A REAL-TIME MONITORING FRAMEWORK FOR COOPERATIVE 3D PRINTING

Cole Mensch, Anuj Swaminathan, Zhenghui Sha*

Walker Department of Mechanical Engineering, University of Texas, Austin, TX 78712

Abstract

Cooperative 3D printing (C3DP) is a primitive form of swarm manufacturing (SM) that utilizes multiple 3D printing robots to build large parts with the goal of overcoming the scalability issues present in traditional 3D printing. Our prior work has established the C3DP framework, with steps such as geometric partitioning, scheduling, placement, and path planning. However, C3DP cannot achieve full autonomy unless its capabilities include live monitoring of the manufacturing environment and feedback-loop control and error correction. In this paper, we present a real-time process monitoring framework for C3DP, capable of detecting defects that occur in FDM 3D printing and providing feedback to the printers in a closed-loop system. We tested various computer vision techniques for this system, including nozzle tracking, object classification, and image similarity comparisons. Nozzle tracking failed to produce reliable data, but the other two methods were integrated into a holistic monitoring system for C3DP. We then demonstrated the effectiveness of this system by monitoring multiple test prints in real-time.

Keywords: Computer Vision, Cooperative 3D Printing, Process Monitoring

Introduction

Cooperative 3D printing (C3DP) is an emerging technology designed to overcome the scalability issues present in traditional gantry 3D printing systems by employing a swarm of mobile 3D printing robots working cooperatively to decrease the build times of large parts [1]. The key steps in this process include chunking [2-3], placement [4], scheduling [5,6], and path planning [7], all of which have been established in our prior work. The cooperation integral to the system is enabled by chunking or geometrically partitioning large objects into smaller sections that are printable by individual printers. This chunking process is broken up into two types: Z-chunking and XY-chunking. Once a print is divided into chunks, they are placed on the factory floor and a schedule is generated to ensure the printers complete them in a time-optimal manner, with path planning being utilized to avoid collisions between printers during movement. This process can be visualized in the diagram shown in Figure 1.

The C3DP platform is operated via a control hub, which is currently able to send commands and receive information from the robots, but such information is limited as it operates under an open-loop system, as shown in Figure 1. Printers can send data such as nozzle temperature, expected nozzle position, and print status, while the control hub can send G-code commands to the printers. While this theoretically enables full control of the printers, none can be automatically verified externally. As such, the real nozzle position could vary from the expected position due to poor calibration, or worse yet, a collision and a print could fail with no way for the printer to know on its own. To solve this problem, we have developed a process monitoring framework for the

* Corresponding Author: zsha@austin.utexas.edu

C3DP system, which can detect print failures and communicate directly with the control hub to stop failed prints, ensuring that they won't cause a system-wide failure of the platform.

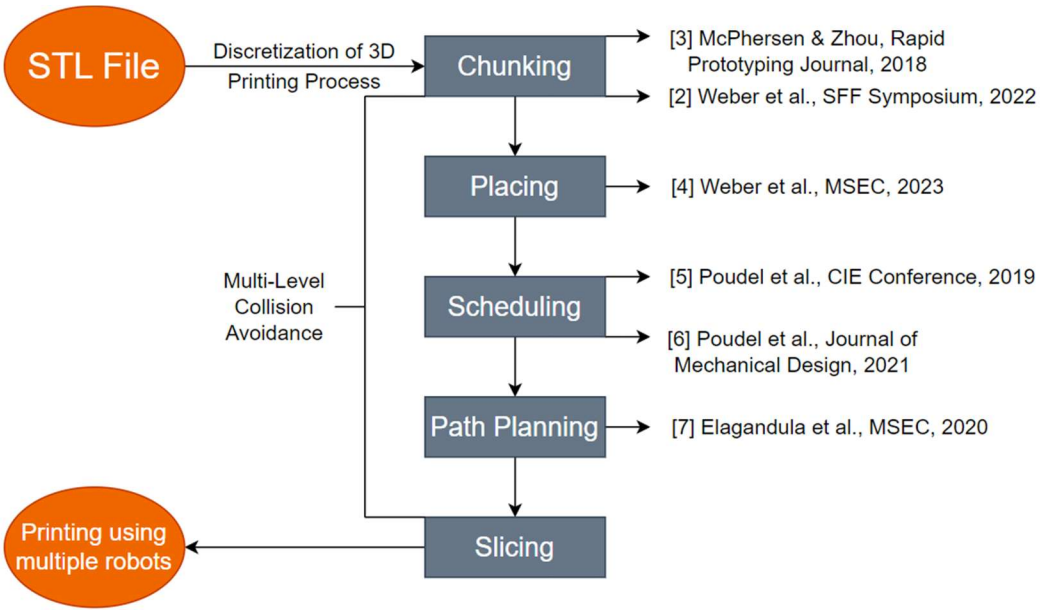


Figure 1: Flow chart of the C3DP Process [4].

Process monitoring for additive manufacturing is currently a growing field of research because one of the significant drawbacks of the process is the inconsistency of parts produced. This has been a heavily discussed area for metal AM specifically, where parts for use in applications such as aerospace must meet strict quality standards that are difficult to verify with additively manufactured parts. As such, thorough process monitoring and control techniques have been researched to guarantee that parts manufactured through this method have minimal defects and meet these high-quality standards [9]. Industrial providers have even begun developing and selling solutions for this issue, such as EOS Smart Monitoring, which is capable of monitoring the build and correcting parameters such as laser power to ensure the final part meets the standards set during pre-processing [10].

This level of quality assurance is typically unnecessary for FDM printing because the parts are typically not used in such demanding environments. However, the process has a larger failure rate than most other manufacturing processes, at around 20% for unskilled users [11]. While the C3DP robots have tested printing parameters specific to the platform and filament used, there are still opportunities for failure. We have noted many of these in our own testing, with extrusion issues, poor bed calibration, warping, and stringing being common problems encountered during test prints. An example of some of these failures is shown in Figure 2, which features some common defects such as layer splitting and interfacial gaps in a cooperative test print of a castle due to poor extrusion and bed calibration. These are common enough issues with FDM printing and typically are not a major problem since the material cost and print times are kept low. However, if FDM printing and its advanced derivatives, such as C3DP, are going to be incorporated into an automated factory setting, there needs to be a system in place to ensure that prints are completed accurately and with no major errors to ensure timely & cost-efficient manufacturing.

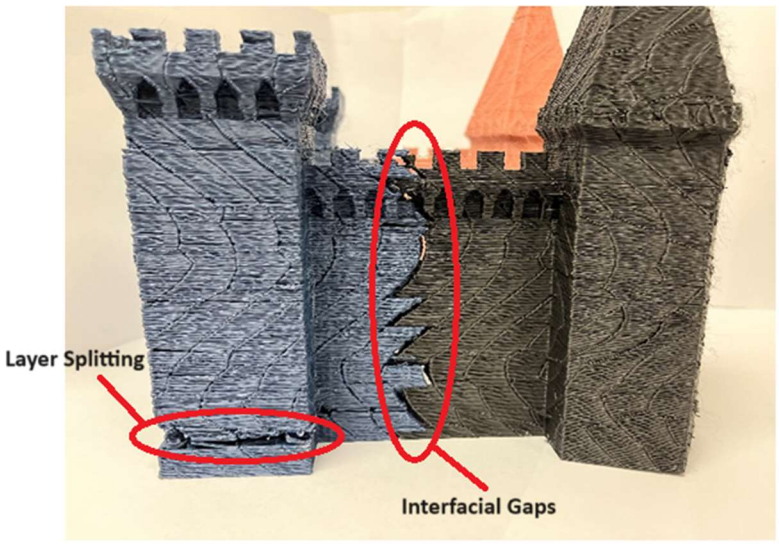


Figure 2: 3-Robot cooperative print of castle demonstrating common FDM printing errors.

