

A TRIDENT SCHOLAR PROJECT REPORT

NO. 502

Tracking Additive Manufacturing using Machine Vision

by

Midshipman 1/C Lenning A. Davis IV, USN



UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

This document has been approved for public
release and sale; its distribution is unlimited.

USNA-1531-2

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.				
1. REPORT DATE (DD-MM-YYYY) 7/12/21		2. REPORT TYPE		3. DATES COVERED (From - To)
4. TITLE AND SUBTITLE Tracking Additive Manufacturing using Machine Vision		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Davis, Lenning A.		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Naval Academy Annapolis, MD 21402		10. SPONSOR/MONITOR'S ACRONYM(S)		
		11. SPONSOR/MONITOR'S REPORT NUMBER(S) Trident Scholar Report no. 502 (2021)		
12. DISTRIBUTION / AVAILABILITY STATEMENT This document has been approved for public release; its distribution is UNLIMITED.				
13. SUPPLEMENTARY NOTES				
14. ABSTRACT This project investigates multiple methods of verifying Additive Manufacturing (AM) products using computer vision, including feature-based visual odometry and image registration. With the rise of AM comes cyber-physical risks and the potential for part defects. Feature-based Visual Odometry (VO) is a field of computer vision research that uses information from images to calculate the relative motion of the camera through space. We developed a VO algorithm to reconstruct extruder position and motion from stereo images as a cost-effective means to monitor AM part construction. Preliminary results in simulation demonstrate feasibility of the proposed VO method and identify factors that may limit performance. In addition to VO, we seek to verify product quality using image registration and template matching techniques. Investigation with feature-based methods led to the development of a print bed tracking pattern, which enables the estimation of the extruder's position and orientation. Combined with knowledge of the product's pose and the position of the camera module a render of the product can be created. This simulated image can be compared to images captured from the camera module using image registration algorithms to check for defects. Preliminary results yield successful image segmentation and detection of structural defects.				
15. SUBJECT TERMS Additive manufacturing, Cyber-physical security, Visual odometry, Product verification, Template matching				
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 43
a. REPORT	b. ABSTRACT	c. THIS PAGE		
				19a. NAME OF RESPONSIBLE PERSON
				19b. TELEPHONE NUMBER (include area code)

U.S.N.A. --- Trident Scholar project report; no. 502 (2021)

TRACKING ADDITIVE MANUFACTURING USING MACHINE VISION

by

Midshipman 1/C Lenning A. Davis IV
United States Naval Academy
Annapolis, Maryland

Certification of Advisers Approval

Associate Professor Michael D.M. Kutzer
Weapons, Robotics, and Control Engineering Department

Assistant Professor John Donnal
Weapons, Robotics, and Control Engineering Department

Acceptance for the Trident Scholar Committee

Professor Maria J. Schroeder
Associate Director of Midshipman Research

Abstract

This project investigates multiple methods of verifying Additive Manufacturing (AM) products using computer vision, including feature-based visual odometry and image registration. With the rise of AM comes the potential for part defects and cyber-physical risks. This research proposes a Visual Odometry (VO) based approach to compare estimated extruder trajectory to the expected result for error identification. Feature-based VO is a field of computer vision research that uses information from sequential images to calculate relative motion of the camera through space. We seek to develop a VO algorithm for extruder trajectory reconstruction as a cost effective means to monitor AM part construction. Preliminary results in simulation demonstrate feasibility of the proposed VO method and identify factors that may limit performance. In addition to VO, we seek to verify product quality using image registration and template matching techniques. A simulated image of the product is rendered with an estimate of the extruder's pose and knowledge of the product's pose relative to the print bed. This simulated image is compared to images captured from the camera module using image registration algorithms to check for defects. Preliminary results yield successful image segmentation and detection of structural defects.

Keywords: Additive manufacturing; Cyber-physical security; Visual odometry; Product verification; Template matching

Acknowledgements

I would first like to thank the Trident Scholar program and committee for affording me the opportunity to pursue my passion for independent research in robotics. Thank you to the Office of Naval Research (ONR) for their support of this project through FY19 Grant No. N0001419WX00587, FY20 Grant No. N0001420WX00934, and FY21 Grant No. N0001421WX00428; and the financial support of Naval Sea Systems Command (NAVSEA) Engineering Directorate (05T). Finally, this project would not be possible without my project advisors, Professor Kutzer and Professor Donnal. For more than a year they have gone out of their way to guide this project, develop me into a better student, and mentor me beyond what is expected from project advisors.

Contents

1	Introduction	4
2	Testbed Overview	6
2.1	Printer Retrofit	6
2.2	Remote Access	7
3	Relative Pose Estimation	8
3.1	Feature-Based Visual Odometry	9
3.1.1	General Process	9
3.1.2	Simulation	10
3.1.3	Camouflage Pattern	17
3.1.4	Discussion	18
3.2	H.264 Motion Vectors	19
3.2.1	Background	19
3.2.2	Investigation	19
3.2.3	Results and Discussion	20
4	Absolute Pose Estimation	21
4.1	Background Patterns and PnP	21
4.1.1	Checkerboard/Grid Pattern	21
4.1.2	AprilTags	22
4.1.3	Discussion	22
4.2	Machine Learning	23
4.2.1	Background	23
4.2.2	General Process	24
4.2.3	Results and Discussion	26
5	Image Comparison and Template Matching	27
5.1	General Process	27
5.2	Image Capture and Segmentation	28
5.3	Initial Pose Estimation	29
5.4	Object Rendering and Visualization	29
5.5	Image Comparison Metrics	30
5.5.1	Contour Mask Method	30
5.5.2	Energy Function Method	32
5.6	Recursive Pose Estimation	33
5.6.1	General Process	33
6	Conclusions	39
7	Publications	40
7.1	Symposium on Additive Manufacturing and Applications 2020	40
7.2	International Mechanical Engineering Congress and Exposition 2021	41

