# Robotic Depowdering for Additive Manufacturing via Pose Tracking

Zhenwei Liu[1], Junyi Geng[2], Xikai Dai[1], Tomasz Swierzewski[1], and Kenji Shimada[1]

***Abstract*— With the rapid development of powder-based additive manufacturing, depowdering, a process of removing unfused powder that covers printed parts, has become a major bottleneck to further improve its productiveness. Traditional manual depowdering is extremely time-consuming and costly, and some prior automated systems lack adaptability to different parts. To solve these problems, we introduce a robotic system to fully automate depowdering. The key component is a visual perception system, which consists of a pose tracking module that tracks the 6D pose of powder-occluded parts in real-time, and a progress estimation module that estimates the completion percentage. The tracking module can be run efficiently on a laptop CPU at up to 60 FPS. Experiments show that our depowdering system can remove powder efficiently for various types of parts without causing any damage. To the best of our knowledge, this is one of the first vision-based, fully automated depowdering systems for additive manufacturing.**

## I. Introduction

Depowdering, a process of removing unfused powder surrounding 3D-printed parts, is an important post-processing step for powder-based additive manufacturing. For example, printed parts need to be extracted from the build box with the residual powder removed, before being sent to subsequent processes such as heat treatment and surface finishing. Traditionally, depowdering is often performed manually, where the human operators use shovels, vacuums, brushes, and blowers to remove powder incrementally, as shown in Fig. 1 (a). However, this process is tedious and time-consuming, resulting in high operational costs. In addition, the airborne powder raised during depowdering may cause damage to the human respiratory system. Therefore, the automation of depowdering has become an urgent need.

Some researchers developed simple automated systems to tackle the depowering problem. For example, the HP automatic unpacking station [1] and Solukon SFP series [2] are able to perform depowdering for small-to-medium sized parts through a combination of vibration, vacuum, and air blasting. However, these solutions may damage fragile parts, such as delicate polymer parts and unsintered metal parts in binder-jet printing. The process also results in unknown part poses, which raises the difficulty for subsequent part handling. While some researchers directly use a compliant gripper to extract parts from the unfused powder [3], the method may not work well for large parts, since the drag force generated when an object moves through powder increases with the object size [4]. Besides, some fragile printed parts may be unsuitable for gripping.

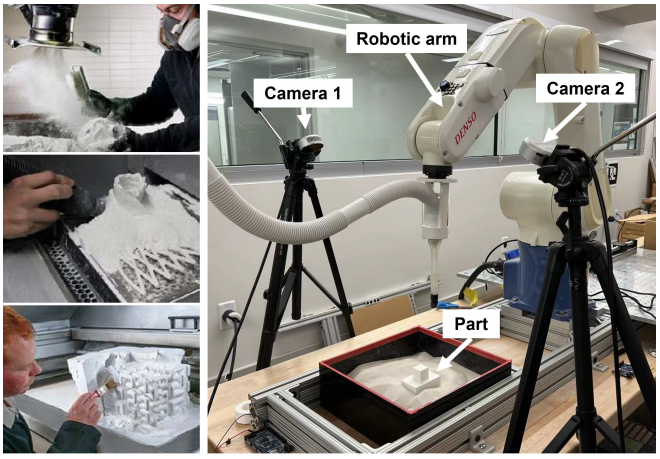


Fig. 1. (a) Examples of manual depowdering. Human operators use blowers, vacuums, and brushes to remove the unfused powder. (b) Overview of our depowdering system. The point cloud sequence from 3D cameras is fed into the visual perception system to track the 6D pose of the powder-occluded part. Based on the estimated pose, a depowdering path is generated for the robot to remove powder through vacuuming and air blasting.

Other semi-automated systems aim to remove the small amount of residual powder that remains on/inside printed parts. The Solukon SFM series [5] rotate the entire part in different orientations to let the powder flow out of the inner structure. However, it requires printed parts to be pre-depowdered. Nguyen et el. [6] used MaskRCNN [7] to localize printed parts and rub them on a brush to remove powder that sticks to the surface. Although it uses vision feedback for part localization, this depowdering solution is not fully automated, because printed parts still need to be pre-depowdered. In addition, it only handles flat shapes and requires a dataset of each specific part for neural network training, which is difficult to obtain for customized parts.

To solve the above problems, we resort to vision-based 6D pose tracking. However, directly applying existing pose tracking methods is challenging due to various reasons. For instance, printed parts appear in a similar color to the surroundings since they consist of the same material. Due to various part shapes and powder occlusion, features such as edges may not be available. In addition, the visible surface of printed parts gradually changes during depowdering, which poses extra challenges. In scenarios shown in Fig. 2, where parts are not anchored to the build box, they might move or lose balance when touched by cleaning tools or when powder support is removed. These challenges indicate that the visual perception system needs to be carefully designed.

[1]Zhenwei Liu, Xikai Dai, Tomasz Swierzewski, and Kenji Shimada are with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA. {zhenweil, xikaid, tswierze, shimada}@andrew.cmu.edu

[2]Junyi Geng is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. junyigen@andrew.cmu.edu

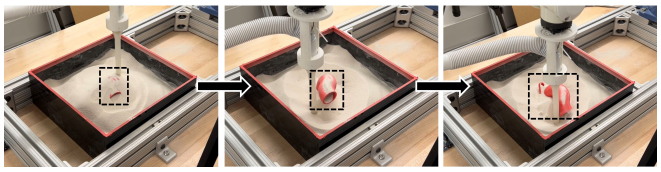

Fig. 2. Example of a printed part losing balance during the depowdering process. As the vacuum constantly removes powder, the part gradually loses powder support and finally falls to one side.

