Real Time Error Quantification of 3D Prints

Adrian Jackson | Rutgers New Brunswick

Introduction

3D printing (i.e., Fused Filament Fabrication or FFF), has huge potential for in-field manufacturing in domains such as space travel, disaster zones, remote locations, and hospital clean areas [2]. However, to ensure efficient and non-defective parts, FFF needs better error detection and correction. I sought to eliminate certain types of defects in using a computer vision system coupled with a reinforcement learning model at the Advanced Manufacturing Sciences Laboratory under Dr. Rajiv Malhotra.

FFF builds parts by extruding thin lines of heated plastic filament into layers. These layers are built on top of and bond to each other, creating a 3D part with the intended geometry over time. However, the extruder can only deposit filament in a certain shape which resembles a beveled rectangle from a top-down view, leaving small air gaps in layers. It also can erroneously under- or over- extruding material due to filament impurities, mechanical inconsistencies, and exogenous disturbances (especially in extreme environments). Miniscule extrusion errors cause significant losses in mechanical strength, which contributes significantly to why FFF struggles to compete with traditional manufacturing methods. There are commercial products that correct these errors on a layer-by-layer scale, but lead to global part integrity losses (i.e., an entire defective layer). Other systems require erroneous samples from a specific part to collect data on how to solve these errors. This entails purposely creating defects in production so that they can be studied, which is antithetical to efficient manufacturing.

To combat this, I designed a part-agnostic error quantification system for extrusion errors and an associated feedback system, as well as constructing the appropriate hardware. Using the quantified error and feedback system, the printer can then take an appropriate action based on the output of a reinforcement learning model - all in real time.

Methods

The algorithm needed to take under a second to identify and propagate its error to the LabVIEW controller and be consistent across various materials and defect types. To facilitate this, I reviewed computer vision algorithms, learned the math behind them, looked into what applications they had been used in previously, and then decided on the best approach to take. Specifically, I read papers that had previously explored error correction with other control systems, like PID control [1]. Additionally, I read through the documentation for many of python's opencv algorithms such as canny, sobel, and Hough transform. I chose to use python open-cv canny edge detection, which is a relatively robust edge detection algorithm that works purely mathematically. This meant it could be executed quickly and parallelized easily, which was essential for making sure the algorithm could operate in real time with minimal delay. Specifically, canny works with multiple stages: blurring the image to reduce noise and false positive edges, gradient calculations to detect the intensity of the image across its dimensions, sharpening edges, and using a threshold to detect edges by areas of high intensity.

Canny proved to be challenging - the image was dim, making edges hard to pick out for the algorithm. After adjusting certain parameters in canny, I quickly realized that my goal was not possible with the current set of data. I broadened my lens and looked for other solutions: adding another external light source and cropping the camera output so we only focused on the most recent segment of printed filament. I additionally positioned an extra light source orthogonal to the road direction, as this would enhance contrast, making road boundaries and defects more clear.

I also tuned canny by changing a variety of parameters. These include kernel size, which affects the resolution of the filter; kernel height, which affects the shape direction of edges detected; and the preprocessing filters used, which included more gaussian blur and brightness control to assist canny.

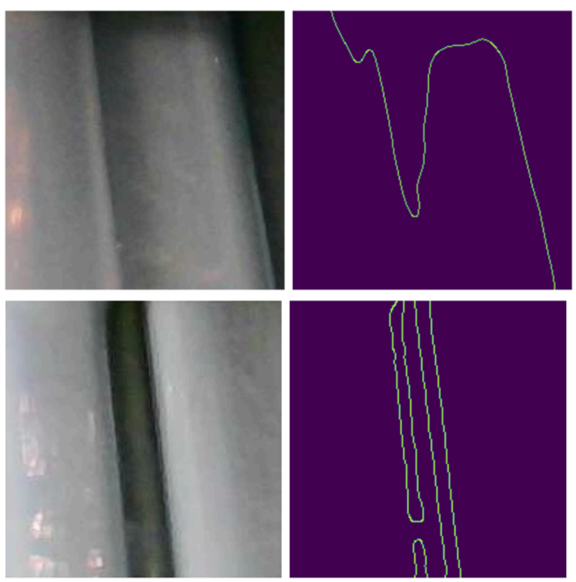

Figure 1: An overextension (top) and underextrusion (bottom), after being processed by the algorithm. These outputs are consistent when cropped and lit the same way.