1 Introduction

Recent advances in Additive Manufacturing methods (AM) have caught the attention of the United States (US) Department of Defense (DOD) for applications in all branches of the US Armed Forces [1–3]. Considerations of efficiency shape the military supply chain, and AM has the potential to bring made-to-order part manufacturing to military bases and on-board naval vessels. Common forms of AM (excluding methods like binder jet printing) utilize a process of creating a product layer by layer by depositing heated material in a three-dimensional space. Previous layers cool and harden during the printing process, and subsequent layers are stacked until the printing process is complete. Additive manufacturing machines are comprised of the same general components: an extruder for heating and depositing material, stepper motors to control the 3D motion of the extruder, a printbed, and software to map the trajectory of the extruder during the printing process. An example Fused Filament Fabrication (FFF) AM machine is shown in Fig. 1.

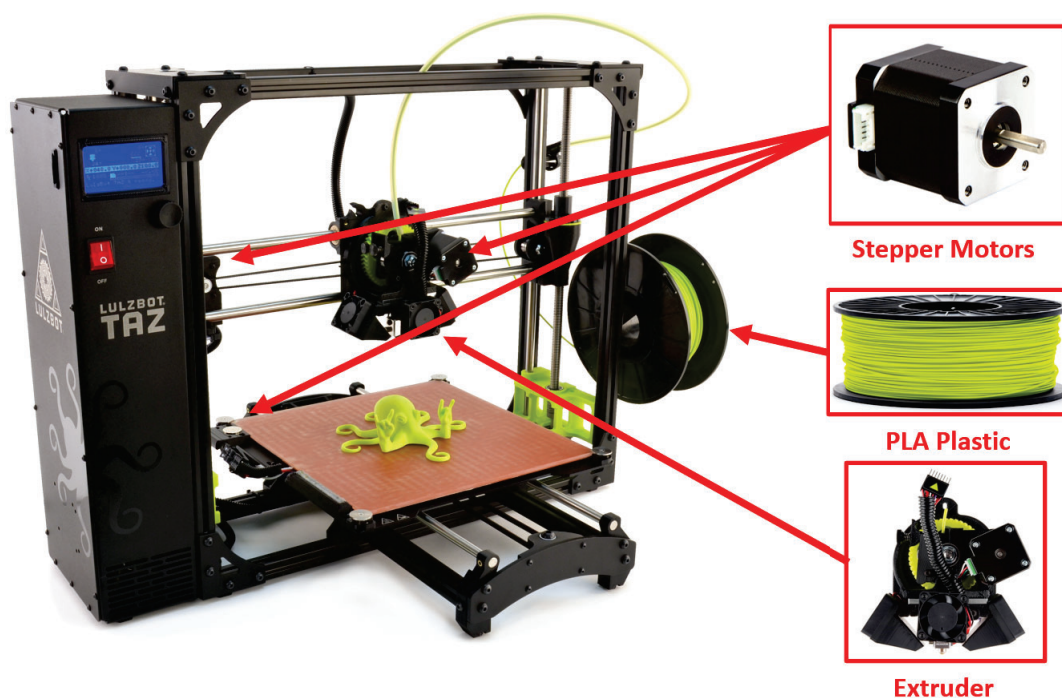


Figure 1: LulzBot TAZ6 PLA Plastic 3D Printer with main components highlighted.

AM is not a perfect process and errors during the printing process should be expected. An AM machine on board a naval vessel may be impacted by heavy seas, shipboard vibrations, and environmental conditions. Beyond environmental concerns, deliberate tampering from an adversary conducting a cyber-physical attack to alter the printer's trajectory or a part's internal geometry is a growing concern for DoD [4, 5]. Because products are created layer-by-layer the internal structure of an AM product can be adjusted without altering the final physical appearance. Fig. 2 shows a collection of incomplete AM products highlighting different infill geometries (i.e. internal structures) to illustrate a variety of possible internal geometries that are hidden upon part completion. Because every part is made-to-order and the infill of an object can be adjusted, AM is unlike other forms of subtractive manufacturing. Defects on the

interior of the part may compromise its structural integrity, but not alter how the part looks or feels [4]. As a result, simply inspecting the part after it is printed may not account for defects that impact the internal structure of the product. Thus, verifying the holistic quality of an AM product is an essential and difficult problem that carries heavy implications for the success of AM in the DoD and manufacturing industry at large.

