IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Estimation of the Caged Object's Posture under Forces Using Stepwise Geometric Calculations

Ryota Yokomura*[1], Yutong Liu*[1] and Rui Fukui*[1].

***Abstract*—In pin–hole assembly processes, precise alignment or compliance mechanisms are typically required. This paper proposes a method for connecting objects by utilizing caging to constrain their motion, enabling the insertion of a pin into a hole to adjust the allowable relative pose for assembly. This approach eliminates the need for force control, even with low-degree-of-freedom manipulators, and reduces deflection caused by misalignment during connection. Although previous research has extensively studied appropriate finger configurations for caging, the behavior of caged objects under external forces remains insufficiently investigated. Furthermore, when connecting caged objects by contact, pose estimation often requires complex collision computations that account for intricate object geometries, which are computationally expensive and may fail to converge when small gaps are present.**

To address this issue, we propose a geometric method that approximates pose changes of caged three-dimensional objects under external forces as rotations about contact points. As a representative case, we focus on objects composed of cuboid elements. The estimated results for simple objects, including caged objects with small clearances, were consistent with geometrically derived theoretical solutions, and are obtained within 0.6 seconds, indicating a practical computation time.

***Index Terms*—Contact Modeling, Caging, Compliant Assembly, Simulation and Animation, Computational Geometry**

I. Introduction

THE use of pins and holes to determine relative positions between objects is widely adopted in constructing a rigid body from multiple components [1]. This method is commonly applied in various fields, including the assembly of mechanical parts, the connection of robots [2], and modular robots' interconnection [3]. In such assembly tasks, ensuring precise relative positioning and maintaining the strength and rigidity of the joints are critical to achieving reliable connections.

When objects are assembled with misalignment in their relative positions, unintended deformation and internal stress can occur. Introducing compliance in the joint to accommodate misalignment, such as Remote Compliance Control [4] or the use of low-stiffness materials for position correction, often results in reduced stiffness and strength and may require additional structural modifications to the parts. To address these challenges, this study proposes a method to correct relative poses within the caging range by bringing caged objects into contact, as illustrated in Fig. 1.

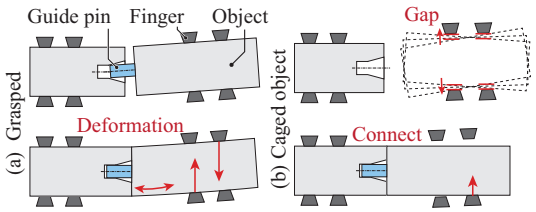

Fig. 1. Assembling a caged object to simplify the alignment and to avoid unintended deformation.

Caging is a grasping method in which an object is geometrically constrained by enclosing it with a robot hand or similar device, relying on position control rather than force control. Makita et al. have described that caged objects can be considered within a subspace of the configuration space [5]. Since gaps exist in caged objects, their positions and orientations are not fully constrained. Therefore, the pose of a caged object changes when subjected to reaction forces upon contact with another object. By designing the caging and contact forces to guide the object's pose toward a target orientation, relative pose correction can be achieved without precise positioning or force control.

Previous research has extensively explored finger configurations for caging objects [6]–[10]. However, most approaches focus on two-dimensional planes [11] [12], and studies addressing caging in three-dimensional space are limited. Additionally, there is little research, to the best of the authors' knowledge, that deeply examines the behavior of caged objects under external forces.

When determining object behaviors under contact or collision, contact analysis is generally employed. Techniques such as finite element methods (FEM) are used to calculate deformations and stresses caused by collisions, utilizing representations like polygon meshes, boundary representations (BRep), and constructive solid geometry (CSG) [13] [14]. Contact models include the penalty method, which models contacts as springs with stiffness [15], the Lagrange multiplier method, which formulates contacts as optimization problems [16], and the augmented Lagrangian method, which combines the strengths of both [17]. These approaches have been applied to simulate robotic grasping [18] [19]. However, force-based calculations for collisions are computationally complex due to penetration handling and velocity calculations, and they require extensive computational resources when dealing with objects with small gaps and fine meshes. In addition, general-

Manuscript received: July, 11, 2025; Revised October, 21, 2025; Accepted December, 1, 2025.

This paper was recommended for publication by Editor Editor A. Banerjee upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by JSPS KAKENHI Grant Number JP25K03202.

[1]Ryota Yokomura, Yutong Liu and Rui Fukui are with Department of Human and Engineered Environmental Studies, School of Frontier Science, the University of Tokyo, Chiba, 277-8563, Japan. yokomuraryota@lelab.t.u-tokyo.ac.jp

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

purpose rigid-body dynamics engines often exhibit numerical instability in narrow-clearance contacts unless unrealistically large damping parameters are introduced, making them unsuitable for quasi-static caging problems.

Unlike rigid grasps that rely on force/form closure and precise contact-force regulation, we confine the object with a cage to permit limited pose adjustment, preventing internal loads on the robot and eliminating the need for active force control. In the proposed method, the robot arm itself does not need to make fine adjustments to the object's orientation, removing the requirement for a full 6-DOF manipulator. This enables the implementation through position control alone, offering advantages in terms of system simplicity and reduced cost.

We thus propose caging as a new foundation for robotic manipulation and present a computationally efficient framework for its quantitative verification. Specifically, we hypothesize that a caged object's motion under external forces can be approximated based on geometrical calculation as a sequence of dominant rotational or translational motions. Furthermore, we demonstrate that this approximation provides a rapid and practical evaluation tool for caging-based manipulation.

The contributions of this study are as follows:

(1) Proposing a method to correct relative poses within the caging range by bringing caged objects into contact, thereby reducing internal stress caused by misalignment during assembly. (2) Implementing a computational method to estimate pose changes efficiently, even in the presence of small gaps, based on the representation of the object as a union of rectangular cuboids consisting of six faces, and the cage as multiple points, using geometric calculations involving intersections between points and surfaces. (3) Representing object slipping by introducing conditional branching based on the angle of the surface normal and external forces for cases where only rotation movement is not valid. (4) Validating the proposed method by comparing it with force-based estimation results in a two-dimensional plane and demonstrating consistency in the range of achievable poses.

II. Pose Estimation of Caged Objects Using Stepwise Geometric Computation

This section discusses a method to approximate the behavior of caged objects under forces using geometric computations.