In-situ process monitoring of FDM prints is another area that has received much research attention and can provide an effective solution to the above FDM failures [12]. Many methods have been explored for this purpose, from 2D and 3D vision techniques to temperature, vibration, and acoustics monitoring [13]. However, due to the relatively low cost of FDM 3D printing compared to other AM techniques, there is a desire to keep the cost of monitoring low so as to not outpace the cost of the printer itself. As such, most of these methods utilize cheap equipment with computationally efficient algorithms that can be run effectively on most standard computers.

2D vision is the most widely explored method for process monitoring of FDM 3D printing. Many different techniques have been explored in this field to detect errors that are commonly seen. For example, deep learning techniques have been used in tandem with a camera focused on the extruder to both detect and correct extrusion errors in real-time, with a success rate of over 98% and a response rate faster than that of a human [14, 15]. Externally mounted cameras can also be used to view the printed object as it is being built and compare the image to the desired print geometry from the STL file [16]. This can also be used to detect warping, and machine learning models have been developed to not only recognize warping when it occurs but also correct it at its onset to ensure it does not cause catastrophic failure [17]. Similar 2D imaging techniques can be done on a layer-wise level by extracting the geometry of the most recently completed layer from the printed part and comparing it to the same layer in the g-code [18]. However, such a process cannot be done in real-time, as it requires the nozzle to move out of the way of the camera to capture an image of the completed layer. Stringing is another common defect that can be indicative of failures in FDM printed parts. A small amount of stringing is a common occurrence in 3D printing, especially when printing thin walls or beams that are spaced apart from one another, but heavy stringing tends to indicate print failure as it occurs when the filament is deposited but does not properly adhere to the part. Fortunately, machine learning methods have been developed to detect stringing in real-time using simple 2D images of the part, although the algorithm must be properly trained if it is to be generalized to a variety of 3D printing environments [19]. While this covers many of the existing techniques for monitoring 3D prints, there is little in the way of a

holistic framework to tie them all together. This paper hopes to bridge that gap by combining some of these techniques into a complete process monitoring system for C3DP.

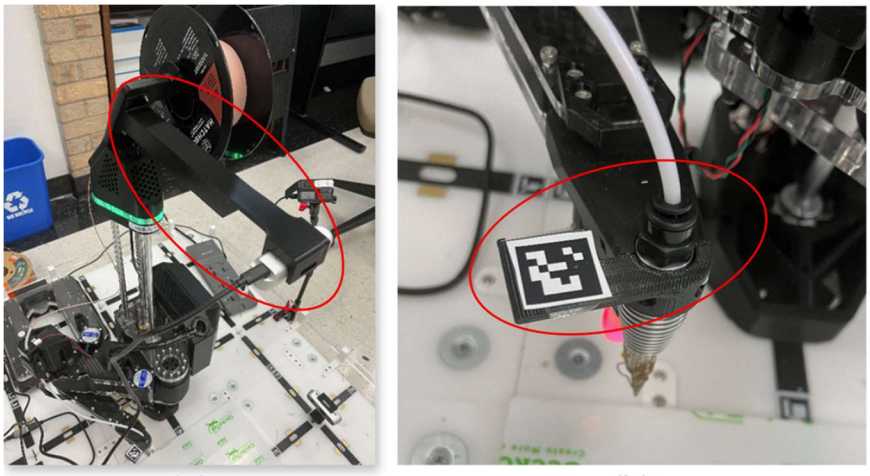
The remaining sections are organized as follows: First, we will discuss the existing 2D vision techniques relevant to the C3DP platform, specifically focused on nozzle tracking, warping, and stringing, and how they can be integrated into a holistic monitoring system for a single print. Next, we present preliminary test cases for each of these vision methods and discuss their effectiveness based on the results. These results form the basis for an integrated C3DP monitoring system, which is demonstrated using three different test cases. Lastly, we present and discuss the results of these test cases before concluding on the effectiveness of the monitoring system. In the Conclusion section, we also present avenues for future work that could improve the existing framework that has been developed.

Methodology

One of the primary benefits of the 2D computer vision techniques is their ease of use and low cost. As such, any standard RGB camera can be used for this purpose. Our testing was conducted using an Intel Realsense D435 camera, shown in Figure 3. This camera provides high RGB image quality with a frame resolution of 1920x1080, a sensor resolution of 2MP, and a maximum framerate of 30 FP [20], and the additional depth camera feature is conducive to future research in which depth data can be integrated into future 2D image processing techniques.

Figure 3: Intel Realsense D435 depth camera [20].

The first 2D vision method we analyzed was object tracking. By determining the position of the nozzle at each point in time, we can ensure that it follows the correct path set by the g-code and can correct for issues such as motor slippage and external collisions. One such object-tracking method uses machine learning methods such as adaptive correlation filters to recognize and track the nozzle in real-time [21]. While this could be an effective strategy, a simpler alternative exists of simply attaching an ArUco marker to the nozzle mount and using an overhead camera to determine the location of this marker using a marker detection script [22]. Using such a marker, as long as the camera position and distance between the marker and nozzle are known, the nozzle position can be easily extracted from the marker position. To this end, a custom camera mount was developed that attaches to the filament spool of the printer, which was chosen because this position is unaffected by the location of the printer and the nozzle. A custom marker base was also developed to attach to the nozzle mount, providing a consistent marker position relative to the nozzle. Both mounts are shown in Figure 4.



(a) (b)

Figure 4: Mounts for use in object tracking. (a) Overhead camera mount, (b) ArUco marker nozzle mount.

The next 2D vision method we investigated was the detection of stringing. Obico is an open-source, network-enabled 3D print monitoring tool that allows users to monitor and control their 3D printers remotely using a mobile app, and it uses a similar AI-driven stringing detection algorithm to predict part failure and warn users it occurs [23]. For our stringing detection method, we utilized the Ultralytics YOLOv8 model, which is capable of object detection, classification, and segmentation [24]. We then trained this model using a stringing dataset from Roboflow, which features images of hundreds of examples of stringing in 3D prints [25]. The training with this dataset took place over 40 iterations to ensure that it would be able to consistently detect stringing from images taken in various different 3D printing scenarios. Due to the use of a trained object detection model to recognize stringing, the camera position for this monitoring method is mostly arbitrary. The only requirements are that the camera must be able to see the print without obstruction and the background must provide good contrast against the filament being used. As such, a side profile of the print was desired to ensure that the print could be monitored in real-time without being occluded by the nozzle while also not confusing the model by having stringing present in front of the filament with the same color, both of which could occur in an overhead image. The mount shown in Figure 5 was developed for this purpose. It can be slotted into the mounting holes on the C3DP factory floor, allowing a side profile of the print to be captured.

Figure 5: Side profile camera mount.

The final method of interest was the 2D image matching of the printed part. This process is done by generating a mask of the part at a certain standardized camera position and comparing this to the rendered mask produced from the STL model at the same orientation [26]. In the image, this is done by filtering out any color hues that do not match the specific filament color we are using. This ensures that only the part is seen in the final image mask, while the shading controls the brightness of the pixels that are filtered to visualize depth features. The two masks can then be compared using the structural similarity index method, generating a similarity score. This tells the user how well the two masks overlap (a similarity score of 1 is perfect) and, therefore, how accurately the print reflects the ground truth model. Because this technique requires masks of both the physical print and 3D model in the same pose, a standardized camera position must be determined. The camera mount generated for stringing detection in Figure 5 is ideal for this application, as it mounts directly to the factory floor and, as such, ensures that the camera always sits at a standardized distance and orientation relative to the build plate. The mask rendered from the STL file can then use this information to ensure the pose generated perfectly matches what the camera will see in the physical system.

