Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Addressing perception uncertainty induced failure modes in robotic bin-picking

Krishnanand N. Kaipa^a, Akshaya S. Kankanhalli-Nagendra^a, Nithyananda B. Kumbla^a, Shaurya Shriyam^a, Srudeep Somnaath Thevendria-Karthic^a, Jeremy A. Marvel^b, Satyandra K. Gupta^{C,*}

^a Maryland Robotics Center, University of Maryland, MD, USA

^b National Institute of Standards and Technology, Gaithersburg, MD, USA

^c Center for Advanced Manufacturing, University of Southern California, CA, USA

ARTICLE INFO

Article history: Received 14 November 2015 Received in revised form 6 April 2016 Accepted 3 May 2016 Available online 14 May 2016

ABSTRACT

We present a comprehensive approach to handle perception uncertainty to reduce failure rates in robotic bin-picking. Our focus is on mixed-bins. We identify the main failure modes at various stages of the bin-picking task and present methods to recover from them. If uncertainty in part detection leads to perception failure, then human intervention is invoked. Our approach estimates the confidence in the part match provided by an automated perception system, which is used to detect perception failures. Human intervention is also invoked if uncertainty in estimated part location and orientation leads to a singulation planning failure. We have developed a user interface that enables remote human interventions when necessary. Finally, if uncertainty in part posture in the gripper leads to failure in placing the part with the desired accuracy, sensor-less fine-positioning moves are used to correct the final placement errors. We have developed a fine-positioning planner with a suite of fine-motion strategies that offer different tradeoffs between completion time and postural accuracy at the destination. We report our observations from system characterization experiments with a dual-armed Baxter robot, equipped with a Ensenso three-dimensional camera, to perform bin-picking on mixed-bins.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Bin-picking is a precursor to kitting [1,2] and assembly operations in many discrete-part manufacturing applications [3–5]. The use of robots for bin picking can enable handling a wide variety of parts without any change in the hardware; hence it offers a flexible automation solution. Machine vision is a key enabling technology in this context [6,7]. Robotic bin-picking, guided by vision and other sensor modalities, has been successfully demonstrated with a high degree of reliability for bins containing a single type of part with a relatively simple shape [8].

When bins are complex, the reliability of robotic bin picking operations is reduced. The complexity in bins might arise due to the presence of multiple different types of parts. Such bins are called mixed bins. Recognizing the desired part in a mixed bin and estimating its location is a much more challenging problem from the perception point of view. Unstructured, randomly distributed, mixed-bins make the perception problem challenging

* Corresponding author. *E-mail addresses: jeremy.marvel@nist.gov* (J.A. Marvel), skgupta@usc.edu (S.K. Gupta).

http://dx.doi.org/10.1016/j.rcim.2016.05.002 0736-5845/© 2016 Elsevier Ltd. All rights reserved. due to the following reasons: (1) parts may lie in widely different three dimensional (3D) postures and (2) parts may be either partially or completely occluded by other parts. The problem is compounded due to factors such as sensor noise, background clutter, shadows, complex reflectance properties, and poor lighting conditions.

The complexity of the bin might also increase because the parts present in the bin have complex shapes, and can only be removed by holding them at certain locations and moving them in certain directions. Uncertainty in part location and orientation estimates may lead to a failure when the robot tries to extract the part from the bin. The potential for part tangling, and occlusion of grasping surfaces, makes the planning problem challenging because of perception uncertainties.

The effect of perception uncertainty propagates through every stage of task execution including, part recognition and pose estimation, singulation, and positioning. This thereby impacts the overall system performance. For example, the detected part match may not correspond to the specified part. Uncertainty in pose estimation may lead to poor singulation plans, and thereby singulation failures. Finally, uncertainty in the initial grasped posture of the part may lead to errors in part posture at the destination after





CrossMark

K.N. Kaipa et al. / Robotics and Computer-Integrated Manufacturing 42 (2016) 17-38



Fig. 1. Flowchart showing how uncertainty is handled at various stages of the bin-picking task.

final drop off. However, many manufacturing applications require parts to be placed in a specified posture, within tight tolerances, before tasks like assembly or packaging can take place [9].

To improve the reliability of the bin-picking operations, we need to characterize the effect of perception uncertainty on binpicking task execution performance. We also need to develop methods to deal with situations when high perception uncertainty requires specialized methods to prevent the failure. In this paper, we present a comprehensive approach to handle perception uncertainty to reduce failure rates in unstructured robotic bin-picking. The main failure modes at various stages of the bin-picking task and methods to recover from them are shown in Fig. 1. We first characterize the uncertainty in estimating the six dimensional (6D) posture of a part match found by using an automated perception system. The input to the system is a CAD model of the part to be singulated and a 3D point cloud of the mixed-bin. The resulting uncertainty information is used to estimate confidence in part recognition and pose estimation. If perception uncertainty results in a part detection failure or singulation planning failure, then human intervention is invoked. We have developed a user interface that enables remote human interventions when necessary. Intervention in this context may correspond to the human finding a good part match and obtaining an improved estimate of the part pose by using appropriate controls present in the user interface. If perception uncertainty results in an unacceptable error in the final posture of the part at the destination, then fine positioning is invoked to achieve the desired postural accuracy. We have developed a fine-positioning planner to correct errors in the destination posture of the part arising due to uncertainty in the initial grasped state. We have developed a suite of fine-motion strategies that offer different tradeoffs between completion time and postural accuracy at the destination.

In our earlier works, we presented preliminary versions of automated perception algorithm [10], perception failure resolution using human intervention [11], singulation planning [12], and fine-positioning [13]. We treated each problem in an isolated manner. This paper significantly improves upon methods reported in our previous works and presents a comprehensive approach to identify and address perception-uncertainty-induced failure modes in robotic unstructured bin-picking.

2. Related work

Many research groups have addressed the problem of robotic bin-picking. Different aspects of robotic bin-picking include perception, grasp-planning, and motion planning. Each of these represents a vast area of research in itself. Therefore, we survey only prior research that integrated these aspects to achieve bin-picking or grasping. A summary of the focus of various works on binpicking is shown in Table 1. In our survey, we also pay attention to whether uncertainty was taken into account, and if so, how it was handled at different stages of task execution. Most of the research in bin-picking considered the problem up to stage where the part is successfully picked from the bin, while ignoring the next stage of delivering the part in a known posture accurately at the destination. We survey the field of sensorless manipulation where this problem was treated separately.

2.1. Perception for robotic bin-picking

Most previous attempts on a systems approach to bin-picking mainly focussed on the perception problem [25,24,23,21,4,20,19,18,16,15,14], while assuming accurate robot grasping. However, model inaccuracies and sensor uncertainties make it difficult for a majority of the perception algorithms to provide reliable object recognition and localization estimates, thereby affecting overall bin-picking performance.

Except for a few, many of these methods ignored the evaluation of perception quality before proceeding to picking the part. Liu et al. [4] presented a directional, chamfer-matching-based, object localization and pose estimation in heavy clutter for robotic bin picking. The accuracy of their method was tested empirically by evaluating the consistency of a pose estimate across multiple viewpoints of the camera. This was achieved by placing an object in the scene, estimating its pose in local frames of different camera viewpoints, transforming them into the world frame, and plotting the histogram of deviations from the median pose estimate in 6D. But there was no mechanism in place to rate the perception result during task execution.

Papazov et al. [33] presented a 3D object-recognition and poseestimation approach for grasping, based on geometric descriptors, hashing techniques, and random-sampling consensus (RANSAC)like sampling strategies. The authors evaluated the quality of a recognition hypothesis by defining an acceptance function,

Table 1

Summary of aspects addressed in various works on bin-picking.