With these edges detected consistently and clearly, the algorithm could then measure the defect appropriately. In the overextrusion case, because the new, overextruded road flowed onto and overlapped the previous road, the width of the previous road could be measured and the ratio of its size to the ideal road size taken as an error metric. This road size is dependent on printer hardware; namely, it is proportional to nozzle width. In the underextrusion case, the gap between roads was measured and used as an error metric.

Upon initial testing, it became clear that though consistent and accurate, my edge detection algorithm was very brittle. Environmental changes such as the room lighting or camera zoom and position would cause incorrect quantifier outputs, which led to nonsensical control actions and failure to correct defects.

I proposed that instead of simply detecting edges by their brightness, I could also segment the images by color and therefore find edges. Since the build plate of the printer is blue and the filament was a natural tone, there was a stark difference in their RGB values (particularly the B value), meaning this approach of color segmentation was viable. An additional strategy of measuring the angle of the road being printed allowed us to normalize with respect to it. We then measured defect size orthogonally to it, leading to more accurate readings.

Once again, I searched for relevant literature, choosing to approach the problem by converting from RGB to a HSV (Hue Saturation Value) space and using masks to select only the roads and denoise the image. Canny edge detection was then applied again and the relevant distance for the defect measured. This resulted in a much higher tolerance to disturbance. We no longer need to crop the image to only view the segment of material that was just extruded, because the entire road of

Real Time Error Quantification of 3D Prints

Adrian Jackson | Rutgers New Brunswick

material can be extracted using color segmentation. The system was also able to ignore rotations of the camera because of the angle normalizer described above, and was less sensitive to changes in lighting.

Figure 2: A dual-road PLA sample of overextrusion correction

Results

This new algorithm worked even better: it was as fast, able to work on a wide range of materials and disturbances, and detected defects more consistently and accurately.This has allowed our system to correct defects such as those in Figure 2 to a non-error state in a single control action - typically taking a duration of around 5 seconds.

An example of an overextrusion defect and correction can be seen in figure 2. The extruder begins at the bottom right, and extrudes the first road. It then wraps around at the left edge, and starts overextruding (the result of a previously defined exogenous disturbance). This is picked up by the camera, sent to a python script, processed, and the quantified error sent to a reinforcement learning algorithm. This algorithm returns a control action via a change in extruder flow rate or printer stage speed, correcting the defect.

Likewise, figure 3 shows an extreme underextrusion which is also corrected in a single control action.

This is a first-of-its-kind achievement: correcting errors on a line-by-line scope in 3D printing.

Figure 3: A dual-road ASA sample of extreme void correction

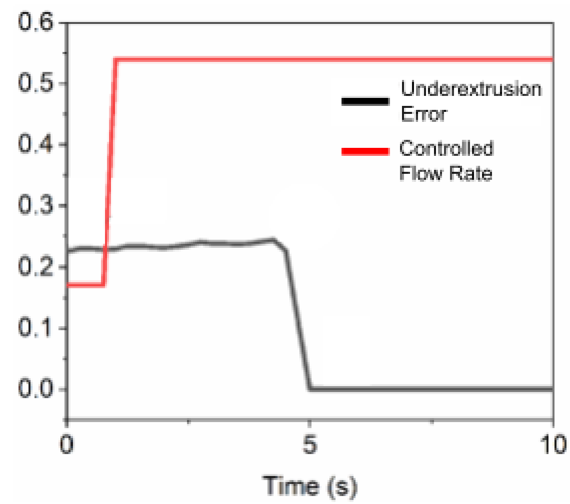

Figure 4: Example of void correction over time (commanded F refers to flow rate change)

Future Direction

Future direction for this work includes correcting for more types of defects, introducing direction-agnostic error quantification, and correcting for a wider range of and more extreme exogenous disturbances. Work is currently ongoing to correct stringing errors, as well as periodic extrusion disturbances.

References:

1. Liu, Chenang, et al. "Image analysis-based closed loop quality control for additive manufacturing with fused filament fabrication." Journal of Manufacturing Systems 51 (2019): 75-86.
2. Mangrolia, B, Cleeman, J, Patel, A, Jackson, A, & Malhotra, R. "Real-Time Recovery From Cyberattacks on Manufacturing Processes." Proceedings of the 2024 ISFA. 2024 International Symposium on Flexible Automation. Seattle, Washington, USA. July 21–24, 2024. V001T01A001. ASME. https://doi.org/10.1115/ISFA2024-139964