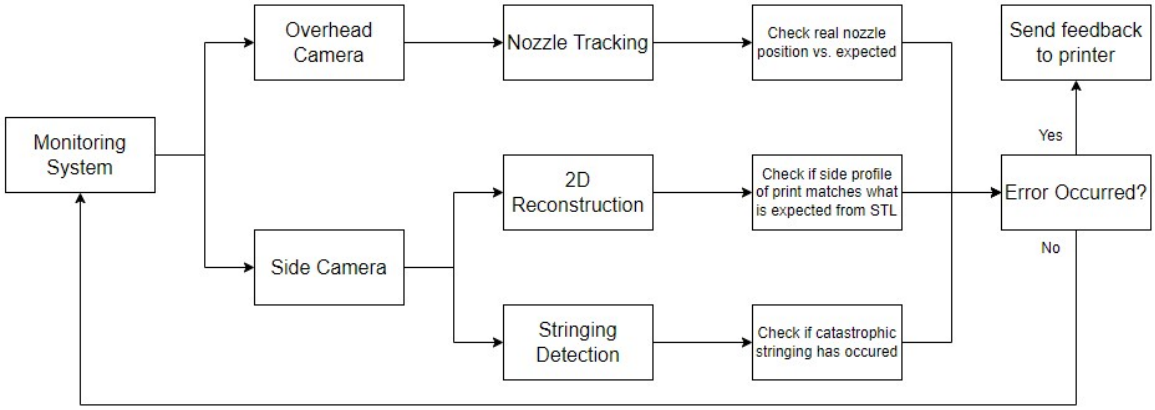


Figure 6: A flow chart of the process monitoring framework.

These three methods of print monitoring can be integrated into a single holistic monitoring framework to detect any possible failure in a print. Object tracking will ensure that the nozzle is always in the right position, stringing detection will ensure that there is good layer adhesion and filament retraction when printers pause, and the 2D image comparison will check for major flaws in the print, such as layer shifting and warping. A flow chart of the proposed monitoring framework is shown in Figure 6.

Preliminary Testing

With the testing methodology established, the next step is to verify that each of the proposed process monitoring techniques works effectively. After the techniques are individually validated for the C3DP platform, they are integrated into a holistic monitoring framework outlined in Figure 6.

The first method we tested was the nozzle tracking using a mounted ArUco marker. The initial results for this study show that the overhead camera can detect the ArUco marker effectively, with detection being measured in over 99% of frames during typical printer movement. This was tested by running a standard square test print and counting the total number of frames captured by

the camera and how many of those frames the computer could detect an ArUco marker in. The overhead camera feed with the marker tracking overlay can be seen in Figure 7.

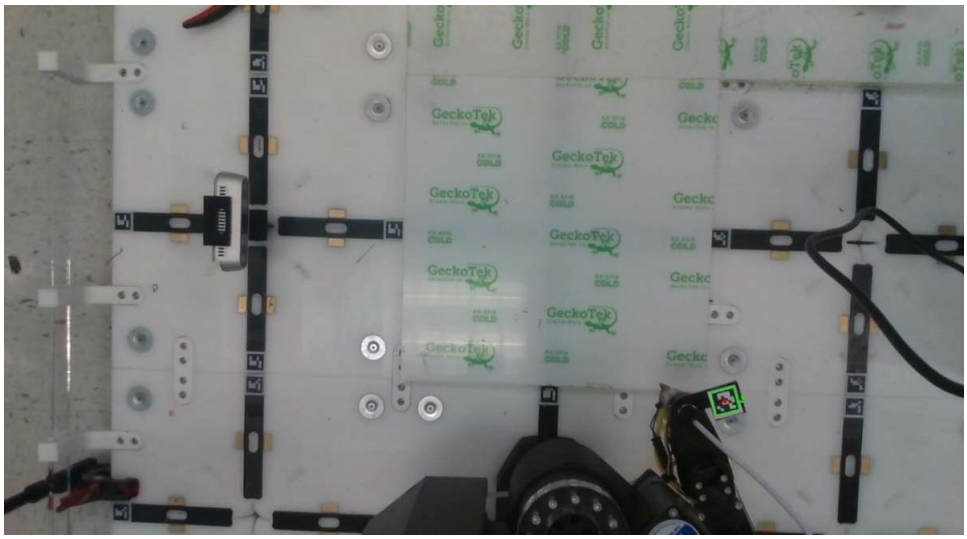


Figure 7: Overhead camera feed during nozzle tracking.

While the marker can be detected, we have noted a significant error in the comparison between the nozzle position from the marker and the ground truth position from the printer. This was tested in three ways: direct marker tracking using the camera position as a reference, using a single reference marker to compare the nozzle position, and finally, a 3D trilateration using markers on all four corners of the build plate. The results from testing each of these methods are shown in Table 1. The table shows that all three methods have significant average errors, making it difficult to get reliable tracking data. While the standard deviations of the measurements from direct tracking, and to a lesser extent, single reference marker comparison, are quite low, the measurements themselves are not accurate enough to justify pursuing this method further. We also looked at the results if the Z-axis is ignored since it is less accurate due to being derived from the marker size, and while it improved slightly for every method, the average error is still far too high. Every method gives an average error greater than 10 mm, with some going as high as 45 mm, and submillimeter errors are required to accurately calibrate a printer and track its position. Even after recalibrating the camera and re-measuring the positions of the camera and nozzle, the results did not improve, and as such, this method is excluded from the final process monitoring system.

Table 1: Nozzle Tracking Results

Method	**Direct Tracking**	**Single Reference**	**3D Trilateration**
Average Error	21.68 mm	14.23 mm	45.16 mm
Average Error (X, Y)	20.77 mm	11.37 mm	30.06 mm
Standard Deviation	1.30 mm	2.39 mm	13.53 mm
Standard Deviation (X, Y)	0.89 mm	1.97 mm	7.02 mm

The stringing algorithm was then evaluated using several test parts that were printed with the goal of demonstrating stringing. This was done in a few different settings to ensure the process was effective regardless of environmental variables such as the lighting, background, and filament color. Examples of these test cases are shown in Figure 8. While this test was not performed on a real-time print, the initial results show that the algorithm is highly sensitive to the previously mentioned variables. For example, the algorithm can detect stringing in darker parts with a light background at a very high level of confidence, while lighter-colored filaments such as orange and yellow do not provide enough contrast for the algorithm to consistently detect the stringing. Similarly, the stringing in parts printed with these light filaments can be easily detected against a black background, but the stringing in the black part goes undetected with this background. While this method of stringing detection was not perfectly consistent, it was reliable enough in the right conditions that it could be included in the final system. However, the inconsistency, paired with the fact that stringing often occurs in non-failed prints, leads us to utilize stringing detection as a quality assurance metric rather than true failure detection.