| | Main focus | Sensor | Bins | Parts | Robot implementation |
|---------|----------------------------------|---------------------------------|-------------|---|----------------------|
| [14] | Perception | Sick Ranger and laser diode | Mixed | Plastic gears, rings, etc. | No |
| [15] | Perception | 3D depth sensor | Homogeneous | | No |
| [16,17] | Perception | Kinect | Homogeneous | Connector pipes | Yes |
| [18] | Perception | 2D monochrome camera | Homogeneous | 2D planar objects | Yes |
| [19] | Perception | 2D camera with multi-lighting | Homogeneous | Shiny brackets | Yes |
| [20] | Perception | Stereo camera | Homogeneous | Wheel hubs, brake disks, piston rods | Yes |
| [4] | Perception | Multi-flash camera | Mixed | Shiny metal parts, texture-less plastic objects | Yes |
| [21] | Perception | 3D camera | Homogeneous | Alternators | No |
| [22] | Perception | Laser scanner | Homogeneous | Joist hangers, plug gauges, piston rods | Yes |
| [23] | Perception | Laser ranging | | | No |
| [24] | Perception | Laser ranging and video imaging | Mixed | Gears, balls, wooden objects | No |
| [25] | Perception | Photometric stereo | Homogeneous | toroid shaped parts | Yes |
| [26] | Grasp planning | 3D camera | Homogeneous | Coil springs, metal linkages | Yes |
| [27] | Grasp planning | 3D camera | Mixed | piston rods, plastic toys | Yes |
| [28] | Grasp planning | 3D camera | Homogeneous | piston rods | Yes |
| [29] | Motion planning | 3D grid scanner | Homogeneous | Metal pipes | Yes |
| [30] | Perception and grasp planning | SICK IVP Ruler E1200 laser line | Homogeneous | piston rods | Yes |
| [21] | Derception and grass planning | solid state range camera | Homogonoous | Power | Vac |
| [21] | Crasp and motion planning | 2D camora | Homogeneous | DUXES | Yes |
| [22] | Barcontion grace motion planning | 2D lasor scappor | Mixed | cans, polyhodral shaped objects | Vos |
| [24] | Sustems approach, human robot | Time of flight comore | Homogonoous | Calls, polyheural shapeu objects | Yes |
| [34] | collaboration | Time-or-mgnit calllera | nomogeneous | riastic pipes | 162 |
| [35] | Learning to improve grasping | 3D camera | Homogeneous | | Yes |

comprising a visibility term and a penalty term. The visibility term was computed as the ratio of transformed model points that fell within a certain threshold band of the scene to the total number of the model points. The penalty term penalized a hypothesis if it violated the condition that a scene point lying behind the localized model cannot be seen through an opaque surface when viewed from the perspective of the camera. The penalty was computed as a ratio of transformed model points, which are between the projection center of the range image and a range-image pixel, to the total number of model points. The hypothesis was accepted if the visibility term was greater than, and the penalty term was lesser than, certain manually set thresholds. The authors addressed uncertainty by empirically testing how the recognition rate of their algorithm varied as a function of zero-mean Gaussian noise added into the noise-free, scene data.

Perception failures were not addressed explicitly in most of the above approaches. The robotic bin-picking system developed by Fuchs et al. [34] has built-in mechanisms to detect object-localization failures. In particular, they assume significant uncertainty in object pose estimation and initiate grasping only when the reliability of the pose hypothesis falls below a given threshold. Otherwise, the localization is restarted from a different view point of the camera. Another relevant work is an algorithm, presented by Pronobis and Caputo [36], which is able to measure its own level of confidence in performing a visual, place-recognition task. Taking a *support vector machine* approach, the authors propose a method for measuring the confidence level of the classification output based on the distance of a test image and the average distance of training vectors.

2.2. Grasping under uncertainty

Grasp planning literature is very vast. Approaches can be broadly divided into analytical and data-driven methods [37]. A review of analytical approaches to grasp synthesis can be found in [38]. A comprehensive survey on data-driven grasp synthesis can be found in [39]. Data-driven approaches have become more popular over the past decade with the advent of simulationbased tools like Graspit! [40]. Grasp evaluation is usually based on a widely used, force-closure, quality metric for precision grips [41].

Given that pose-estimation error impacts grasping performance in practice, many researchers have addressed the problem of grasp planning under perception uncertainty [42–46] and uncertainty in object shape due to manufacturing tolerances, and mechanics, due to limits on sensing during grasping [47]. Nguyen [42] incorporated contact-location uncertainty into force-closure analysis. Roa and Suarez [43] proposed a grasp-synthesis approach that accounted for the fact that the real fingers can never contact the object at the computed points. Zheng and Qian [44] considered both friction uncertainty and position uncertainty in their force-closure analysis. Each of these approaches considered the effects of contact location uncertainty independently, but did not address the effects of calibration error and object pose uncertainty on grasp quality. Grasping under pose uncertainty was addressed in the works of Berenson et al. [45] and Weisz and Allen [46].

Kehoe et al. [47] considered uncertainty in object shape due to manufacturing tolerances, and mechanics, due to limits on sensing during grasping. Their algorithm takes three inputs: an approximate object outline, Gaussian uncertainty around each vertex, and center of mass. The grasp-quality metric was defined as a lower bound on the probability of achieving force closure, and was computed using Monte Carlo sampling. Glover et al. [48] proposed probabilistic models of object geometry for grasp planning. Bayesian models have been used by Goldberg and Mason [49] and Hsiao et al. [50] to address pose uncertainty in grasping.

Most approaches described above focus on grasping problems that involve interactions between the gripper and the single object to be grasped. However, we are interested in the class of binpicking problems where performance is measured as a composite of (1) quality of approach toward the object, (2) grasp quality, and (3) quality of extraction of the grasped object. Dupius et al. [28] proposed a two-fingered, grasp-planning method in the context of vision-guided robotic bin-picking. Off-line grasp candidates are generated for each instance of the object based on the quality metrics provided by the Graspit! simulator. Each candidate instance is evaluated online based on how many pre-generated grasps are collision-free with respect to neighboring objects. Their approach dealt with bins with the same type of parts.

2.3. Sensorless manipulation to reduce uncertainty in positioning

Sensorless manipulation approaches can be broadly divided into two categories based on whether part manipulation is induced by the robot through external surfaces, or achieved by the robot directly using non-prehensile manipulation moves. Erdmann and Mason [51] presented a robot motion planner that generated tray-tilting plans to orient planar objects. In particular, a robot tilted the tray containing the randomly oriented object causing the object to slide into walls, along walls, and into corners, until the object settled into a desired orientation. Erdmann et al. [52] extended the tilting strategy to three-dimensional polyhedral parts. Akella et al. [53] used a combination of a controlled one degree-offreedom joint and a constant-velocity conveyor belt to orient planar parts.

In all the above approaches, the robot induced part motions through an external surface rather than applying the forces directly on the part. Later, non-prehensile manipulation methods were proposed where the gripper fingers interacted with the part without encompassing it [54,55]. Goldberg [54] presented a sensorless manipulation method, in which a simple parallel-jaw gripper applied forces on the part directly, resulting in its specified orientation in a finite sequence of steps. Similarly, Erdmann [55] presented a non-prehensile, two-palm, manipulation method for orientation of polyhedral objects. The entire palms were used rather than the fingertips alone. Dogar and Srinivasa [56] used the notion of task mechanics to introduce push-grasp plans for dexterous hands in the presence of object-pose uncertainty and high clutter. However, this was presented as a grasping technique as opposed to post-grasp manipulation. Kristek and Shell [57] extended the sensorless, non-prehensile manipulation to deformable polygonal parts. Other recent examples of sensorless manipulation include [58,59], and [60].

3. Problem formulation

3.1. Hybrid cell

Hybrid cells support different human–robot collaboration (HRC) modes [61–64]. State-of-the-art HRC approaches have mainly considered humans and robots physically sharing the workspace inside the hybrid cell. Human and robot may be working concurrently on a task, sequentially on the task, or one of them may perform most of the operations, while the other plays an assistive role. We take a different approach to achieving HRC in hybrid cells. We are mainly interested in a *human-on-the-call* mode that enables a remotely located human to take over when the robot needs help.

However, implementing this mode requires the robot to be capable of detecting an impending failure and invoking human intervention. Currently, robots have difficulty in assessing their own capability to complete a task. Consider the following case. A robot is capable of picking a part if the part is presented to it at a certain location. However, if the part has shifted from its nominal location, the robot might not be able to pick it. The robot does not simply know where the transition boundary between task-execution success and failure lies. As it attempts to pick the part, it might bump into it, push it further away, and jam the material handling system. This can, in turn, trigger a system fault and cause a shut-down of the system.

To use robots in small-production batch operations, they must be able to estimate the probability of task completion before beginning the task. This will enable robots to assess their confidence in doing a task. If the robot does not have a high confidence in completing a task, then it can call for help. This will enable human operators to provide the robot the needed assistance and prevent major system faults that result from task-execution failures. Providing such task assistance to robots is cheaper than recovering from a system shutdown. We illustrate these concepts in the context of robotic bin-picking in this paper.

The experimental setup used (Fig. 2) consists of RoboSAM¹, a ROBotic Smart Assistant for Manufacturing and a user interface that allows remote human interventions. The RoboSAM system is built using a Baxter Research Robot² and an Ensenso 3D camera. We are mainly interested in part-order problems that specify multiple quantities of different parts to be singulated from a bin of randomly scattered parts and to be delivered in a known posture at a destination location as rapidly as possible. An illustration of the singulation task is shown in Fig. 3. Successful singulation, given a noisy estimate of part posture, primarily depends on (1) planning the approach of the gripper toward the part such that it does not collide with other nearby parts, (2) determining grasp postures that result in force-closure of the grasped part, and (3) performing tangle-free extraction. Different singulation-failure scenarios are shown in Fig. 4. We consider representative industrial parts that afford different recognition and grasping complexities to illustrate various challenges encountered during the bin-picking task. Fig. 5 shows examples of part placements in a bin with varying degrees of perception and singulation complexities. In this context, the quality of the point cloud is a function of the fraction of points on the surface of the part that are exposed to the camera, and thereby, get registered in the point cloud. For example, in Fig. 5(a), the part to be singulated gives a good point cloud, since a large fraction of its surface area is visible from the camera. However, in Fig. 5(b), the part is in an orientation such that relatively few points get registered, resulting in a poor quality of the point cloud. Finally, fine positioning is invoked if needed to achieve the desired postural accuracy. Fig. 6 illustrates the problem of fine-positioning.