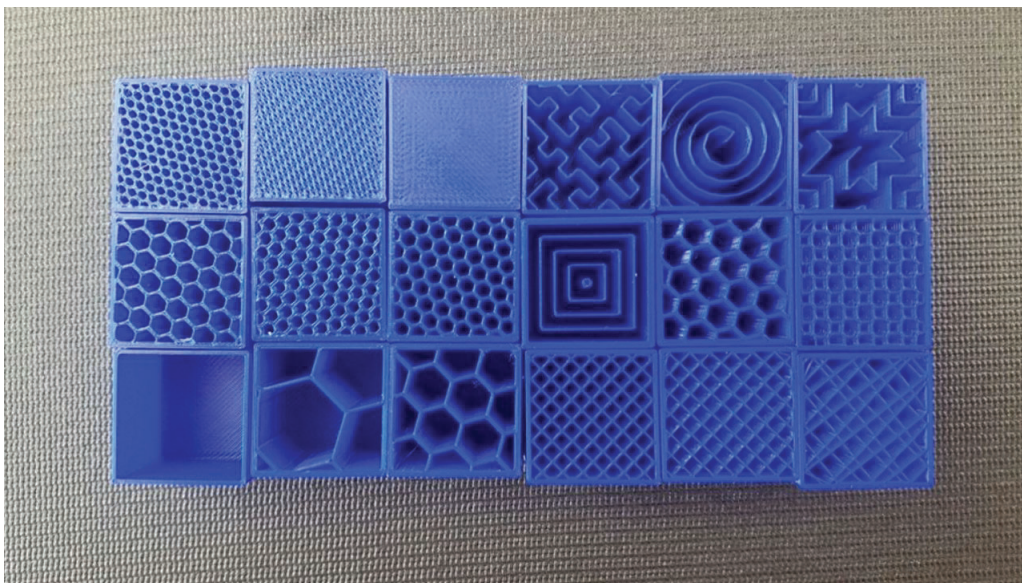
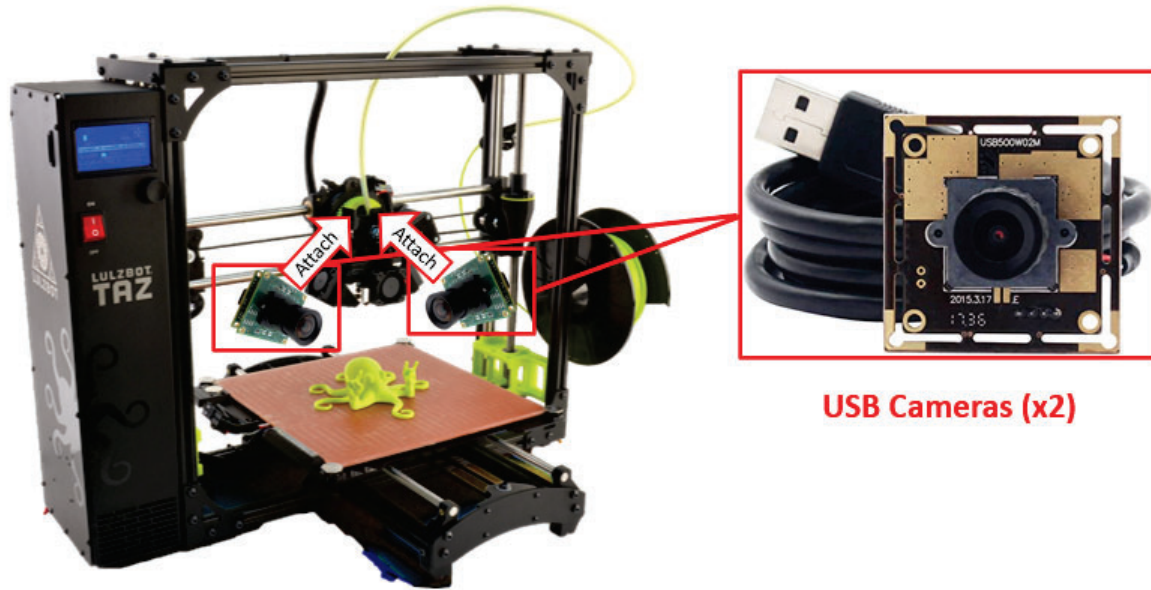


Figure 2: Comparison of AM infill patterns from incomplete AM products [6]. Note that these structures are not visible in a final product.

We present a vision-based approach of AM product verification to address this problem. Our research is founded upon a low-cost, minimally invasive retrofit to AM machines with commercial off the shelf (COTS) hardware. We are proposing a low-cost alternative to destructive analysis, and analysis using imaging tools like computed tomography (CT). We propose attaching COTS USB cameras to the extruder of an AM machine to track and verify the product throughout the printing process. A graphic of the cameras attached to an AM machine is shown in Fig. 3. This research proposes two means of verifying product quality: extruder pose estimation and model registration. Pose estimation is the understanding of the extruder's position and orientation inside the print area through the course of the printing process. The position and orientation of the extruder is estimated from the attached cameras and compared to the expected trajectory. This enables the detection of anomalous extruder motion during the printing process. Model registration involves comparing an image of the product to an expected digital render of the object. An estimate of extruder pose is recovered during the model-based registration process to render a simulated image of the product and several comparison metrics are applied to compare the render to the captured image of the physical part. Using the methods proposed in this work, the simulated image can be rendered for any layer in the printing process, thus internal deformities can be detected before they are covered by subsequent layers.



USB Cameras (x2)

Figure 3: Graphic showing conceptual COTS USB camera mounting locations on an AM machine. Cameras are fixed to the printhead of the AM system allowing them to move relative to the part during fabrication.

2 Testbed Overview

The physical testbed is designed around a LulzBot TAZ6 (Aleph Objects, Inc., Loveland, CO). The TAZ 6 provides a reliable AM platform with open hardware and software, which is leveraged to retrofit the printer with motor encoders, a stereo camera module, and a custom printbed pattern.

