

HANDLING PERCEPTION UNCERTAINTY IN SIMULATION BASED SINGULATION PLANNING FOR ROBOTIC BIN PICKING

Nithyananda B. Kumbla

Department of
Mechanical Engineering
University of Maryland
College Park, MD USA
nkumbla@umd.edu

Shantanu Thakar

Department of Aerospace
& Mechanical Engineering
University of Southern California
Los Angeles, CA USA
sthakar@usc.edu

Krishnanand N. Kaipa

Department of Mechanical
& Aerospace Engineering
Old Dominion University
Norfolk, VA USA
kkaipa@odu.edu

Jeremy Marvel

Intelligent Systems Division
National Institute of
Standards and Technology
Gaithersburg, MD USA
jeremy.marvel@nist.gov

Satyandra K. Gupta*

Department of Aerospace
& Mechanical Engineering
University of Southern California
Los Angeles, CA USA
skgupta@usc.edu

ABSTRACT

Robotic bin picking requires using a perception system to estimate the pose of the parts in the bin. The selected singulation plan should be robust with respect to perception uncertainties. If the estimated posture is significantly different from the actual posture, then the singulation plan may fail during execution. In such cases, the singulation process will need to be repeated. We are interested in selecting singulation plans that minimize the expected task completion time. In order to estimate the expected task completion time for a proposed singulation plan, we need to estimate the probability of success and the plan execution time. Robotic bin picking needs to be done in real-time. Therefore, candidate singulation plans need to be generated and evaluated in real-time. This paper presents an approach for utilizing computationally efficient simulations for generating singulation plans. Results from physical experiments match well with predictions obtained from simulations.¹

1 Introduction

The use of robots for bin picking applications has been effective to singulate a wide variety of parts when the estimate of the part posture in the bin is accurate or the part is of simple geometry and is not entangled with neighboring parts. Singulation

is defined as the concatenation of five stages, including approach, grasp, extract, transport, and drop-off. Singulation planning consists of first synthesizing candidate plans and then using an evaluation method to score each candidate plan and select the best one for execution. A computationally efficient evaluation method is a crucial requirement in this context since all the candidate plans need to be evaluated in near real-time so that the robot's idle time during planning can be minimized. This paper is focused on the evaluation aspect of singulation planning.

Fig. 1 illustrates an example of a singulation plan. If the part posture is known accurately, then the singulation plan will succeed if the robot constraints are not violated. If the estimated part posture is uncertain (as illustrated in Fig. 2), the singulation planner might fail to extract the part from the bin. Fig. 3 illustrates possible outcomes of a singulation plan under perception uncertainty. The frame denoted by \hat{x} , \hat{y} , and \hat{z} indicates the grasp point computed based on the posture estimate obtained from the perception system and the frame denoted by x_p , y_p , and z_p indicates the actual grasp point on the part. The gripper frame is denoted by x_g , y_g , and z_g . The outcome of a singulation plan under perception uncertainty can be one of the following:

1. **Successful grasp.** Some uncertainties get updated during the execution of the singulation plan. Fig. 3(a) illustrates a case in which the uncertainty in rotation about z_g is reduced when the grippers are closed. Some uncertainties might propagate to further stages of the singulation task. Fig. 3(b) illustrates a case in which the uncertainty in rotation about y_g and translation in z_g is propagated after a successful grasp.

*Address all correspondence to this author.

¹DISCLAIMER: Any commercial product or company name in this paper is given for informational purposes only. Their use does not imply recommendation or endorsement by NIST or University of Maryland or Old Dominion University or University of Southern California.

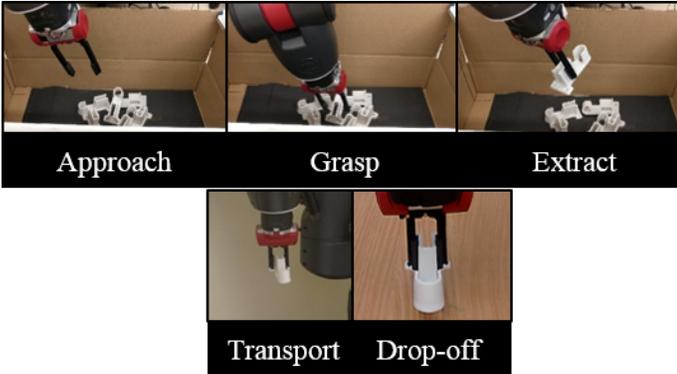


FIGURE 1. Example of a singulation plan



FIGURE 2. Illustration of perception uncertainty in the part location - Estimated posture might differ from the actual part posture.

2. **Failure due to collision.** Fig. 3(c) illustrates uncertainty in the y_g direction leading to collision with the gripper.
3. **Failure due to grasp miss.** - Fig. 3(d) illustrates high uncertainty in translation along the x_g direction leading to a grasp miss.

In case of Fig. 3(a) and Fig. 3(b), the time for completing the task would be equal to the time required to execute the singulation plan once; that is, T_e . There can be some side effects caused when the exact location of the part is not known within the gripper. For instance, the robot has to account for high tolerances for waypoints in the singulation plan so that the part held within the gripper does not collide with the bin during extraction. In addition, these propagated uncertainties might lead to a drop-off with bouncing effects on the part and the part settling outside acceptable tolerance limits. This causes uncertainty to propagate to the tasks following bin picking. In case of Fig. 3(c) and Fig. 3(d), the robot has to execute the singulation plan again.

Since the outcome of the singulation plan is probabilistic in the presence of perception uncertainty, p and T_e can be considered as the main factors during the evaluation of the expected

task completion time. The singulation plan being different for every grasp strategy defined for a part introduces the need to estimate expected task completion time online after each plan has been generated. T_e for a singulation plan can be computed online, using the joint angle differences at consecutive waypoints and joint velocity. Monte Carlo simulations can be used for online estimation of success probability. After all the candidate plans are evaluated, the singulation plan with the least expected task completion time will be selected for execution. This paper is an extended version of our previously published conference paper [1] and presents an approach for utilizing computationally efficient simulations for generating singulation plans.

2 Related Work

Pose uncertainty is a well-addressed problem in bin picking applications. Many previous attempts on a systems approach to bin picking mainly focused on the perception problem [2–8], decoupled from motion planning of the robot. Akizuki et al. [9] used a simulator to compute the observability factor of a 3D vector pair from multiple viewpoints to estimate the pose of the part. In reality, the pose estimate from the perception system is not accurate. There is always an uncertainty associated with the estimate. The work done by Fuchs et al. [10] assumed significant uncertainty in object pose estimate and the object was grasped only when the reliability of the pose hypothesis was below a certain threshold. Otherwise, a different camera was used to have a different viewpoint. Harada et al. [11] applied probabilistic properties to the pick-and-place motion planner of an object. They planned a pick-and-place motion with a set of regions in combination with the probabilistic properties. Liu et al. [12] presented a directional, chamfer-matching-based, object localization and pose estimation in the presence of heavy clutter in the bin. Papazov et al. [13] evaluated recognition hypothesis quality by defining an acceptance function, comprising a visibility term and a penalty term. Another work by Pronobis and Caputo [14] used a support vector machine approach to quantify the level of confidence in performing a visual, place-recognition task.