A. Representation of Caged Objects

To achieve approximating the behavior of caged objects subjected to external forces through geometric computation, the following conditions are defined: (A) The object, Obj, is approximated as a union of rectangular "colliders," Col. (B) The structure "Cage" that grasps the object is represented as a set of multiple points. (C) An external force with a direction vector $\boldsymbol{f}$ is applied to a point ${}^{\mathrm{o}}\boldsymbol{p}_{\mathrm{f}}$ on the object. The magnitude of the external force is assumed to be significantly larger than gravity. (D) Contact points between the object and the cage are denoted as T_k. (E) Pose changes of the object within the cage are assumed to be sufficiently small, specifically within 45 degrees for each axis in ZYX Euler angles.

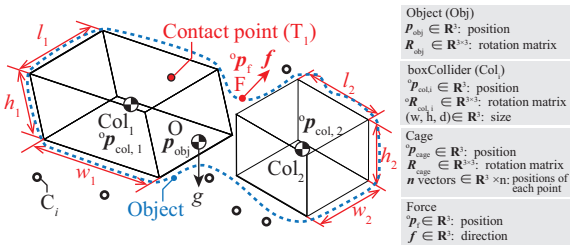

Fig. 2. Parameters and model used in the simulation.

These conditions and parameters are illustrated in Fig. 2. The object Obj is approximated by combining multiple colliders Col_i, and it has parameters representing the position of its center of gravity O as a vector $\boldsymbol{p}_{\mathrm{obj}}$ and its orientation as a rotation matrix $\boldsymbol{R}_{\mathrm{obj}}$. This is a reasonable assumption for artificial structures composed of box-like components commonly found in construction and machinery. Here, i denotes the collider's ID. Each collider is defined by its dimensions, width w_i, height h_i, and depth d_i, as well as its relative position ${}^{\mathrm{o}}\boldsymbol{p}_{\mathrm{col},i}$ and rotation matrix ${}^{\mathrm{o}}\boldsymbol{R}_{\mathrm{col},i}$ from the center of gravity O. The cage is represented as points C_i $(1 \leq i \leq n)$.

Contact points are denoted as T_i with coordinates $\boldsymbol{p}_{\mathrm{T},i}$, where i indicates the order of contact, starting from $i = 1$ for the first contact. Since the trajectories of the moving cage points and the contact points can be geometrically computed as intersections with the surfaces of colliders, this method can provide solutions for objects with small gaps, unlike mesh-based spatial discretization methods. The external force is applied to a relative coordinate ${}^{\mathrm{o}}\boldsymbol{p}_{\mathrm{f}}$ on the object in the direction of a unit vector $\boldsymbol{f}$ with a norm of 1.

B. Approximation of Object Behavior under Forces and Contact Using Geometric Transformations

The behavior of objects under external forces, gravity, and contact with the cage is approximated using geometric transformations. The movement of the object is constrained by the number of contact points, and the behavior is categorized based on this number. Initially, the conditions for non-slipping at the contact points are examined.

Fig. 3(a) illustrates the case where there are no contact points, meaning the object does not touch the cage. The object has six degrees of freedom (DOF), and it undergoes uniformly accelerated linear motion in the direction of $\boldsymbol{f}$. In the proposed method, this motion is represented as a linear translation parallel to the direction of $\boldsymbol{f}$.

Fig. 3(b) shows the case with one contact point. The object's DOF is reduced to three rotational DOFs about the contact point T_1, and it rotates around T_1 without slipping. The direction of this rotation is determined mechanically by the equation $I\dot{\omega} = M$, where I is the moment of inertia of the object, ω is the angular velocity, and M is the moment about T_1 caused by the external force.

$$\boldsymbol{p}_{\mathrm{f}} = \boldsymbol{R}_{\mathrm{obj}}{}^{\mathrm{o}}\boldsymbol{p}_{\mathrm{f}} + \boldsymbol{p}_{\mathrm{obj}} \tag{1}$$

$$\boldsymbol{a}_1 = ((\boldsymbol{p}_{\mathrm{f}} - \boldsymbol{p}_{\mathrm{T1}}) \times \boldsymbol{f})/||(\boldsymbol{p}_{\mathrm{f}} - \boldsymbol{p}_{\mathrm{T1}}) \times \boldsymbol{f}|| \tag{2}$$

In the proposed method, the direction vector $\boldsymbol{a}_1$ of the rotational axis axis 1 is determined from the condition that

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

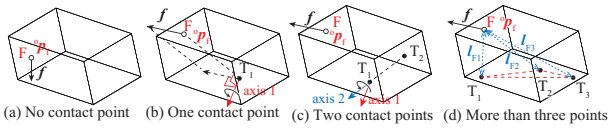

Fig. 3. Variations of moving an object considering the contact points.

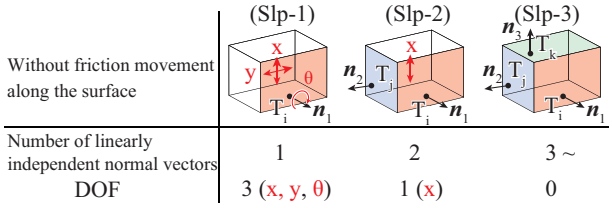

Fig. 4. DOF of an object without friction.

$\boldsymbol{f}$ is perpendicular to the line segment FT_1, as expressed in Eq. (2). The object then rotates about axis 1 in the direction of a right-hand screw until it contacts the cage.

Fig. 3(c) illustrates the case with two contact points. The object's DOF is further reduced to one rotational DOF about the axis passing through T_1 and T_2. The object rotates in the direction of the moment caused by the external force. The direction of the axis axis 2, represented by $\boldsymbol{a}_2$, is calculated by the difference between $\boldsymbol{p}_{\mathrm{T}1}$ and $\boldsymbol{p}_{\mathrm{T}2}$. However, the rotation direction may not always align with the right-hand screw rule. The rotation direction is determined based on whether the rotation induced by the moment aligns with the right-hand screw direction of axis 2. This is evaluated by comparing axis 1 and axis 2, and $\boldsymbol{a}_2 = \boldsymbol{p}_{\mathrm{T}1} - \boldsymbol{p}_{\mathrm{T}2}$ when $(\boldsymbol{p}_{\mathrm{T}1} - \boldsymbol{p}_{\mathrm{T}2}) \cdot \boldsymbol{a}_1 \geq 0$, and $\boldsymbol{a}_2 = -(\boldsymbol{p}_{\mathrm{T}1} - \boldsymbol{p}_{\mathrm{T}2})$ when $(\boldsymbol{p}_{\mathrm{T}1} - \boldsymbol{p}_{\mathrm{T}2}) \cdot \boldsymbol{a}_1 < 0$,

C. Determining Stable States and Handling Unstable Cases

Fig. 3(d) illustrates the case where there are three or more contact points. In this scenario, the object may either remain stationary or undergo rotational movement. In statics, the object remains stationary if the forces and moments from the external force and the reaction forces from the cage are in equilibrium.