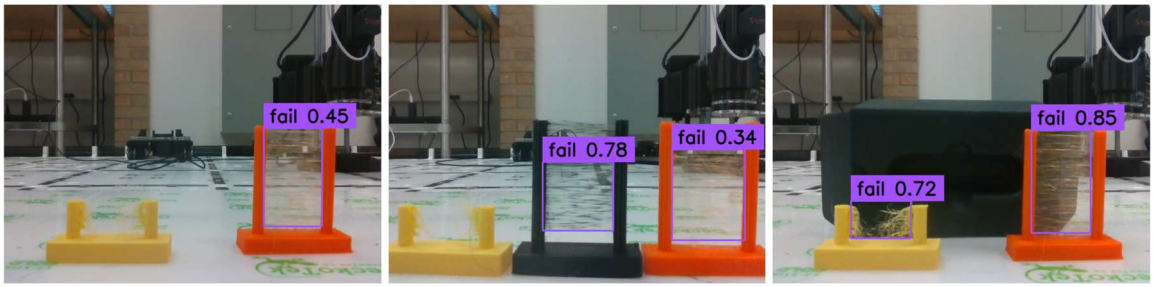


Figure 8: Detection of stringing in 3D printed parts with different filament colors and backgrounds.

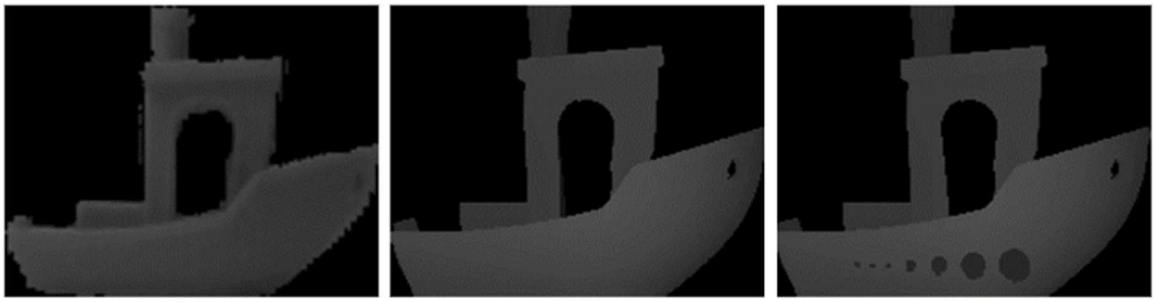


Figure 9: Masks of 3Dbenchy: (a) Mask generated from the camera image, (b) Mask generated from the correct STL file, (c) Mask generated from the altered STL file.

To test the 2D image matching framework, the side profile of a printed 3DBenchy model was used to generate an image mask of the part. This was then compared to rendered masks of both the STL file used to print the model and a slightly altered STL file with holes to simulate an error in the final print when compared to the ground truth model. In doing this, we hope to verify that the image matching can differentiate between a completed print that is correct and one that has errors. The similarity scores generated are 0.808 for the comparison between the object and the correct (original) STL file and 0.743 for the comparison between the object and the altered STL file with simulated errors. The masks generated for this test are shown in Figure 9. The main

reason for the imperfect comparison is the resolution of the camera itself, which causes the lines in the image mask to appear jagged, while the lines from the rendered mask are perfectly smooth. The grayscale filter not matching perfectly also causes some errors, as the pixels are not perfectly identical when compared, even when they should be. However, this is still the most promising method tested thus far, and as long as the similarity score is tuned properly, it should be capable of detecting failures without being prone to false positives.

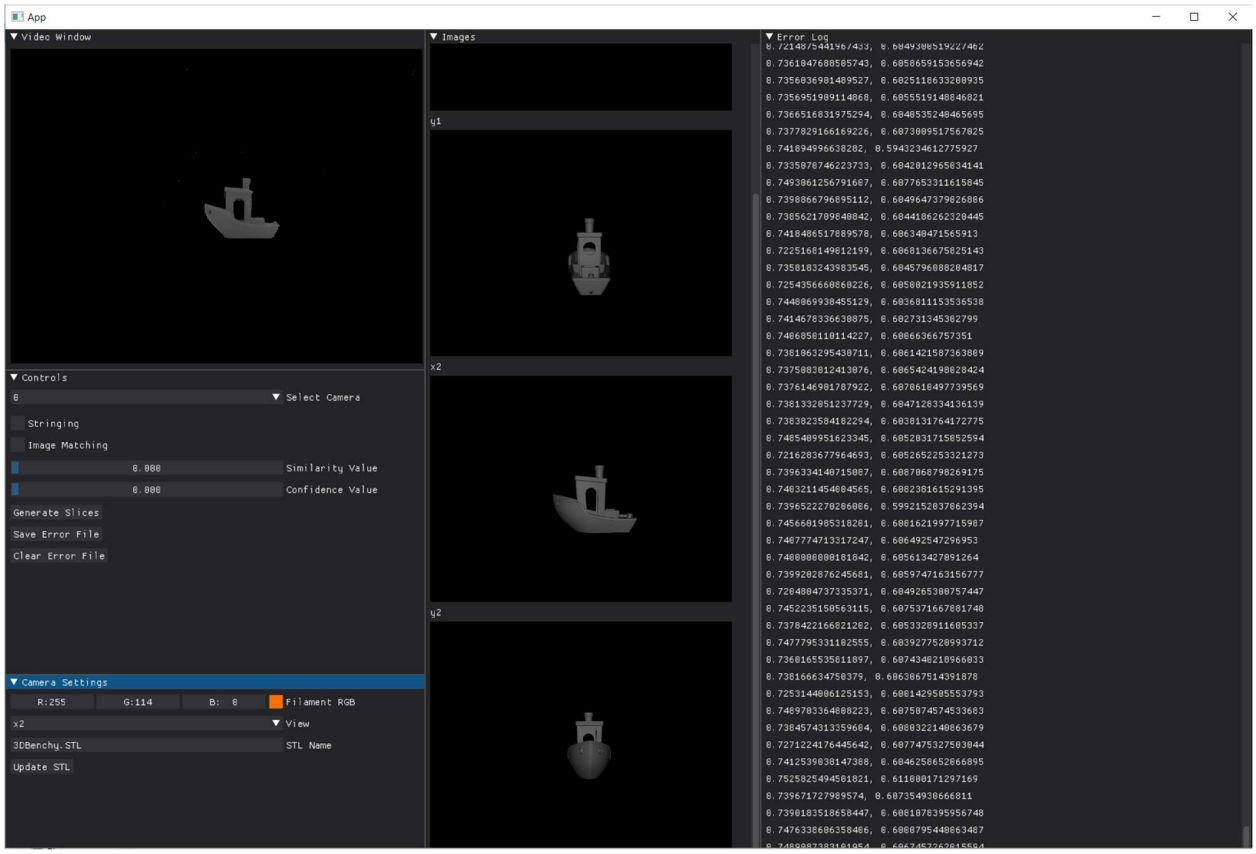


Figure 10: GUI of the integrated C3DP monitoring system.

Combined Monitoring System

Using the results from the individual testing cases, we combined the most functional systems into a holistic monitoring framework that can be applied to the C3DP platform. Since the nozzle tracking system lacked the accuracy required, it was not implemented. This left the stringing and 2D similarity matching algorithms to be integrated, which did not require additional hardware since they both could be run on the same camera using the mount from Figure 5. In order to run both scripts simultaneously while also providing real-time feedback that can be fed to the printers, a graphical user interface (GUI) was developed using the Python library DearPyGui [27]. This GUI provides useful information to the user, such as showing the filtered image mask and rendered mask to allow for visual comparison while also including numerous changeable settings to adjust how the monitoring system behaves. This includes changing the color of the filament that the image processor filters for, turning the stringing and 2D similarity algorithms on and off, and adjusting tolerances for the stringing confidence and similarity index. Finally, it has the functionality to detect and adjust settings specific to each camera connected to the computer, which

allows the camera perspective and the STL file of the printed part to be customized for each unique camera location and printed in a collaborative printing environment. A picture of this GUI is shown in Figure 10.