Many attempts have been made to define grasp strategies offline and evaluate them online using collision checks. A review of analytical approaches to grasp synthesis can be seen in [15]. A survey on data-driven grasp synthesis can be found in [16]. Dupuis et al. [17] presented a two-finger grasp generation method and target selection for bin picking of randomized parts. They generated a dense set of grasp points and evaluated them using factors like the sensitivity of grasp point with a small variation in grasp point and based on the feasibility and stableness of neighboring grasps. Ellekilde et al. [18] used learning methods to improve success probability of grasps for bin picking. They chose a set of grasps offline and applied the learning process for a day or two in an industrial environment to improve success

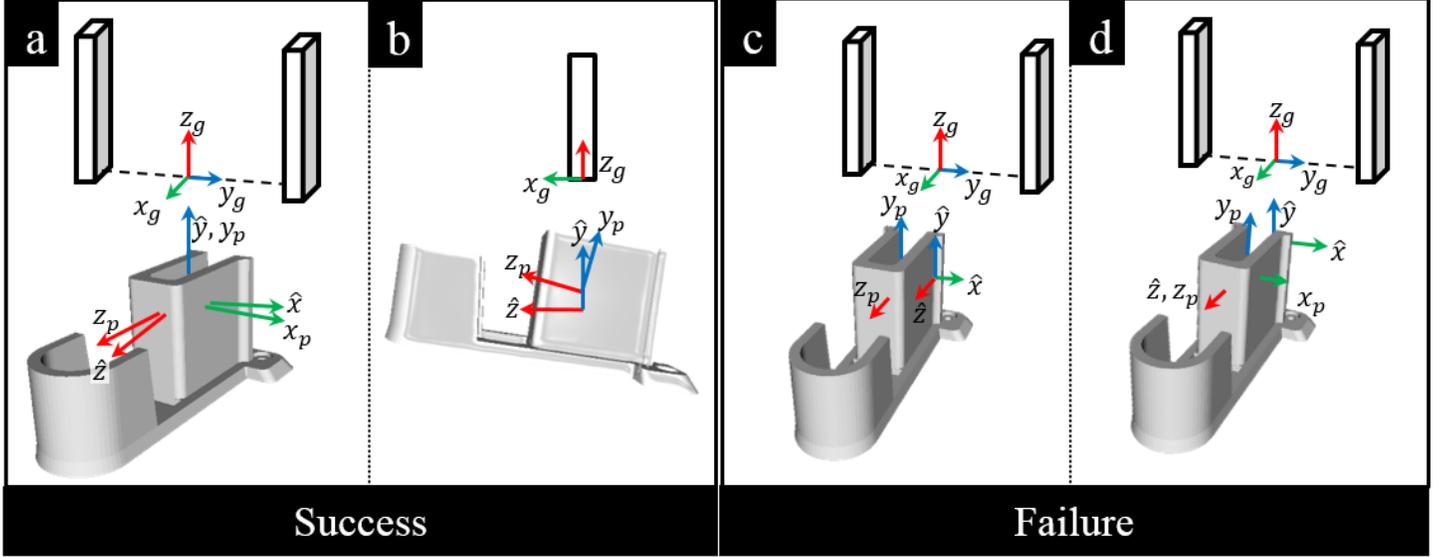


FIGURE 3. Possible outcomes of a singulation plan: [a] Successful grasp; certain uncertainties like rotation about z_g being updated during grasp. [b] Successful grasp; certain uncertainties like rotation about y_g and translation along z_g will propagate through the successive stages. [c] Failure due to collision with the part upon lowering the gripper due to uncertainty in translation along y_g . [d] Failure due to grasp miss as the uncertainty in translation along x_g is high.

probability. Kendall et al. [19] used a Bayesian convolutional neural network to estimate relocalization uncertainty and regress the pose obtained from the camera. Zheng et al. [20] presented a force closure analysis approach to handle uncertainties in friction and contact position. Berenson et al. [21] used Task Space Regions (TSRs) and modified them to handle pose uncertainty.

Chang et al. [22] developed a framework for representing possible strategies of interactive singulation of items from a pile of cluttered objects. They used a perception module to evaluate the outcome of pushing actions. In our previous work [23, 24], any failure in the automatic pose estimation was handled with the help of human assistance. The human operator used an interface to estimate the pose of the part and sent it back to the robot. Simple sensorless fine positioning moves were used after the part was dropped off to correct uncertainties present in the final part location. The extraction planning presented in [25] used the relationship between object geometries and executed simple extraction motions to resolve occlusions. In [26], a data-driven approach was used to evaluate singulation plans based on perception uncertainty, grasp quality, approach quality, and extraction quality. The singulation plan with minimum probability of failure was chosen for execution.

3 Problem Formulation

Consider a bin with known dimensions containing an instance of the part chosen for singulation. Define a finite set of

grasp strategies $G = [G_1, G_2 \dots G_n]$ for the chosen part. Let the estimate of the six-dimensional (6D) pose of the part be represented as $P = \{x, y, z, \alpha, \beta, \gamma\}$. Let $\mathbb{U} = [\mathbb{U}_t, \mathbb{U}_o]$ represent the uncertainty in P , where \mathbb{U}_t represents uncertainty in position estimate and \mathbb{U}_o represents uncertainty in orientation estimate.

Given P and \mathbb{U} , the task is to build a singulation plan \mathbb{S} for every strategy in G , evaluate its expected task completion time by estimating p and T_e , and execute the best plan based on this evaluation. Expected task completion time, $\mathbb{E}(T_c)$, can be computed as

$$\mathbb{E}(T_c) = pT_e + (1-p)(2pT_e) + (1-p)^2(3pT_e) + \dots, \quad (1)$$

where pT_e estimates the time required for plans that succeed in the first attempt, $(1-p)(2pT_e)$ estimates the time required for plans that succeed in the second attempt, and so on. For an arithmetico-geometric sequence, the sum to infinite terms is given by

$$\lim_{n \rightarrow \infty} S_n = \frac{a}{1-r} + \frac{dr}{(1-r)^2}. \quad (2)$$

Substituting $a = pT_e$; $r = (1-p)$; $d = pT_e$, we get

$$\mathbb{E}(T_c) = \frac{pT_e}{1-(1-p)} + \frac{pT_e(1-p)}{(1-(1-p))^2},$$

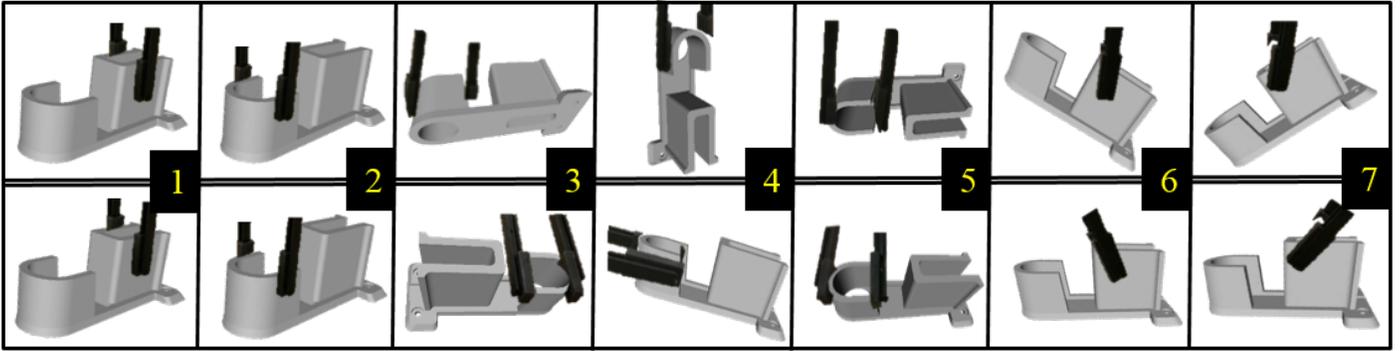


FIGURE 4. Illustration of seven grasp strategies for the chosen part (row 1) and corresponding seven drop-off strategies (row 2).

$$\mathbb{E}(T_c) = \frac{T_e}{p}. \quad (3)$$

To select the singulation plan \mathbb{S}_{best} corresponding to the grasp strategy with minimum $\mathbb{E}(T_c)$, we use the following approach.