In the proposed method, when there are n contact points, all pairs of two contact points T_j $(1 \leq j \leq n)$ are evaluated. The axis passing through each pair is used as a candidate rotation axis, and the object is rotated by a small angle around each axis to check for interference with the cage. If interference occurs in all rotations, the object is considered to be in a stable state.

If there exists a rotation axis without interference, the object is not in a stable state, and the process transitions back to finding new contact points. For cases with multiple candidate rotation axes, the selection of the rotation axis for the next pose change is based on the following method.

In the proposed method, as shown in Fig. 3(d), the rotation axis is chosen to maximize the total distance $l_{\mathrm{F}j}$ from the external force application point ${}^{\mathrm{o}}\boldsymbol{p}_{\mathrm{f}}$ to the two cage points forming the rotation axis. This approach assumes that, as depicted in Fig. 3(d), under the typical scenario where the object Obj is supported by the cage on the bottom face and, the load on the object is better supported at points farther horizontally from the application point F.

Although the equilibrium of forces and moments can be calculated precisely, such computations are complex. Since the purpose of this study is not to fully simulate pose transitions but to simplify the process, a heuristic method is employed.

D. Pose Changes Involving Slipping

Next, cases where the object slips at the contact points are considered. Mechanically, static friction proportional to the normal force at the contact point resists slipping, but slipping occurs when the external force exceeds this frictional limit.

In the proposed method, the angle α_i between the external force $\boldsymbol{f}$ and the surface normal vector $\boldsymbol{n}_i$ at the contact point T_i is expressed as $\alpha_i = \arccos\left((\boldsymbol{n}_i \cdot \boldsymbol{f})/(||\boldsymbol{n}_i||||\boldsymbol{f}||)\right)$ and used to classify contact behavior. A contact point is classified as a (no-slip-point) if $0 \leq \alpha_i < \theta_{\mathrm{th}}$, a (slip-point) if $\theta_{\mathrm{th}} \leq \alpha_i \leq \pi/2$, and a (contactless-point) if $\alpha_i > \pi/2$. (Contactless-point)s are excluded from pose change computations as they effectively do not involve contact.

Pose changes are determined based on the classified contact point types. If at least one no-slip point exists, the object rotates about the no-slip point, following the procedure described in Section II-C.

If no (no-slip-point) exists, meaning only (slip-points) are present, it becomes computationally complex to consider the changes in the object's DOF due to the positions and number of contact points. To simplify, the degrees of freedom are categorized based on the number of linearly independent surface normal vectors $\boldsymbol{n}_i$ associated with the (slip-points).

For one normal vector (Slp-1), where one surface of the object's collider is in contact with and slips along several (slip-points), the DOF is three, as shown in Fig. 4. The object can either translate along the surface or rotate about the normal vector $\boldsymbol{n}_1$. Assuming that the center of gravity and the application point of the external force are generally not aligned, rotational motion is presumed to occur, with rotation about one of the contact points on the surface.

For two normal vectors (Slp-2), where two surfaces are in contact and slipping, the object translates along the direction orthogonal to the normal vectors $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$.

For three normal vectors (Slp-3), where three surfaces are in contact and slipping, the object's DOF reduces to zero, and a stable state determination is performed as in Section 2.C.

III. Implementation of Pose Simulation for Caged Objects under External Forces

Based on the design principles in Chapter 2, this section describes the implementation of translational and rotational movements, as well as methods to stabilize the calculations for determining stable states.

A. Simulation Framework

Fig. 5 illustrates the overall process of the proposed calculation method. First, in step (A), the initial position and orientation under gravity are determined. Subsequently, in step (B), the position and orientation changes under the applied force are computed.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

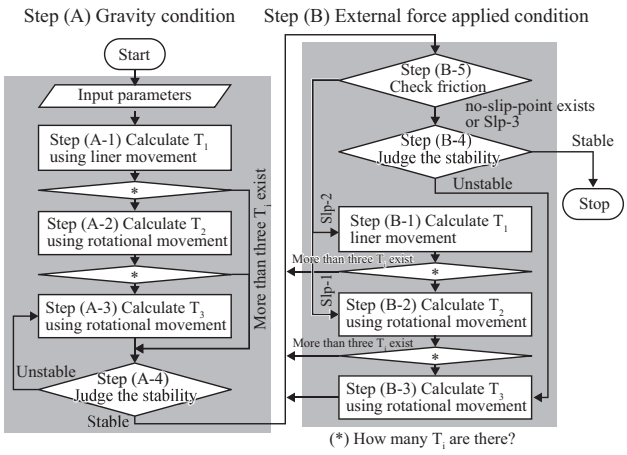

Fig. 5. Overview of the simulation process.

Algorithm 1 Calculate T_1 via translational movement

function TranslationalMovementCol, Cage, $\boldsymbol{v}_{\mathrm{dir}}$
  for all collider i in object **do**
    if No T1 exists
      $\boldsymbol{v}_{\mathrm{dir}} = (0,\ 0,\ -1)$
    pos_point_r = ${}^{\mathrm{o}}\boldsymbol{R}_{\mathrm{col},i}^{-1}(\boldsymbol{p}_{\mathrm{cage},k} - \boldsymbol{p}_{\mathrm{col},i})$
    dir = ${}^{\mathrm{o}}\boldsymbol{R}_{\mathrm{col},i} \times (-\boldsymbol{v}_{\mathrm{dir}})$
    for all plane j in collider **do**
      $L_{\mathrm{i,j}}$ = calculate from cage to plane
      if intersection is out of the plane
        $L_{\mathrm{i,j}}$ = Inf
    end for
  end for
  $minL$ = positive_min(L)
  $result$ = Cage_ID(minL)
  $\boldsymbol{p}_{\mathrm{obj}} = \boldsymbol{p}_{\mathrm{obj}} + \mathrm{minL}\ \boldsymbol{v}_{\mathrm{dir}}$
  Add T_i if Object interfered with other cages

For step (A), the following substeps are executed: (A-1) Translational movement along the direction of gravity is used to identify T_1. (A-2) Rotational movement around T_1 is performed to identify T_2. (A-3) Rotational movement around the axis passing through T_1 and T_2 is conducted to identify T_3. At this stage, (A-4) determines whether the object is in a stable state. If the object is stable, the process transitions to step (B). If further rotation is possible, new T_1 and T_2 are updated for the rotation axis, and the process returns to (A-3) to recompute T_3. After each step is executed, step (*) checks for other contact points. If three or more contact points are detected, the process proceeds to step (A-4). Slipping is intentionally disregarded in step (A), whose sole role is to determine the unique static contact pose required by step (B). Incorporating slipping at this stage would increase computational cost without affecting the resulting pose.