In this paper, We develop a vision-based robotic system that fully automates depowdering for various types of parts, as shown in Fig. 1 (b). The depowdering system consists of two major components: a visual perception system (VPS) and a motion planning system (MPS). This paper mainly focuses on the development of the VPS, which is to track the 6D poses of powder occluded parts in real-time, identify the powder contour around printed parts, and estimate the depowdering progress. To demonstrate the effectiveness of our VPS, we apply a simple motion planner to generate a depowdering path for the robotic arm based on the part pose and the powder contour provided by VPS.

In particular, we design an algorithm that leverages iterative closest point (ICP) [8] and template update [9] to track the part pose. It constantly registers a template to the point cloud scan, with the template derived from the CAD model based on the current visual appearance of the part. In addition, we propose a conditional template update strategy to avoid accumulated template update error. We then extract the powder contour around the printed part and estimate the depowdering progress based on the 6D pose. Due to the nature of template matching, our approach does not rely on feature extraction, thus is robust to point cloud noise and occlusion. Different from other learning-based methods that require a long time of neural network training, our method only requires the CAD model and the initial object pose, which are usually available in additive manufacturing.

To summarize, the main contributions of this paper are:

1) We develop a vision-based robotic system that fully automates depowdering, which is applicable to different part shapes and avoids damaging fragile parts. To the best of our knowledge, this is one of the first vision-based, fully automated depowdering systems.
2) We design an efficient pose tracking algorithm, which combines ICP and conditional template update without the need for a large dataset or neural network training.
3) We present thorough real-time experiments to evaluate the tracking performance on various parts. The tracking pipeline is computationally efficient, which can be run on a laptop CPU at maximum 60 FPS.

## II. Related Work

### A. Automated Depowdering

Traditionally, depowdering tasks are accomplished by human labor. However, the huge demand due to the recent growth of 3D printing technologies raises the need for automated depowdering. As mentioned previously, prior works address automated depowdering through a combination of vibration, vacuum, and air blasting [1] [2], using a compliant gripper to directly extract parts [3], rotating parts in different orientations [5], and rubbing parts on a brush based on visual localization [6]. Recently, Lim et el. [10] introduced a pipeline as a subsequent work of [6] that generates synthetic images of powder-occluded parts, which can be used to train the Mask-RCNN for part localization in [6] and save some effort for dataset preparation. Compared with previous works, our robotic system is applicable to printed parts with different shapes and does not require a large dataset for neural network training.

### B. 6D Pose Estimation

6D pose estimation provides critical information for robots to perform planning and execution. Current methods regarding 6D pose estimation can be categorized as *instance-level* [11]–[16], where the CAD models of known instances are given as priors, or *category-level* [17]–[22], where the CAD model is not available and the target object may be unseen during training. Depending on whether deep learning is applied, 6D pose estimation can also be classified as *geometry-based* [23]–[28] or *learning-based* [11], [13], [19]–[21]. Each category has its own pros and cons. For example, learning-based methods can generalize to different backgrounds, occlusion, and lighting conditions. However, they require a large amount of data for neural network training, which may not be efficient for some applications.

Compared to learning-based methods, geometry-based methods are simple, accurate, and can be quickly implemented and deployed. Iterative closest point (ICP) [8], a well-known geometry-based algorithm for point cloud registration, estimates the 6D pose by aligning the source point cloud to the target point cloud. While several variants of ICP have been proposed to improve the registration performance [29]–[31], they either require feature selection or cannot achieve real-time performance. In this paper, we develop an adaptive algorithm that leverages ICP and template update. The algorithm does not rely on feature extraction, and is robust to sensor noise and occlusion.

## III. System Overview

### A. Hardware Platform

The main components of our hardware platform are:

- a Denso VS-6577 6-axis manipulator, with a cleaning nozzle attached to the end-effector.
- two Realsense L515 cameras, which generate a point cloud sequence of the scanned object.
- a build box with printed parts and powder[1] inside.

### B. System Architecture

The architecture of the depowdering system is shown in Fig. 3. It consists of a visual perception system (VPS), which is the main focus of this paper, and a motion planning system (MPS). The VPS tracks the part pose in real-time, extracts

[1]We use children-play sand in the experiment since it is safe and easy to handle while mimics the property of the actual polymer and metal powder.