2.1 Printer Retrofit

The five stepper motors that actuate the TAZ6 are replaced with dual shaft equivalent motors, each with quadrature encoders attached to the rear shaft [7]. Encoder measurements from the TAZ6 retrofit are used as ground-truth to validate estimates derived from pose estimation algorithms. A stereo camera module consisting of two USB cameras is mounted to the TAZ6 extruder [7]. The camera module records stereo images to be used for the VO algorithm. An annotated image of the printer retrofit is shown in Fig. 4 and the camera module is shown up close in Fig. 5

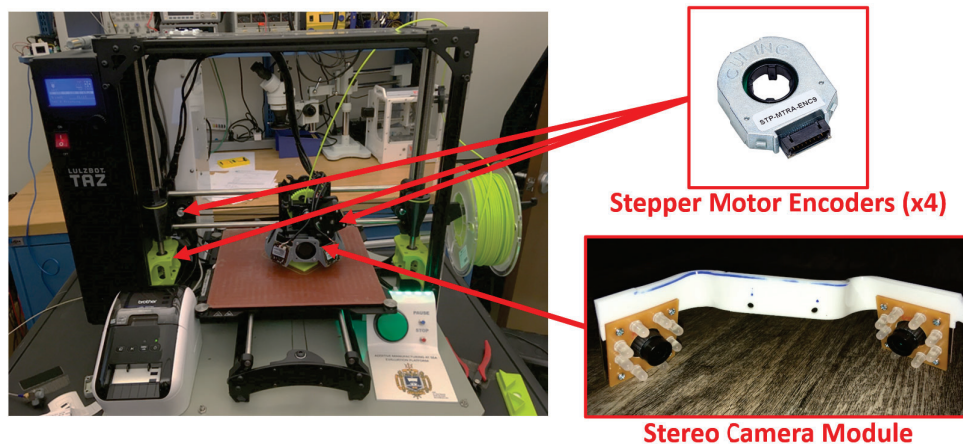


Figure 4: TAZ6 printer showing the encoder, camera module, and printbed pattern retrofit.

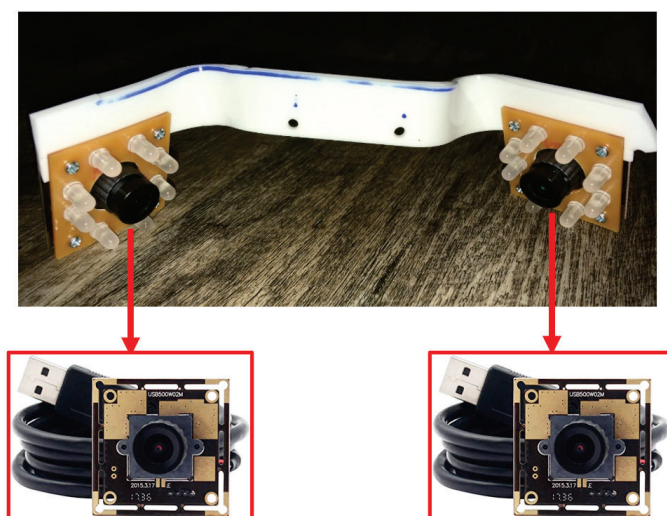


Figure 5: Camera module attached to the TAZ6 extruder, mounted with two USB cameras. Note LED lights are attached around the cameras for improved illumination.

2.2 Remote Access

The TAZ6 workstation for this project is configured for remote access due to the unique challenges of 2020. All printer control is accessible from an online Graphical User Interface (GUI) shown in Fig. 6. This enables control of all printer functions without being physically present in the lab space, to include starting/stopping prints and recording data. Furthermore, the USB cameras attached to the stereo camera module are accessible through this interface, shown in Fig. 7. A third wide angle camera is attached to the TAZ6 to provide a view of the full print area.

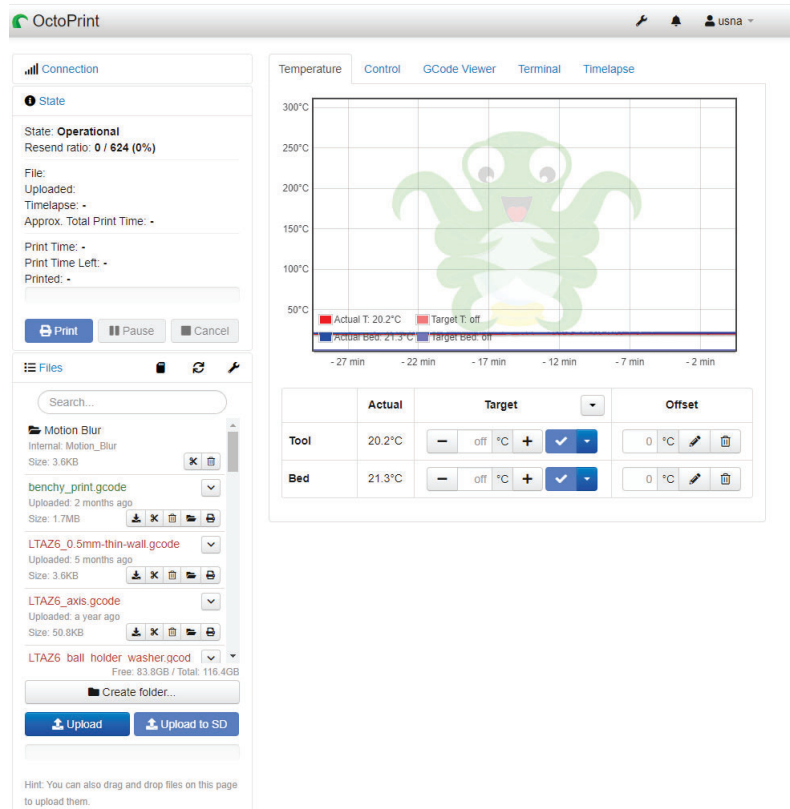


Figure 6: Screen capture of the remote control GUI.