3.2. Definitions

Definition 1. A general 6D posture is represented by $\ell \in \mathbb{R}^6 = \{x, y, z, \alpha, \beta, \gamma\}$ where (x, y, z) and (α, β, γ) represent the position and orientation (in Euler angle representation), respectively in 3D.

Definition 2. A mixed-bin $\mathcal{B}(\kappa, n, \{n_i\})$ is a bin of randomly scattered pile of *n* parts, comprising multiple instances n_i of κ different part types:

$$\mathcal{B}(\kappa, n, \{n_i\}) = \{p_i^{(j)} : j = 1, 2, ..., n_i, i = 1, 2, ..., \kappa\} | \mathcal{B}$$
$$| = \sum_{i=1}^{\kappa} n_i = n$$
(1)

where part $p_i^{(j)}$ represents the *j*th instance of part type *i*.

Definition 3. Given a mixed-bin \mathcal{B} , we define a *part-order* $\mathcal{P}(I, \{\bar{n}_i\})$ as an order placed by a customer requesting a set of parts $\mathcal{P} \subseteq \mathcal{B}$, while requiring each part $p_i^{(j)} \in \mathcal{P}$ to be transferred and positioned at a destination posture $\mathcal{E}_i^{(j)}$ within an expected postural accuracy $\Delta \mathcal{E}$, where $j = 1, 2, ..., \bar{n}_i \leq n_i$, $i \in I$, $I(|I| \leq \kappa)$ represents the set of indices of the part types to be selected.

¹ Video link: https://www.youtube.com/watch?v=ZfcCmijILsw

² DISCLAIMER: 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 the University of Maryland.



RoboSAM

Fig. 2. Hybrid Cell showing the RoboSAM system built using the Baxter robot and the Ensenso 3D camera and the remote human operator.



Fig. 3. Illustration of the singulation task: (a) Robot gripper in the initial approach posture. (b) Part grasped. (c) Part successfully singulated.

Definition 4. A *sequenced-order* is defined as a part-order that requires the parts to be delivered in a specified sequence.

Definition 5. *Singulation* is defined as the concatenation of four stages including positioning the gripper at an appropriate posture above the bin, approaching the gripper toward the part, grasping the part, and extracting the part out of the bin.

Definition 6. We define *tangle-free-singulation* as a singulation of a part from a bin such that it is not tangled with other neighboring parts in the bin during extraction, thereby ensuring singulation of only one part at a time (Fig. 3).

Definition 7. *Fine-positioning* refers to the act of applying appropriate sliding forces on the part until its posture is within the desired accuracy limits.

3.3. High-level decision making in the hybrid cell

Given a mixed bin $\mathcal{B}(\kappa, n, \{n_i\})$ and a sequenced part-order $\mathcal{P}(\mathcal{I}, \{\bar{n}_i\})$, our goals in this paper are to achieve tangle-free singulation of the first part $p \in \mathcal{P}$ in the sequence and to position that part at a destination posture within a specified postural accuracy $\Delta \mathcal{E}$. The inputs to the system are a CAD model of the part p_d to be singulated and a 3D point cloud of the mixed-bin \mathcal{D} .

The steps in the high-level planner are given below:

1. Characterize the uncertainty in estimating the 6D posture of a part instance $p^{(j)}$ that is detected by using an automated

perception system (Section 4), while reporting the following:

- Estimate of part posture $\hat{\ell}^{(j)}$ with postural uncertainty $\sigma^{(j)}$.
- Confidence in the part match by using a signature based method (Section 4.2).
- 2. If confidence is acceptable, then
 - Perform singulation planning (Section 5) to generate and evaluate singulation plans, while accounting for
 - Uncertainty in the estimated part pose $\sigma^{(j)}$.
 - \circ Grasp-approach quality q_a .
 - \circ Grasp quality based on force-closure q_{g} .
 - Part being tangle-free during singulation.
 - If a singulation plan exists (for given $\sigma^{(j)}$), then
 - Execute the singulation plan.
 - Proceed to fine-positioning if needed (Section 6): Given σ^(j) and Δℓ, select a fine-positioning strategy.

else-if $\sigma^{(j)}$ is low (plan fails as the part is in a difficult-to-reach posture)

• Randomize-bin and restart perception characterization

else Initiate human intervention (Section 7) to detect part with high confidence

• If match found, then proceed with singulation planning else Send *Randomize-bin* command to robot

else Initiate human intervention (Section 7) to reduce $\sigma^{(j)}$

• If match found, then proceed with singulation planning else Send *Randomize-bin* command to robot



Fig. 4. Different failure scenarios during singulation of a desired part from the bin: (a) Failure during the approach phase. The two striped circles represent the relative positions of the two fingers with respect to the part to be grasped during approach. (b) Failure during the grasping phase. (c) Failure during the extraction phase. (d) Failure as a result of the part being tangled with another part.

A flow chart of the high-level planner is shown in Fig. 7.

4. Confidence estimation in perception

Given a mixed bin $\mathcal{B}(\kappa, n, \{n_i\})$ and a desired part p_d to be singulated, the first step in the high-level decision making in the hybrid cell (Section 3.3) is to characterize the uncertainty in estimating the 6D posture of a part instance $p^{(j)}$. This part instance is detected by using an automated perception system, while reporting pose estimate $\hat{\ell}^{(j)}$ with postural uncertainty $\sigma^{(j)}$.

4.1. Automated perception algorithm

Given a CAD model of the desired part to be singulated and the 3D point cloud of the mixed-bin, the automated perception system attempts to identify both an instance of that part in the bin and its 6D posture. Let $\mathcal{P} = \{p_i : p_i \in \mathbb{R}^3, \}$ be the point cloud of the bin of parts captured from the 3D sensor. Let $Q = \{q_i : q_i \in \mathbb{R}^3\}$ be the point cloud obtained by uniform surface sampling of the CAD model of the part to be identified. Our approach consists of extracting features (e.g., edges) available in the sensed data and

exploiting these features to collapse the problem from a 6D search to a finite number of line searches. Feature extraction [65–67] is one of the preprocessing procedures used in many scene reconstruction tasks. The extracted features help in docking the CAD model of the desired part at possible postures in the point cloud of the scene where a part match is likely to be found. The algorithm steps are given below:

- 1. Estimate surface normals at each point in the point cloud.
- 2. Cluster surface normals into a Gauss map to recognize planes.
- 3. Use intersection of planes to extract oriented edges.
- 4. For each oriented edge
- (a) Align the part CAD model along the oriented edge.
- (b) Filter the CAD model to contain only the points perceivable from the camera for that orientation of the CAD model.
- (c) Obtain a part match by moving the filtered CAD model Q_f along the edge where it is docked as a function of a translation parameter *s*, and finding the *s** that minimizes the mean point-to-point distance ρ from the filtered CAD model to the point cloud from the sensor.

$$\rho = \min_{s} \frac{1}{|Q_f|} \sum_{i=1}^{|Q_f|} d(q_i, \mathcal{P})$$
(2)



Perception complexity

Fig. 5. Graph showing examples of bin scenarios with varying degrees of perception and singulation complexities: (a) The part is easy to perceive as it gives a good point cloud. It is also easy to singulate as it enables collision-free approach, a good quality grasp, collision-free extraction, and finally a tangle-free singulation. (b) The part is difficult to perceive as it does not give a good point cloud. However, it is easy to singulate. (c) The part is easy to perceive but difficult to singulate. (d) The part is difficult to perceive and singulate as well.

where
$$d(q_i, \mathcal{P}) = \min_i || q_i - p_j ||, \quad q_i \in Q_f, p_j \in \mathcal{P}$$
 (3)

5. Select the match that gives the minimum ρ .

$$\frac{1}{2} + \int_{\frac{L}{2} - x_m}^{0} \frac{\exp(-x^2)}{\sigma^2} dx$$
(4)

$$\delta_m \notin (\mu - 3\sigma, \mu + 3\sigma) \tag{5}$$

 \Rightarrow low confidence (6)

We illustrate the working of the algorithm by applying it to detect the part shown in Fig. 8(a) from a simple bin shown in Fig. 8 (b). This part presents recognition as well as grasping

complexities. In particular, the quality of the point cloud corresponding to this part is heavily influenced by its orientation relative to the 3D camera. Whereas the part is symmetric along its longitudinal axis, it is asymmetric along its lateral axis making the grasping problem nontrivial. Fig. 8(c) shows the corresponding point cloud obtained from a 3D camera. Figs. 9(a)-(d) show the steps in the part matching algorithm. Fig. 10 shows the matching results by running the algorithm on some representative bin scenarios. In particular, this experiment reveals how the matching performance (ρ value) changes as a function of bin complexityparts of same type not touching with each other (Fig. 10(a, b)), parts of same type overlapping with each other (Fig. 10(c, d)), and parts of different type overlapping with each other (Fig. 10(e, f)). Fig. 11 illustrates a bin scenario that results in a part matching failure, where the desired part model (highlighted) is localized erroneously.