1. Obtain the point cloud of the scene, the pose of the part to be singulated, and the drop-off location.
2. For every available grasp strategy, compute the grasp points, grasp orientation, and the approach vector based on the pose of the part. Compute the drop-off offset and the drop-off orientation for the corresponding drop-off strategies (see Section 4 for details).
3. Generate singulation plan for every grasp strategy and check if the generated plans are feasible (see Section 5 for details).
4. Invoke Monte Carlo simulator to estimate p and estimate T_e (see Section 6 for details). Compute $\mathbb{E}(T_c)$ of every singulation plan using Eqn. 3.
5. Choose the singulation plan which has the minimum $\mathbb{E}(T_c)$.

4 Representing Singulation Plans

4.1 Grasp Strategies

Grasp strategy is the strategy with which the gripper approaches the part and grasps it. Grasp strategy directs the singulation planner to orient the gripper to a specific point on the part during approach and grasp phases. In this paper, we have considered seven grasp strategies for the chosen part (Fig. 4 - Row 1) to be able to approach and grasp the part in seven different ways. Each grasp strategy can be parameterized by the following four components:

1. **Grasp offset** (G_x, G_y, G_z) specifies the translational offset between the part origin and the grasp point expressed in the part frame.

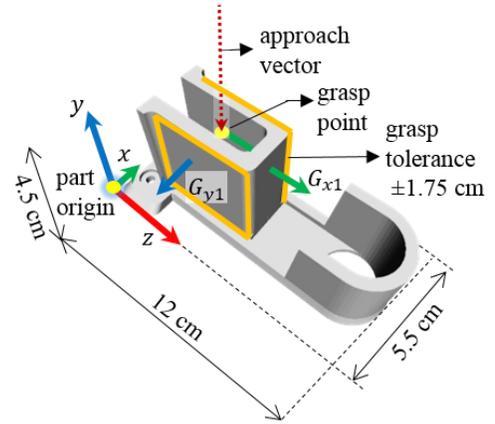


FIGURE 5. Illustration of grasp parameters for strategy 1.

2. **Grasp orientation** ($G_\alpha, G_\beta, G_\gamma$) specifies the orientation of the grippers with respect to the part frame during grasp.
3. **Approach vector** G_γ specifies the direction in which the grippers must approach the part for grasp.
4. **Tolerance** G_t specifies the tolerance in the x_g and z_g direction for a grasp to be successful (Fig. 3(a) and Fig. 3(b)). Every grasp strategy has different tolerance value; that is, they handle different levels of uncertainty based on the grasp offset and orientation. The closing action of the gripper corrects small uncertainties in the y_g direction. If uncertainty in the y_g direction is large, it either leads to collision between the part and the gripper (Fig. 3(c)) or a grasp miss (Fig. 3(d)). These cases are explained in section 6.1.

Table 1 lists parameters for seven grasp strategies considered for the chosen part. Grasp orientation terms follow ZYX Euler angle ordering. Grasp parameters for strategy 1 are shown pictorially in Fig. 5. These parameters correspond to row 1 in Table 1. The

TABLE 1. Parameters for grasp and drop-off for seven strategies

Strategy	G_x, G_y, G_z (cm)	$G_\alpha, G_\beta, G_\gamma$ (degrees)	G_v	G_t (cm)	D_x, D_y, D_z (cm)	$D_\alpha, D_\beta, D_\gamma$ (degrees)
1	2.5, 2.5, 3.5	-90, 90, 0	0, 1, 0	± 1.75	2.5, 2.5, 3.5	-90, 90, 0
2	2.5, 2.5, 9.0	-90, 90, 0	0, 1, 0	± 1.00	2.5, 2.5, 9.0	-90, 90, 0
3	2.5, 1.5, 9.0	90, 0, -90	1, 0, 0	± 0.50	2.5, 1.5, 9.0	-45, 0, 90
4	2.5, 1.5, 9.0	-90, 180, 0	0, 0, 1	± 1.00	2.5, 1.5, 9.0	90, 0, 180
5	2.5, 1.5, 9.0	-90, 0, -90	-1, 0, 0	± 1.00	2.5, 1.5, 9.0	45, 0, 90
6	2.5, 2.5, 3.5	-90, 135, 0	0, 0.707, 0.707	± 1.75	2.5, 2.5, 3.5	-90, 90, 0
7	2.5, 2.5, 3.5	-90, 45, 0	0, 0.707, -0.707	± 1.75	2.5, 2.5, 3.5	-90, 90, 0

planner generates waypoints such that the gripper frame exactly coincides with the grasp point, computed relative to the estimated part posture.

4.2 Drop-off Strategies

After the part has been extracted from the bin and transported to the drop-off location, the planner has to plan the moves to drop the part gently. The strategy to place the part on a flat surface depends on the position and orientation of the grasped part within the grippers. So, each grasp strategy has a unique drop-off strategy as illustrated in Fig. 4 - Row 2. Drop-off strategy can be parameterized by the following two components:

1. **Drop-off offset** (D_x, D_y, D_z) specifies the offset between the gripper frame and the drop-off point (specified by the user). This compensates for the offset that was used during the grasping operation.
2. **Drop-off orientation** ($D_\alpha, D_\beta, D_\gamma$) specifies the gripper orientation just before the drop-off such that the dropping action introduces minimal uncertainties in the final part position. For some grasp strategies (like strategies 3, 4, and 5), the gripper might not be able to place the part gently because of kinematic constraints of the arm. In such cases, the arm will orient as closely as possible to the drop-off frame.

Table 1 lists the parameters for seven drop-off strategies corresponding to the seven grasp strategies. Drop-off orientation terms follow *ZYX* Euler angle ordering.

4.3 Pose Uncertainty Updates During Plan Execution

Assuming that the singulation task is performed using a parallel-jaw gripper and the part is being gripped on a flat surface, for every successful case, it is certain that the closing action of the gripper jaws updates some of the uncertainties in the part pose. For instance, Fig. 6(a) illustrates that the gripping action updates uncertainty in translation along the y_g direction. In

a similar way, the uncertainty in the rotation about x_g is updated as illustrated in Fig. 6(b). In Fig. 6(c), upon gripper closure, the uncertainty in the rotation about z_g is updated. These cases may occur in unison or individually based on the uncertainty in the posture estimate. Gentle drop-off at the endpoint also updates some uncertainties. Fig. 6(d) illustrates this scenario in which the vertical constraints posed by the flat surface at drop-off update the pose uncertainty in translation along z_g and the rotation about y_g . The only uncertainty that is not updated during the execution is translation along x_g . This uncertainty can be corrected with fine positioning moves [27].

Let us consider that the part is being grasped according to strategy 1. For a successful plan, the uncertainty in $x_p, \alpha_p,$ and β_p is updated when the part is grasped within the parallel jaw grippers. For a gentle drop-off, the uncertainty in z_p and γ_p is updated by the constraints introduced by the horizontal drop-off surface. Upon performing fine positioning moves, uncertainty in y_p is updated. Table 2 summarizes the effect of these actions on the uncertainty of the part pose.