Test Cases

With the monitoring framework built, the next step is to test it on a live print. To do this, we utilized three main test cases, two of which were done on a single printer, while the final was a cooperative printer performed with two printers working together. Two cameras are used for every test to allow for different perspectives of each to be compared. For the single printer tests, the cameras were set up at a 90-degree angle from each other, with one set up opposite the printer and the other set up to the right side of the printer. This testing setup can be seen in Figure 11.

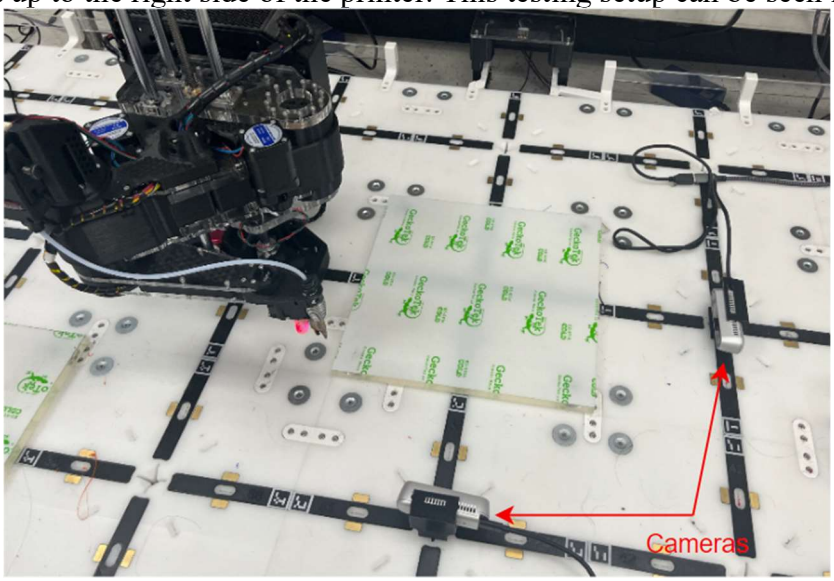


Figure 11: Single printer testing setup with two Intel Realsense D435 cameras.

The first test case is a highly simplified model of the Roman Colosseum, with dimensions of $100mm \times 100mm \times 38.1mm$. This model is fully symmetrical about the Z-axis, meaning that the print should look the same to both cameras regardless of the differing perspective. It also means that unless any errors occur in the print, any difference in the similarity index should be due to environmental factors rather than differences in the print geometry. As such, this is a good baseline test for determining the sensitivity of the 2D image comparison to environmental conditions such as the lighting and background. The Colosseum test can be seen in Figure 12.

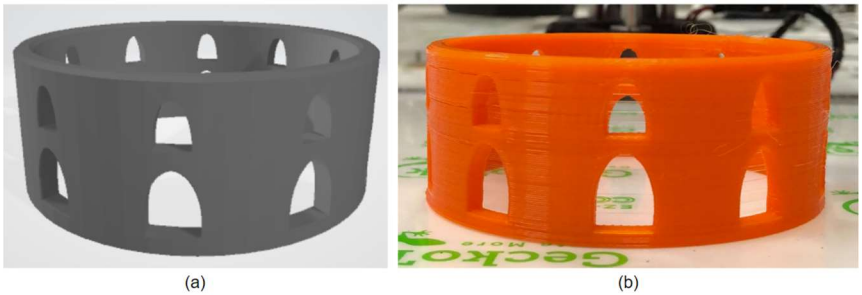


Figure 12: Colosseum test print. (a) Rendered STL file, (b) Completed print.

The second test case is a model of the UT tower with dimensions of $50.8mm \times 50.8mm \times 80mm$. This model has four distinct sides with the opposite sides mirrored, while the adjacent sides are different, which means that each camera will capture a unique face, unlike the previous test case. One of these sides has fine details in the form of a clock and windows, while the other side replaces these features with holes that cut through the entire part. As such, this test case helps determine the effectiveness of the image comparison system in monitoring prints with depth features, from full-length through-holes to finer surface details. The tower test is shown in Figure 13.

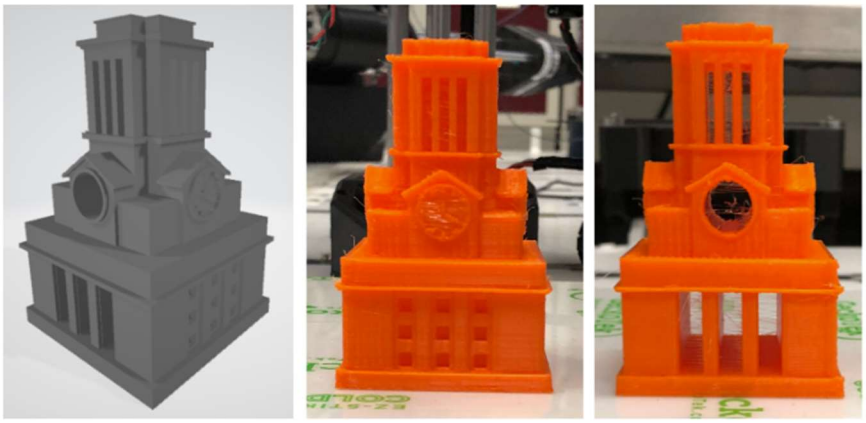


Figure 13: Tower test print. (a) Rendered STL file, (b, c) Unique perspectives of completed print.

The final test case performed was a cooperative print utilizing two printers. The printers were set up facing each other with two build plates between them to allow each printer to work on its own section. Since only two cameras were used, each section of the print was monitored by a single camera, which was set up to the left of its respective printer. This testing setup can be seen in Figure 14.

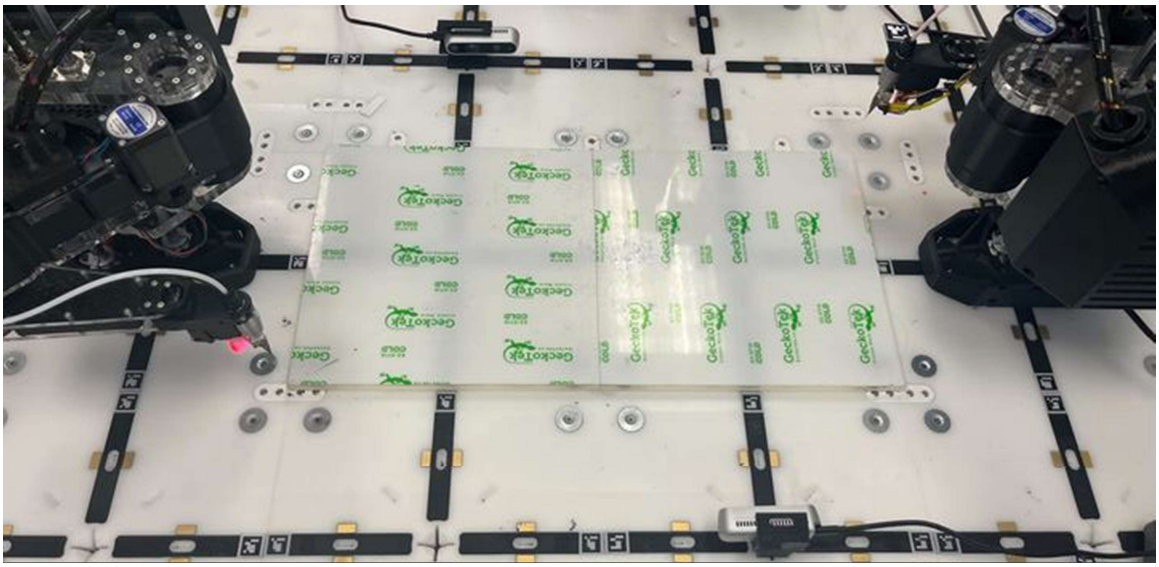


Figure 14: Collaborative testing setup with two printers and two Intel Realsense D435 cameras.