4.2. Confidence estimation

We compute confidence estimate in the part-matching result of the perception algorithm by using a signature based method. This involves obtaining (1) the ideal part match signature, (2) reference signatures based on synthetically generated point clouds, (3) the probability distribution of dissimilarity between ideal and reference signatures, and (4) the observed signature based on the test point cloud.

Given a sample point cloud of a single part and its CAD model, a part match signature is defined as the fraction of points ξ for which the minimum point-to-point distance $d(q_i, \mathcal{P})$ given in Eq. (3) is below a threshold distance d_t , plotted as a function of d_t . Note that this is a monotonically non-decreasing function.

The ideal signature is generated by performing calibration experiments to obtain the sensor noise model. Note that points from a sampled CAD model are used in the computation of ρ , which degrades the approximation of true ρ . To address this issue, we use a perfect cuboid-shaped object (Fig. 12(a)) in the calibration experiments. The CAD model of the object can be approximated by orthogonal planes. This enables us to compute point-to-plane distances, which gives a better approximation of ρ by isolating the sampling noise and discretization error and only accounting for sensor noise. The experiment is performed by placing the object in the scene such that three orthogonal planes are exposed to the sensor and obtaining a point cloud. Next, the automated



(7)

Fig. 6. (a) Part in the initial grasped state. (b) Specified final location of the part. (c) Good placement. (d) Bad placement.



Fig. 7. Flow chart of the high-level planner.

perception algorithm described above is run to match the point cloud with the plane-fitted CAD model. The match is shown in Fig. 12(b). Now, $d(q_i, \mathcal{P})$ is computed as the minimum point-to-plane distance and used to generate an ideal part match signature.

Fig. 13 shows an ideal signature and part match signature obtained by placing a real part in the scene. Note from the figure that the signature deviates as the part is modified (80% shrunk and 120% elongated). Also, the part match signature changes significantly for a different part. The dissimilarity of each part match signature from the ideal signature can be obtained by computing the corresponding difference in the area-under-the-curve of the two signatures.

Next, we must model the probability distribution of dissimilarity for a given part. First, a reference signature for the part of interest is obtained based on a synthetic point cloud that is representative of a real point cloud. This is generated by placing a part CAD model at an appropriate relative distance from a virtual camera in a simulated scene. There are mainly five sources of error that deviate the synthetic signature from the reference signature of the real part:

- 1. CAD model sampling error.
- 2. Algorithm moves in discrete steps.
- 3. The CAD model dimensions differ slightly from that of the real part.
- 4. Gaussian sensor noise.
- 5. Some points (mainly near part boundaries) are not visible due to sensor noise

The first two errors are taken care of by using the same CAD model sampling and the same discretization steps of the matching



Fig. 8. Inputs to the pose estimation algorithm: (a) CAD model of the part to be singulated. (b, c) Raw image and the corresponding 3D point cloud obtained from the Ensenso 3D camera.



Fig. 9. Example illustrating the various stages of the automated perception algorithm: (a) Gauss map used to detect planes. (b) Edge extracted from intersection of two planes. (c) Initial docking of the CAD model along an oriented edge. (d) Final match obtained by translation of the CAD model along the oriented edge.



Fig. 10. Representative bin scenarios and corresponding matches: (a, b) Multiple parts of same type not touching with each other. (c, d) Multiple parts of same type overlapping with each other. (e, f) Multiple parts of different type overlapping with each other.

algorithm as used for the real part. The third source of error is accounted for by measuring the dimensions of the real part manually and using them to create a better approximation of the real part. The fourth error is addressed by adding Gaussian noise into the synthetic point cloud. The final source of error is accounted for by randomly culling a few percent of points such that points near boundaries have much higher probability of removal than interior points. The signatures for the synthetic part and a real part, each in five different postures are shown in Fig. 14. Note from the figure that the synthetic signatures closely approximate the signatures of the real part.

By using the above procedure, a set of 100 synthetic signatures was obtained and a histogram of the corresponding dissimilarities, along with dissimilarities for the real part in 10 different postures,



Fig. 11. Bin scenario that results in a part matching failure.



Fig. 12. (a) Object used for calibration. (b) Match obtained between the point cloud of the scene and the filtered CAD model.



Fig. 13. Different signature curves: ideal signature, signatures for original real part and its modified versions, and signature for a different part.

was used to approximate the probability distribution of dissimilarity between ideal and reference signatures (Fig. 15). The resulting dissimilarity distribution can be approximated as a normal distribution with a mean μ =0.9751 and standard deviation σ =0.0659. The standard deviation in position σ_p =0.51 mm and standard deviation in orientation σ_o =0.43°. Given an observation, which is a point cloud of the bin and the filtered CAD model of the desired part, the observed signature is obtained and its dissimilarity with ideal is computed. This observed dissimilarity is used in conjunction with the dissimilarity probability distribution for the purpose of confidence estimation. If the measured dissimilarity is not in the range [$\mu - 3\sigma$, $\mu + 3\sigma$] (=[0.77, 1.17]), then it implies that the confidence in the part match is low, thereby declaring the part match as a failure.



Fig. 14. Signature for a synthetic part along with signatures for a real part in 10 different postures.

Another parameter that influences matching performance, and thereby the confidence measure, is the percentage of points in the point cloud of the CAD model that are filtered either due to self occlusions or occlusions due to other neighboring parts. Therefore, whenever the filtered points are above a certain threshold (arbitrarily, we chose 70%), we declare the part match as a failure.

5. Singulation planning under perception uncertainty

Per Definition 5, singulation is the concatenation of the four stages of positioning, approaching, grasping, and extracting. The success of tangle-free singulation depends on postural uncertainty, grasp-approach quality q_a , grasp quality based on force-closure q_g ,



Fig. 15. Probability distribution of dissimilarity approximated as a normal distribution with mean μ =0.9751 and standard deviation σ =0.0659 based on the histogram of dissimilarities.

and whether the part is tangle-free during singulation or not. Accordingly, we present a method that incorporates all the above factors to generate and evaluate singulation plans. In particular, each singulation plan is evaluated by estimating the over all probability of successful tangle-free singulation $\wp(s|p_i^{(j)}, \hat{\ell}_i^{(j)}, \sigma_i^{(j)}, \rho_i^{(j)}, \varepsilon)$ for each part instance $p_i^{(j)} \in \mathcal{P}_v$. Fig. 16 shows the overall system architecture used for plan generation and evaluation.

5.1. Plan generation

Each singulation plan is constructed by using four key postures: initial approach posture, pre-grasp posture, grasp posture, and extraction posture. Intermediate waypoints are generated through linear interpolation between neighboring postures. Note that only the position of the gripper changes during motion through the waypoints, while its orientation remains the same. However, we allow orientation changes at the transition between two postures. Between pre-grasp and grasp, the location of the gripper remains constant, and the separation between the fingers decreases until the part is grasped. The orientation of the gripper depends on a grasping strategy for the part which is computed offline.

5.1.1. Offline computation of grasp strategies

We use the popular force-closure quality metric [41] to evaluate the grasp candidates. For each contact point, following [42,68,69], we verify if it satisfies a force-closure constraint. Consider two points *a* and *b* where the two fingers make contact with the part's surface. The grasp at the contact pair is said to satisfy the force-closure constraint if each point lies in the friction cone of the other point (Fig. 17(a)). The friction cone at each point is oriented about the inward normal making a semi-angle $\tan^{-1}(\mu)$, where μ is the coefficient of friction. Now, we compute the grasp quality as the number of points that satisfy force-closure divided by the total number of points that project onto the finger surface.

For a particular grasp configuration, the pair of points where the center-axis of the gripper (along the pinching direction)



Fig. 16. Singulation planning architecture.



Fig. 17. (a) Friction cone illustration. (b) A representative grasp configuration where points with black normals are not in contact with the gripper finger-pads, red ones make contact but do not satisfy friction-cone property, and green ones are both in contact with gripper and stable grasps based on friction-cone concept described below. (c) Top 20 grasp postures displayed on the CAD model of a given part. Grasps are sampled randomly and those which have non-zero grasp quality are plotted by depicting the corresponding contact pairs. Green grasps represent the top 20 grasp qualities. The rest are represented as red. The size of the line segment is proportional to the quality of each grasp. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



Fig. 18. Mixed-bins used in the experiments.

intersects the part's surface is uniquely determined (Fig. 17(b)). Therefore, we can represent each grasp candidate by such point pairs. Fig. 17(c) shows the grasp quality of the best 20 grasp pairs evaluated using the above method for an industrial part. Sampled point clouds are generated for the CAD models of the part and the gripper and used in the grasp quality computations. In the figure, the length of each line segment at a grasp point is proportional to the corresponding grasp quality of that grasp pair.