In step (B), the same procedure is applied with the external forces instead of gravity. Before transitioning to (B-4), step (B-5) evaluates slipping conditions at the contact points. If a no-slip-point exists, or if the number of linearly independent surface normal vectors associated with slip-points is three (Slp-3), the process proceeds to (B-4). Otherwise, slipping is assumed, and the process transitions to rotational movement (B-2) for (Slp-1) or translational movement (B-1) for (Slp-2).

B. Computing Contact Points for Translational Motion

The algorithm for deriving T_1 in step (A-1) of Fig. 5 is shown in Algorithm 1. A line parallel to the external force $\boldsymbol{f}$ is drawn from each C_j, and the intersection of the line with the six planes that constitute each Col_i is calculated. The algorithm then checks whether the intersection point lies within the plane.

If the intersection point exists within the plane, the distance $L_{\mathrm{i,j}}$ between the point on the plane and C_j is calculated. This process is repeated for all cage points and colliders.

The first contact point during translational movement is defined as the point with the smallest distance $L_{\mathrm{i,j}}$, and this point is identified as T_1. Finally, the object is moved by $L_{\mathrm{i,j}}$ such that the collider and C_1 are in contact.

C. Computing Contact Points for Rotational Motion

Algorithm 2 outlines the derivation process of T_2 and T_3 through the rotational movement of the object in Step (A-2) and Step (A-3) of Fig. 5. The vector of the rotation axis differs between Step (A-2) and Step (A-3) as shown in section 2.

The object's coordinate system is rotated such that the x-axis aligns with the rotation axis $\boldsymbol{a}_t$ $(t = 1 \text{ or } 2)$. The cage is then rotated inversely about $\boldsymbol{a}_t$ to find the contact points between the cage and collider. This approach simplifies the computation by fixing the collider while calculating the intersections between the circular trajectory of the rotating cage and the collider planes.

First, the algorithm calculates whether an intersection exists between the circular trajectory of the cage rotated about $\boldsymbol{a}_t$ and the planes of the collider. If an intersection exists, the rotation angle θ_{rot} required to reach the intersection is recorded. In Algorithm 2, $\boldsymbol{n}_{\mathrm{vec},j}$ represents the normal vector of each collider plane, and $\boldsymbol{p}_{\mathrm{cop},j}$ represents the plane's center coordinates.

This calculation is repeated for all cages and colliders, and the smallest negative angle, referred to as $\mathrm{minAngle}$, is selected. This angle is negative because the rotation is applied to the cage rather than the collider. The cage is then rotated by $-\mathrm{minAngle}$, and the resulting contact point is recorded as T_2 in Step (A-2) and T_3 in Step (A-3).

D. Stability Verification and Rotational Adjustment

Two points among n contact points are selected as the rotation axis, and the object is rotated by a small angle $\delta\theta$. To enable verification at a resolution of $0.01\ \mathrm{deg} = \pi/1800\ \mathrm{rad}$, the small angle is set as $\delta\theta = \pi/10000\ \mathrm{rad}$.

After each rotation, the algorithm checks whether the cage is inside the collider. If the cage is not inside, the rotation is considered feasible, and the rotation axis is recorded.

If interference occurs for all small rotations about every axis, the state is regarded as stable, and the calculation ends. If a non-interfering rotation axis exists, the algorithm transitions to the next rotation step, updating T_1 and T_2 accordingly.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Algorithm 2 Calculate T_2 or T_3 via rotational Movement

function RotationalMovementCol, Cage
  if only T1 exists
    $\boldsymbol{a}_1 = (\boldsymbol{p}_\mathrm{f} - \boldsymbol{p}_{\mathrm{T}1}) \times \boldsymbol{f} / ||(\boldsymbol{p}_\mathrm{f} - \boldsymbol{p}_{\mathrm{T}1}) \times \boldsymbol{f}||$
  else
    $\boldsymbol{a}_2 = (\boldsymbol{p}_{\mathrm{T}1} - \boldsymbol{p}_{\mathrm{T}2}) / ||\boldsymbol{p}_{\mathrm{T}1} - \boldsymbol{p}_{\mathrm{T}2}||$
    if rotation is not in the right hand screw direction
      $\boldsymbol{a}_t = -\boldsymbol{a}_t$
  calculate rotation matrix $\boldsymbol{R}_\mathrm{k}$ from $\boldsymbol{a}_t$ to [1;0;0]
  for all i and j in colliders and planes of collider **do**
    ${}^\mathrm{k}\boldsymbol{p}_\mathrm{cage} = \boldsymbol{R}_\mathrm{k}(\boldsymbol{n}(j) - \boldsymbol{p}_{\mathrm{T}1})$
    ${}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j} = \boldsymbol{R}_\mathrm{k}{}^\mathrm{o}\boldsymbol{R}_\mathrm{obj}\boldsymbol{n}_{\mathrm{vec},j}$
    ${}^\mathrm{k}\boldsymbol{p}_{\mathrm{cop},j} = \boldsymbol{R}_\mathrm{k}(\boldsymbol{R}_\mathrm{obj}\boldsymbol{p}_{\mathrm{cop},j} + \boldsymbol{p}_\mathrm{obj} - \boldsymbol{p}_{\mathrm{T}1})$
    $B = {}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j}.y\,{}^\mathrm{k}\boldsymbol{p}_{\mathrm{cage}.y} + {}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j.z}\,{}^\mathrm{k}\boldsymbol{p}_{\mathrm{cage}.z}$
    $C = {}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j.y}\,{}^\mathrm{k}\boldsymbol{p}_{\mathrm{cage}.z} - {}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j.z}\,{}^\mathrm{k}\boldsymbol{p}_{\mathrm{cage}.y}$
    $\theta_{\mathrm{rot}1} = \arcsin(({}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j} \cdot {}^\mathrm{k}\boldsymbol{p}_{\mathrm{cop},j} - {}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j.x} \cdot {}^\mathrm{k}\boldsymbol{p}_{\mathrm{cage}}.x)/\sqrt{B^2+C^2}) - \arctan 2(B,C)$
    $\theta_{\mathrm{rot}2} = \pi - \arcsin(({}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j} \cdot {}^\mathrm{k}\boldsymbol{p}_{\mathrm{cop},j}) - {}^\mathrm{k}\boldsymbol{n}_{\mathrm{vec},j.x}\,{}^\mathrm{k}\boldsymbol{p}_{\mathrm{cage}.x})/\sqrt{B^2+C^2} - \arctan 2(B,C)$
    Convert the output values to a real number or a value in the range between -2π to 0
    if intersection is out of the plane
      θ_rot = -Inf
  end for
  $minAngle$ = negative_max(θ_rot)
  $result$ = - Cage_ID(minAngle)
  rotate(Object, minAngle around $\boldsymbol{a}_t$)
  Add T_i if Object interfered with other cages