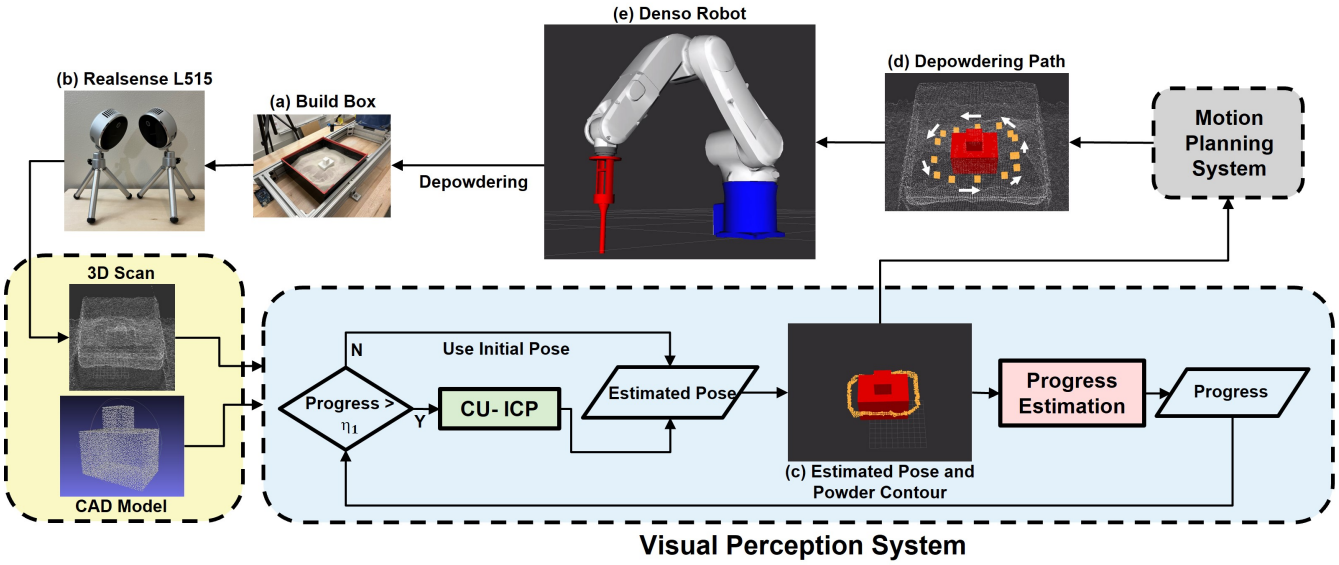


Fig. 3. The architecture of our system. The inputs to the system are the CAD model and the point cloud can. When the depowdering progress is less than $\eta_1$, no tracking is performed, and the initial pose is used. Once the progress exceeds $\eta_1$, CU-ICP starts running. The robot first removes powder through vacuuming, and then finishes up depowdering by air blasting.

the powder contour around printed parts, and estimates the depowdering progress. The progress is defined as the height ratio between the visible portion of the part and the complete part, which will be further explained in Sec. IV-C. Then, based on the estimated pose and the powder contour, as shown in Fig. 3 (c), the MPS generates a depowdering path[2] along the outer contour of the visible part surface, shown in Fig. 3 (d). Overall, the depowdering process can be divided into three phases based on the progress $\eta$:

*Phase 1:* $0 < \eta \leq \eta_1$. No pose tracking is performed because not enough part surface is available for tracking. Printed parts are assumed to stay at the initial pose[3]. The powder contour and the path are generated based on the initial pose. The robotic arm follows the path and removes powder through vacuuming. We empirically select $\eta_1 = 30\%$ based on the observation from the depowdering experiments.

*Phase 2:* $\eta_1 < \eta \leq \eta_2$. Pose tracking starts. The powder distribution and depowdering path are generated based on the estimated part pose. The robot follows the path and removes powder through vacuuming. Note that the generated path automatically adjusts to a new pose when the part moves. We select $\eta_2 = 85\%$ as we observe that printed parts usually appear to be fully uncovered with progress larger than 85%.

*Phase 3:* $\eta_2 < \eta \leq 100\%$. The majority of the printed part has been uncovered. To finish up depowdering, the robot goes over the entire part surface and removes the residual powder through air blasting from the top[4]. After this step, parts can be sent for further post-processing.

[2]The path is collision-free because it is always certain distances away from the outer contour of the visible part surface.

[3]During this stage, the majority of the printed part is still buried under powder. Therefore, the powder will hold the part in place.

[4]Currently the vacuum is manually switched to air blower by reconnecting the hose. The power source switching can be automated in the future work.

## IV. Visual Perception System

The printed parts are completely covered by unfused powder initially, with the powder surface slightly higher than the top of the part. The initial part pose $T_{init}$ can be obtained from the 3D printing system. With the CAD model $P_{cad}$, the initial pose $T_{init}$, and a sequence of point cloud scan $P_i, i = 1, ..., t$ as input, the goals of the VPS are to:

- track the current 6D part pose $T_i$ in real-time.
- identify the powder contour around the part and estimate the depowdering progress based on $T_i$.

### A. Mismatch Dilemma in ICP

With the known initial pose and the CAD model, we formulate the pose tracking problem as a real-time registration problem. Given an initial pose $T_{init}$, the traditional ICP is able to align a source point cloud to a target point cloud by minimizing the $L_2$ error without relying on feature extraction. For our pose tracking task, the source is the template point cloud $P_{tmpl}$ derived from the CAD model, and the target is the current point cloud scan $P_i$. By constantly aligning $P_{tmpl}$ to $P_i$ with the previous pose estimate $T_{i-1}$ as a prior, ICP outputs a relative transformation $T_{i,i-1}$ between the current pose and the previous pose. The new pose $T_i$ can then be derived by $T_i = T_{i,i-1} \times T_{i-1}$.

One major challenge for ICP is the target appearance variation, i.e., printed parts start from fully occluded to fully visible. A common practice to address this is to update the template based on the current object appearance. While many researchers investigated the template update strategies [9], most of them only address the scenario for 2D images. We explore further by extending these strategies from 2D to 3D.

**Strategy 1** A naive way is to use the entire CAD model as a template and does not update the template. However, this could result in a mismatch between the template and the target due to the partially visible appearance, see Fig. 4 (c).

**Strategy 2** One way to account for the appearance variation is to update the template every frame by taking the overlap between the CAD model and the current point cloud scan, shown in Fig. 4 (d). Nonetheless, this strategy still may not achieve ideal performance, as the template quickly becomes erroneous, as shown in Fig. 4 (e) and Fig. 5. This is because the sensor noise causes the instability of alignment and finally leads to template mismatch.

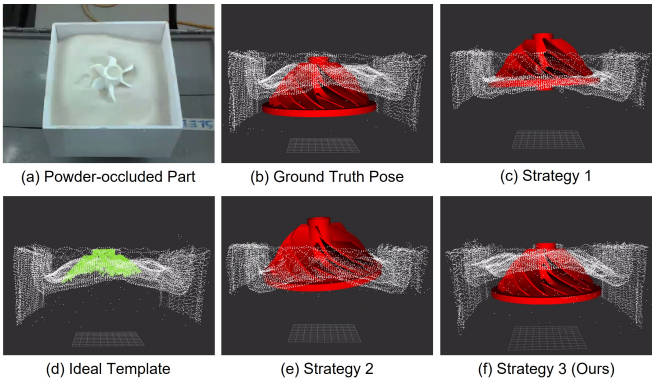


Fig. 4. A comparison of different template update strategies. (a) A propeller partially covered by powder. (b) The CAD model rendered according to the ground truth pose. (c) Pose tracking result with strategy 1. (d) Template taken from the overlapping area between the CAD model and the scan. (e),(f) Pose tracking results with Strategy 2 and our strategy. With Strategies 1 and 2, the bottom of the propeller is mistakenly aligned to the powder surface, resulting in incorrect estimated pose. Our strategy resolves this issue.