5 Generating Singulation Plans

The singulation planner is responsible for generating waypoints for approach, grasp, extract, transport, and drop-off of a chosen part from the bin, checking the feasibility of these waypoints, invoking the evaluation module for the feasible singulation plans, and choosing a singulation plan for execution based on this evaluation. In *Algorithm 1*, a singulation plan \mathbb{S} is generated for each grasp strategy. The point cloud of the scene consisting of the part to be singulated is referred to as *ScenePointCloud*. If the waypoints in \mathbb{S} are infeasible, then the singulation plan is discarded. If all the waypoints are feasible, the singulation planner estimates its expected task completion time (Section 6). The singulation plan with minimum expected task completion time (\mathbb{S}_{best}) is executed.

The following two conditions need to be validated to ensure

TABLE 2. Action and corresponding uncertainty update

Action	Uncertainty update					
	Translation			Rotation		
	x_p	y_p	z_p	α_p	β_p	γ_p
Grasp	✓			✓	✓	
Drop-off			✓			✓
Fine-positioning	✓	✓		✓		

Algorithm 1 SingulationPlanner

```

1: Input: Part pose  $P$ , pose uncertainty  $\mathbb{U}$ , set of grasp
   strategies  $G = [G_1, G_2, \dots, G_n]$ , ScenePointCloud
2:  $\mathbb{S}_{best} \leftarrow \{\emptyset\}$ 
3: for  $\forall G_i \in G$  do
4:   Generate  $\mathbb{S} \hat{=} G_i$ .
5:   if ( $FeasiblePlan(\mathbb{S}) = 1$ ) then
6:      $\mathbb{E}(T_c) \leftarrow EvaluatePlan(\mathbb{S}, P, \mathbb{U}, ScenePointCloud)$ 
7:     if  $\mathbb{E}(T_c)$  is lowest then  $\mathbb{S}_{best} \leftarrow \mathbb{S}$ 
8:   end if
9: end for
10: Execute ( $\mathbb{S}_{best}$ )

```

feasibility of waypoints in the singulation plan:

Condition 1a- Inverse kinematics: Each waypoint in the generated singulation plan is specified as position and orientation of the gripper frame with respect to the robot base frame B . The generated waypoints have to be validated if the robot can actually move to these points. A feasible singulation plan is one which has an inverse kinematic solution at every waypoint. If the inverse kinematic solution does not exist at any waypoint, then the singulation plan is termed infeasible.

Condition 1b- Checking for collision with the bin: The collision detection module checks for any possible collision of the robot arm with the bin during the execution of the singulation plan. For checking the validity of condition 1b, the following representation is used:

Robot arm - The computer-aided design (CAD) model of the robot arm is approximated with a set of rectangular bounding boxes.

Bin - The CAD model of the bin is represented as a point cloud. This point cloud is referred to as *BinPointCloud* in the paper.

The motion of the arm during approach and extraction is simulated by computing the swept volume of the approximated bounding box of the arm along the approach vector. If any point in the *BinPointCloud* lies within this swept volume, then the singulation plan is discarded because of possible collisions. *Algorithm 2* validates condition 1a and condition 1b for a given

singulation plan.

6 Evaluating Singulation Plans

Algorithm 2 FeasiblePlan(\mathbb{S})

```

1: Input: BinPointCloud
2:  $\mathbb{J} \leftarrow$  Inverse kinematic solution for waypoints in  $\mathbb{S}$ .
3:                                      $\triangleright$  Condition 1a
4: Compute swept volume of the robot arm along  $G_v$ .
5:  $\mathbb{C} \leftarrow$  Check for collision between computed swept volume
   and BinPointCloud                                      $\triangleright$  Condition 1b
6: if  $\{\exists \mathbb{J} \forall \text{ waypoints in } \mathbb{S} \ \& \ \mathbb{C} = \{\emptyset\}\}$  then
7:   return 1                                              $\triangleright$  Plan is feasible
8: else
9:   return 0                                              $\triangleright$  Plan is infeasible
10: end if

```

The Monte Carlo simulator estimates success probability of a singulation plan by simulating the gripper motion during the approach, the grasp, and the extract phases and checking for any occurrences of collision. The parallel jaw grippers are approximated with a set of two bounding boxes as shown in Fig. 7. The computation of the swept volume of the approximated bounding box of the gripper during approach, grasp, and the extract phases is shown in Fig. 8. During the extract phase, the CAD model becomes an integral part of the gripper indicating that the part has been gripped. Hence the bounding box of the gripper will include the part as well.

The algorithm takes the singulation plan, the estimate of the part pose, the uncertainty in the pose estimate, and the point cloud of the scene as input. A Gaussian noise is added to the pose of the part, to represent that the reality might differ from the camera pose estimate. The simulator checks for failure conditions (explained in condition 2a and condition 2b) and if both conditions are negative, the singulation plan is considered a pass. The simulator outputs the ratio of the number of trials being marked successful over the total number of trials as success probability of the singulation plan.

6.1 Estimating Probability of Success for a Singulation Plan

A singulation plan is termed successful if the gripper is able to grasp the part without colliding with the part or the neighboring parts. Each simulation trial should introduce uncertainty in the part pose and check if the grippers collide with any part in the bin. If the uncertainty is too high, the evaluator must check the condition for a grasp miss and evaluate success probability for a singulation plan accordingly.