Algorithm 3 Determine slipage

$\alpha_i = \arccos((\boldsymbol{n}_i \cdot \boldsymbol{f})/(||\boldsymbol{n}_i||||\boldsymbol{f}||))$
if any α_i satisfies $0 \leq \alpha_i < \theta_\mathrm{th}$
  Judge the stability
else
  $slipNumber$ = count_linearly_independent_normal_vectors($\boldsymbol{n}_i$)
  $frictionNumber$ = count(slip-point)
  if $frictionNumber$ == 0
    Judge the stability
  switch on $slipNumber$
    case:1
      Rotation($\boldsymbol{a}_t \leftarrow \boldsymbol{n}_i$)
    case:2
      LinerMovement($\boldsymbol{v}_\mathrm{dir} \leftarrow \boldsymbol{f}$)
    default
      Judge the stability

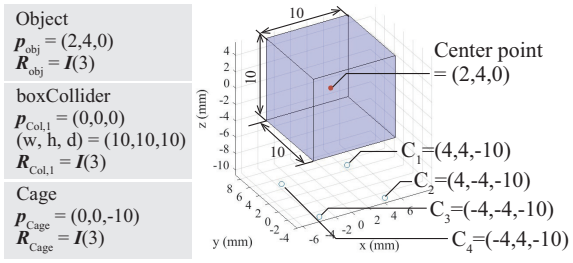

Fig. 6. Input to the simulator for testing.

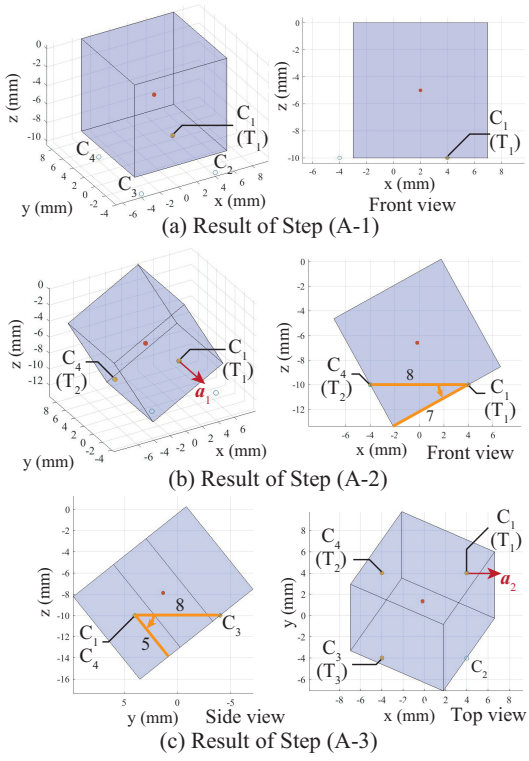

Fig. 7. Snapshot of simulation.

E. Pose Estimation under Slip Conditions

Algorithm 3 outlines the condition-based computation for pose changes of objects considering slipping. For each contact point, the algorithm calculates α_i, which indicates the angle between the normal vector $\boldsymbol{n}_i$ and the external force $\boldsymbol{f}$. If α_i is under θ_th, indicating the presence of at least one (no-slip point), the object is assumed to undergo rotational movement, and the algorithm transitions to the stability determination.

If (no-slip point) does not exist, the algorithm determines the number of linearly independent normal vectors among the (slip-points) as $slipNumber$ and the total number of (slip-points) as $frictionNumber$.

When $frictionNumber$ is 0 or $slipNumber$ is more than 3, the algorithm transitions to the stability determination When $slipNumber$ is 1, the object rotates about the normal vector of the contacting plane, following the procedure in Fig. 4 (Slp-1). When $slipNumber$ is 2, the object translates along the two contacting planes, as shown in Fig. 4 (Slp-2).

F. Verification of the simulator

To validate the proposed method, a simulator was implemented using the software Matlab. The simulator was used to compare the geometric computation results under gravity (A) with theoretical expectations. The sample object shape and the parameters are shown in Fig. 6. The movement of the object under gravity (A) as outlined in Fig. 5 was verified. Here, $\boldsymbol{I}(3)$ represents a 3×3 identity matrix.

The simulation results are presented in Fig. 7. In Fig. 7(a), the translational movement corresponding to (A-1) shows that the object moves -5 mm along the z-axis, with the contact point T_1 identified as C_1.

In Fig. 7(b), the operation corresponds to (A-2), where the object rotates 28.96 deg around the axis $\boldsymbol{a}_1 = (0, -1, 0)$ passing through point $\mathrm{C}_1 = (4, 4, 0)$, resulting in the contact point T_2 being identified as C_4. This result matches the theoretical rotation angle $\arccos(7/8) = 0.5054$ rad $= 28.96$ deg calculated for the xz-plane, as shown in Fig. 7(b).

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

In Fig. 7(c), the operation corresponds to (A-3), where the object rotates 51.32 deg around the axis $\boldsymbol{a}_2 = (1, 0, 0)$ passing through point $\mathrm{C}_1 = (4, 4, 0)$, resulting in the contact point T_3 being identified as C_3. This result aligns with the theoretical rotation angle $\arccos(5/8) = 0.8957$ rad $= 51.32$ deg calculated for the yz-plane. The simulation illustrates the object rotating around two contact points and tipping over to one side, as shown in Fig. 7(c).

Finally, in (A-4), the state was determined to be stable. The top view in Fig. 7(c) shows that the center of gravity lies within the triangle formed by the three contact points C1, C3, and C4, consistent with the physical stability of the object.