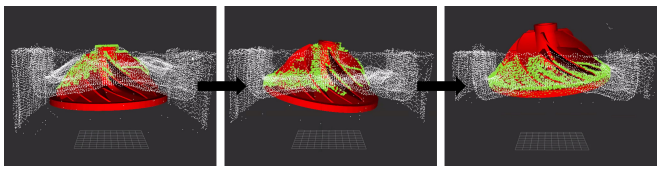

Fig. 5. Visualization of template error accumulation with Strategy 2. The initial template aligns well with the point cloud (left). As tracking starts, although the target remains stationary, the sensor noise causes the oscillation of the estimated pose. The oscillation results in the template error accumulation, leading to an incorrect template and pose estimate.

### B. Conditional Update ICP

We propose a conditional template update strategy for ICP to avoid accumulated template update errors. Specifically, the template is updated only when the part appearance has changed significantly. In depowdering, this happens either when (1) more part surface becomes visible as more powder is cleaned or (2) printed parts have moved either by the depowdering tool or due to the loss of balance. Either scenario causes the powder distribution to change.

**Strategy 3** The template is updated only when the orientation $R$, translation $t$, or depowdering progress $\eta$ change by a significant amount:

$$\begin{aligned} &\text{If } |R_i - R_{last}| > \delta_1 \text{ or } |t_i - t_{last}| > \delta_2 \text{ or } |\eta_i - \eta_{last}| > \delta_3, \\ &\text{then } P^i_{tmpl} = (P_{cad} \cdot T_i) \cap P_i \\ &\text{else } P^i_{tmpl} = P^{i-1}_{tmpl}, \end{aligned} \tag{1}$$

where $R_{last}$, $t_{last}$, and $\eta_{last}$ are the orientation, translation, and progress at the last time when the template was updated, $P^i_{tmpl}$ is the current template, and $P_{cad} \cdot T_i$ is the CAD model transformed to the current pose. $\delta_1$, $\delta_2$, and $\delta_3$ are user-defined thresholds that control the template update frequency. Since this rule avoids unnecessary template update, the shape of the template is preserved throughout depowdering.

With this new template update rule, we develop our *Conditional Update ICP* (CU-ICP) algorithm shown in Alg. 1. The template is initialized (Lines 4-6) by transforming the CAD model to the initial part pose and running Alg. 2. Then, ICP constantly registers the template to the current scan, updates the current pose, and transforms the CAD model to the new pose (Lines 8-10). Based on the new pose, the depowdering progress is calculated, which will be discussed further in Sec. IV-C. If the condition is satisfied, the template will be updated (Lines 12-16). Specifically, for each point $p_{cad}$ in the CAD model, Alg. 2 finds its nearest neighbor $p_{nb}$ from the scan (Line 3). If the distance is within a threshold, $p_{cad}$ is considered to be matched and is added to the updated template (Lines 5-6).

**Algorithm 1** Conditional Update ICP

1: **while** *true* **do**
2: $P_i \leftarrow$ **GetCurrentScan**;
3: **if** *first iteration* **then**
4: $T_i \leftarrow T_{init}$;
5: $P_{cad} \leftarrow$ **TransformPointCloud**$(P_{cad}, T_{init})$;
6: $P_{tmpl} \leftarrow$ **TemplateUpdate**$(P_{cad}, P_i)$;
7: **end if**
8: $T_{i,i-1} \leftarrow$ **ICP**$(P_{tmpl}, P_i)$;
9: $T_i \leftarrow T_{i,i-1} \times T_i$;
10: $P_{cad} \leftarrow$ **TransformPointCloud**$(P_{cad}, T_{i,i-1})$;
11: $\eta_i \leftarrow$ **ProgressEstimation**$(P_{cad}, P_i)$;
12: **if** $R_i - R_{last} > \delta_1 \parallel t_i - t_{last} > \delta_2 \parallel \eta_i - \eta_{last} > \delta_3$ **then**
13: $P_{tmpl} \leftarrow$ **TemplateUpdate**$(P_{cad}, P_i)$;
14: $R_{last} \leftarrow R_i$;
15: $t_{last} \leftarrow t_i$;
16: $\eta_{last} \leftarrow \eta_i$;
17: **end if**
18: **end while**

**Algorithm 2** Template Update

1: $P_{tmpl} \leftarrow \emptyset$
2: **for** each $p_{cad}$ in $P_{cad}$ **do**
3: $p_{nb} \leftarrow$ **FindNearestNeighbor**$(p_{cad}, P_i)$;
4: $d \leftarrow$ **CalcEuclideanDist**$(p_{cad}, p_{nb})$;
5: **if** $d < \xi$ **then**
6: *Add* $p_{cad}$ to $P_{tmpl}$;
7: **end if**
8: **end for**
9: **return** $P_{tmpl}$;

### C. Progress Estimation

Progress estimation monitors the percentage of completion for depowdering, with 0% meaning parts are completely

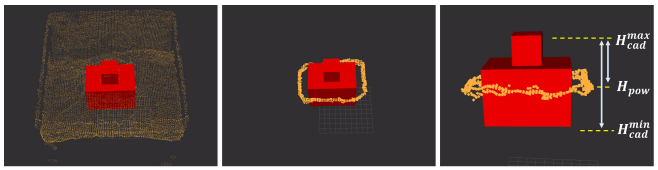


Fig. 6. Powder contour extraction and progress estimation. (a) Powder (orange) segmented from the scan. (b) Powder contour generated by selecting points surrounding the part contour. (c) Progress estimation based on the height ratio of the visible surface and the whole part.

covered, and 100% meaning depowdering is completed. It is also an important factor for conditional template update, as described in Sec. IV-B. Given an entire CAD model and an updated template that represents the visible portion, an intuitive way is to calculate the ratio between their surface area or volume. However, these two metrics may not accurately reflect the depowdering progress due to the potential non-linear changing pattern for parts whose cross section area varies significantly in the vertical direction. In fact, the progress is directly related to the powder surface height, because the unfused powder is removed layer by layer with a constant thickness drop through vacuuming. Therefore, we select the height ratio between the template and the CAD model to reflect the progress.