5.1.2. Sampling based planner

We use a sampling based planner that generates several random plans. The initial approach posture is sampled at a safe height from the bin and in a small region around a nominal posture that corresponds to the grasp candidate that is ranked best by the above grasp quality metric. The pre-grasp posture is sampled in a small region around the estimated posture of the target part. The extraction posture is uniformly sampled at a safe height from the bin.

A Monte Carlo simulator evaluates each sampled plan by computing its probability of failure. The point cloud obtained from the 3D sensor is split into two: one consisting of only the points in the bin, excluding the part to be picked, and the other point cloud consisting of those points of the part that were captured by the 3D camera. The simulation scene involves the gripper, part CAD model, point cloud of the bin excluding the part, and the point cloud of the part. The CAD models of the part and gripper are also converted to sampled point clouds before adding them to the scene.

Each simulation run works in the following way. Given that an automated perception system provides an estimate of the pose of target part $\hat{\ell}_i^{(j)}$ with an error that follows a Gaussian distribution $\mathcal{N}(0, \sigma_i^{(j)})$, this is simulated by placing a CAD model of the part at



Fig. 19. Snapshots from an animation of a successful singulation plan. (a-c) Approach. (d-f) Pre-grasp to grasp. (g-i) Extraction.



Fig. 20. Snapshots from an animation of a failed singulation plan.

the estimated posture and shifting it by a value drawn from the above distribution of pose-error. Now, a candidate plan is evaluated by moving the gripper through the way-points, while checking for a collision at each way-point. The Point Cloud Library is used in C++ to check for collisions. A collision is said to occur between two point clouds when the minimum clearance between them falls below a certain threshold. If the way-point belongs to approach phase, then collision is checked between gripper and the entire scene. If the way-point belongs to the grasping phase, then collision is checked

between gripper and the bin excluding part. If the way-point belongs to the extraction phase, then collision is checked between gripper and the bin excluding part, as well as between part and the rest of the bin. These collision check conditions ensure that we achieve tangle-free singulation of the part. If a collision is returned for at least one way-point during a trial, then that trial is classified as a failure. If there are *m* such failure runs out of a total of *n* runs, then the probability of failure for the specified plan is $\frac{m}{n}$. The plan which minimizes the probability of failure is chosen as the execution plan.



Fig. 21. Graph of average clearance as a function of step number for plans with varying probabilities of failure.



Fig. 22. Graph of singulation success rate as a function of increasing perception uncertainty.

5.2. Characterization of influence of perception uncertainty on singulation plans

We first report results from illustrative experiments to show the working of the sampling based plan generation and Monte Carlo-based plan evaluation. Bin 3 in Fig. 18 is used in these experiments. We use the same part (Fig. 8(a)) that was used in Section 4. Simulations were performed to compute failure probabilities P_f of different plans generated by the planner. Fig. 19 shows snapshots from an animation of a sample Monte Carlo trial showing different stages of a successful singulation plan. Fig. 20 shows snapshots from an animation resulting in a failed singulation plan where gripper collides with the part to be singulated during approach.



Fig. 24. The two types of deviation considered in the initial grasped state.

It took 0.5 s to compute each trial. The computer running the simulations consisted of Intel Core i7 2600 @ 3.4 GHz CPU and 8-GB Dual-Channel DDR3 RAM memory. For each plan, a set of 100 trials was simulated with a position uncertainty of 2 mm and orientation uncertainty of 4° in each axis, added into the estimated 6D posture of the target part in each trial. Fig. 21 shows the graph of average clearance as a function of step number (1–5 Approach, 6–10 Pre-grasp to Grasp, 11–15 Extraction) for five sampled plans with varying probabilities of failure. When the extraction location was directly above the estimated location of the target part, $P_f=1$ ("
__"-marked curve). This was due to collision with a neighboring part in the bin during extraction in every trial. But as the extraction point was moved away from this location, $P_{f}=0$ ("o"-marked curve). Whenever the average minimum clearance dips below a threshold of \approx 3 mm, we flag the state as collision and the plan is aborted. The clearance values after this point for each plan are only averaged over the trials that have been reported as success by the simulator. For another plan with $P_f = 1$ (" \star "-marked curve), some of the trials failed during approach and the remaining during extraction at step 12. For the plan with $P_f = 0.86$ (" \diamond "-marked curve, most of the trials failed indicating that it was a bad plan for the current uncertainty model. The plan with $P_f=0$ is representative of an ideal plan for this uncertainty model. At every step in the plan, the average minimum clearance is safely above the threshold value. For the plan with $P_f=0.166$ (" \triangle "-marked curve), some of the



Fig. 23. Robot using the pose estimated by the system to proceed with the part singulation task.



Fig. 25. Five sample cases illustrating different initial grasp states and corresponding postures of the part after dropped off by the robot.



Fig. 26. Average positional error after direct drop off with varying uncertainties in the initial grasped posture.

trials failed due to collision during approach as a result of uncertainty in pose estimation.

Next, we analyzed how the probability of success of a successful plan degrades with increasing uncertainty introduced into the estimated posture of the part. We considered standard deviation increments of 2 mm in position along each axis and a fixed standard deviation of 4° about each orientation axis. Note that the perception uncertainty levels (0.51 mm in position and 0.43° in orientation) found in Section 4 are well within the uncertainty values considered for analyzing the singulation plans. We conducted this experiment for four bin samples and one plan with success probability equal to one when uncertainty is zero. A set of 100 Monte Carlo runs was used to compute the probability for each uncertainty level. To accommodate uncertainty in the estimated part pose, maximum gripper pad separation was used during the approach phase to guarantee the encompassing of the part. But in doing so, the fingers might collide with neighboring parts resulting in a singulation failure. Fig. 22 shows the graph of success probability as a function of postural uncertainty for the four bins shown in Fig. 18. The success probability of singulation plans starts degrading from a standard deviation of 9 mm in positional error. Note that as the bin gets more cluttered in the neighborhood of the part to be picked, the degradation will begin at a smaller positional uncertainty.

The uncertainty information provided by the perception algorithm (from Section 4) is integrated into the above analysis to compute success probability, which is used as a decision variable during the singulation-plan execution phase. That is, for a given uncertainty found from the perception result, if the success probability, computed based on the above evaluation, is above τ_s (=0.99), then robot proceeds with executing the singulation plan. Otherwise, either human intervention or randomize-bin command is invoked based on the uncertainty level. For example, if the uncertainty is high, then human intervenes and adjusts the part



Fig. 27. Average orientation error after direct drop off with varying uncertainties in the initial grasped posture.



Fig. 28. Illustration of the fine-positioning subtask: (a,b) Moves for rotational error correction. (c) Moves for translational error correction.

posture so that the robot can detect the part with higher confidence. However, if the uncertainty is low and still the planner does not find a successful plan, then the part could be in a difficult-to-reach posture. Hence, it is best to randomize the bin and restart perception characterization. An example of a successful singulation plan implementation is shown in Fig. 23.

6. Correcting destination posture errors using sensorless finepositioning

From the problem definition in Section 3, after each part $p_i^{(j)} \in \mathcal{P}$ has been singulated from the bin, it must be placed at a destination posture $\mathcal{E}_i^{(j)}$ within an expected postural accuracy $\Delta \mathcal{E}$. Factors like initial grasped postural uncertainty induced by perception uncertainty, positioning accuracy of the end-effector, and momentum imparted to the part during drop-off degrade the

postural accuracy that can be achieved at the destination. According to the second step in the high-level decision making in the hybrid cell (Section 3.3), a fine-positioning method is invoked to correct these postural errors.

6.1. Characterization of destination postural error in terms of initial grasped posture uncertainty

We considered varying uncertainty in the initial grasped state and examined the error in the destination posture after drop-off. To minimize the robot positional error, a sequence of two robot moves, with coarse and fine motion-planning parameters, were used before drop-off. (This corresponds to the second fine-positioning strategy that is considered in the next subsection.) Standard deviations of 5 mm, 8 mm, 10 mm, 12 mm, and 15 mm were considered in the initial position along the vertical direction. Standard deviations of 3°, 5°, 8°, and 10° were considered in the orientation in the pinch axis. Deviations in position in the



Fig. 29. Snapshots from a video showing the execution of one sample plan that results in the positioning of the part within the accuracy limits: (a) Initial grasped state. (b) Gripper at drop-off location. (c) Part dropped off at final location. (d) Rotational error correction in the clockwise direction. (e) Rotational error correction in the counter clockwise direction. (f) Gripper moved to fine-positioning begin location. (g) Gripper rotated by 90°. (h) Gripper opened by fixed amount to correct translational error. (i) Gripper moved to neutral location.



Fig. 30. T-shaped tool used to push a flat surface.

horizontal plane and orientations in the approach and longitudinal axes of the gripper were not considered, since any perception uncertainty in these directions are disambiguated when the part is grasped. The two deviations considered are shown in Fig. 24. Five sample cases illustrating different randomly initialized grasped states and corresponding postures of the part after drop off by the robot are shown in Fig. 25. The average positional error and orientation error after direct drop off, with varying uncertainties in the initial grasped posture, for 30 trials are shown in Figs. 26 and 27, respectively.