These results confirm that the proposed method successfully describes the position and pose changes of the object.

IV. Verification of Caging Object

In this section, we apply the simulator developed in Chapter 3 to the connection mechanism of the modular rail structure, ManipuRailer [20]. To assess the reproducibility, simulation results are compared with pose changes predicted by planar geometric analysis.

While validation with physical prototypes is left for future work, this study focuses on geometric simulation of object behavior to assess the proposed modeling approach.

A. Simulation Setup and Parameters

Fig. 8(a) shows the rail connection part of the ManipuRailer, a system developed to facilitate the deployment of rigid scaffolds for bridge maintenance. By combining modular rail structures that serve as scaffolding for robots, the system deploys the rail structure through rotation and linear motion, similar to a robotic arm. Subsequently, the front ends of opposing rail structures are connected to construct a scaffold.

The upper surface of the rail structure includes a central pin rack and guide rails on both sides. The V-guide rollers in the rail sliding mechanism constrain the motion to one axis while the sprocket drives the pin rack, enabling linear motion of the entire rail structure. The front end of the rail structure is equipped with four rough guide pins for positioning relative to the tapered holes on the opposite connection surface. In this verification, it is assumed that the tip of the first contacting pin is subjected to a horizontal force. Additionally, gaps of 0.5 mm on both sides are set between the V-guide rollers and the guide rails to create a caging condition for the guide rail. Because the sprocket–pin clearance is much larger than the guide-rail/roller gap, we replicate the sprocket's constraint by aligning the cages by 1 mm forward and backward of the guide rail, thereby blocking translation along the x-axis.

The shape and parameters of the model are defined as shown in Fig. 8(b), based on the scale model developed in previous studies. The threshold for sliding determination, θ_{th}, is set to 0.1 based on the friction coefficient of metal.

The connection procedure for caged objects assumed in this study is described in Fig. 9. The pin rack attached to the upper surface of the rail structure is translated using the sprocket in the rail sliding mechanism, aligning the connection surface with the pin-side surface such that the normal vector passing

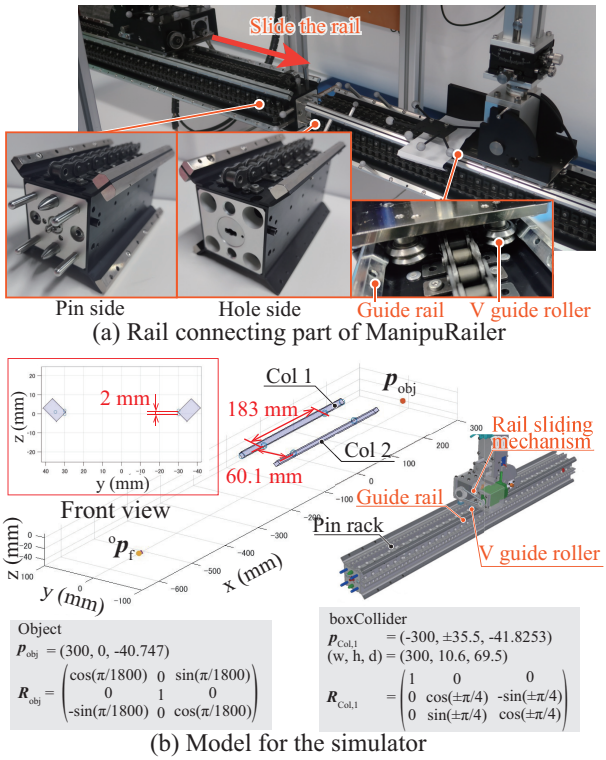

Fig. 8. Model of connecting mechanism.

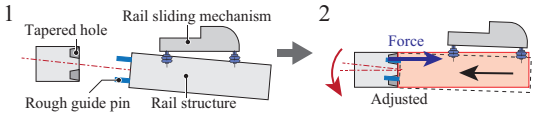

Fig. 9. Rail connecting procedure.

through the center of the connection surface aligns with the center of the pin-side surface. In this state, the positioning pins are inserted into the tapered holes to finalize the alignment. At this stage, the positioning pins receive reaction forces from the tapered holes, causing the pose of the rail structure to change within the range of poses allowed by the caging condition.

This study examines how the rail structure's pose changes due to external forces during contact when the pitch, yaw, or roll angles of the rail sliding mechanism are not parallel to the mating surface.

B. Verification in pitch angle

The simulation results are verified for consistency with the geometric model calculations in a two-dimensional plane. The pitch angle is examined to evaluate how the reaction forces at the rail structure's front end affect its pose change.

1) Theoretical value: In the rail docking experiment, the relative motion between the rail structures is geometrically constrained to a planar subspace, where out-of-plane rotations are physically restricted. Therefore, the comparison with two-dimensional geometric calculations is sufficient for validating the pose estimation accuracy.

For the scale model configuration, a horizontal force $\boldsymbol{f} = (1,\ 0,\ 0)$ is applied to the tip of the pin while the rail sliding mechanism is tilted in the pitch direction. The pose changes of

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

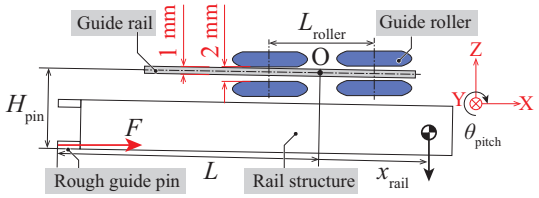

Fig. 10. Geometrical model of rail structure for changes in pitch angle.

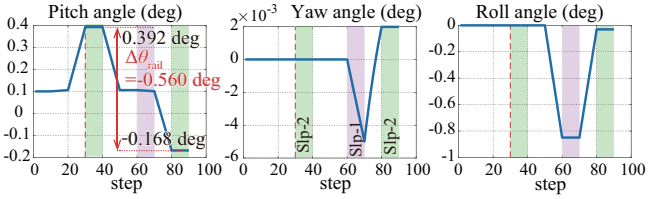

Fig. 11. Change in the object's posture when oriented in the pitch direction.

the rail structure predicted through planar geometric analysis are compared with those obtained from the simulation to confirm consistency.

First, the theoretical conditions for pose correction and the associated angular changes are derived. The geometric model is shown in Fig. 10. Point O represents the rotation center of the rail structure, and an external force F is applied to the lower end of the pin at the front end. From the cross-sectional view of the guide rails and V-guide rollers attached to the upper surface of the rail structure, the gap between the upper and lower rollers is set to 2 mm, and the rail section is 1 mm.

