Rigid Grasp Candidate Generation for Assembly Tasks*

Suhan Park1, Jiyeong Baek1, Seungyeon Kim1 and Jaeheung Park1,2

Abstract—In this study, a new method is proposed to generate rigid grasp candidates for assembly tasks. Although the majority of the recent research is aimed at the generation of candidates for grasping several types of objects, our method focuses on tasks requiring precise and rigid contact. For instance, assembling furniture requires higher grasping force and precision than bin picking tasks or pick and place tasks. This means that the pose information of the object should be accurate for assembly tasks, and the object should not slip while being held. Given these constraints, we generated possible grasp pose candidates. The proposed algorithm is verified by performing real robot experiments in which furniture parts are assembled.

I. INTRODUCTION

Grippers transmit force to an object through contact between the manipulator and object. This transfer of force comprises normal contact force and static friction of the contact area. However, the maximum value of the normal contact force is limited by the torque saturation of the actuator. In humans, this limitation can be overcome by powerfully grasping the object with their palms [1]. This fixes the object in the palm and increases the contact area. Robots also contain robotic grippers with palms in order to mimic the advantages of the power grasp. In the case shown in Fig. 1 (a), red plates acting as a palm are installed in the jaw gripper to perform the furniture assembly tasks that require high force. There are soft grippers that utilize the palm, but it is still difficult to exert high power.

Figure 1 (b) illustrates our target application performing a furniture assembly task [2]. Most furniture is built with wood; assembling the parts requires rigid contact between the robot and object in order to transmit high force and torque. This rigid contact can be achieved in various ways. For the jaw gripper illustrated in Fig. 1 (a), the palm and fingers of the gripper comprise of a plane surface coated with high friction material. This condition is exploited by enlarging the contact area and thereby generating high friction. When creating grasp pose candidates, the finger surface is aligned with the object surface and the insertion depth is maximized.

Figure 2 depicts two cases in which the gripper grasps the object. It is difficult to transmit torque to the object when there are point contacts between gripper and object (Fig. 2 (a)); however, the contact in Fig. 2 (b) is rigid.

Once generated, multiple candidates can be used to re-grasp objects, grasping them with multiple grippers, or handing them over [3]–[5]. While planning the grasping motion, it must be noted that changing the object pose while grasping introduces additional uncertainties to the planned motion. These could necessitate re-planning the motion or even rendering it infeasible. Therefore, it is best to minimize changes in the pose when the robot grasps an object.

In this study, a rigid grasp pose candidate generation algorithm is proposed that considers the object model and jaw gripper geometry for assembly tasks. The majority of the previous work performs random sampling under the guide of the object model, which focuses on obtaining any possible grasps rather than utilizing the palm. The main contribution of this study is the determination of rigid grasp pose candidates for the grippers with the palm, which is obtained by analyzing the model data of the object (especially its outline) and the insertion depth of the gripper.

This paper is organized as follows. First, a survey of prior work related to grasp pose candidate generation is presented in Section II. Subsequently, the method used in this study is explained in Section III. The results of the candidates are then presented in Section IV. Finally, the conclusions are drawn in Section V.

II. RELATED WORK

Extensive research has been conducted on the generation of grasp pose candidates as a prior process of grasp tasks. Diankov and Kuffner [6] proposed OpenRAVE, which generates grasp poses by projecting cube-shaped grid points

*This work was supported by Industrial Strategic Technology Development Program (No. 20004953) funded By the Ministry of Trade, Industry & Energy(MOTIE, Korea)
1Suhan Park, Jiyeong Baek, Seungyeon Kim, and Jaeheung Park are with Department of Transdisciplinary Studies, Graduate School of Convergence Science and Technology, Seoul National University, Republic of Korea. {psh117, jy951111, ksy0711, park73}@snu.ac.kr
2Jaeheung Park is also with Advanced Institutes of Convergence Technology(AICT), Republic of Korea. He is the corresponding author of this paper.

Fig. 1. (a) Gripper used for furniture assembly tasks (red plates act as a palm), (b) Assembly task where rigid grasp of an object is required.
on the surface of the object. These points are useful when grasping the outer surface on the object, but they may not be accurately reproduced on the inside surface. Mahler et al. [7] proposed DexNet 1.0, which uniformly samples contact points on the mesh model and obtains antipodal contact points in the random directions uniformly sampled in $S^2$.