To estimate the depowdering progress based on height ratio, the first step is to identify the powder contour around the visible part surface. As shown in Fig. 6 (a), based on the estimated pose, the point cloud scan can be segmented into part and powder. The powder contour is then extracted by selecting powder points that are certain distances away from the part segment[5], as shown in Fig. 6 (b). By calculating the average height of the powder contour $H_{pow}$, the depowdering progress $\eta$ can be calculated by:

$$\eta = \frac{H_{cad}^{max} - H_{pow}}{H_{cad}^{max} - H_{cad}^{min}}, \tag{2}$$

where $H_{cad}^{max}$ is the height of the CAD model's top surface and $H_{cad}^{min}$ is the height of the CAD model's bottom surface. If $H_{pow}$ is higher than $H_{cad}^{max}$, $H_{pow}$ is truncated to $H_{cad}^{max}$.

## V. Experiments

In this section, we first introduce the experimental setup. Then, we perform several experiments to validate the tracking performance of our CU-ICP. Finally, we present several complete demonstrations of the overall depowdering processes achieved by the robotic system.

### A. Experimental Setup

**Evaluation Objects**. We evaluate our system on five 3D printed parts with different shapes: cube, cup, propeller, owl, and pipe, shown in Fig. 7. The sizes of these objects, defined as the diagonal length of the object's minimum 3D bounding box, vary from 8 centimeters to 15 centimeters. We cover

[5]In scenarios where the printed part is fully covered, the segmentation can still be performed, as the points in the scan that are close to the top of the part will be grouped as the part segment. Therefore, the powder contour can still be generated based on the segmentation result at this stage.

Fig. 7. Parts selected for evaluation. From left to right are: cube, cup, propeller, owl, and pipe. Although these parts are printed in different colors, the color information is not used.

printed parts with various amounts of powder to generate a range of surface visibility. For convenience, we define visibility the same as depowdering progress, see Eq. (2).

**Evaluation Metrics**. For stationary objects, we report the following metrics for pose tracking: 1) $\mathbf{R}_{\text{err}}$: mean rotation error in degrees, and 2) $\mathbf{t}_{\text{err}}$: mean translation error in centimeters. For moving objects, we employ the success rate and maximum trackable speed.

**Baseline Comparison**. We compare our algorithm with two baselines: (1) Vanilla ICP, where the entire CAD model is used as a template and no template update is performed; (2) Continuous ICP, where the template is updated in every frame based on method specified by Alg. 2.

**Key Parameters**. We empirically select $\delta_1 = 30$deg, $\delta_2 = 5$cm, and $\delta_3 = 15\%$, as we notice that an overly large threshold leads to an outdated template caused by the less frequent template update, while an overly small threshold ends with very frequent update that results in error accumulation as explained in Sec. IV-A. In addition, we select $\eta_1 = 30\%$ and $\eta_2 = 85\%$ for the depowdering progress thresholds, as mentioned in Sec. III-B.

### B. Pose Tracking for Static Objects

In a real depowdering environment, printed parts are partially visible. The robot end-effector introduces extra interference and occlusion while moving around parts. To test the pose tracking accuracy under this condition, we manually move a nozzle around stationary parts, with the visibility ranging from 20% to 100%, as shown in Fig. 8.

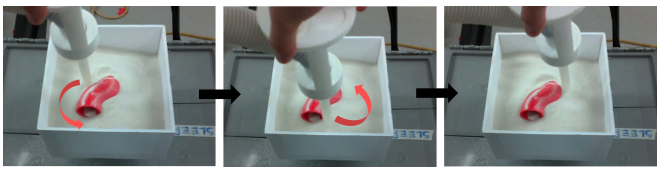

Fig. 8. Experimental setup for tracking static objects at different visibility. We manually move the nozzle around parts to simulate the actual depowdering environment with point cloud occlusion and interference. The red arrow shows the nozzle moving direction.

As shown in Table I, our algorithm achieves significant better performance compared to Continuous ICP and vanilla ICP, especially when the surface visibility is less than or equal to 60%. Thanks to the conditional template update, the template formulation is robust to additional point cloud occlusion and interference. In fact, in this static object scenario where no significant changes occur on either $\eta$ or $T_i$, CU-ICP does not perform unnecessary template update and therefore is able to maintain a stable template.