The print itself was developed to be a good test case for both the 2D image matching and stringing detection aspects of the monitoring framework. The 2D image matching is tested with holes of various sizes and shapes placed sporadically throughout the print, while the placement of two tall, thin beams on each end of the print should generate some stringing in each. Also, the use of two different filament colors will help determine how the filament color affects both stringing and image comparison. This collaborative test print can be seen in Figure 15.

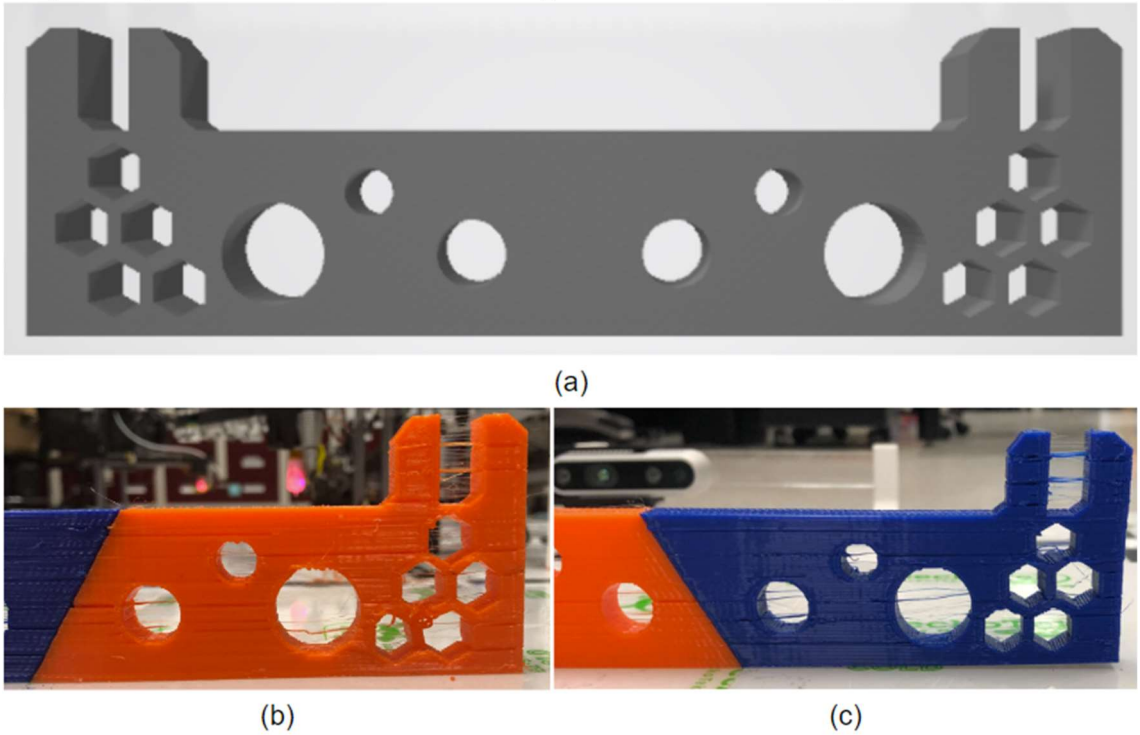
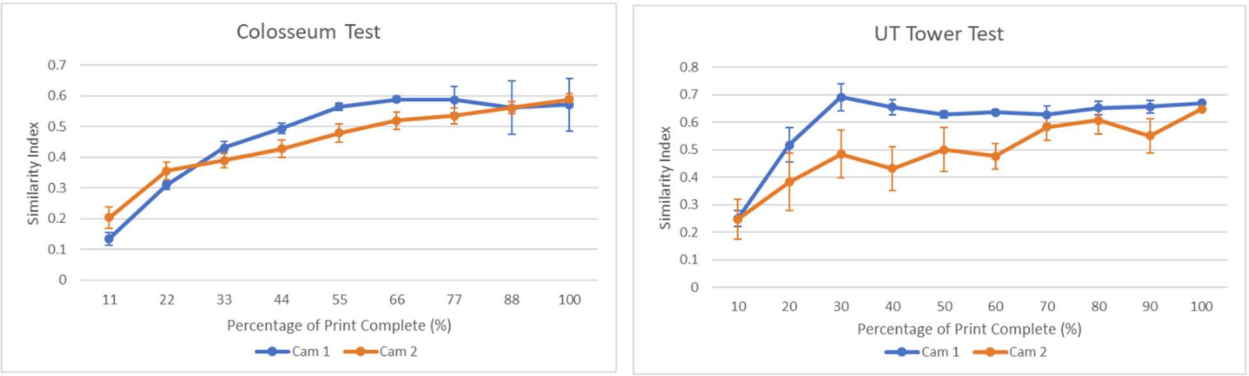


Figure 15: Collaborative test print. (a) Rendered STL file, (b, c) Both unique perspectives of completed print.

Results

Results for the test cases described in the previous section were gathered in real time during the printing process, with the only exception being the final data point which was gathered after the print was complete. In order to ensure that the data could be easily interpreted, we chose to run the monitoring system over a single layer at discrete points during the print. For example, the UT tower print had exactly 200 layers, so data was gathered every 20th layer or every 10% of the print. This way, the data could be divided based on print completeness, and as such, it provides us with a full picture of the entire printing process. The similarity data for the first two test cases is shown in Figure 16. No noteworthy stringing was detected for these test cases.

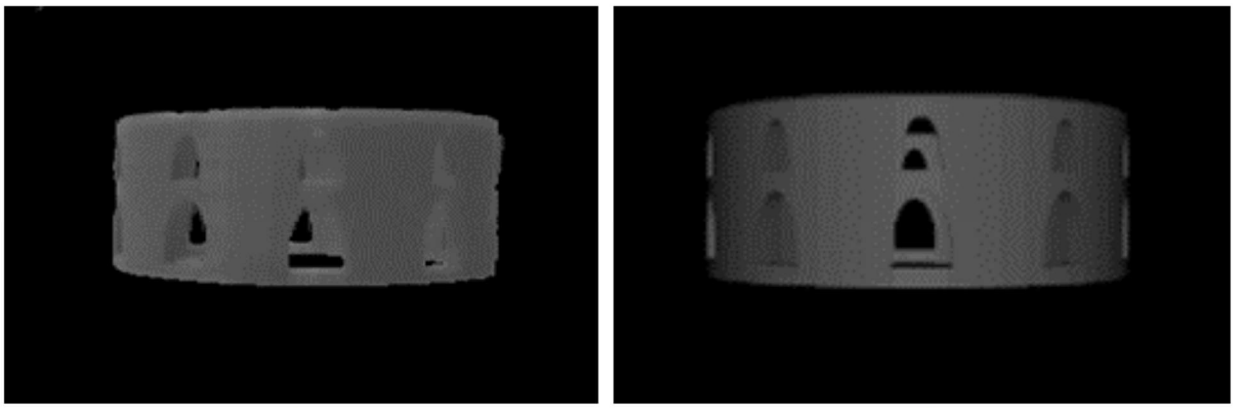
Both the graphs above show similar trends, with the similarity index being quite low in the early stages of the print but increasing as more of the print is completed. As such, we can safely assume that a certain percentage of the print must be completed before the comparison between the physical print and the rendered model becomes accurate enough to detect any deformities. However, the similarity index begins to level out around 0.6-0.7, indicating that very high similarity scores (> 0.8) are not consistently measured, even for relatively error-free prints.