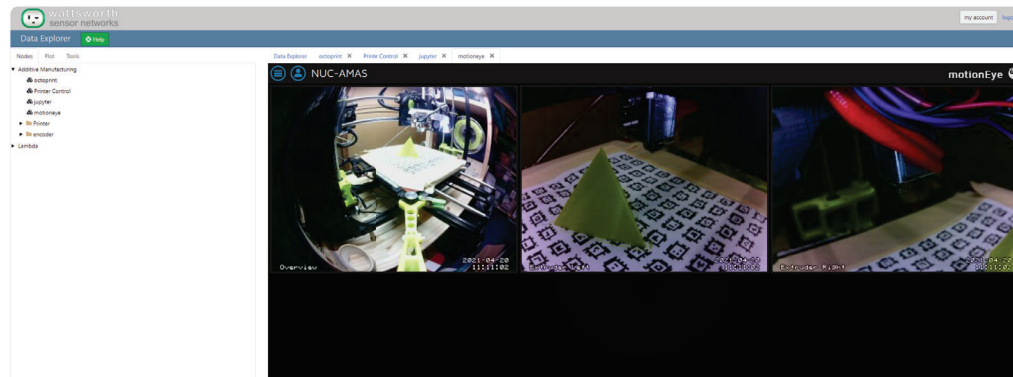


Figure 7: Screen capture of remote camera feeds of the TAZ6 set up. These video feeds are accessible from any computer with an internet connection and enabled remote work on this project.

3 Relative Pose Estimation

Our method of verifying additive manufacturing products consists of two fundamental problems: extruder pose estimation and model registration. Pose estimation is the understanding of the extruder's

position and orientation inside the print area through the course of the printing process. We investigated a variety of methods that fall in two general approaches to pose estimation: relative pose estimation and absolute pose estimation. Relative pose estimation involves solving for the extruder's pose at a given time step relative to a previous time and reconstructing absolute position as a product of intermittent poses.

3.1 Feature-Based Visual Odometry

Feature-based visual odometry (VO) is the process of estimating the motion of a camera by tracking key-points in successive images. Feature-based VO is a topic of computer vision research with history dating back to the 1980's [8]; there is a plethora of literature covering a variety of methods, algorithms, and approaches. Feature-based VO has traditionally been associated with mobile robotics, and not additive manufacturing.

3.1.1 General Process

The feature-based visual odometry developed for this work entails the following general steps:

1. Capture a stereo image pair.
2. Identify features within the image pair and describe them using the Speeded Up Robust Features (SURF) algorithm [9].
3. Match features across the stereo image pair.
4. Estimate position of the features in 3D space using the pinhole camera model as represented in Eq. 1. This involves translating the features from the image coordinates (x^m, y^m) to 3D coordinates relative to the camera $(x^c, y^c, \text{ and } z^c)$ using camera intrinsics (A_c^m) and extrinsics $(R_c^c \text{ and } d_c^c)$.

$$\begin{bmatrix} x^m \\ y^m \\ 1 \end{bmatrix} = \frac{1}{z^c} A_c^m \begin{bmatrix} R_c^c & d_c^c \end{bmatrix} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} \quad (1)$$

5. Correlate shared features from the stereo image pair to features from an adjacent instance in time using the SURF algorithm [9].
6. Estimate the relative pose of the extruder across a time step using the difference in 3D position of features from different instances in time [10] [11].
7. Calculate the pose of the extruder at a specific instance in time as the product of the relative transformations for each time step, as shown in Eq. 2. $H_0^{t_n}$ corresponds to the relative transformation of the extruder from its starting position to its pose at time t_n .

$$H_0^{t_n} = H_{t_{n-1}}^{t_n} H_{t_{n-2}}^{t_{n-1}} \dots H_{t_1}^{t_2} H_{t_0}^{t_1} \quad (2)$$

3.1.2 Simulation

Before applying our feature-based approach directly to the TAZ6 testbed, we verified our developed approach in simulation to investigate the impact of various factors that could affect visual odometry accuracy. We created a simulation environment in MATLAB to calculate printer trajectory using simulated cameras and features. The simulated printbed is shown in Fig. 8 and an example pair of simulated images are shown in Fig. 9. As the simulated extruder “moves” over the testbed, the figures change in real time for visualization. Once the process is complete, the actual trajectory and the estimated trajectory of the extruder are plotted in 3D.