TABLE I

POSE TRACKING ERROR WITH DIFFERENT LEVELS OF SURFACE VISIBILITY

| | | $\mathbf{R_{err}}$(degree) | | | | | $\mathbf{t_{err}}$(cm) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Surface Visibility | 20% | 40% | 60% | 80% | 100% | 20% | 40% | 60% | 80% | 100% |
| CU-ICP (ours) | Cube | 16.8 | **1.69** | **0.73** | **2.87** | **4.29** | **1.05** | **0.15** | **0.19** | 0.28 | 0.35 |
| | Cup | **3.85** | **4.05** | **3.76** | **3.06** | **5.03** | **0.94** | **1.07** | **0.68** | **0.67** | **0.60** |
| | Propeller | **6.60** | **2.62** | **2.12** | **2.26** | **2.45** | **0.83** | **0.39** | **0.36** | **0.15** | **0.19** |
| | Owl | **47.9** | **1.55** | 12.4 | **16.4** | **5.85** | **2.20** | **1.55** | **0.66** | **0.80** | **0.40** |
| | Pipe | **19.6** | **8.53** | **6.74** | **5.35** | **3.93** | **1.17** | **0.30** | **0.38** | **0.32** | **0.34** |
| | Overall | **19.0** | **3.69** | **5.15** | **5.99** | **4.31** | **1.24** | **0.69** | **0.45** | **0.44** | **0.38** |
| Continuous ICP | Cube | 16.8 | 3.76 | 1.80 | 3.77 | 4.46 | 1.38 | 0.22 | 0.24 | **0.27** | 0.35 |
| | Cup | 50.3 | 36.1 | 34.3 | 13.4 | 5.66 | 6.54 | 2.08 | 1.13 | 0.85 | 0.79 |
| | Propeller | 11.1 | 12.7 | 4.04 | 12.7 | 10.8 | 4.14 | 2.88 | 1.70 | 0.70 | 0.49 |
| | Owl | 74.0 | 39.1 | **12.3** | 26.7 | 7.01 | 4.79 | 2.66 | 0.79 | 1.31 | 0.42 |
| | Pipe | 55.5 | 29.7 | 18.5 | 6.79 | 4.16 | 2.32 | 1.25 | 1.38 | 0.60 | 0.38 |
| | Overall | 41.5 | 24.3 | 14.2 | 12.7 | 6.42 | 3.83 | 1.82 | 1.05 | 0.75 | 0.49 |
| Vanilla ICP | Cube | 18.1 | 3.52 | 2.43 | 3.49 | 4.49 | 1.41 | 0.23 | 0.27 | 0.30 | 0.35 |
| | Cup | 42.0 | 36.3 | 36.0 | 13.3 | 5.80 | 2.94 | 2.04 | 1.15 | 0.91 | 0.80 |
| | Propeller | 10.4 | 7.61 | 4.47 | 16.9 | 16.2 | 4.02 | 2.85 | 1.78 | 0.62 | 0.44 |
| | Owl | 75.1 | 69.9 | 27.0 | 28.9 | 6.75 | 4.91 | 4.64 | 1.69 | 1.40 | 0.41 |
| | Pipe | 57.2 | 30.0 | 18.9 | 6.86 | 4.25 | 2.38 | 1.30 | 1.14 | 0.61 | 0.38 |
| | Overall | 40.6 | 29.5 | 17.8 | 13.9 | 7.50 | 3.13 | 2.21 | 1.21 | 0.77 | 0.48 |

[1] Surface visibility here is defined the same as depowdering progress.
[2] $R_{err}$ and $t_{err}$ represent rotation error and translation error, respectively. The lower the better.

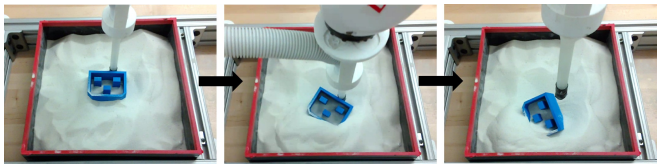

Fig. 9. Experimental setup for tracking moving objects, with the initial surface visibility ranging from around 40% to 60%. We use the robotic arm to push the part until it reaches the box edge, and test if the part pose can be tracked during the motion.

TABLE II

SUCCESS RATE (%) OF TRACKING MOVING OBJECTS IN POWDER

| Method | Cube | Cup | Propeller | Owl | Pipe | Overall |
|---|---|---|---|---|---|---|
| **CU-ICP (ours)** | 100 | **80** | **100** | 100 | **80** | **92** |
| **Continuous ICP** | 100 | 0 | 40 | 100 | 20 | 52 |
| **Vanilla ICP** | 60 | 0 | 0 | 100 | 0 | 32 |

### C. Pose Tracking for Moving Objects

When a printed part moves through powder, the powder distribution changes as the powder piles up on one side and spreads out on the other side. To examine the tracking performance for moving objects, we use the robotic arm to push printed parts at different initial contact points, till they reach the edge of the build box. The resulting motion is a combination of translation and rotation. With the initial pose given, we run pose tracking while parts are moving. For each selected part in Fig. 7, we randomly sample five initial contact points. Hence, 25 trials are performed in total. A trial is defined as successful if the final orientation error is within 15deg and the final translation error is within 15mm.

As shown in Table II, the overall success rate of CU-ICP is significantly higher than other baselines. Note that both Continuous ICP and Vanilla ICP have low success rate on propeller, cup, and pipe, while CU-ICP shows better performance among all different parts. The reason is that the inner structures of the cup and propeller cannot be scanned by the camera, while the CAD models still contain the inner structures. Without the conditional template update, the inconsistency between the CAD model and the scan easily leads to a pose estimation error and template mismatch. For the pipe, the curved pointcloud cannot be exactly reconstructed due to the camera limitation, which again challenges the baseline approaches to handle error accumulation. The fact that our CU-ICP is less affected by these drawbacks demonstrates its robustness to point cloud occlusion as well as distortion.

### D. Maximum Trackable Speed

We test the maximum part moving speed at which the tracking algorithms lose track of the selected parts. We use a linear actuator and a turntable to generate adjustable translation and rotation[6], as shown in Fig. 10 (a).

As shown in Fig. 10 (b), it is obvious that CU-ICP results in higher maximum trackable speed compared with Continuous ICP and Vanilla ICP with visibility of less than

[6]Due to the hardware limit, the maximum moving speed that the linear actuator and turntable can generate is limited. Therefore, in order to achieve arbitrary moving speed, we playback the recorded point cloud sequence at faster frame rates instead of physically moving parts. Then we uniformly downsample the sequence to the normal frame rate.