Wan et al. [8] proposed a method that samples preliminary points on the mesh model of the object according to contact size and computes the grasp candidates that have force-closure. Furthermore, Wan et al. [9] significantly developed the method by analyzing the mesh model. They improved the quality of sampled points by preprocessing the mesh and sampling antipodal points. Mousavian et al. [10] proposed 6-DOF GraspNet, which samples preliminary points on the object mesh surface and aligns the z-axis of the gripper (approach direction, see Fig. 5) with the surface normal and its orientation around the z-axis drawn from a uniform distribution. Subsequently, a variational autoencoder is utilized for refinement of sampled grasp poses. Chu et al. [11] proposed a method that predicts multiple graspable locations by utilizing ResNet in $SE(2)$. Marcus et al. [12] proposed grasp pose detection (GPD), which randomly selects preliminary points from the point cloud of the model and generates potential grasp pose candidates by aligning the normal vector of the preliminary points with the z-axis of the gripper. At each orientation, GPD pushes the gripper forward from a collision-free configuration until the fingers of the gripper touch the input point cloud.

These methods can generate grasp pose candidates for arbitrary types of objects and successfully perform pick and place tasks with cluttered objects. Since grasping in pick and place tasks and bin picking tasks has already been well developed, we focus on grasping in assembly tasks. First, the change of the object pose should be minimized when the robot grasps the object because it can cause uncertainties, as illustrated in Fig. 3. Previously existing methods that consider only the approach direction face this problem. In addition, assembly of furniture typically requires a great amount of force and most furniture parts are plane-rich. Thus, we take advantage of the antipodal-based methods and approach-based methods to increase the contact area and consider using the grippers with the palm.

III. RIGID GRASP CANDIDATE GENERATION

In this section, we explain the generation process of grasp pose candidates. The candidates in the proposed method have

![Fig. 2. (a) Grasp pose that can hardly transmit torque to the grasped object along the green line because the gripper grasps the object with point contact, (b) Grasp pose that can transmit force and torque in every direction.](image)

![Fig. 3. Grasp pose that could reorient the object when the gripper grasps the object.](image)

![Fig. 4. (a) 3D mesh model of furniture parts, (b) wireframe view of the object comprising triangles. Red lines represent the outline of the object.](image)
The generation process, where function \( \text{edges}(\mathcal{T}) \) returns a set of edges comprising two points in the input triangle plane \( \mathcal{T} \), function \( \text{normal}(\mathcal{T}) \) returns the normal vector of the input triangle plane \( \mathcal{T} \), and function \( \text{genPrePoints}(\mathcal{P}_{A1}, \mathcal{P}_{A2}, d, c) \) returns a set of preliminary points sampled in \( [\mathcal{P}_{A1} + d e, \mathcal{P}_{A2} + d e] \) and \( \mathcal{P}_{A1}, \mathcal{P}_{A2} \subseteq \mathbb{R}^3 \) is described in Algorithm 1. For every triangle \( \mathcal{T}_A \) in the mesh \( \mathcal{M} \), we can obtain its normal vector \( \mathbf{n}_A \in \mathbb{S}^2 \) and a set of point pairs of each edge \( \mathcal{E} = \{ \mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3 \} \), where \( \mathcal{P}_i = [\mathcal{P}_{A1}, \mathcal{P}_{A2}], i \in \{ 1, 2, 3 \} \). Subsequently, for each edge vector \( \mathbf{e} = \mathcal{P}_{A2} - \mathcal{P}_{A1} \), we calculate the approach direction \( \mathbf{c} = \mathbf{n}_A \times (\mathbf{e} / \|\mathbf{e}\|_2) \). Using the approach direction \( \mathbf{c} \), we create preliminary points \( \mathcal{P}_{A,new} = \{ \mathcal{P}_1, \mathcal{P}_2, \ldots \} \) between \( \mathcal{P}_{A1} + d e \) and \( \mathcal{P}_{A2} + d e \) where \( d \in \mathbb{R} \) is the distance between the center of the surface of a finger and palm so that the gripper can utilize the palm (see Fig. 5). Afterwards, function \( \text{makePair}() \) uses the preliminary points to generate pairs. Figure 6 specifies the location of the generated points. The gray points indicate preliminary points acquired from the function \( \text{genPrePoints}(\cdot) \) and yellow points represent the result of the function \( \text{makePair}(\cdot) \), which involves the pairing of contact points.

In previous works, the antipodal point was obtained by using the direction in the friction cone or the direction sampled randomly [7], [8]. However, in this study, a stricter limit is placed on finding the antipodal point in order to ensure that the object is firmly grasped. In other words, the direction vector that is used to find the antipodal point is set to \( -\mathbf{n}_A \), and the normal vector of the triangle to which the antipodal point belongs should be \( -\mathbf{n}_A \) with a tolerance \( \epsilon_n \).