Let θ_{rail} denote the pitch angle of the rail structure in the fixed coordinate system, θ_{roller} the pitch angle of the rail sliding mechanism in the fixed coordinate system, and θ_{dif} the relative pitch angle of the rail structure with respect to the rail sliding mechanism. These relationships can be expressed as $\theta_{\text{rail}} = \theta_{\text{dif}} + \theta_{\text{roller}}$.

For $\theta_{\text{rail}} = 0$, the following conditions must be satisfied: (Rq.1) The external moment must allow the relative pitch angle θ_{dif} of the rail structure with respect to the rail sliding mechanism to change. (Rq.2) $\theta_{\text{roller}} = -\theta_{\text{dif}}$.

(Rq.1) implies that the rail structure rotates under a force applied to its front end. This occurs when the moments due to the external force and gravity act in opposite directions. Assuming the positive moment direction is around the y-axis, the gravitational moment M_{G} is expressed as $M_{\text{G}} = m \cdot x_{\text{rail}}$, where m is the weight of the rail structure and x_{rail} is the horizontal distance from the rotation center O of the rail structure to its center of gravity.

The external force moment T is expressed as $T = F \cdot (H_{\text{pin}} \cos\theta_{\text{rail}} - L \sin\theta_{\text{rail}})$, where $H_{\text{pin}} = 60.747$ mm is the distance from O to the lower end of the rough guide pin, and $L = 600$ mm is the distance from O to the pin tip. For $\theta_{\text{rail}} < 5$ deg, $H_{\text{pin}} \cos\theta_{\text{rail}} - L \sin\theta_{\text{rail}} > 0$, which makes $T < 0$. Therefore, to satisfy (Rq.1), $M_{\text{G}} > 0$ must hold, implying that $x_{\text{rail}} > 0$. This means the center of gravity of the rail structure must be located behind the rotation center O.

For (Rq.2), the range of θ_{dif} is derived from the cross-sectional view of the rollers and guide rails, where the gap is $\epsilon_{\text{gap}} = 1$ mm, and the roller distance is $L_{\text{roller}} = 183$ mm. This relationship is expressed as $\Delta\theta_{\text{dif}} = \frac{\epsilon_{\text{gap}}}{L_{\text{roller}}} = 0.626$ deg. Thus, to achieve $\theta_{\text{rail}} = 0$ deg, the rail sliding mechanism must satisfy $|\theta_{\text{roller}}| \leq 0.313$ deg.

2) Comparison to simulation results: Based on the conditions in Section 4.B.1, the simulation was conducted with $\theta_{\text{roller}} = 0.1$ deg to evaluate whether the rail structure achieves $\theta_{\text{rail}} = 0$ deg. Fig. 11 illustrates the pose changes of the object obtained from the simulation, represented in ZYX Euler angles. The horizontal axis represents the simulation steps, with one rotational movement divided into 10 steps. The red dashed line indicates the end of process (A), representing the initial position obtained under gravity. The purple and green regions indicate slipping, with Slp-1 corresponding to rotation about a surface-normal axis and Slp-2 to translation orthogonal to the two surface normals.

At step 40, the rail structure has rotated 0.392 deg in the pitch direction. This corresponds to the sum of the initial rail sliding mechanism pitch angle $\theta_{\text{roller}} = 0.1$ deg and the angular tolerance 0.313 deg due to the gap between the rail and the wheels.

The pitch angle changes by -0.560 deg, from 0.392 deg to -0.168 deg, which is consistent with the theoretical value of $\Delta\theta_{\text{dif}} = 0.626$ deg. This confirms that the simulator accurately reflects the behavior of caged objects. Furthermore, the computation time was 0.554 s, indicating that the simulation can be completed within a practical time frame.

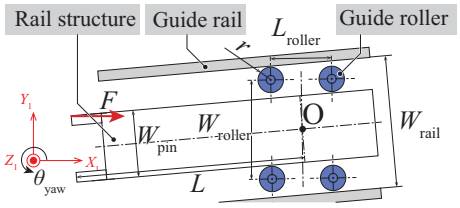

Fig. 12. Geometrical model of rail structure for changes in yaw angle.

C. Verification of rotational movement including slip

This section examines the behavior of an object significantly influenced by slipping, using rotational movement in the yaw direction as an example. The simulation results are compared with theoretical values to evaluate the discrepancies.

1) Theoretical values: For the scale model configuration, the yaw direction of the rail sliding mechanism is tilted, and a horizontal force $\boldsymbol{f} = (1, 0, 0)$ is applied to the pin tip, as shown in the model in Figure 12. Let O represent the rotation center of the rail structure, ϕ_{rail} denote the yaw angle of the rail structure in the fixed coordinate system, ϕ_{roller} denote the yaw angle of the rail sliding mechanism, and ϕ_{dif} denote the relative yaw angle of the rail structure with respect to the rail sliding mechanism. These relationships are expressed as $\Delta\theta_{\text{dif}} = \epsilon_{\text{gap}}/L_{\text{roller}} = 0.626$ deg.

As in the previous section, the following conditions must be satisfied for the rail structure to achieve $\phi_{\text{rail}} = 0$ deg:

(Rq.1) The external moment must allow the relative yaw angle ϕ_{dif} of the rail structure with respect to the rail sliding mechanism to change. (Rq.2) $\phi_{\text{roller}} = -\phi_{\text{dif}}$.

For (Rq.1), when $\phi_{\text{rail}} > 0$, the external force moment M_{F} is expressed as $M_{\text{F}} = F \cdot (W_{\text{pin}} \cos(\phi_{\text{rail}})/2 - L \sin|\phi_{\text{rail}}|)$,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

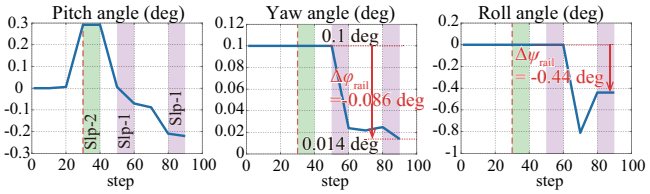

Fig. 13. Change in the object's posture when oriented in the yaw direction.

where W_{pin} is the distance between the guide pins, and L is the distance from O to the pin tip. To achieve $\phi_{\text{rail}} = 0$ deg, $M_{\text{F}} < 0$ must hold, which requires $|\phi_{\text{rail}}| < 1.91$ deg.

