Analyzing Videos of 3D Printing from a Depth Camera

Sophia Cao[1]
Sreepathi Pai, Ph.D.[2], and Rongcui Dong, Ph.D.[2]
[1] University of Michigan; [2] University of Rochester

Abstract

3D printing offers a rapid and cost-effective method for manufacturing parts. However, print failures occur often due to the volatile nature of melting plastic. To address this, we devised a method for acquiring measurements of printed objects for quality assurance. By comparing these measurements with those from the intended print model, failed prints can be detected and halted. My method utilizes an Intel RealSense depth camera to record videos and produce layer-by-layer reconstructions of printed parts, from which measurements can be extracted. Our findings demonstrate the feasibility of this process and reveal that extracted X and Y measurements are accurate. The successful integration of depth cameras for measurement acquisition opens up promising avenues for future work.

Keywords— 3D printing, quality assurance, depth camera, 3D reconstruction

1 Introduction

In recent years, 3D printing has emerged as a transformative technology that is used in various industries, ranging from manufacturing and aerospace to healthcare and consumer goods. 3D printing, also known as additive manufacturing, is the construction of physical parts from three-dimensional models. The process begins with the creation of a digital 3D model using computer-aided design (CAD) software, which produces a stereo-lithography (STL) file. Next, this STL file is run through a slicer, where the object is divided into 2D layers and translated into G-code instructions. The resulting G-code file is then provided to the 3D printer to print.

While there are many types of 3D printing, this project utilizes Fused Deposition Modeling (FDM). FDM is one of the most popular and accessible 3D printing methods, making it an ideal choice to study (Carolo, 2023). In the FDM process, a thermoplastic filament, such as Acrylonitrile Butadiene Styrene (ABS), is fed into the printer's extrusion nozzle. The material is then heated to its melting point and deposited in layers onto the bed platform. As the deposited material cools and solidifies, it fuses with previous layers, building a three-dimensional object. A Voron Switchwire printer and ABS were used in this project.

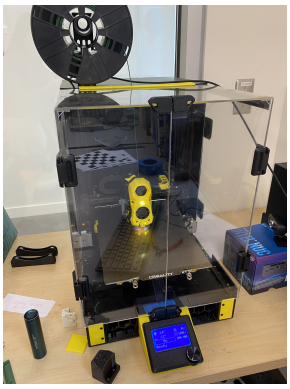

Figure 1: Voron Switchwire printer

1.1 Motivation

Failed prints occur frequently, with failure rates as as high as 20% (Pearce, 2022). This wastes both time and material. Some examples of problems commonly encountered are stringing, blobs/zits, and warping. Stringing occurs when the print head leaves thin strands of filament between different parts of the object. Blobs and zits are imperfections left on the surface of the print. Warping poses a significant problem, as it causes the printed object to deform or detach from the build plate due to uneven cooling and shrinkage.

Since prints are often lengthy, ranging from

several hours to even days for large-scale projects, print failures can be frustrating and time-consuming. Therefore, incorporating real-time quality assurance would allow users to stop prints upon detection of defects, saving time and material.

1.2 Intel RealSense D405

The depth camera used in this project is an Intel RealSense D405, a short-range stereo camera for close-range computer vision and defect detection. It utilizes structured light to gather depth information. It operates at an optimal range of 7 cm to 50 cm, with object detection as low as 0.1 mm at 7 cm (dep, nd). For this reason, we aim to keep the camera 7 cm-8 cm away from the object in question. The camera is used to record depth and color videos simultaneously. It outputs a .bag file, which can be replayed in the Intel RealSense Viewer or converted to .png images and .csv files using rs-convert.

2 Data Collection & Techniques

At a high level, the data collection and analysis pipeline involves video recording, depth acquisition, layer extraction, object reconstruction, and measurement comparison. In this section, the process is described in further detail.

2.1 Data Collection

To capture depth data of a print, the first step is to mount a depth camera onto the 3D printer.