(a) (b)

Figure 16: Single printer testing results. (a) Colosseum, (b) UT Tower.

Another notable finding is that environmental factors can affect the image mask enough to change the similarity index considerably. If this weren't the case, the similarity indexes measured by each camera at the layers of the colosseum test would overlap, but camera 2 is generally lower even after accounting for the standard deviation. This is only false near the beginning of the print, when it is too small to accurately compare with the rendered model, and at the very end, where we noted some data points where the similarity value measured by camera 1 dropped considerably (< 0.2). This seems to be due to the red light in the BLTouch sensor on the printer, which is close enough to the filament color that it occasionally would not be filtered out, instead being present in the filtered image and ruining the comparison. Once these values are filtered out, the standard deviation of these points decreases to a level similar to that of the other data points, and the similarity index increases to a value more in line with the existing trend.



(a) (b)

Figure 17: Filtered image of colosseum test from (a) physical print, (b) rendered model.

A final thing to note with these tests is that the presence of through holes in a print generally makes the comparison worse since any inaccuracy with the camera position or rendered slice generation will cause a significant error in the final similarity index. This can be seen in the fact that camera 1 for the UT tower test quickly reaches its maximum similarity and levels out, while camera 2 for this test and both cameras in the colosseum test have a general trend of increasing similarity over time as more material is deposited. This is due to the fact that image comparison is done on a pixel-by-pixel basis, and as such, the slightest shift in angle or perspective means that

some pixels that should be filled are not, while those that should be empty are filled. This can be seen in Figure 17, which shows the filtered image and rendered model for the colosseum test. The slight shift in the angle of the camera relative to the print leads to the holes not being properly aligned and causes significant errors in the comparison. This is not true when through holes are not present, as even though the fine details might be shifted in the camera frame relative to the model, the lack of empty pixels means that the error is less severe and, therefore, has a minor impact on the similarity index.

The final collaborative test was performed last and involved monitoring both the stringing and performing the real-time image comparison. One print was done with blue filament (viewed by camera 1), and the other with orange filament (viewed by camera 2). This way, each camera can distinguish its own part of the print without the results from the two overlapping. The results for each of these methods are shown in Figure 18.

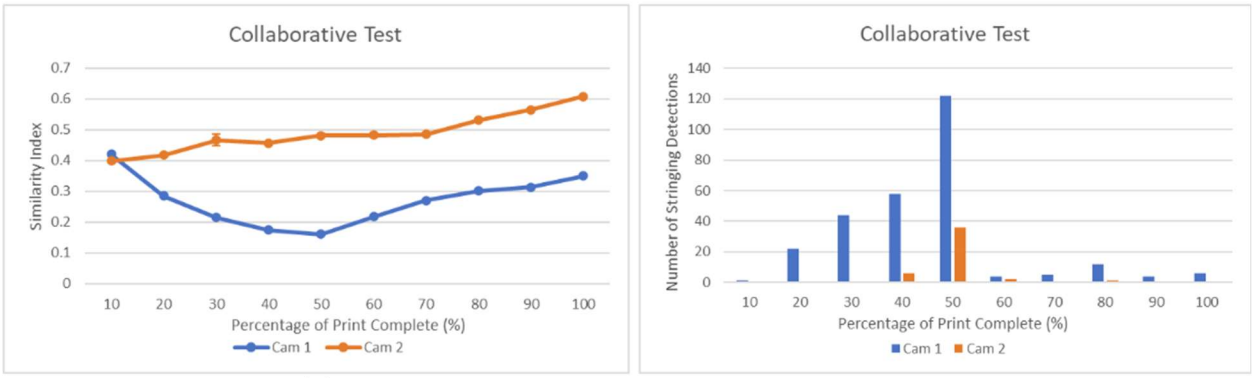


Figure 18: Collaborative print testing results. (a) Image similarity, (b) Stringing.

Starting with image matching, the most notable finding here is that the color of the filament used plays a major role in the ability of the software to generate an accurate mask. The measured similarity index drops to less than 0.2 until around 50% of the print is complete. This is not observed in other tests with brighter-colored filaments, which maintain a similarity of around 0.5. This is because the test environment has overhead lighting, which perfectly illuminates the top surface of the print but fails to provide enough light to the sides. For brighter filaments, this issue does not matter as much, but for darker filaments, the shading causes the pixel intensity of the filtered image to not accurately match that of the rendered STL model. We verified this by shining a light on the front surface of the part, and when better illuminated, the similarity index for the blue section of the part nearly matches that of the orange section. The image mask of both sides of the print under standard lighting can be seen in Figure 19.

Before we look at the stringing results, it is important to note that the part was not manually cleaned or fixed in any way during the printing process. This is pointed out because, without external intervention, we would expect that once stringing is detected at one point of the print, it will continue to be detected over the entire course of the print. This does not happen, as the results show the detection of stringing increases until around half of the layers are completed before dropping off considerably. This is around the point that the two largest holes in the test part are finished. As such, it seems that the lighting might play a factor here as well, as once the top of a

hole is closed off, the ability of the software to detect the stringing inside drops off considerably. Also, there was very little detection of stringing in the beams on top of the prints, even though these features had more visible stringing than the holes. This is either due to the angle at which the camera viewed them or the short distance between the beams not being long enough for the detection algorithm to reliably detect the strings. Lastly, while the print quality of both parts was very similar, much more stringing was detected from the blue filament. This is likely due to the darker color contrasting better with the bright background, while the orange filament seems to blend in.

Figure 19: Image mask of both sides of the collaborative print. (a) Orange, (b) Blue.

Conclusion

This paper presented a holistic process monitoring framework for the C3DP system. It utilizes two primary monitoring techniques: stringing detection using machine learning and image similarity comparisons using a structured similarity index. These two techniques are then integrated into a custom-made GUI and tested on the C3DP system using three unique test cases. The results from these test cases can be summarized as follows:

- Stringing
  - Capable of detecting both catastrophic and non-catastrophic stringing in parts
  - Highly sensitive to variables such as filament color, lighting, and background
  - Since all levels of stringing are treated the same, this metric should focus on quality assurance rather than definitive failure detection.
- Image matching
  - Can detect errors in completed parts
  - Must be carefully tuned to avoid false positives while still detecting major errors, meaning that very fine errors will often go undetected
  - Highly sensitive to filament color and lighting conditions
  - Comparison struggles in the early stages of a print, which is where failures are most likely to occur

While these results are promising, there is clearly a lot of room for improvement. As such, we have numerous ideas for future work that will build off of the foundation presented here. First of all, as noted previously, the conditions of the environment heavily affect how accurately error detection can be done. An obvious solution is to normalize the environment, but in a busy manufacturing setting, it is impossible to guarantee perfect lighting and background conditions at all times. To address this challenge, one must ensure the detection algorithm is adaptable to the

wide range of conditions that might be experienced. Currently, the image mask is generated using only the hue as a filter, while shading controls pixel intensity to visualize depth. However, since lighting conditions can change and filaments can have different tones as well as hues, we are exploring options to allow both factors of the color to be set while still accounting for depth. However, a major limitation that cannot be fixed is the fact that white and black filaments do not work with this system. While stringing detection is possible with the right background, black and white filaments have no hue and cannot be currently detected. While adding the ability to control for tone would allow for them to be detected, these colors are too prevalent in the system to realistically distinguish the filament from its background.