6.2. Design of fine-positioning strategies

We describe an empirical methodology based on a representative part (Fig. 8(a)) to select from a suite of fine-positioning strategies that offer different tradeoffs between completion time and postural accuracy at the destination. Once the part is dropped off at the destination, the basic fine-positioning task consists of the gripper applying a finite set of sliding forces and moments on the part until its posture is within the desired postural limits. We consider parts with the following properties: (1) When the part is lying on a flat surface in one of its stable postures, two of the three orientation parameters are frozen (or become zero by using an appropriate coordinate frame assignment), leaving only one rotation parameter about the vertical axis. (2) When the part is in a stable posture, there exists at least one vertical flat face to which sliding forces can be applied. Given the above assumptions on the part, there are two translational errors and one rotational error associated with the posture of the part at the goal location. The fine-positioning steps that achieve rotational and translational error correction are illustrated in Fig. 28.

The specific algorithmic steps to achieve transport and finepositioning are given below:

Robot moves the grasped part from the singulation location to a
position vertically above the desired location. This is performed
in two steps. Initially a coarse motion plan is used that transports the part rapidly to the desired location, but with some



Fig. 31. Translation and orientation errors and execution times for five different regimes: (1) drop-off with coarse motion parameters, (2) drop-off with coarse and fine motion parameters, (3) drop-off with coarse and fine motion parameters and rotational corrective fine positioning moves, (4) drop-off with coarse and fine motion parameters and translational corrective fine positioning moves, and (5) drop-off with coarse and fine motion parameters and both rotational and translational corrective fine positioning moves.

positional error. Next, a fine motion plan is generated that reduces the positional error at the drop-off location.

- Robot drops the part at the desired location. Considering the uncertainty in the grasp location, the object drop-off location is offset by the maximum uncertainty value in the negative *Y* direction. This offset will ensure that the final location after drop-off will always have an error in the positive *Y* direction and hence the sliding would only be necessary in that direction.
- The grippers are closed with low gripping force to correct the translational error in the positive and negative *X* directions.
- A rotational error correction is applied by rotating the gripper by an appropriate amount about the vertical. Since the system is sensor-less, the system is unaware if the error is in the clockwise or in the counter clockwise direction. Thus the rotational error correction is performed in both directions. In all the cases, only one of the moves would be contributing to the error correction. For example, assume that the robot's gripper first performs a clock-wise rotation, which is followed by a counterclockwise rotation. Further assume that the angular offset of the part is in the clock-wise direction. We can see that the part remains at rest during the first move, while the part's angular position gets adjusted during the second move as shown in Fig. 28(a).
- The gripper is lifted off and positioned behind the vertical face of the part. This position is chosen by considering the maximum expected positional error. This thereby ensures that the gripper is behind the part in all cases.
- The gripper moves a fixed distance in the positive *Y* direction that is sufficient enough to nudge the part into its postural accuracy limits.

Fig. 29 shows snapshots from a video showing the execution of one sample plan that results in the positioning of the part within the accuracy limits. Note from Fig. 29 that the sliding forces are directly applied by the finger pads. A push tool with a matched shape with respect to the part being pushed can be used to ensure better sliding movements. For example, a T-shaped tool is used in Fig. 30 to push a flat surface. Similarly a convex tool front can be used for a part with a concave surface and so on.

Next, we characterized the performance of different fine-positioning strategies We considered five fine-motion strategies:

- 1. Drop-off with coarse motion parameters.
- 2. Drop-off with coarse and fine motion parameters.
- 3. Drop-off with coarse and fine motion parameters and rotational corrective fine positioning moves.
- 4. Drop-off with coarse and fine motion parameters and translational corrective fine positioning moves.
- Drop-off with coarse and fine motion parameters and both rotational and translational corrective fine positioning moves.

Fig. 31 shows the average translation errors, orientation errors, and execution times for each motion strategy across 10 trials. We observe that as more corrective moves are added, the accuracy improves both in position and orientation, while the completion time increases.

Now, we use the above results to select a fine-positioning strategy. For example, from Figs. 26 and 27, an initial grasping uncertainty of 10 mm standard deviation in position and 10° standard deviation in orientation leads to an average translation error of about 5 mm and an average orientation error of 3°. Now, from Fig. 31 if this meets the desired postural accuracy requirements, we use the second fine-positioning strategy. Otherwise, if an accuracy of (2 mm, 2°) is required, we use the fifth fine-positioning strategy.

It took about 20 min to conduct the fine positioning experiments for this part. Whenever we have a new part, we use the above experiments based methodology to select a fine-positioning strategy based on the accuracy and completion time requirements.



Fig. 32. (a) Robot sending Skype call to a remotely operating human requesting for help. (b) 2D image of the bin and CAD model of the part to be identified. (c) User interface used by the remote human to resolve part recognition and pose estimation failures.



Fig. 33. Illustration of human identifying correspondences between edges in the image with those in a CAD model.

7. Design of user interface to enable remote human interventions

We have developed a new user interface (Fig. 32) that allows a remote human to perform pose estimation in scenes with high clutter where the automated perception system may fail. The system makes a Skype call to the remote human when help is

needed and sends three pieces of information: the raw camera image of the scene, the corresponding point cloud, and the CAD model of the part to be picked.

The human operator selects features (edges) from the 2D image and shows a correspondence in the CAD model (Fig. 33). The algorithm uses these features to estimate the part location and orientation in 3D and dock the CAD model at this pose. The human can do minor adjustments to the pose using a joystick. The *x* and *y* information in the image space is transformed to point cloud coordinates using scaling and translation operations.

7.1. Accuracy/time tradeoff

There is a tradeoff between accuracy and time needed to extract the data. Orientation accuracy impacts grasping performance. The accuracy needed to grasp a part successfully depends on its shape complexity and its particular posture. This information is pre-determined for each part and conveyed to the human operator, who can stop the estimation process once a good enough orientation accuracy is obtained. For this purpose, we placed a single instance of the target part on a tripod and used a digitalinclination meter to set the orientation of the part at a known posture. In one sample experiment, we used a nominal orientation of 30° about the longitudinal axis of the part and 35 degrees about the lateral axis of the part. Next, we manually introduced 2-degree



Fig. 34. Average time taken (in seconds) by the human to complete the perception task for 10 trials in each regime across four different parts. Success rates of 100%, 80%, and 100% were achieved in the first, second, and third regimes, respectively, for all the four parts.



Fig. 35. Comparison of average time taken (in seconds) by two users to complete the perception task for 10 trials in each regime for the white part.

increments of perception error about each axis and observed its impact on grasping performance. For the part shown in Fig. 8(a), we noticed that the robot was able to successfully grasp up to an error of $\pm 8^{\circ}$ about the longitudinal axis. We noticed a high asymmetry about the lateral axis with successful grasping up to 8° in the clockwise direction and only 2° in the counter clockwise direction. This accuracy characteristic is made available to the human during training, which makes the human aware of how closely to match, and hence decide how much time to spend on the task.

7.2. Evaluation of the user interface

In these experiments, we considered complex bin examples where the failure rate of the automated perception system was more than 50%. Fig. 32 shows the user interface used by the human to perform part matching whenever the automated perception system failed (Fig. 11). Fig. 32(c) shows the CAD model docking using the edge-selection method. The user interface provides different functions that allow the human operator to achieve the part matching task. We conducted experiments to analyze the influence of different combinations of these features on both the time taken to solve the problem and the overall success rate of the

singulation task. Accordingly, the effectiveness of the user interface was evaluated cross three experimental regimes:

- 1. Usage of only joystick to move the CAD model and dock it at an appropriate posture in the point cloud.
- 2. Usage of only the edge selection method to directly dock the CAD model.
- 3. Usage of the edge selection method to dock the CAD model, and subsequently the joystick to do any fine adjustments if necessary.

We conducted 120 experimental trials. Each trial consisted of the human using one of the three methods to perform the part matching task. Each trial was validated by sending the extracted postural information to the robot and verifying whether or not the robot could singulate the specified part by using this information. We conducted ten trials for each regime and across four parts with different geometries. We expect that this task will be performed by experts in real industrial settings. Therefore, all trials were carried out by a well-trained user. The singulation success rate was 80% in the second regime where only edge selection was used to register the part. In the first and third regimes, the success rate was 100%. Because of high success rates, 10 user trials per regime was sufficient to validate the effectiveness of the user interface. The time taken (in seconds) by the human to complete the perception task over ten trials in each regime for all the four parts is shown in Fig. 34. Similar performance was observed across the parts for all the regimes. The edge-selection only took the least time for all the parts, but with some failure rate. Therefore, the third regime that ranked second in terms of time, and with 100% success rate was chosen as the best solution.