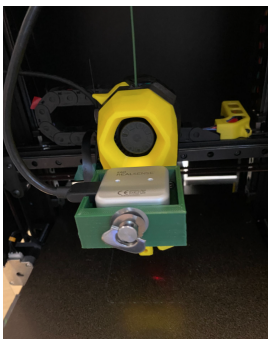

Figure 2: Depth camera mounted to the printer

Before starting a print, a custom printerdata python script was executed. This script collects the position and timestamps of the print head as it moves and outputs it as a .csv file. Next, a print was started. Simultaneously, a stopwatch was started. This was to keep track of the time elapsed between when the print starts and when the video starts for subsequent frame extraction (see section 2.3). After the bed and extruder heated up, a video recording was started and the time on the stopwatch was written down.

Since the camera was mounted in front of the extruder, the object being printed was not automatically within the frame. To address this, a Gcode macro was integrated during the slicing of the STL file, adjusting the print head's position after each layer. This ensured that after each layer was printed, the frame was adequately captured by the depth camera.

After the completion of the print, two data outputs were obtained. The first was a .bag file generated by the depth camera, containing the recorded video with depth information. The second output was a CSV file produced by the printerdata script, encompassing timestamps and the corresponding print head's positions.

2.2 Storage

During the data collection process, the use of the depth camera presented storage challenges. The Intel RealSense depth camera takes approximately 2MB per frame, making longer videos at higher frame rates slower to work with. For example, a 30-minute video recorded at 15 FPS takes about 54 GB. To address this, I recorded at 5 FPS and started the recording after the printer had reached its target temperature (the camera remained static and non-relevant information was omitted). As a result, a video capturing a 12-layer print took about 10 minutes to record and consumed approximately 6 GB of storage.

2.3 Post-processing

For processing the outputted data, the first step was to identify the timestamps when the camera paused during the print. Utilizing the CSV file generated by the printerdata script, a Python script named get_timestamps was implemented. This script identified instances when the print head moved to specific coordinates (x = 115 and y = 180), indicating that the camera was paused to capture the layer. A tolerance of $5e - 14$ was employed to account for minor discrepancies.

Utilizing the identified timestamps, the next step was to extract the relevant frames from the .bag file recorded by the depth camera. This was done by using the rs-convert tool with additional start and end times parameters.

```
\rs-convert.exe
-i <PATH.bag>
```

```
Gets .bag file
-p <PATH/Name of Files>
Returns color and depth PNG
-v <PATH/Name of Files>
Returns CSV of depth info
-s <Start Time>
Start time in seconds
-e <End Time in seconds>
End time in seconds
```

To calculate the start and end times, let t be the timestamp extracted from the CSV and d be the time recorded on the stopwatch. Then, $t_{start} = t - d - 0.2$ and $t_{end} = t - d$. This ensured accurate synchronization between the depth camera recording and the 3D printing process.

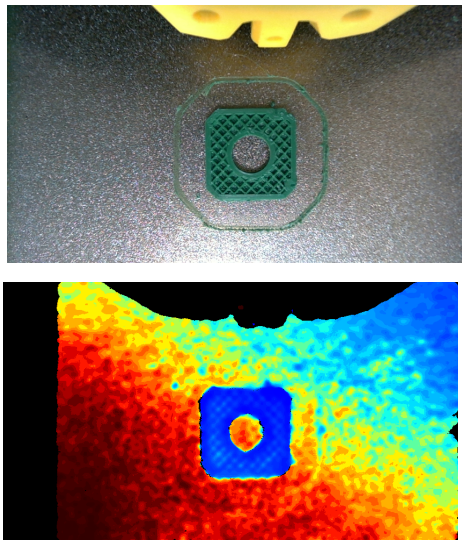

Figure 3: Depth and color PNGs extracted from layer 12 of a cube print

With the extracted frames, the next stage focused on converting the 2D frames into 3D XYZ coordinates, also called "deprojection". This conversion was accomplished by applying the Brown-Conrady distortion model[1] and the camera intrinsics, which were acquired using the rs-sensor-control tool. The intrinsics for the stream profile used in this project's experiments are as follows:

```
Video Stream: Z16 848x480@ 5Hz
Principal Point         : 430.037, 240.698
Focal Length            : 429.084, 429.084
Distortion Model        : Brown Conrady
Distortion Coefficients : [0,0,0,0,0]
```