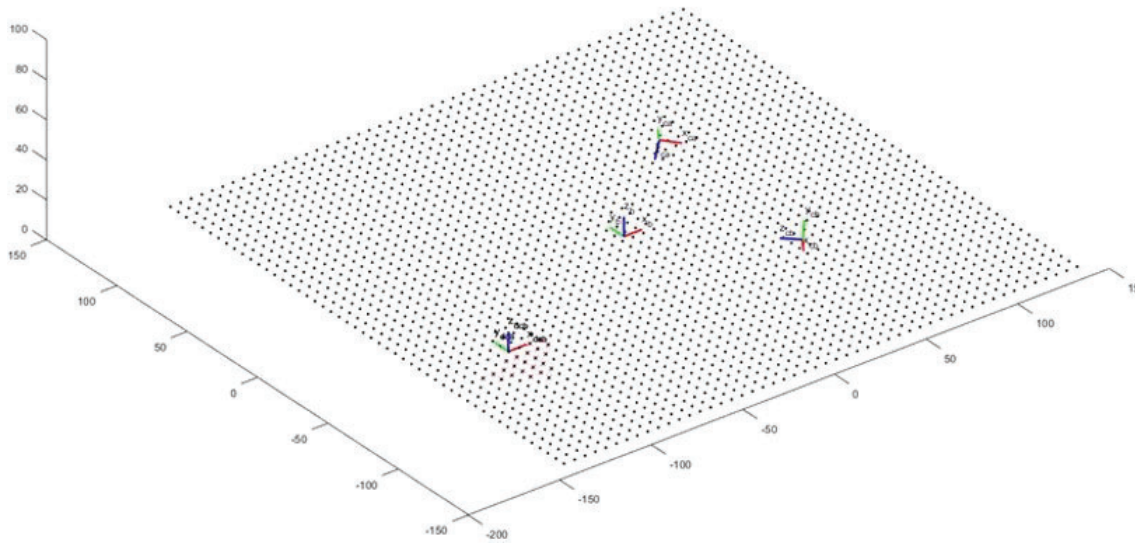


Figure 8: Visualization of the simulated printbed. Features are shown as black dots. The center of the printbed, simulated cameras, and position of the extruder are represented as red, green and blue coordinate frames.

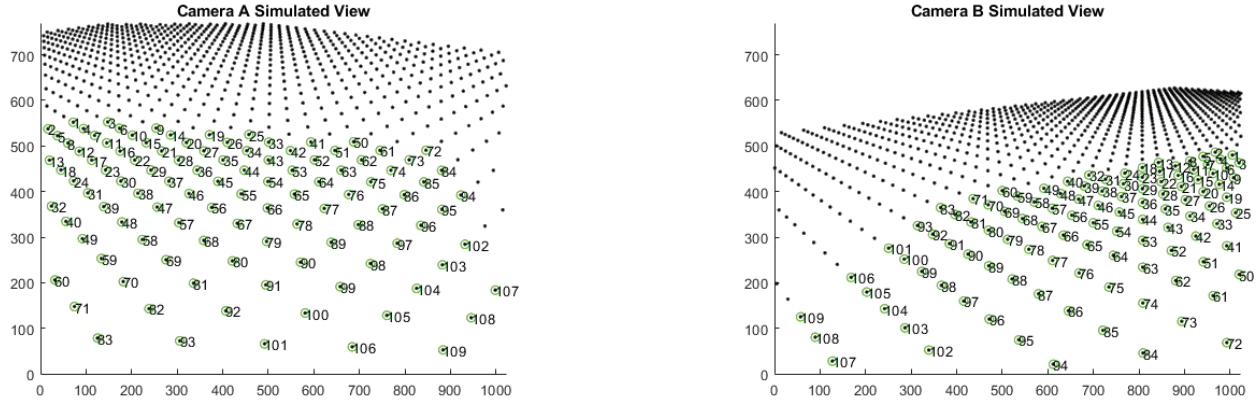


Figure 9: Simulated stereo image pair. Simulated features are represented as black dots, and shared features are highlighted with a green circle.

As a proof of concept for VO trajectory reconstruction, a simulation was performed under ideal circumstances, with: an abundance of features, features matched perfectly across cameras, features matched perfectly across time, and perfectly synchronized camera images. The results yield an accurate reconstruction of printhead trajectory as shown in Fig. 10.

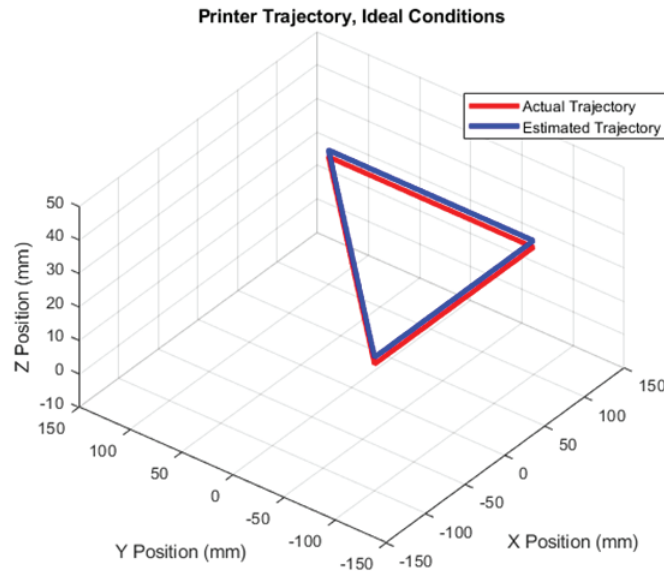


Figure 10: Estimated position of the simulated extruder head in comparison with the actual position in simulation. The estimated trajectory has been shifted 2mm in the z-axis for visibility.

The simulation environment underscores the feasibility of a vision-based approach to printer trajectory reconstruction. Under ideal circumstances, the position of the extruder is estimated through 400 frames

with an average root mean squared accuracy of $.21 \mu m$. Error in position estimation for this simulation is given in Fig. 11.