**B. Point pairing**

First, we find a triangle \( \mathcal{T}_B \) with the normal vector in the opposite direction as \( \mathbf{n}_A \). Next, the antipodal point is obtained by calculating the intersection of the triangle \( \mathcal{T}_B \) and the line from the preliminary point \( \mathcal{P}_{A,new} - \epsilon \mathbf{n}_A \) where \( \epsilon > 0 \) (note that the normal vector of the triangle is set to face outward). Finally, a grasp pose candidate \( g \) is generated using the pair of points, normal vector, and given approach direction. Algorithm 2 illustrates this pairing process.

**Algorithm 2: Point pairing**

```plaintext
Function makePair(\( \mathcal{M}, \mathcal{T}_A, \mathcal{P}_{A,new}, \mathbf{c}, \mathcal{G} \)):
1. \( \mathbf{n}_A \leftarrow \text{normal}(\mathcal{T}_A) \)
2. \( \forall \mathcal{T}_B \in \mathcal{M} \) do
3. \( \mathbf{n}_B \leftarrow \text{normal}(\mathcal{T}_B) \)
4. if \( \|\mathbf{n}_A + \mathbf{n}_B\|_2 < \epsilon_n \) then
5. \( \mathcal{P}_{B,new} \leftarrow \text{intersection}(\mathcal{T}_B, \mathcal{P}_{A,new}, -\mathbf{n}_A) \)
6. \( \mathbf{R} \leftarrow [\mathbf{n}_A \times \mathbf{n}_A \times \mathbf{c}] \)
7. \( \mathbf{g} \leftarrow (\mathcal{P}_{A,new}, \mathcal{P}_{B,new}, \mathbf{R}) \)
8. \( \mathcal{G} \leftarrow \mathcal{G} \cup \mathbf{g} \)
9. end
10. end
11. end
```

**C. Rejection of impossible pairs and obtaining the grasp candidate subspace**

The generated candidates are validated by examining the collision between the object meshes and the grippers placed on the preliminary grasp candidates. The gripper is approximated by boxes and the FCL library is used to examine the collision between the gripper and object meshes [16]. After the rejection process, the valid candidates are examined to derive a graspable outline (red lines of Fig. 8 (a)). If the
candidates cover all available area on a line, we determine the line as a graspable outline. The graspable outlines produce grasp candidates subspace G wherein a grasp candidate \( g(i, x) \in G, i \in \mathbb{N} \) indicates the index of the grasp candidate line, and \( x \in \mathbb{R} \) is the position of the grasp candidate on the line.

IV. GENERATION RESULTS AND EXPERIMENTS

A. Rigid Grasp Candidates for IKEA Stefan parts

The proposed algorithm was used to generate rigid grasp pose candidates for the IKEA Stefan chair. The source codes are available on Github at https://github.com/psll17fgpg. Figure 8 (a) depicts the result of the proposed method. The first row illustrates the subspace of the grasp candidates. The second row visualizes the candidates using the 3D gripper model shown in Fig. 7. The third row shows enlargements of the red boxes of the second row. Each column illustrates the back part, seat part, apron part, and a side part of the IKEA Stefan chair. Figure 8 (b) depicts the candidate generation results using antipodal-based method [7] in the same order. The first row illustrates the preliminary points. Figure 9 shows the average insertion depth of the grippers shown in Fig. 8.

It is evident that the grasp candidates depicted in Fig. 8 (b) can barely utilize the palm of the gripper whereas the grasp candidates represented in Fig. 8 (a) completely exploit the surface of the fingers and palm, thereby achieving more rigid contact.

B. Evaluation

The proposed algorithm is evaluated by conducting an entropy evaluation to demonstrate the diversity of the generated candidates. Mahler et al. introduced the metric for grasp coverage [17]. However, our approach limits the number of relevant candidates from among the available candidates and decreases the grasp coverage. Therefore, we have used the entropy evaluation to evaluate how the proposed method can provide sufficient usable options. Addition of the direction vector aspect to previous work enables the discretization of the position and direction vector and makes it possible to define the discrete probability density function of grasp pose candidates [18]:

\[
p(i, j, k, l, m, n) = \frac{q(i, j, k, l, m, n)}{\sum_{i,j,k,l,m,n} q(i, j, k, l, m, n)}
\]