After deprojection, the topmost layer of each frame was extracted. This was achieved by selecting the points that fall within a specified z range (in meters), with experimental values chosen based on visual inspection of the background and topmost layer. The z_{min} used was 0.0765. For layers below 10, $z_{max} = 0.0777 + ((layer\_number - 1) * 0.0001)$, while for subsequent layers, $z_{max} = 0.0785$.

Subsequently, each layer was consolidated into a single CSV file of points. This concatenated file was then plotted as a point cloud representation of the printed part.

Finally, measurements were acquired from the point cloud using either the Euclidean distance formula or the point picking tool in CloudCompare. Using a caliper, real-life measurements of the part were recorded and compared with the point cloud measurements.

3 Results

By following the process outlined above, 3D reconstructions of prints were successfully generated. The data collected and processed came from the first 12 layers of a Voron Design Cube with 30% infill.

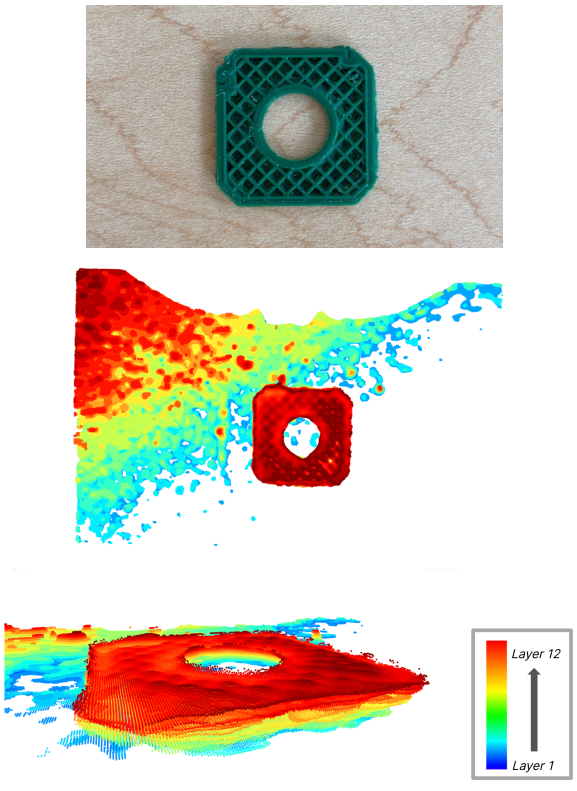

Figure 4: 3D reconstruction of 12-layer Voron Design Cube, 30% infill

During the layer extraction process, we observed imperfections on the print, which were clearly evident in the generated point clouds. Specifically, layer 5 of the print displayed notice-

[1]More about deprojection in Intel RealSense can be found at https://dev.intelrealsense.com/docs/projection-in-intel-realsense-sdk-20.

able blobbing on the bottom left, which was reflected in the reconstruction.

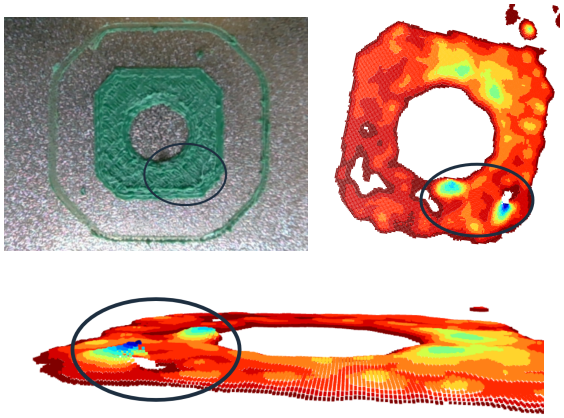

Figure 5: Imperfection in layer 5 visualized in point cloud

Similarly, in layer 6, there was observed blobbing on the top right, also captured in the reconstruction.