For (Rq.2), the range of ϕ_{dif} is calculated using the roller radius $r = 7.92$ mm, roller gap $W_{\text{roller}} = 43.2$ mm, roller distance $L_{\text{roller}} = 183$ mm, and guide rail gap $W_{\text{rail}} = 60.1$ mm. This is expressed as $\Delta\phi_{\text{dif}} = (W_{\text{rail}} - W_{\text{roller}} - 2r)/L_{\text{roller}} = 0.640$ deg. Thus, for $\phi_{\text{rail}} = 0$ deg, the rail sliding mechanism must satisfy $|\phi_{\text{roller}}| \leq 0.320$ deg.

These results indicate that within the range of $|\phi_{\text{roller}}| \leq 0.320$ deg, the reaction force brings ϕ_{rail} closer to zero.

2) Comparison with simulation results: Based on the conditions in Section 4.C.1, the simulation was conducted with $\phi_{\text{roller}} = 0.1$ deg to evaluate whether the rail structure achieves $\phi_{\text{rail}} = 0$ deg. Figure 13 shows the results when the rail sliding mechanism is offset in the yaw direction. The pitch direction changes similarly to Section 4.B, with the pitch angle decreasing in the negative direction.

Regarding yaw angle changes, as predicted, the external force causes ϕ_{rail} to approach zero, decreasing by 0.086 deg from its initial value of 0.1 deg to 0.014 deg. Notably, during the second sliding process (Slp-1), where the object rotates along the plane, the yaw angle changes significantly, demonstrating that sliding reproduction has a substantial impact on representing the object's behavior.

However, the yaw angle did not fully reach $\phi_{\text{rail}} = 0$ deg. Simultaneously, the roll angle ψ_{rail} changed to $\psi_{\text{rail}} = -0.44$ deg under the external force. This suggests that the forces acting on the rail influenced both yaw and roll angles. two-dimensional geometric calculations are useful for estimating simple rotations, but cannot account for simultaneous rotations in multiple directions, such as roll angle changes. In contrast, the proposed simulation demonstrated the ability to represent pose changes across multiple directions.

V. Conclusion

This study proposed a method to estimate the pose changes of caged objects under forces using geometric computations. The method was shown to be effective in describing the behavior of objects in three-dimensional space.

In the proposed approach, the object is represented as a union of rectangular cuboids consisting of six faces, and the cage is represented as multiple points. By calculating intersections between the point and the faces using geometric computations, the method enables stable and efficient calculations of pose changes within the cage.

Additionally, the method incorporates conditional branching based on the angle between the surface normal of the object and the external force applied to the cage points in contact with the object's surface. This allows the representation of sliding movements within the cage when rotational motion alone cannot adequately describe the object's behavior.

To validate pose changes caused by contact between caged objects, simulations were conducted using dimensions-based on a scale model of a modular segmented robot. For displacements in the pitch direction, where the influence of sliding is minimal, the results closely matched theoretical predictions derived from planar analysis.

As future work, we plan to verify whether the proposed simulation remains effective for more complex geometries composed of multiple colliders.

References

[1] K. Zhang *et al.*, "Force control for a rigid dual peg-in-hole assembly," *Assembly Automation*, vol. 37, no. 2, pp. 200–207, 2017.
[2] N. Tanishima *et al.*, "Concept and design of the caging-based debris gripper for paf capturing," *Journal of Space Safety Engineering*, vol. 7, no. 3, pp. 358–363, 2020.
[3] W. Saab, P. Racioppo, and P. Ben-Tzvi, "A review of coupling mechanism designs for modular reconfigurable robots," *Robotica*, vol. 37, no. 2, pp. 378–403, 2019.
[4] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," 1982.
[5] S. Makita and W. Wan, "A survey of robotic caging and its applications," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1071–1085, 2017.
[6] H. A. Bunis, E. D. Rimon, Y. Golan, and A. Shapiro, "Caging polygonal objects using formationally similar three-finger hands," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3271–3278, 2018.
[7] A. Rodriguez *et al.*, "From caging to grasping," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 886–900, 2012.
[8] T. Egawa *et al.*, "Two-and three-dimensional caging-based grasping of objects of various shapes with circular robots and multi-fingered hands," in *Proceedings of IECON 2015-41st Annual Conference of the IEEE Industrial Electronics Society*. IEEE, pp. 643–648.
[9] D. Kim, Y. Maeda, and S. Komiyama, "Caging-based grasping of deformable objects for geometry-based robotic manipulation," *ROBOMECH Journal*, vol. 6, pp. 1–13, 2019.
[10] Y. Liu *et al.*, "Spatial grasp synthesis for complex objects using model-based simulation," *Industrial Robot: An International Journal*, vol. 32, no. 1, pp. 24–31, 2005.
[11] W. Wan, B. Shi, Z. Wang, and R. Fukui, "Multirobot object transport via robust caging," *IEEE transactions on systems, man, and cybernetics: systems*, vol. 50, no. 1, pp. 270–280, 2017.
[12] W. Wan and R. Fukui, "Efficient planar caging test using space mapping," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 278–289, 2016.
[13] D. H. Laidlaw *et al.*, "Constructive solid geometry for polyhedral objects," in *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, 1986, pp. 161–170.
[14] J. Bender *et al.*, "Position-based simulation of continuous materials," *Computers & Graphics*, vol. 44, pp. 1–10, 2014.
[15] Zienkiewicz *et al.*, "The hierarchical concept in finite element analysis," *Computers & Structures*, vol. 16, no. 1-4, pp. 53–65, 1983.
[16] H. T. Yang, S. Saigal, A. Masud, and R. Kapania, "A survey of recent shell finite elements," *International Journal for numerical methods in engineering*, vol. 47, no. 1-3, pp. 101–127, 2000.
[17] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, "On augmented lagrangian methods with general lower-level constraints," *SIAM Journal on Optimization*, vol. 18, no. 4, pp. 1286–1309, 2008.
[18] A. T. Miller *et al.*, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
[19] B. León, Ulbrich *et al.*, "Opengrasp: a toolkit for robot grasping simulation," in *Proceedings of Simulation, Modeling, and Programming for Autonomous Robots: Second International Conference, SIMPAR*. Springer, 2010, pp. 109–120.
[20] R. Yokomura and R. Fukui, "Manipurailer: A long-reach robotic arm comprising linear and rotational joints equipped with multiple rails for deploying robots to hard-to-access," *Journal of the Robotics Society of Japan*, vol. 43, no. 8, pp. 781–790, 2025.