," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 1933–1939.

[20] Z. Sun, K. Yuan, W. Hu, C. Yang, and Z. Li, "Learning pregrasp manipulation of objects from ungraspable poses," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2020, pp. 9917–9923.
[21] 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. Automat. (ICRA)*, May 2016, pp. 5499–5504.
[22] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Preparatory object reorientation for task-oriented grasping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 893–899.
[23] D. Pavlichenko, D. Rodriguez, C. Lenz, M. Schwarz, and S. Behnke, "Autonomous bimanual functional regrasping of novel object class instances," in *Proc. IEEE-RAS 19th Int. Conf. Humanoid Robots (Humanoids)*, Oct. 2019, pp. 351–358.
[24] W. Wan, K. Harada, and F. Kanehiro, "Preparatory manipulation planning using automatically determined single and dual arm," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 442–453, Jan. 2020.
[25] J. Xiao and X. Ji, "Automatic generation of high-level contact state space," *Int. J. Robot. Res.*, vol. 20, no. 7, pp. 584–606, Jul. 2001.
[26] I. A. Sucan and S. Chitta. *Moveit*. Accessed: Sep. 28, 2021. [Online]. Available: https://moveit.ros.org/
[27] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*.

INHYUK BAEK received the B.S. degree from the Department of Electronic Systems Engineering, Hanyang University, South Korea, in 2015, and the M.S. degree from the Department of Interdisciplinary Engineering Systems, Hanyang University, in 2017, where he is currently pursuing the Ph.D. degree with the Department of Mechatronics Engineering. His research interest includes manipulation planning.

KYOOSIK SHIN received the B.S. degree from the Department of Mechanical Engineering, Hanyang University, Seoul, South Korea, in 1983, and the M.S. and Ph.D. degrees from the Department of Mechanical Engineering, The University of Texas at Austin, in 1990 and 1995, respectively. From June 1995 to March 2008, he worked at Samsung SDS as a Product Development Consultant. From March 2008 to August 2009, he was the General Manager of the Pohang Institute of Intelligent Robotics (PIRO). In September 2009, he joined Hanyang University, Ansan, Gyeonggi-do, South Korea, as an Associate Professor with the Department of Mechanical Engineering, where he is currently a Professor with the Department of Robot Engineering. His research interests include robot manipulator design, robot design methodology, and energy efficient robot systems.

HYUNJUN KIM received the B.S. degree from the Department of Robot Engineering, Hanyang University, South Korea, in 2019. He is currently pursuing the M.S. degree with the Department of Interdisciplinary Robot Engineering Systems, Hanyang University. His research interests include mobile robot and manipulation planning.

SEUNGHOON HWANG received the B.S. degree from the Department of Mechanical Engineering, Hanyang University, South Korea, the M.S. degree from the Department of Interdisciplinary Engineering Systems, Hanyang University, and the Ph.D. degree from the Department of Mechatronics Engineering, Hanyang University, in 2020. Then, he worked as a Postdoctoral Researcher with Hanyang University from 2020 to 2021. He is currently a Postdoctoral Researcher with Arizona State University. His research interests include rehabilitation robotics, physical human–robot interaction, automation construction, and exoskeleton robotics.

ERIC DEMEESTER (Member, IEEE) received the Graduate degree in mechanical engineering with a focus on mechatronics from KU Leuven, in 1999. His Ph.D. research targeted user-adapted plan recognition and shared control for wheelchair driver assistance. Since 2018, he has been an Associate Professor at KU Leuven. He is one of the coordinators of the ACRO Research Group, which focuses on automation, computer vision and robotics in various sectors, such as manufacturing, logistics, care, pharma, nuclear, and agriculture. His research interests include probabilistic state estimation, computer vision, collision-free path planning and decision-making under uncertainty, and increasingly also machine learning.

MIN-SUNG KANG (Member, IEEE) received the B.S. and M.S. degrees from the Department of Mechanical Engineering, Hongik University, Seoul, South Korea, in 2002, and the Ph.D. degree from the Department of Mechatronics Engineering, Hanyang University, Seoul, in 2013. He worked with the Korea Institute of Industrial Technology (KITECH), as a Researcher, from 2003 to 2009. In 2014, he joined Hanyang University, Ansan, South Korea, as a Collaboration Professor with the Department of Interdisciplinary Engineering Systems. He is currently an Assistant Professor with the School of Smart Convergence Engineering, Hanyang University. His research interests include robot manipulator control, robot mechanism design, autonomous mobile robot, construction robot, computer vision, and their applications.