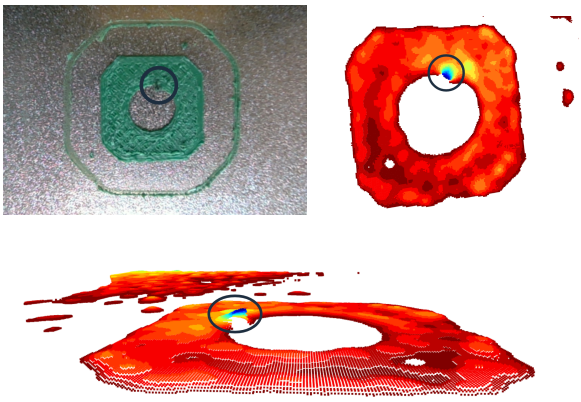

Figure 6: Imperfection in layer 6 visualized in point cloud

To evaluate the accuracy of the reconstruction, the point-picking tool in Cloud Compare was used to extract X and Y measurements from the point cloud. These measurements were then compared with caliper measurements obtained from the real-life object.

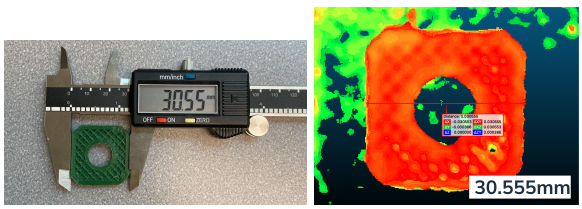

Figure 7: X measurement comparison

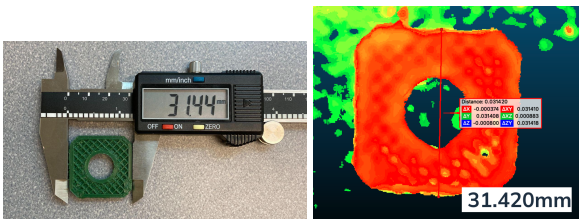

Figure 8: Y measurement comparison

3.1 Infill

During analysis of the reconstruction, a notable observation was that the infill was not clearly visible in the point cloud. To gain a better understanding of this issue, we decided to conduct a follow-up experiment by printing the first 12 layers of a Voron Design Cube with a reduced infill of 10%. In this reconstruction, the infill was far more apparent and distinguishable.