There are also other methods of doing image similarity comparisons that could considerably improve the accuracy of error detection. For example, a deep learning approach could be beneficial, especially because a major issue preventing accurate comparisons is the resolution of the cameras. We also wish to break down the image similarity comparison into smaller segments to detect localized defects and hopefully make the system more sensitive to small errors. This way, the system is more likely to detect common defects such as warping and poor layer adhesion, which are very difficult to notice on the scale of a full part. Lastly, improving the method for generating the rendered STL model slices and ensuring they more closely mirror the camera perspective would allow the comparison to be much more accurate, especially at the beginning of the print.

Finally, the current system simply monitors the print, and once it begins to consistently detect a similarity value below a threshold set by the user, it pauses the printer. This is functional, but it generally takes a catastrophic print failure to drop the similarity low enough that failure is registered. Setting the similarity threshold higher can allow the system to detect small errors more quickly, but it also leads to frequent false positives. We hope that the changes discussed previously will eventually allow the system to detect errors as they occur, at which point we can tune print parameters such as nozzle temperature, extrusion rate, and printing speed to correct them in real-time. This would be a major addition to the C3DP platform and would realize our goal of a real-time process monitoring system for C3DP.

Acknowledgments

We would like to thank our senior design team, consisting of Travis Bouchard, Rafiq Hamzeh, Kyle Hodowany, and Manuel Salazar for helping with the preliminary testing. Lastly, we would like to thank Kenta Yoshizaki for helping run the final test cases.

References

[1] Poudel, L., Marques, L. G., Williams, R. A., Hyden, Z., Guerra, P., Fowler, O. L., Sha, Z., & Zhou, W. (2022). Toward swarm manufacturing: Architecting a cooperative 3D printing system. *Journal of Manufacturing Science and Engineering*, *144*(8). https://doi.org/10.1115/1.4053681

[2] Weber, D., Zhou, W., & Sha, Z. (2022). Z-Chunking for Cooperative 3D Printing of Large and Tall Objects. Proceedings of the 33rd Annual International Solid Freeform Fabrication Symposium (SFF). http://dx.doi.org/10.26153/tsw/44190

[3] McPherson, J., & Zhou, W. (2018). A chunk-based Slicer for cooperative 3D printing. Rapid Prototyping Journal, 24(9), 1436–1446. https://doi.org/10.1108/rpj-07-2017-0150

[4] Weber, D., Zhou, W., & Sha, Z. (2023). Job Placement for Cooperative 3D Printing. *Manufacturing Science and Engineering Conference (MSEC)*. https://doi.org/10.1115/MSEC2023-104613

[5] Poudel, L., Zhou, W., & Sha, Z. (2019). Computational design of scheduling strategies for multi-robot cooperative 3D printing. Volume 1: 39th Computers and Information in Engineering Conference (CIE). https://doi.org/10.1115/detc2019-97640

[6] Poudel, L., Zhou, W., & Sha, Z. (2021). Resource-constrained scheduling for multi-robot cooperative three-dimensional printing. Journal of Mechanical Design, 143(7). https://doi.org/10.1115/1.4050380

[7] Elagandula, S., Poudel, L., Sha, Z., & Zhou, W. (2020). Multi-robot path planning for cooperative 3D printing. International Manufacturing Science and Engineering Conference (MSEC). https://doi.org/10.1115/msec2020-8390

[8] Poudel, L., Sha, Z., & Zhou, W. (2018). Mechanical strength of chunk-based printed parts for cooperative 3D printing. Procedia Manufacturing, 26, 962–972. https://doi.org/10.1016/j.promfg.2018.07.123

[9] Usha, S. (2021). In situ monitoring of Metal Additive Manufacturing Process: A Review. *Additive Manufacturing*, 275–299. https://doi.org/10.1016/b978-0-12-822056-6.00007-2

[10] *Eos Smart Monitoring for 3D printing*. EOS. https://www.eos.info/en-us/enablement/software/eos-smart-monitoring

[11] Wittbrodt, B. T., Glover, A. G., Laureto, J., Anzalone, G. C., Oppliger, D., Irwin, J. L., & Pearce, J. M. (2013). Life-Cycle Economic Analysis of distributed manufacturing with open-source 3-D printers. *Mechatronics*, *23*(6), 713–726. https://doi.org/10.1016/j.mechatronics.2013.06.002

[12] Fu, Y., Downey, A., Yuan, L., Pratt, A., & Balogun, Y. (2021). In situ monitoring for fused Filament Fabrication Process: A Review. *Additive Manufacturing*, *38*, 101749. https://doi.org/10.1016/j.addma.2020.101749

[13] Oleff, A., Küster, B., Stonis, M., & Overmeyer, L. (2021). Process monitoring for material extrusion additive manufacturing: A state-of-the-art review. *Progress in Additive Manufacturing*, *6*(4), 705–730. https://doi.org/10.1007/s40964-021-00192-4

[14] Brion, D. A., & Pattinson, S. W. (2022). Generalisable 3D printing error detection and correction via multi-head neural networks. *Nature Communications*, *13*(1). https://doi.org/10.1038/s41467-022-31985-y

[15] Jin, Z., Zhang, Z., & Gu, G. X. (2019). Autonomous in-situ correction of fused deposition modeling printers using computer vision and Deep Learning. *Manufacturing Letters*, *22*, 11–15. https://doi.org/10.1016/j.mfglet.2019.09.005

[16] Straub, J. (2015). Initial work on the characterization of Additive Manufacturing (3D printing) using software image analysis. *Machines*, *3*(2), 55–71. https://doi.org/10.3390/machines3020055

[17] Brion, D. A. J., Shen, M., & Pattinson, S. W. (2022). Automated recognition and correction of warp deformation in extrusion additive manufacturing. *Additive Manufacturing*, *56*, 102838. https://doi.org/10.1016/j.addma.2022.102838

[18] Petsiuk, A. L., & Pearce, J. M. (2020). Open source computer vision-based layer-wise 3D printing analysis. *Additive Manufacturing*, *36*, 101473. https://doi.org/10.1016/j.addma.2020.101473

[19] Paraskevoudis, K., Karayannis, P., & Koumoulos, E. P. (2020). Real-time 3D printing remote defect detection (stringing) with Computer Vision and artificial intelligence. *Processes*, *8*(11), 1464. https://doi.org/10.3390/pr8111464

[20] *Depth camera D435*. Intel® RealSenseTM Depth and Tracking Cameras. (2022, December 5). https://www.intelrealsense.com/depth-camera-d435/

[21] Bolme, D., Beveridge, J. R., Draper, B. A., & Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2010.5539960

[22] *Detection of ArUco Markers*. OpenCV. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

[23] *Your 3D printer is dumb. make it smart!*. Obico. https://www.obico.io/

[24] Jocher, G. *Ultralytics YOLOv8*. GitHub. https://github.com/ultralytics/ultralytics?tab=readme-ov-file

[25] QA, A. M. (2022, December 29). *3DPrinting Computer Vision Project*. Roboflow. https://universe.roboflow.com/additive-manufacturing-qa/3dprinting

[26] Lyngby, R. A., Wilm, J., Eiriksson, E. R., Nielsen, J. B., Jensen, J. N., Aanæs, H., & Pedersen, D. B. (2017). In-line 3D print failure detection using computer vision. Euspen. https://www.euspen.eu/knowledge-base/AM17133.pdf

[27] Hoffstadt. (2024). DearPyGui. GitHub. https://github.com/hoffstadt/DearPyGui