where \( q(i, j, k, l, m, n) \) denotes the number of points contained in the \([i, j, k, l, m, n]\) 6D grid of positions and direction vector of the candidates, that is, the number of points that satisfy \( i = \lfloor x/\lambda_p \rfloor, j = \lfloor y/\lambda_p \rfloor, k = \lfloor z/\lambda_p \rfloor, l = \lfloor c_x/\lambda_o \rfloor, m = \lfloor c_y/\lambda_o \rfloor, \) and \( n = \lfloor c_z/\lambda_o \rfloor \) where \( x, y, \) and \( z \) are the position of the candidate, \( c_x, c_y, \) and \( c_z \) are the components of approach direction vector \( c, \) and \( \lambda_p, \lambda_o \) are the grid size of the position and direction vector.

Three methods are compared: the proposed method, an antipodal-based method [7], and an approach-based method [12]. The proposed method samples points in the line and the distance of these sampled points is set equally to 0.025m, which provides a deterministic solution. For comparison, we conducted 100 trials of the entropy from the antipodal-based method and the approach-based method to illustrate their mean and deviation. The result is shown in Fig. 10. It is evident that the difference in entropy is small between the methods. Therefore, we can conclude that our method can also provide various and well-distributed grasping options.

C. Real robot experiments

In order to validate the proposed algorithm, furniture assembly tasks were executed with a Franka Emika Panda robot having 7 degrees of freedom (DOF). First, one of the generated candidates was randomly selected. This is used by the robot to generate a collision-free path from the initial configuration of the robot to the grasp pose. Moveit [19] is used for collision check and motion planning. The next step involves grasping the object and moving it to the assembly location. There are five frames in this scene as indicated in Fig. 11, \( \{E\}, \{O\}, \{G\}, \{A\}, \) and \( \{D\} \) are the end-effector frame of the robot, the base frame of the object, the grasp pose frame of the object, the assembly pose frame attached to the object, and the desired assembly pose frame attached to the ground, respectively. Our goal is to align \( \{A\} \) to \( \{D\} \), that is, \( W_T E \hat{T}_{G}^{-1} O_{A} = W_{T_{D}} \) where \( ATB \in SE(3) \) is the transformation matrix from frame \( \{A\} \) to frame \( \{B\} \) (we assume that the end-effector frame \( \{E\} \) is equal to \( \{G\} \)). After aligning \( \{A\} \) with \( \{D\} \), peg-in-hole control is used to assemble the object to another object fixed to the ground [20], [21].

These tasks were conducted on the apron part, into which two pins were inserted. The aim was to attach the apron part to the side part using the pins. The experiments were successfully performed: a grasp success rate of 100% and an assembly success rate of 75% were exhibited in 20 trials. The peg-in-hole task failed when the distance between the grasp frame \( \{G\} \) and assembly frame \( \{A\} \) was too extensive. These failures occurred because of kinematic uncertainties. Therefore, we can prioritize the grasp pose, which involves contact near the assembly point, in order to minimize uncertainties in the real application.

V. CONCLUSIONS

We proposed a grasp pose candidate generation algorithm to enable better grasping of known objects. We validated the results of this algorithm experimentally. The proposed
Fig. 8. Grasp pose candidates are drawn on the object model. (a) The result of the proposed method, (b) the result of the antipodal-based method. The first row: (a) the subspace of the grasp candidates (b) preliminary contact points. The second row: sampled grasp pose candidates. The third row shows enlargements of the red boxes of the second row.
algorithm is applicable for plane-rich objects in which the gripper can access a large contact area and utilize the palm area, which helps the robot transfer force to the object. Furniture assembly tasks can be executed using the candidates generated by our algorithm. While planning constrained motion that involves grasping, the subspace produced by the proposed method can be used to alleviate the constraints required for re-grasping. Typically, grasping in $SE(3)$ space requires equality constraints of six DOF, which can be relaxed by replacing the equality constraint of one DOF with an inequality constraint because the candidates in the subspace are represented by the position in the graspable line.

Fig. 9. Average insertion depth of the grippers shown in Fig. 8.

Fig. 10. Entropy comparison of three different methods with four object models.

Fig. 11. Real robot experiments. ($\{O\}$ denotes the base frame of the object, $\{E\}$ denotes the end-effector frame of the robot, $\{G\}$ denotes the grasp pose frame of the object, $\{A\}$ denotes the assembly pose frame attached to the object, and $\{D\}$ denotes the target assembly pose frame attached to the ground.)

References