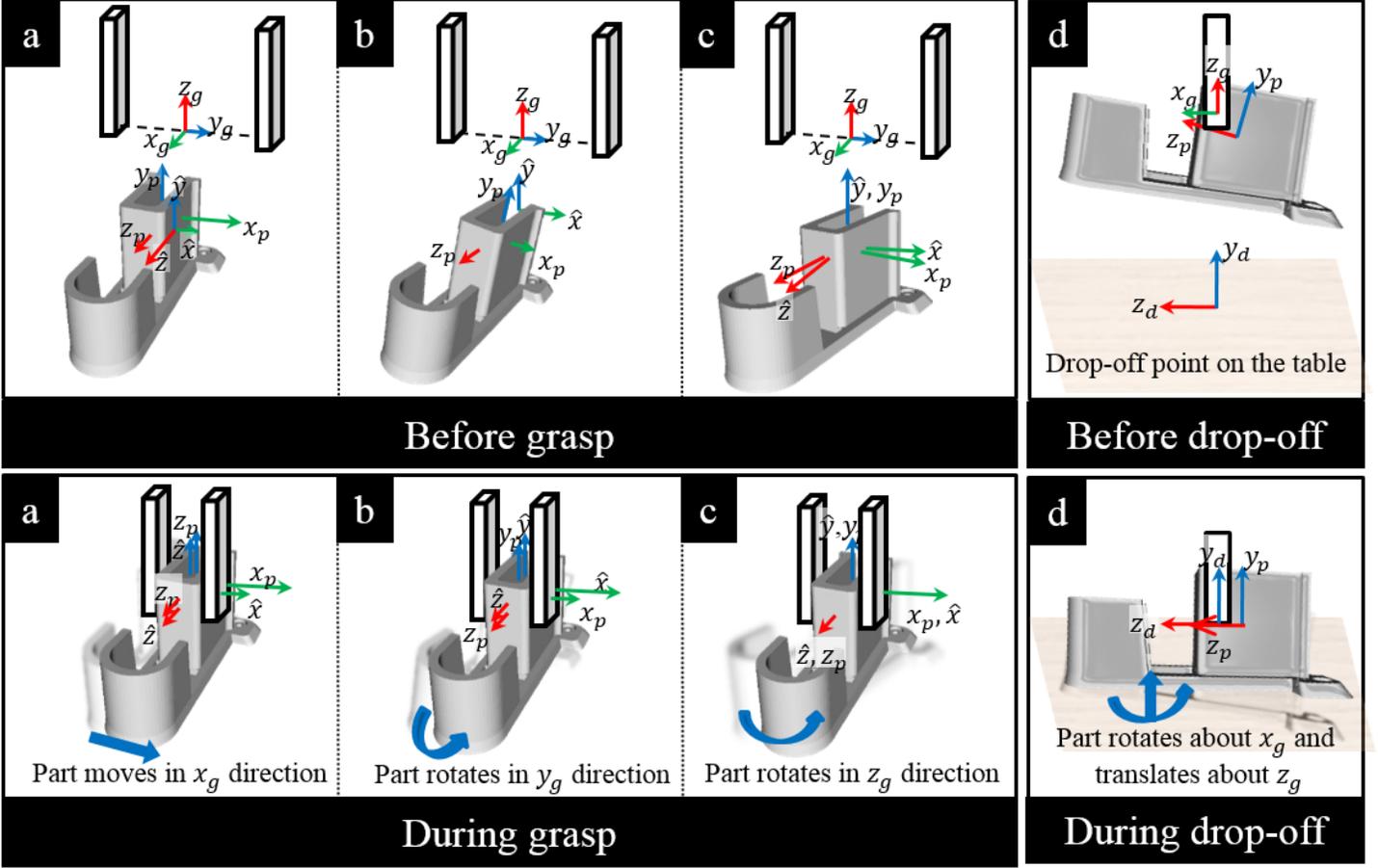


FIGURE 6. Illustration of uncertainty update during plan execution. [a] Uncertainty in translation along y_g being updated during grasping action; [b] Uncertainty in rotation about x_g being updated during grasp. [c] Uncertainty in rotation about z_g being updated during grasp. [d] Uncertainty in rotation about y_g and translation about z_g being updated during drop-off on a flat surface.

Algorithm 3 EvaluatePlan ($\mathcal{S}, P, U, ScenePointCloud$)

- 1: Initialize $s = 0$ ▷ Number of trials being successful
- 2: Initialize N ▷ Total number of trials
- 3: $T_e \leftarrow 0$
- 4: **for** every trial in Monte Carlo simulation **do**
- 5: $P_{new} \leftarrow P \sim \mathcal{N}(0, \mathbb{U}^2)$ ▷ Adding Gaussian noise to P
- 6: Check condition 2a ▷ Condition for collision check
- 7: Check condition 2b ▷ Condition for grasp miss
- 8: **if** condition 2a & 2b is negative **then**
- 9: $s \leftarrow s + 1$
- 10: **end for**
- 11: $p \leftarrow s/N$
- 12: Estimate T_e using Eqn. 4 and Eqn. 5
- 13: $\mathbb{E}(T_c) \leftarrow T_e/p$
- 14: return $\mathbb{E}(T_c)$

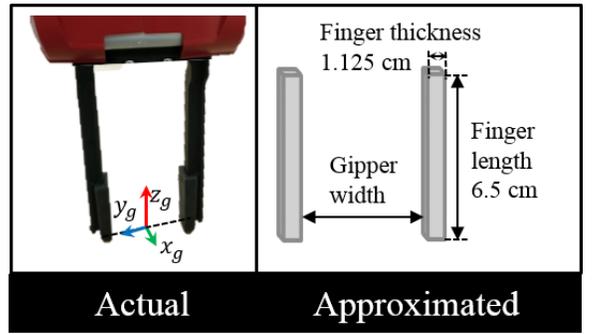


FIGURE 7. Parallel jaw grippers of the robot and the bounding box approximation of the grippers for collision check.

Condition 2a- Failure due to collision with the part or neighboring parts: This condition checks for any occurrence of

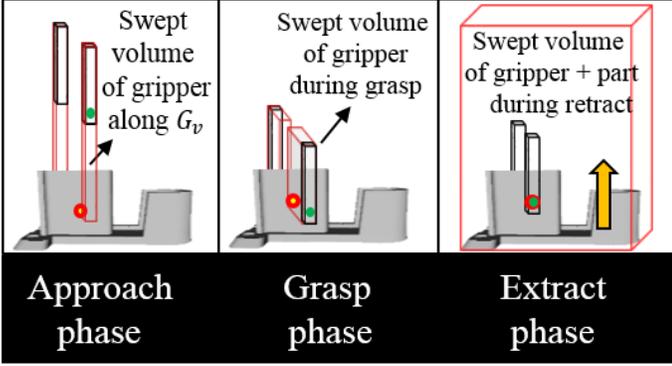


FIGURE 8. Swept volume of the approximated gripper bounding box during [a] Approach, [b] Grasp, and [c] Extract phase. The black cuboids indicate the approximated bounding box of the gripper and the red cuboids indicate the swept volume of the approximated bounding box during the execution. The yellow-red dot indicates the grasp point on the part from the side view and the green dot indicates the grasp point on the gripper from the side view.

collision between the gripper and the parts in the bin. For checking the validity of condition 2a, the following representation is used:

Robot grippers - Parallel jaw grippers are approximated with a set of two rectangular bounding boxes.

Neighbor parts - Represented as a point cloud by subtracting the point cloud belonging to the part chosen for singulation from the point cloud of the scene.

CAD model of the part- The CAD model of the part is represented as a point cloud. In simulations, the point cloud of the CAD model is rendered at the estimated pose of the part. The combined point cloud of the neighboring parts and the CAD model of the part to be singulated constitute the *ScenePointCloud*.

The swept volume of the gripper computed for approach, grasp, and extraction phases is aligned along the axes of the robot base frame and the *ScenePointCloud* is transformed to match the relative orientation. This simplifies the collision detection to just checking if any point in the *ScenePointCloud* lies within the x , y , and z limits of the swept volume. If the number of points in the swept volume is non-zero, then the grasp strategy is discarded because of a possible collision.

Condition 2b- Failure due to grasp miss: The grasp point and the tolerance region are represented using a bounding box as shown in Fig. 9. For a successful plan, the tolerance bounding box and the swept volume of the grippers should not touch or intersect during approach. During grasp, the tolerance bounding box touches the swept volume of the gripper motion. Fig. 9 - Row 1 illustrates examples of successful singulation plans even under uncertainty. In all these cases the bounding boxes touch

with the swept volume during grasp. For failure cases due to collision, these two regions would intersect each other during approach indicating that the grippers would collide with the part if the plan was executed. In the case where the uncertainty is too high and the estimated pose is way too off from the actual part location, the plan fails due to grasp miss. For these cases, the tolerance bounding box and the bounding box of the grippers do not touch each other during approach or grasp. Fig. 9 - Row 2 illustrates examples where the singulation plan fails because of grasp miss.