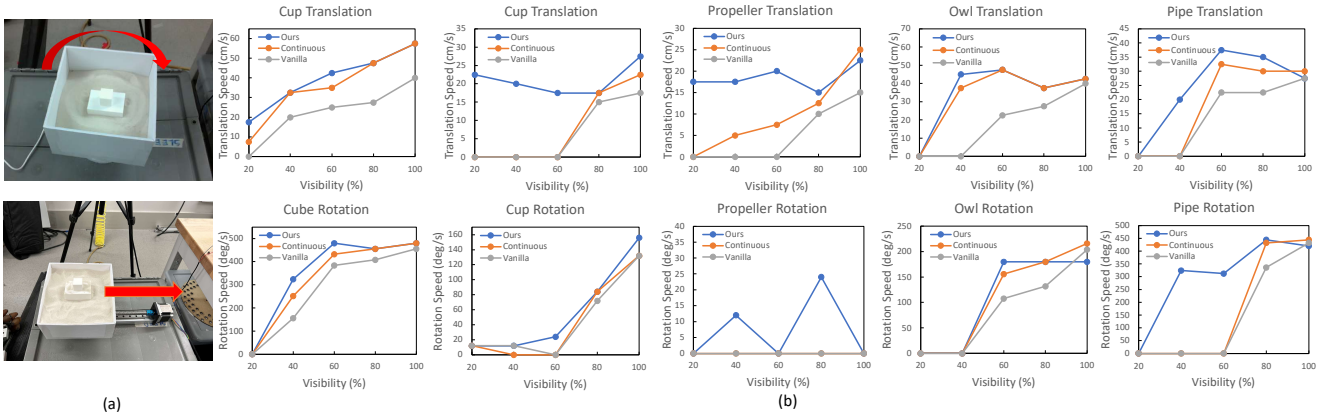


Fig. 10. (a) Experimental setup for maximum trackable moving speed test. We use a linear actuator and a turntable to generate translation and rotation for printed parts. (b) Maximum trackable moving speed in translation and rotation for different printed parts with various surface visibility.

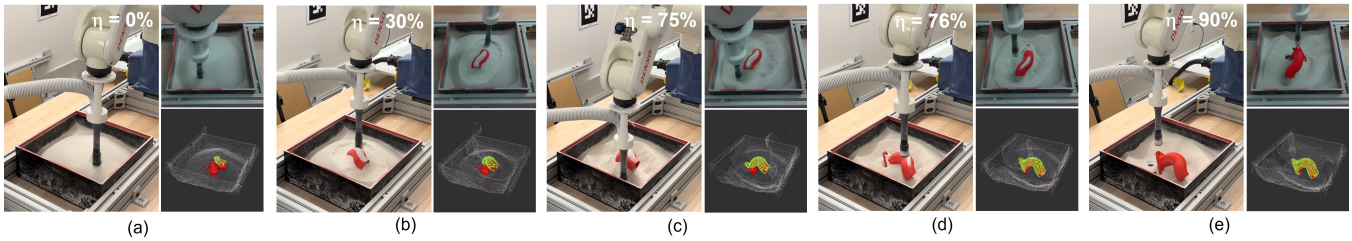


Fig. 11. A complete depowdering process for the pipe. The lower right picture of each snapshot shows the estimated pose in red and the corresponding template in green. (a) Progress is 0%. The pipe is entirely covered by powder. (b) Progress reaches 30%. CU-ICP starts tracking the part pose. (c) The part becomes more visible, and the robot continues to remove powder along the outer contour of the pipe. (d) The pipe loses balance due to the loss of powder support. The robot adjusts to the new pose and avoids collision. (e) The robot removes the residual powder through air blasting.

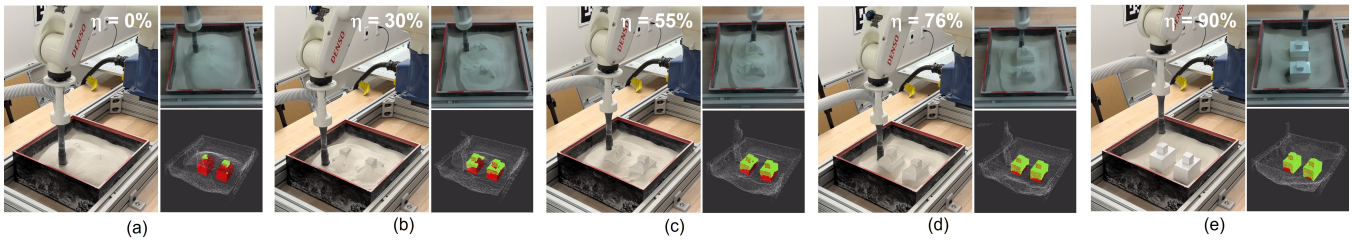


Fig. 12. A complete depowdering process for two cubes. We run two tracking processes in parallel to improve tracking efficiency. The robot first performs depowdering on the left cube for one cycle, and then moves on to the right cube for another cycle. (a) Progress is 0%. The cubes are entirely covered by powder. (b) Progress reaches 30% and two cubes starts being tracked in parallel. (c)-(d) CU-ICP keeps tracking poses and two cubes gradually become more visible. (e) The residual powder is removed through air blasting.

80%. As the visibility increases, the maximum trackable speeds for the three algorithms gradually become close to each other. This is because the updated template is more similar to the original CAD model. Also, note that the maximum trackable speed for CU-ICP and Continuous ICP may not strictly increase along with the visibility, such as the cup translation with visibility from 20% to 80% and the pipe translation with visibility from 40% to 100%. This indicates that increasing target visibility does not necessarily result in a higher trackable speed. In fact, as the target visibility increases, the number of points in the template to be processed also increases, which slows down the computation and reduces the tracking capability of the algorithms. Overall, CU-ICP can be run on a laptop CPU with the maximum processing rate of around 60fps.