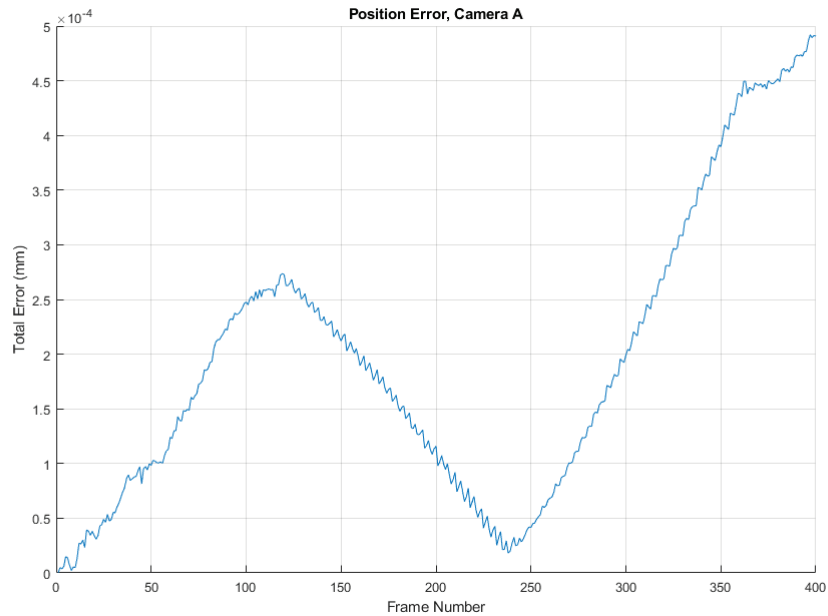


Figure 11: Shows error in position of the simulation estimate vs the simulation actual position. Note that the estimation of position does not exceed $1 \mu m$

When running preliminary tests on the TAZ6 testbed to determine ideal camera frame rates and exposure, we noticed inconsistencies in camera synchronization. To investigate, we conducted tests by recording videos of a stopwatch visible in both camera frames. We used the Optical Character Recognition functionality available in the MATLAB Computer Vision Toolbox to extract the time from the stopwatch for each frame in the video. We discovered a camera synchronization offset of approximately $1/10^{\text{th}}$ of a second. Furthermore, this synchronization offset was not constant and the cameras would drop in and out of sync approximately every 200 frames. In the test reflected by Fig. 12, between frames 196 and 197 *Camera A* dropped a frame and continued recording one frame behind *Camera B*.

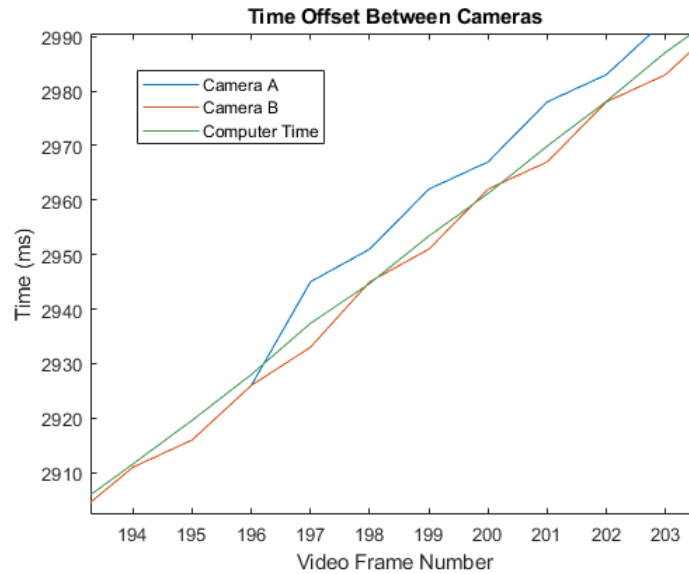


Figure 12: Time is annotated from two videos of the same stopwatch on the printbed. Note the divergence in time between frames 196 and 197. The clock maintained from the computer that records the video is included for reference.

We hypothesize that this synchronization error is a result of the camera hardware. The cameras that we are using for this research are low-cost, commercially available USB cameras - similar to those you would use for web conferencing. We theorize that these cameras are programmed to drop frames when they fall behind in processing the large amount of video data. While there is hardware available that can synchronize the cameras to the microsecond, a stated goal of this project is to avoid hardware constraints as we see them detracting from the broader impact of this research: to develop a generalized, minimally invasive VO process that can be applied to a variety of hardware systems.

To understand how this synchronization error impacts the reconstruction of the printer trajectory, we implemented various synchronization offsets into the MATLAB simulation. With a constant offset of one frame, the estimation of extruder position drifts away from the actual position visible in Fig. 13. The error in position estimation is not constant through the course of the print, and changes with the direction of travel. While the error in motion estimation is relatively small for each time step, the error in position grows through the course of the print as in the feature-based method, position is calculated by summing the relative transformations at each time step. This growing drift is illustrated in Fig. 14.

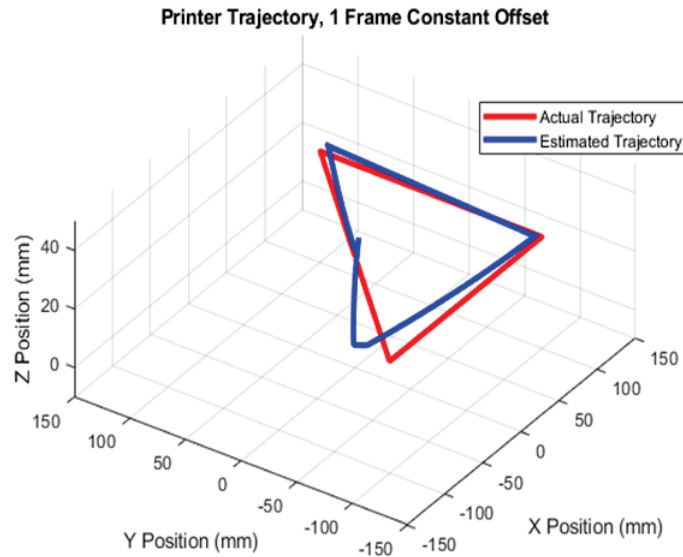


Figure 13: Printer trajectory with a simulated constant camera synchronization offset.