6.2 Estimating Execution Time for a Singulation Plan

The execution time of a singulation plan is the time required by the robot to move from the first waypoint to the last waypoint in the singulation plan. The time required for executing a singulation plan is dependent on the joint velocity of the robot arm and the joint angle differences between the waypoints. When all of the joint angles are commanded simultaneously at the same joint velocity, the execution time depends only on the joint which has the maximum difference among the others. If the waypoints in the singulation plan and the joint velocity of the robot arm are known, the time required for execution can be estimated before executing the singulation plan. Execution time is the ratio of the sum of the maximum difference in joint angles to the joint velocity. If \mathbb{J} represents the joint angles obtained from the inverse kinematic solver at a particular waypoint and \mathbb{J}_{prev} represents the joint angles at the previous waypoint, then $\Delta\mathbb{J}$ is the maximum of the difference between \mathbb{J} and \mathbb{J}_{prev} . Equation 4 and Eqn. 5 mathematically represent the estimation of time for execution of a singulation plan.

$$\Delta\mathbb{J} = \max[\mathbb{J} \sim \mathbb{J}_{prev}] , \quad (4)$$

$$T_e = \frac{\sum_{w \in S} \Delta\mathbb{J}}{Joint\ Velocity} . \quad (5)$$

Using T_e and p , *Algorithm 3* computes $\mathbb{E}(T_c)$ for a singulation plan. Let us consider a case as shown in Fig. 10. Table 3 shows the output of singulation planner. Each line indicates the evaluation result for a grasp strategy. Since the singulation plan corresponding to strategy 6 has lowest expected time for completion, it is chosen for execution (row marked by ‘*’ in Table 3).

7 Experimental Results

The estimated success probability from the simulator was validated with physical experiments. The experimental setup included a Baxter Research Robot with an Asus Xtion Pro 3D camera mounted on the left arm and a parallel jaw gripper mounted on the right arm as shown in Fig. 11. The joint resolution of

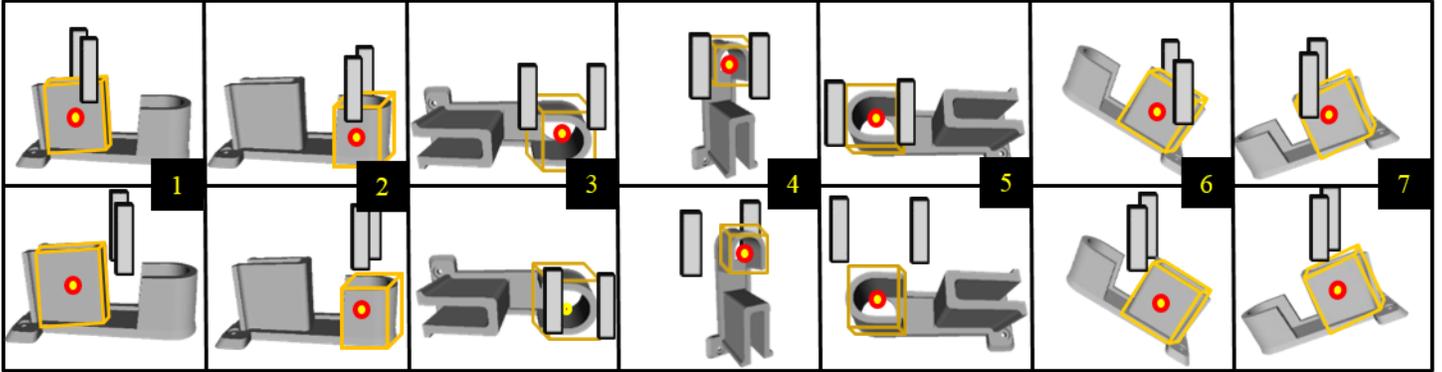


FIGURE 9. Illustration of successful grasp under uncertainty (row 1) and failure due to grasp miss (row 2). The yellow-red dot indicates the ideal grasp point on the part. The yellow bounding box indicates the tolerance region for every grasp strategy. In row 1, the bounding box of the grasp tolerance touches the bounding box of the grippers during grasp. In row 2 the bounding box of the grasp tolerance does not touch the bounding box of the grippers during grasp.

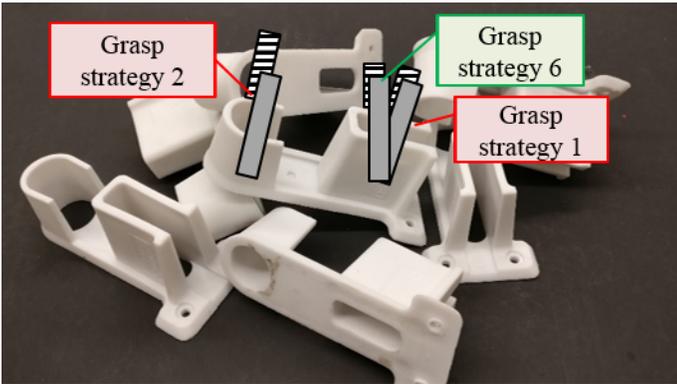


FIGURE 10. Illustration of grasp strategies that are feasible for an estimated part pose. Grasp strategy 6 is chosen for execution as it has minimum expected completion time (Table 3).

the robot arm was set to achieve best possible precision. The joint velocity of all seven joints in the robot arm was set to 0.35 radians/second.

Thirty trials were conducted for each grasp strategy. The part was placed in the bin and the boundary of the part was marked such that the part could be placed in the same position and orientation for later parts of the experiment. The point cloud of the scene was captured and a user interface designed in MATLAB was used to estimate the pose of the part by manually docking the CAD model of the part in the point cloud [23]. The points belonging to the part were deleted from the point cloud and the point cloud of the CAD model was rendered at the pose of the part. Pose uncertainty was simulated by adding Gaussian noise of $\sigma = 5$ mm for \mathbb{U}_l and 5 degrees for \mathbb{U}_o to the estimated pose of the CAD point cloud. The feasibility of every singulation plan

TABLE 3. Output of singulation planner for the case shown in Fig. 10.

Strategy	Evaluation outcome	$\mathbb{E}(T_c)$
1	$p = 0.85, T_e = 9.3$ s	10.9411 s
2	$p = 0.75, T_e = 9.1$ s	12.1333 s
3	Condition 1a invalid	-
4	Condition 1a invalid	-
5	Condition 1b invalid	-
6	$p = 0.95, T_e = 9.25$ s	9.7368 s*
7	Condition 1b invalid	-

was checked for condition 1a and 1b. For every feasible plan, the simulation checked if the gripper was able to extract the part by validating condition 2a and 2b. Fig. 12 illustrates collision check during the approach phase of the singulation plan. At the end, the simulator outputs success probability of the singulation plan. For the physical trials, the part was placed in the marked region and the same noise was added as that of the simulation trials. The robot was commanded to pick the part using the noisy pose and the number of successful grasps were counted. The comparison of the probability of success estimated by the simulator and the physical trials is shown in Fig. 13. The plot shows that the online simulator gives a good estimate of success probability for every grasp strategy.

Next, experiments to validate the estimated execution time of the singulation plan were conducted. Thirty trials were conducted for each grasp strategy and the average error between esti-

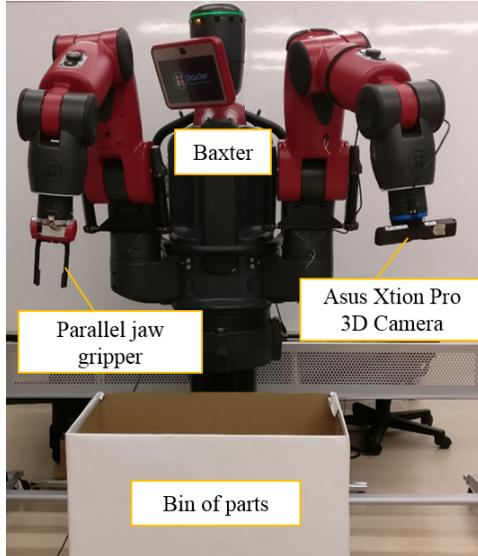


FIGURE 11. Experimental setup with Baxter Research Robot equipped with an Asus Xtion Pro camera mounted on the left arm and a parallel jaw gripper on the right arm.