In the third regime, the user spends about 10 s in edge selection and subsequently about 25 s using the joystick to improve the estimated posture. Note that about 80% success rate can be expected with only edge selection (from second regime). This information can be exploited by the user to reduce the time spent in using joystick to achieve a level of accuracy, which may be redundant.

Next, we tested the trainability of the interface. For this purpose, we trained a second user and conducted ten trials for the white part to compare the user's performance with that of the first user in all the three regimes. To have a common benchmark, the same data used by the first user was presented to the second user. The comparison was only limited to the part-matching task in these experiments, since the same bin settings were no longer available to proceed with the singulation task. Instead, difference in transformations was computed and used as a comparison metric. Fig. 35 shows the comparison of time taken by the two users to complete the perception task. The second user took an average of 36.7 s to complete the perception task for the white part, in the third regime, which is very close to that of the first user. Similar performance was observed in first and second regimes.

8. Conclusions

We presented an approach that treats coping with uncertainty as a key step to handle failures and enhance performance in robotic unstructured bin-picking. The principal contributions in the paper include:

- 1. A method to characterize uncertainty in pose estimation of a part match found by using an automated perception system.
- A mechanism for the rationalized basis to detect failures and invoke human interventions.
- 3. A new user interface that allows the remote human to provide distinguishing cues to reduce uncertainty in part pose estimates. This also enables part detection.
- 4. A singulation planner that incorporates uncertainty into plan evaluation.
- 5. A fine positioning planner for correcting errors in destination part posture.

In our previous work, we have developed other modules including an ontology for task partitioning in human-robot collaboration for kitting operations [70], resolving occlusions in robotic bin-picking [71], sequence planning for complex assemblies [72], instruction generation for human operations [73], bimanual robotic cleaning [74,75], and ensuring human safety [76]. Future work consists of investigating how to integrate these modules to develop hybrid work cells, where humans and robots collaborate to carry out non-repetitive industrial tasks. Currently, we used a single-human-single-robot model. We would like to extend this to a single-human-multi-robot model, where a single human is remotely bailing out multiple robots that may be stationed either at the same work cell or different work cells. An interesting future research direction lies in exploring whether the robot can learn, on-the-fly, as more information becomes available from the mistakes made and the corresponding human interventions. Other future works include exploring alternative sensor modalities, handling of heavy parts with more complex geometries, and sensor-based fine-positioning strategies.

Acknowledgments

This work is supported in part by National Science Foundation Grants #1200087 and #1527220 and National Institute of Standards and Technology Cooperative Agreement #70NANB15H250. Opinions expressed are those of the authors and do not necessarily reflect opinions of the sponsors.

References

- C. Schlenoff, Z. Kootbally, A. Pietromartire, M. Franaszek, S. Foufou, Intention recognition in manufacturing applications, Robot. Comput.-Integr. Manuf. 33 (2015) 29–41, Special Issue on Knowledge Driven Robotics and Manufacturing.
- [2] S. Balakirsky, Z. Kootbally, C. Schlenoff, T. Kramer, S.K. Gupta, An industrial robotic knowledge representation for kit building applications, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2012, pp. 1365–1370.
- [3] Z. Kootbally, C. Schlenoff, C. Lawler, T. Kramer, S. Gupta, Towards robust assembly with knowledge representation for the planning domain definition language (pddl), Robot. Comput.-Integr. Manuf. 33 (2015) 42–55, Special Issue on Knowledge Driven Robotics and Manufacturing.
- [4] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T.K. Marks, R. Chellappa, Fast object localization and pose estimation in heavy clutter for robotic bin picking, Int. J. Robot. Res. 31 (8) (2012) 951–973.
- [5] J.A. Marvel, K. Saidi, R. Eastman, T. Hong, G. Cheok, E. Messina, Technology readiness levels for randomized bin picking, in: Proceedings of the Workshop on Performance Metrics for Intelligent Systems, PerMIS, New York, NY, USA, ACM, 2012, pp. 109–113.
- [6] H. Golnabi, A. Asadpour, Design and application of industrial machine vision systems, Robot. Comput.-Integr. Manuf. 23 (6) (2007) 630–637.
- [7] A.M. Pinto, L.F. Rocha, A.P. Moreira, Object recognition using laser range finder and machine learning techniques, Robot. Comput.-Integr. Manuf. 29 (1) (2013) 12–22.
- [8] A. Pochyly, T. Kubela, V. Singule, P. Cihak, 3d vision systems for industrial binpicking applications, in: 15th International Symposium on MECHATRONIKA, December 2012, pp. 1–6.
- [9] K. Bohringer, B. Donald, L. Kavraki, F. Lamiraux, Part orientation with one or two stable equilibria using programmable force fields, IEEE Trans. Robot. Autom. 16 (April) (2000) 157–170.
- [10] K.N. Kaipa, A.S. Kankanhalli-Nagendra, S.K. Gupta, Toward estimating task execution confidence for robotic bin-picking applications, in: AAAI Fall Symposium: Self-Confidence in Autonomous Systems, 2015.
- [11] K.N. Kaipa, S.S. Thevendria-Karthic, S. Shriyam, A.M. Kabir, J.D. Langsfeld, S.K. Gupta, Resolving automated perception system failures in bin-picking tasks using assistance from remote human operators, in: Proceedings of IEEE International Conference on Automation Science and Engineering, August 2015.
- [12] K.N. Kaipa, S. Shriyam, N.B. Kumbla, S.K. Gupta, Automated plan generation for robotic singulation from mixed bins, in: IROS Workshop on Task Planning for Intelligent Robots in Service and Manufacturing, 2015.
- [13] K.N. Kaipa, N.B. Kumbla, S.K. Gupta, 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, 2015.
- [14] G. Sansoni, P. Bellandi, F. Leoni, F. Docchio, Optoranger: a 3d pattern matching method for bin picking applications, Opt. Lasers Eng. 54 (2014) 222–231.
- [15] H.-Y. Kuo, H.-R. Su, S.-H. Lai, C.-C. Wu, 3d object detection and pose estimation from depth image for robotic bin picking, in: Proceedings of IEEE International Conference on Automation Science and Engineering (CASE), 2014, pp. 1264– 1269.
- [16] D. Holz, M. Nieuwenhuisen, D. Droeschel, J. Stückler, A. Berner, J. Li, R. Klein, S. Behnke, Active recognition and manipulation for mobile robot bin picking, in: Gearing Up and Accelerating Cross-fertilization between Academic and Industrial Robotics Research in Europe, Springer, 2014, pp. 133–153.
- [17] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stuckler, A. Berner, J. Li, R. Klein, S. Behnke, Mobile bin picking with an anthropomorphic service robot, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 2327–2334.
- [18] A. Pretto, S. Tonello, E. Menegatti, Flexible 3d localization of planar objects for industrial bin-picking with monocamera vision system, in: Proceedings of IEEE International Conference on Automation Science and Engineering (CASE), 2013, pp. 168–175.
- [19] J.J. Rodrigues, J.-S. Kim, M. Furukawa, J. Xavier, P. Aguiar, T. Kanade, 6d pose estimation of textureless shiny objects using random ferns for bin-picking, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 3334–3341.