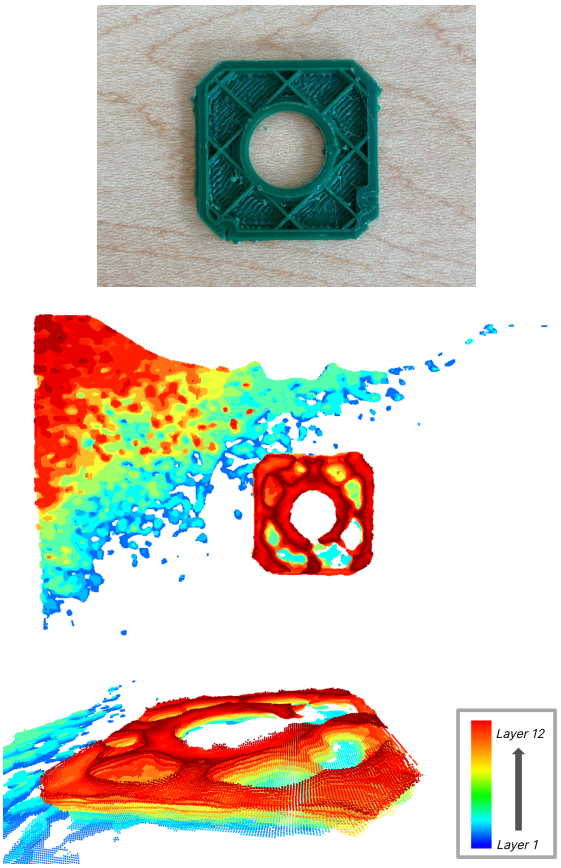

Figure 9: 3D reconstruction of 12-layer Voron Design Cube, 10% infill

However, this raised the question of whether the infill measurements obtained from the camera were accurate, and if there was a potential issue with the camera rather than the reconstruction pipeline. To address this, we devised a method to determine the camera's infill accuracy. Examining the 12th (and last) layer, we selected pixels from

the topmost layer and pixels within the infill holes, then calculated the depth difference between them. This provided us with the camera's infill measurement, which we then compared to the real-life infill measurement.

The actual infill measurement of the part, measured with a caliper, was 1.2 mm. In the bottom left, the infill measurement from the camera was $79.4 - 79.1 = 0.3mm$, while in the bottom right, the infill measurement was $79.5 - 78.3 = 1.2mm$.

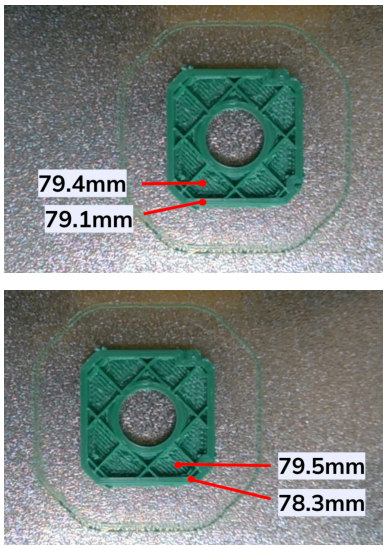

Figure 10: Varying infill measurements from camera, 10% infill

These results indicate that the camera's infill measurements were inconsistent and inaccurate. This finding prompted us to address the limitations of our approach and identify future investigation, as will be discussed in the following section.

4 Conclusion & Future Work

The results of this project demonstrate the feasibility of utilizing a depth camera for generating 3D reconstructions of printed parts. Moreover, the X and Y measurements extracted from the point cloud have shown to be accurate. However, the discrepancy in infill measurements indicate that future investigation of the Z measurements is required.

Several potential factors may contribute to the discrepancy in Z measurements. One possible factor is inconsistent lighting, as a visible glare was seen on the print bed during the printing process. Additionally, surface reflectivity of printed parts may lead to the scattering of infrared light. Since the camera relies on structured light, these factors might complicate its depth determination process. A higher infill percentage could also pose challenges in distinguishing between the surface and the infill, and variations in bed tilt might result in different areas of the part being at varying distances from the camera.

Other future work includes reducing background noise in the reconstructions. This could involve exploring the experimental extraction range to optimize the data captured during layer extraction. Looking ahead, converting post-processing into a real-time pipeline would enable measurements to be acquired during a print. By doing so, users would be able to make immediate adjustments and optimize printing outcomes.

In conclusion, continued research and development are needed to improve the accuracy of depth measurements and to streamline post-processing techniques. Current findings, however, offer promising insights into using depth cameras for 3D reconstructions of printed parts.

Acknowledgments

I would like to acknowledge my supervisor, Sreepathi Pai, my project partner, Jesus Diaz, and my graduate student mentor, Rongcui Dong. I would also like to acknowledge the Kearns Center at the University of Rochester, the Goergen Institute for Data Science, and the NSF for making this research project possible.

References

(n.d.). Depth camera d405 – intel® realsense™ depth and tracking cameras. Intel RealSense.

Carolo, L. (2023). What is fdm 3d printing? – simply explained. All3DP.

Pearce, J. M. (2022). Average failure rate for 3d printing. ResearchGate.