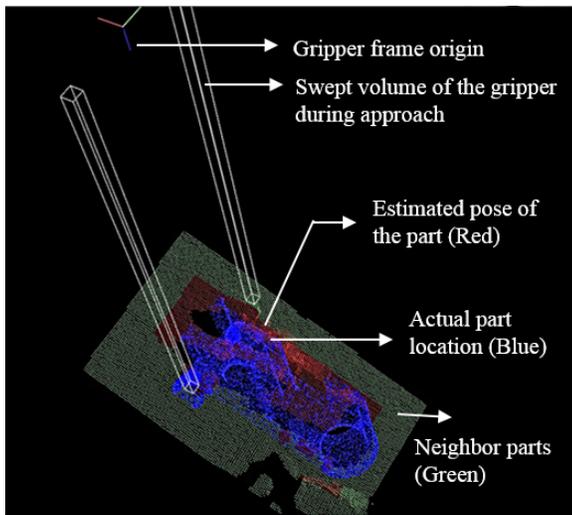


FIGURE 12. Experimental data: Collision check between gripper and part during approach.

mated execution time and actual execution time were computed. For every trial, joint angles were computed at every waypoint using Baxter’s inbuilt Inverse Kinematics solver to estimate the execution time of the singulation plan as explained in section 6.2. The robot was commanded to move through these waypoints (with zero settling time at intermediate waypoints) and the time taken by the robot to completely execute the singulation plan was calculated. Errors obtained by these two cases were

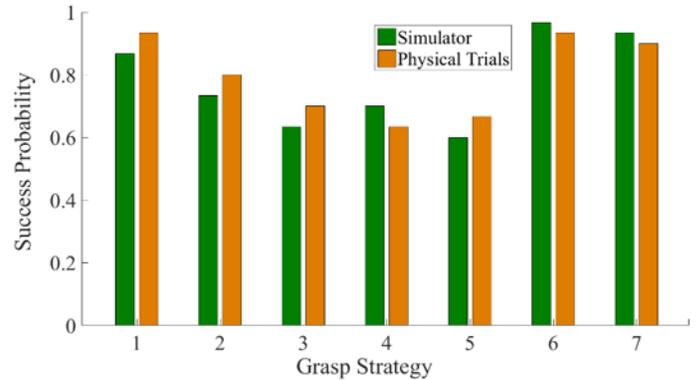


FIGURE 13. Comparison of success probability estimated by the simulator and physical trials.

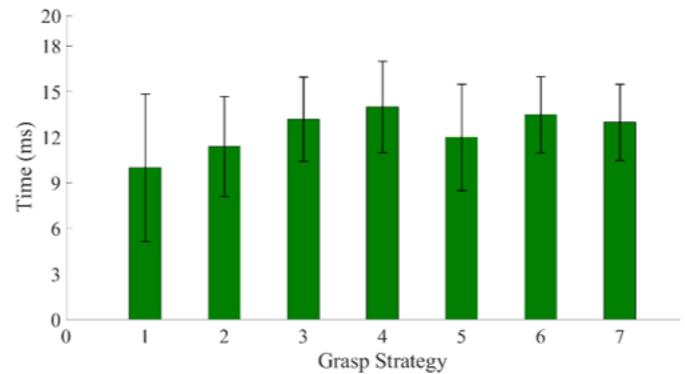


FIGURE 14. Error between estimated execution time and actual time for execution.

averaged over thirty trials for every grasp strategy. The error between estimated execution time and actual time taken is shown in Fig. 14. The vertical bar represents mean and the error bar represents standard deviation. The plot shows that the mean error in the method used for estimation of execution time is within 14 ms.

Experiments were also conducted to use the estimate of success probability to select an appropriate gripper width during the execution of the singulation plan. For each grasp strategy, a cluttered scene was considered (shown in Table 4 Row 1). The success probability was estimated for a singulation plan with various gripper widths. Experiments showed an initial increase in success probability with the increase in gripper width. The values showed a steep dip due to collisions with the neighboring parts as the gripper width was increased beyond a certain value. The outcome of the experiment for seven grasp strategies is shown in Table 4.

Experiments were also conducted to observe the variation of success probability with the variation in the uncertainty of the

TABLE 4. Variation of success probability for each grasp strategy with gripper width. The scene considered for each grasp strategy is shown in row 1. The part of interest is marked with a red dot.

Grasp Strategy Gripper Width (cm)	1	2	3	4	5	6	7
							
3	0.4	0.3	0.2	0.033	0.266	0.4	0.4
4	0.833	0.666	0.5	0.5	0.466	0.833	0.833
5	0.933	0.8	0.266	0.733	0.233	0.833	0.933
6	0.966	0.866	0.066	0.9	0.1	0.933	0.933
7	1	0	0	0.966	0	1	1
8	1	0	0	0	0	0	1
9	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0

TABLE 5. Variation of success probability with uncertainty levels for grasp strategy 1.

U_o (deg) U_t (mm)	5	8	10	12	15
5	0.9667	0.9667	0.9	0.8333	0.8
8	0.9667	0.9	0.8333	0.8	0.7333
10	0.8667	0.8	0.7333	0.6667	0.6333
12	0.8333	0.7333	0.7	0.6333	0.5667
15	0.7333	0.6333	0.6333	0.6	0.5333

pose estimate. U_t and U_o were varied separately and success probability was estimated in each case. A point cloud was captured with a single part in the scene and grasp strategy 1 with gripper width as 6 cm was chosen during the experiment. The experimental results are shown in Table. 5. It can be seen that the success probability decreases as the uncertainty level increases.

8 Conclusions

This paper presented a simulation-based approach to evaluate singulation plans. This approach can be used to select a grasp strategy and associated motion plan based on the estimated expected time of completion. Each plan is evaluated with a Monte Carlo simulation that runs in real time to estimate the expected time for completion. The proposed method uses simplified collision detection to speed up computation. Comparison of estimated success probability with physical trials showed that the

two numbers were very close. The maximum mean error between the estimated execution time and actual execution time was very small. The method can also be used to select optimum gripper width such that success probability is maximum for the chosen grasp strategy. The method can also be used to check the performance of any system for a given perception uncertainty.

We assumed the pose uncertainty to follow a Gaussian distribution with a sigma level of 5 mm and 5 degrees in translation and orientation, respectively. Empirical experiments demonstrate that this choice of the uncertainty distribution resulted in a close agreement between the success probability estimated by the simulator and the physical trials. More experiments need to be conducted in the future to obtain a better characterization of pose uncertainty. The future work in this area also includes extending the approach to handle other types of grippers and using active perception to reduce uncertainty. We can use the work presented in [28–30] to replicate the collision detection model.

ACKNOWLEDGMENT

This work is supported in part by National Science Foundation Grants #1634431 and #1713921, and National Institute of Standards and Technology Award #70NANB15H250. Opinions expressed are those of the authors and do not necessarily reflect opinions of the sponsors.