- [20] J.-K. Oh, S. Lee, C.-H. Lee, Stereo vision based automation for a bin-picking solution, Int. J. Control Autom. Syst. 10 (2) (2012) 362–373.
- [21] S. Lee, J. Kim, M. Lee, K. Yoo, L.G. Barajas, R. Menassa, 3d visual perception system for bin picking in automotive sub-assembly automation, in: Proceedings of IEEE International Conference on Automation Science and Engineering (CASE), 2012, pp. 706–713.
- [22] D. Buchholz, S. Winkelbach, F.M. Wahl, Ransam for industrial bin-picking, in: Proceedings of 41st International Symposium on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK), 2010, pp. 1–6.
- [23] K. Boehnke, Object localization in range data for robotic bin picking, in: Proceedings of IEEE International Conference on Automation Science and Engineering, CASE, 2007, pp. 572–577.
- [24] F. Boughorbel, Y. Zhang, S. Kang, U. Chidambaram, B. Abidi, A. Koschan, M. Abidi, Laser ranging and video imaging for bin picking, Assembly Autom. 23 (1) (2003) 53–59.
- [25] B.K. Horn, K. Ikeuchi, Picking Parts Out of a Bin, Tech. Report, Massachussetts Institute of Technology, 1982.
- [26] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, T. Hirai, Fast graspability evaluation on single depth maps for bin picking with general grippers, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1997–2004.
- [27] D. Buchholz, D. Kubus, I. Weidauer, A. Scholz, F.M. Wahl, Combining visual and inertial features for efficient grasping and bin-picking, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 875– 882.
- [28] D.C. Dupuis, S. Léonard, M.A. Baumann, E.A. Croft, J.J. Little, Two-fingered grasp planning for randomized bin-picking, in: Proceedings of the Robotics: Science and Systems Manipulation Workshop-Intelligence in Human Environments, 2008.
- [29] L.-P. Ellekilde, H.G. Petersen, Motion planning efficient trajectories for industrial bin-picking, Int. J. Robot. Res. 32 (9–10) (2013) 991–1004.
- [30] D. Buchholz, M. Futterlieb, S. Winkelbach, F.M. Wahl, Efficient bin-picking and grasp planning based on depth data, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 3245–3250.
- [31] S. Kristensen, S. Estable, M. Kossow, R. Brösel, Bin-picking with a solid state range camera, Robot. Auton. Syst. 35 (3–4) (2001) 143–151.
- [32] K. Harada, K. Nagata, T. Tsuji, N. Yamanobe, A. Nakamura, Y. Kawai, Probabilistic approach for object bin picking approximated by cylinders, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 3742–3747.
- [33] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, D. Burschka, Rigid 3d geometry matching for grasping of known objects in cluttered scenes, Int. J. Robot. Res. (2012), DOI:10.1177/0278364911436019.
- [34] S. Fuchs, S. Haddadin, M. Keller, S. Parusel, A. Kolb, M. Suppa, Cooperative binpicking with time-of-flight camera and impedance controlled dlr lightweight robot iii, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010, pp. 4862–4867.
- [35] L.-P. Ellekilde, J. Jorgensen, D. Kraft, N. Kruger, J. Piater, H.G. Petersen, et al., Applying a learning framework for improving success rates in industrial bin picking, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012, pp. 1637–1643.
- [36] A. Pronobis, B. Caputo, Confidence-based cue integration for visual place recognition, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2394–2401.
- [37] A. Sahbani, S. El-Khoury, P. Bidaud, An overview of 3d object grasp synthesis algorithms, Robot. Auton. Syst. 60 (3) (2012) 326–336.
- [38] A. Bicchi, V. Kumar, Robotic grasping and contact: a review, in: Proceedings of IEEE/RSJ International Conference on Robotics and Automation, vol. 1, 2000, pp. 348–353.
- [39] J. Bohg, A. Morales, T. Asfour, D. Kragic, Data-driven grasp synthesis—a survey, IEEE Trans. Robot. 30 (April) (2014) 289–309.
- [40] A. Miller, P. Allen, Graspit! a versatile simulator for robotic grasping, IEEE Robot. Autom. Mag. 11 (December) (2004) 110–122.
- [41] C. Ferrari, J. Canny, Planning optimal grasps, in: Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, May 1992, pp. 2290– 2295.
- [42] V.-D. Nguyen, Constructing force-closure grasps, Int. J. Robot. Res. 7 (3) (1988) 3–16.
- [43] M. Roa, R. Suarez, Computation of independent contact regions for grasping 3-d objects, IEEE Trans. Robot. 25 (August) (2009) 839–850.
- [44] Y. Zheng, W.-H. Qian, Coping with the grasping uncertainties in force-closure analysis, Int. J. Robot. Res. 24 (4) (2005) 311–327.
- [45] D. Berenson, S. Srinivasa, J. Kuffner, Addressing pose uncertainty in manipulation planning using task space regions, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2009, pp. 1419–1425.
- [46] J. Weisz, P. Allen, Pose error robust grasping from contact wrench space metrics, in: Proceedings of IEEE International Conference on Robotics and Automation, May 2012, pp. 557–562.
- [47] B. Kehoe, D. Berenson, K. Goldberg, Toward cloud-based grasping with uncertainty in shape: estimating lower bounds on achieving force closure with zero-slip push grasps, in: Proceedings of IEEE International Conference on

Robotics and Automation, May 2012, pp. 576-583.

- [48] J. Glover, D. Rus, and N. Roy, Probabilistic models of object geometry for grasp planning, in: Proceedings of Robotics: Science and Systems, 2008.
- [49] K. Goldberg, M.T. Mason, Bayesian grasping, in: Proceedings of IEEE International Conference on Robotics and Automation, vol. 2, May 1990, pp. 1264– 1269.
- [50] K. Hsiao, M. Ciocarlie, P. Brook, Bayesian grasp planning, in: ICRA Workshop on Mobile Manipulation: Integrating Perception and Manipulation, 2011.
- [51] M. Erdmann, M.T. Mason, An exploration of sensorless manipulation, IEEE J. Robot. Autom. 4 (August) (1988) 369–379.
- [52] M. Erdmann, M. Mason, J. VaněČek, Mechanical parts orienting: the case of a polyhedron on a table, Algorithmica 10 (2–4) (1993) 226–247.
- [53] S. Akella, W. Huang, K. Lynch, M.T. Mason, Sensorless parts orienting with a one-joint manipulator, in: Proceedings of IEEE International Conference on Robotics and Automation, vol. 3, April 1997, pp. 2383–2390.
- [54] K. Goldberg, Orienting polygonal parts without sensors, Algorithmica 10 (2–4) (1993) 201–225.
- [55] M. Erdmann, An exploration of nonprehensile two-palm manipulation, Int. J. Robot. Res. 17 (5) (1998) 485–503.
- [56] M. Dogar, S. Srinivasa, Push-grasping with dexterous hands: mechanics and a method, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2010, pp. 2123–2130.
- [57] S. Kristek, D. Shell, Orienting deformable polygonal parts without sensors, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 2012, pp. 973–979.
- [58] A. Holladay, J. Barry, L. Kaelbling, T. Lozano-Perez, Object placement as inverse motion planning, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 3715–3721.
- [59] F. Panahi, A.F. van der Stappen, Bounding the locus of the center of mass for a part with shape variation, Comput. Geom. 47 (8) (2014) 847–855.
- [60] Y. Bai, C. Liu, Dexterous manipulation using both palm and fingers, in: Proceedings of IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1560–1565.
- [61] F. Duan, J. Tan, J.G. Tong, R. Kato, T. Arai, Application of the assembly skill transfer system in an actual cellular manufacturing system, IEEE Trans. Autom. Sci. Eng. 9 (1) (2012) 31–41.
- [62] S. Takata, T. Hirano, Human and robot allocation method for hybrid assembly systems, {CIRP} Annals—Manuf. Technol. 60 (1) (2011) 9–12.
- [63] F. Chen, K. Sekiyama, H. Sasaki, J. Huang, B. Sun, T. Fukuda, Assembly strategy modeling and selection for human and robot coordinated cell assembly, in: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, pp. 4670–4675.
- [64] M. Morioka, S. Sakakibara, A new cell production assembly system with human-robot cooperation, {CIRP} Annals—Manuf. Technol. 59 (1) (2010) 9–12.
- [65] C. Weber, S. Hahmann, H. Hagen, Sharp feature detection in point clouds, in: Proceedings of Shape Modeling International Conference (SMI), 2010, pp. 175– 186.
- [66] K. Demarsin, D. Vanderstraeten, T. Volodine, D. Roose, Detection of closed sharp edges in point clouds using normal estimation and graph theory, Comput.-Aided Des. 39 (4) (2007) 276–283.
- [67] S. Gumhold, X. Wang, R. Macleod, Feature extraction from point clouds, in: Proceedings of the 10th International Meshing Roundtable, 2001, pp. 293–305.
- [68] B. Faverjon, J. Ponce, On computing two-finger force-closure grasps of curved 2d objects, in: Proceedings of IEEE International Conference on Robotics and Automation, 1991, pp. 424–429.
- [69] A. Blake, A symmetry theory of planar grasp, Int. J. Robot. Res. 14 (5) (1995) 425–444.
- [70] A.G. Banerjee, A. Barnes, K.N. Kaipa, J. Liu, S. Shriyam, N. Shah, S.K. Gupta, An ontology to enable optimized task partitioning in human-robot collaboration for warehouse kitting operations, in: Proceedings of SPIE, Next-Generation Robotics II; and Machine Intelligence and Bio-inspired Computation: Theory and Applications IX, 2015.
- [71] K.N. Kaipa, S. Shriyam, G.S.K., Resolving occlusions through simple extraction motions in robotic bin-picking, in: ASMEs 11th Manufacturing Science and Engineering Conference, June 2016.
- [72] C. Morato, K.N. Kaipa, S.K. Gupta, Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters, Comput.-Aided Des. 45 (11) (2013) 1349–1364.
- [73] K.N. Kaipa, C. Morato, B. Zhao, S.K. Gupta, Instruction generation for assembly operations performed by humans, in: Proceedings of ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2012, pp. 1121–1130.
- [74] A.M. Kabir, J.D. Langsfeld, C. Zhuang, K.K.N., G.S.K., Automated learning of operation parameters for robotic cleaning by mechanical scrubbing, in: ASMEs 11th Manufacturing Science and Engineering Conference, June 2016.
- [75] J.D. Langsfeld, A.M. Kabir, K.K.N., G.S.K., Online learning of part deformation models in robotic cleaning of compliant objects, in: ASMEs 11th Manufacturing Science and Engineering Conference, June 2016.
- [76] C. Morato, K.N. Kaipa, B. Zhao, S.K. Gupta, Toward safe human robot collaboration by using multiple kinects based real-time human tracking, J. Comput. Inf. Sci. Eng. 14 (1) (2014).