### *E. Depowdering with the Robotic System*

Finally, we demonstrate several complete depowdering processes achieved by our robotic system. Two depowdering scenarios are considered here: (1) single-part depowdering, and (2) multiple-part depowdering, as shown in Fig. 11 and Fig. 12. In the former scenario, the printed part loses balance during the process and falls to one side. The robot is able to adjust to the new pose and successfully avoids collision. In the latter scenario, the pose tracking for different parts is run in parallel to improve the computation efficiency. The robot successfully removes powder for all parts. The two demonstrations prove that our system fully automates depowdering without the concern of damaging fragile parts.

## VI. Conclusions And Future Directions

This paper proposes a robotic system that fully automates depowdering using visual feedback. The main component is a visual perception system that takes in the scanned point cloud from the depth camera, tracks the 6D pose of powder occluded parts, and identifies the powder contour for progress estimation and path generation. Experiments show that the pose tracking module is robust to occlusion and sensor noise. In particular, our tracking algorithm, named Conditional Update ICP (CU-ICP), achieves higher tracking performance compared to other baselines. The depowdering system is able to fully automate depowdering without the concern of damaging fragile parts. Future work includes investigating more advanced path planning strategies to further improve the depowdering efficiency for various types of parts.

## ACKNOWLEDGMENT

We are grateful to Dr. Chen Wang at the Robotics Institute, Carnegie Mellon Univeristy, for his advice and support to the research. This work was funded by Carnegie Mellon University Manufacturing Futures Initiative and NASA University Leadership Initiative.

## References

[1] M. Guthrie. (2020) Hp and am solutions present the innovative hp jet fusion 5200 series 3d automatic unpacking station. Accessed: 2022-02-08. [Online]. Available: https://press.hp.com/us/en/press-releases/2020/hp-am-solutions-present-new-automatic-unpacking-solution.html

[2] Unpacking systems for plastic parts by solukon. Accessed: 2022-02-08. [Online]. Available: https://www.solukon.de/en/depowdering-plastic-systems/

[3] J. Cormack, M. Fotouhi, G. Adams, and T. Pipe, "Automated extraction of 3d printed parts from unfused pa12 powder using a one-shot 3d printed compliant gripper*," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8655–8662, 2021.

[4] I. Albert, J. G. Sample, A. J. Morss, S. Rajagopalan, A. L. Barabasi, and P. Schiffer, "Granular drag on a discrete object: shape effects on jamming." *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 64 6 Pt 1, p. 061303, 2001.

[5] Solukon: Automated depowdering of laser melted metal parts. Accessed: 2022-02-08. [Online]. Available: https://www.solukon.de/en/depowdering-metal/

[6] H. Nguyen, N. Adrian, J. L. Xin Yan, J. M. Salfity, W. Allen, and Q.-C. Pham, "Development of a robotic system for automated decaking of 3d-printed parts," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 8202–8208.

[7] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988.

[8] P. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[9] L. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 26, no. 6, pp. 810–815, 2004.

[10] J. X.-Y. Lim and Q.-C. Pham, "Automated post-processing of 3d-printed parts: artificial powdering for deep classification and localisation," *Virtual and Physical Prototyping*, vol. 16, pp. 333 – 346, 2021.

[11] Y. Labbe, J. Carpentier, M. Aubry, and J. Sivic, "Cosypose: Consistent multi-view multi-object 6d pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2020.

[12] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking," *IEEE Trans. Robot.*, vol. 37, no. 5, pp. 1328–1342, 2021.

[13] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," *ArXiv*, vol. abs/1711.00199, 2018.

[14] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2019.

[15] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "DeepIM: deep iterative matching for 6D pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018.

[16] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, "Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1530–1538.

[17] X. Deng, J. Geng, T. Bretl, Y. Xiang, and D. Fox, "icaps: Iterative category-level object pose and shape estimation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1784–1791, 2022.

[18] B. Wen and K. Bekris, "Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models," in *IEEE Int. Conf. Intell. Robots Syst.*, 2021, pp. 8067–8074.

[19] C. Wang, R. Martín-Martín, D. Xu, J. Lv, C. Lu, L. Fei-Fei, S. Savarese, and Y. Zhu, "6-pack: Category-level 6d pose tracker with anchor-based keypoints," in *IEEE Int. Conf. Robot. Autom.*, 2020, pp. 10 059–10 066.

[20] W. Chen, X. Jia, H. J. Chang, J. Duan, L. Shen, and A. Leonardis, "Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1581–1590.

[21] D. Chen, J. Li, Z. Wang, and K. Xu, "Learning canonical shape space for category-level 6d object pose and size estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 970–11 979.

[22] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized object coordinate space for category-level 6d object pose and size estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2637–2646.

[23] A. Collet, D. Berenson, S. S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *IEEE Int. Conf. Robot. Autom.*, 2009, pp. 48–55.

[24] C. Choi and H. I. Christensen, "3d pose estimation of daily objects using an rgb-d camera," in *IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 3342–3349.

[25] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3d sensor," in *IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1724–1731.

[26] C. Choi and H. I. Christensen, "3d textureless object detection and tracking: An edge-based approach," in *IEEE Int. Conf. Intell. Robots Syst.*, 2012, pp. 3877–3884.

[27] T. Drummond and R. Cipolla, "Real-time visual tracking of complex structures," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 24, no. 7, pp. 932–946, 2002.

[28] K. Pauwels, L. Rubio, J. Díaz, and E. Ros, "Real-time model-based rigid object pose estimation and tracking combining dense and sparse visual cues," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2347–2354.

[29] A. W. Fitzgibbon, "Robust registration of 2d and 3d point sets," in *BMVC*, 2001.

[30] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2241–2254, 2016.

[31] G. Sharp, S. Lee, and D. Wehe, "Icp registration using invariant features," *IEEE Trans. Pattern. Anal. Mach. Intell.*, vol. 24, no. 1, pp. 90–102, 2002.