References

- [1] Kumbla, N.B., Thakar, S., Kaipa, K.N., Marvel, J.A., and Gupta, S.K., 2017. “Simulation based on-line evaluation of singulation plans to handle perception uncer-

- tainty in robotic bin picking”. In ASME 2017 Manufacturing Science and Engineering Conference, American Society of Mechanical Engineers, pp. V003T04A002–V003T04A002.
- [2] Buchholz, D., 2016. “Depth map based pose estimation”. In *Bin-Picking*. Springer, pp. 39–56.
- [3] Kuo, H.Y., Su, H.R., Lai, S.H., and Wu, C.C., 2014. “3D object detection and pose estimation from depth image for robotic bin picking”. In Automation Science and Engineering (CASE), 2014 IEEE International Conference on, IEEE, pp. 1264–1269.
- [4] Sansoni, G., Bellandi, P., Leoni, F., and Docchio, F., 2014. “Optoranger: A 3D pattern matching method for bin picking applications”. *Optics and Lasers in Engineering*, **54**. Elsevier, pp. 222–231.
- [5] Pretto, A. and Tonello, S. and Menegatti, E., 2013. “Flexible 3d localization of planar objects for industrial bin-picking with monocular vision system”. In Automation Science and Engineering (CASE), 2013 IEEE International Conference on, IEEE, pp. 168–175.
- [6] Rodrigues, J.J., Kim, J.S., Furukawa, M., Xavier, J., Aguiar, P., and Kanade, T., 2012. “6D pose estimation of textureless shiny objects using random ferns for bin-picking”. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE, pp. 3334–3341.
- [7] Boughorbel, F., Zhang, Y., Kang, S., Chidambaram, U., Abidi, B., Koschan, A., and Abidi, M., 2003. “Laser ranging and video imaging for bin picking”. *Assembly Automation*, **23**(1), pp. 53–59.
- [8] Horn, B. and Ikeuchi, K., 1983. “Picking parts out of a bin”. DTIC Document.
- [9] Akizuki, S. and Hashimoto, M., 2015. “Stable position and pose estimation of industrial parts using evaluation of observability of 3D vector pairs”. *The Journal of Robotics and Mechatronics*, **27**(2), pp. 174–181.
- [10] Fuchs, S., Haddadin, S., Keller, M., Parusel, S., Kolb, A., and Suppa, M., 2010. “Cooperative bin-picking with time-of-flight camera and impedance controlled dlr lightweight robot iii”. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, pp. 4862–4867.
- [11] Harada, K., Nagata, K., Tsuji, T., Yamanobe, N., Nakamura, A., and Kawai, Y., 2013. “Probabilistic approach for object bin picking approximated by cylinders”. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, IEEE, pp. 3742–3747.
- [12] Liu, M.Y., Tuzel, O., Veeraraghavan, A., Taguchi, Y., Marks, T.K., and Chellappa, R., 2012. “Fast object localization and pose estimation in heavy clutter for robotic bin picking”. *International Journal of Robotics Research*, **31**(8), SAGE Publications, pp. 951–973.
- [13] Papazov, C., Haddadin, S., Parusel, S., Krieger, K., and Burschka, D., 2012. “Rigid 3D geometry matching for grasping of known objects in cluttered scenes”. *International Journal of Robotics Research*. SAGE Publications, pp. 538 - 553.
- [14] Pronobis, A. and Caputo, B., 2007. “Confidence-based cue integration for visual place recognition”. In Intelligent Robots and Systems (IROS), 2007 IEEE/RSJ International Conference on, IEEE, pp. 2394–2401.
- [15] Bicchi, A., and Kumar, V., 2000. “Robotic grasping and contact: a review”. In Robotics and Automation (ICRA), 2000 IEEE International Conference on, IEEE, pp. 348–353.
- [16] Bohg, J., Morales, A. and Asfour, T. and Kragic, D., 2014. “Data-driven grasp synthesis—a survey”. *IEEE Transactions on Robotics*, **30**(2), pp. 289–309.
- [17] Dupuis, D.C., Léonard, S., Baumann, M.A., Croft, E.A., and Little, J.J., 2008. “Two-fingered grasp planning for randomized bin-picking”. In Robotics: Science and Systems 2008 Manipulation Workshop-Intelligence in Human Environments.
- [18] Ellekilde, L.P., Jorgensen, J.A., Kraft, D., Kruger, N., Piater, J., and Petersen, H. G., 2012. “Applying a learning framework for improving success rates in industrial bin picking”. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, IEEE, pp. 1637–1643.
- [19] Kendall, A., and Cipolla, R., 2016. “Modelling uncertainty in deep learning for camera relocalization”. In Robotics and Automation (ICRA), 2016 IEEE International Conference on, IEEE, pp. 4762–4769.
- [20] Zheng, Y., and Qian, W.H., 2005. “Coping with the grasping uncertainties in force-closure analysis”. *The International Journal of Robotics Research*, **24**(4). SAGE Publications, pp. 311–327.
- [21] Berenson, D., Srinivasa, S.S., and Kuffner, J.J., 2009. “Addressing pose uncertainty in manipulation planning using task space regions”. In Intelligent Robots and Systems (IROS), 2009 IEEE/RSJ International Conference on, IEEE, pp. 1419–1425.
- [22] Chang, L., Smith, J.R., and Fox, D., 2012. “Interactive singulation of objects from a pile”. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, pp. 3875–3882.
- [23] Kaipa, K.N., Kankanhalli-Nagendra, A.S., Kumbla, N.B., Shriyam, S., Thevendria-Karthic, S.S., Marvel, J.A. and Gupta, S.K., 2016. “Addressing perception uncertainty induced failure modes in robotic bin-picking”. *Robotics and Computer-Integrated Manufacturing*, **42**. Elsevier, pp. 17–38.
- [24] Kaipa, K.N., Kankanhalli-Nagendra, A.S., Kumbla, N.B., Shriyam, S., Thevendria-Karthic, S.S., Marvel, J.A. and

- Gupta, S.K., 2016. "Enhancing robotic unstructured bin-picking performance by enabling remote human interventions in challenging perception scenarios". In Automation Science and Engineering (CASE), 2016 IEEE International Conference on, IEEE, pp. 639–645.
- [25] Kaipa, K.N., Shriyam, S., N.B, and Gupta, S.K., 2016. "Resolving occlusions through simple extraction motions in robotic bin picking". In ASME 2016 Manufacturing Science and Engineering Conference, American Society of Mechanical Engineers, pp. V002T04A002-V002T04A002.
- [26] Kaipa, K.N., Shriyam, S., Kumbla, N.B, and Gupta, S.K., 2015. "Automated plan generation for robotic singulation from mixed bins". In IROS Workshop on Task Planning for Intelligent Robots in Service and Manufacturing.
- [27] Kaipa, K.N., Kumbla, N.B, and Gupta, S.K., 2015. "Characterizing Performance of Sensorless Fine Positioning Moves in the Presence of Grasping Position Uncertainty". In IROS Workshop on Task Planning for Intelligent Robots in Service and Manufacturing.
- [28] Ilies, H.T., 2009. "Continuous collision and interference detection for 3D geometric models". *ASME Journal of Computing and Information Science in Engineering*, **9**(2). American Society of Mechanical Engineers, pp. 021007.
- [29] Akgunduz, A., Banerjee, P., and Mehrotra, S., 2005. "A linear programming solution for exact collision detection". *ASME Journal of Computing and Information Science in Engineering*, **5**(1). American Society of Mechanical Engineers, pp. 48–55.
- [30] Kim, B., and Rossignac, J., 2003. "Collision prediction". *ASME Journal of Computing and Information Science in Engineering*, **3**(4). American Society of Mechanical Engineers, pp. 295–301.