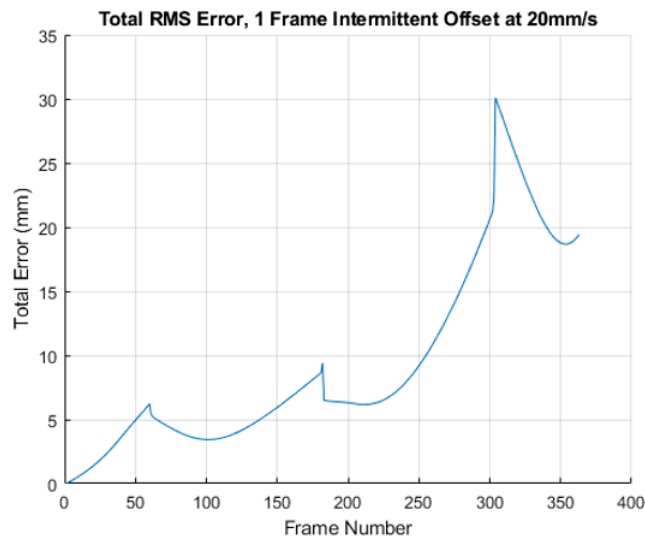


Figure 14: Position error of the simulation with a constant camera synchronization offset. Note that the error grows with time, and the rate of drift changes when the printer changes direction.

In addition to testing a constant frame offset, we simulated an intermittent frame offset. We ran tests with the simulated cameras dropping in and out of sync every 10th and 200th frame. As expected, the resulting estimation of position contained error, as shown in Fig 15. Similar to the constant synchronization offset simulation, the drift for intermittent synchronization error grew over time, shown in Fig 16. To note however, the drift associated with an intermittent synchronization offset was less than the drift of a constant synchronization offset, shown in Fig 17.

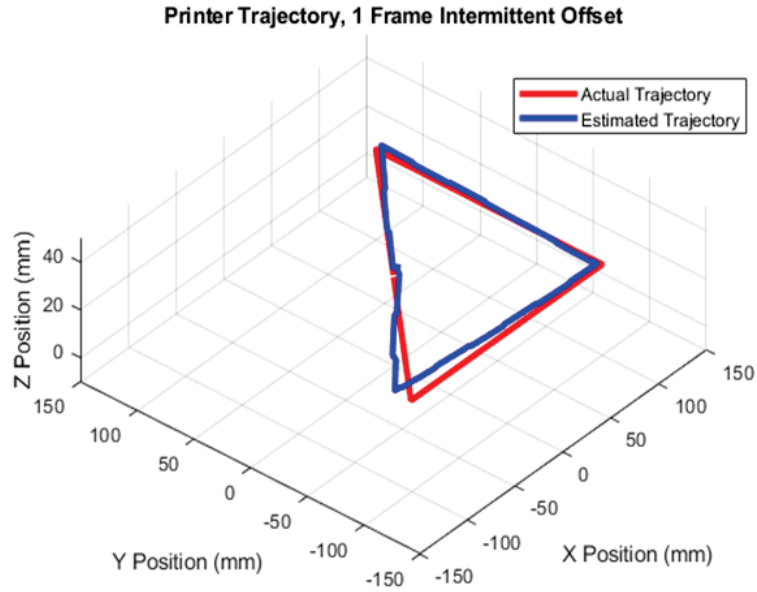


Figure 15: Estimated position of the extruder head in simulation with a 1 frame offset between camera A and camera B. Plots include the ground-truth position and estimated trajectories with the cameras dropping in and out of sync every 10th and 200th frame.

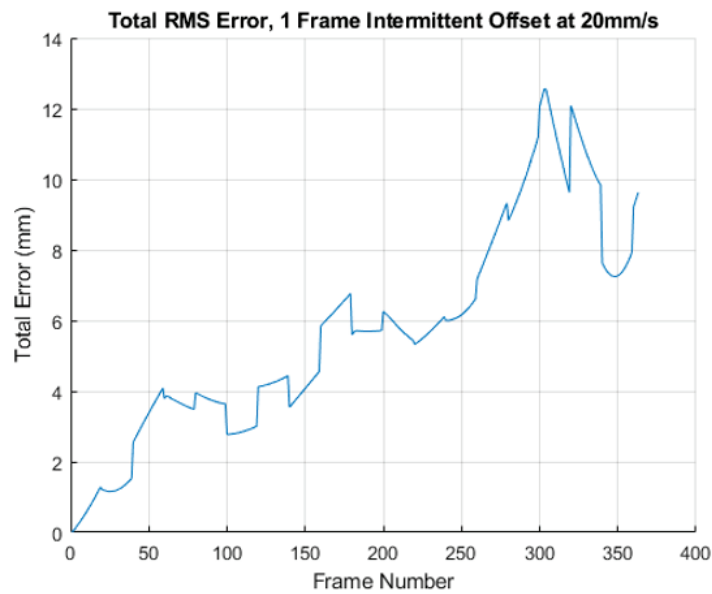


Figure 16: Position error of the simulation with a intermittent camera synchronization offset occurring every 20 frames.

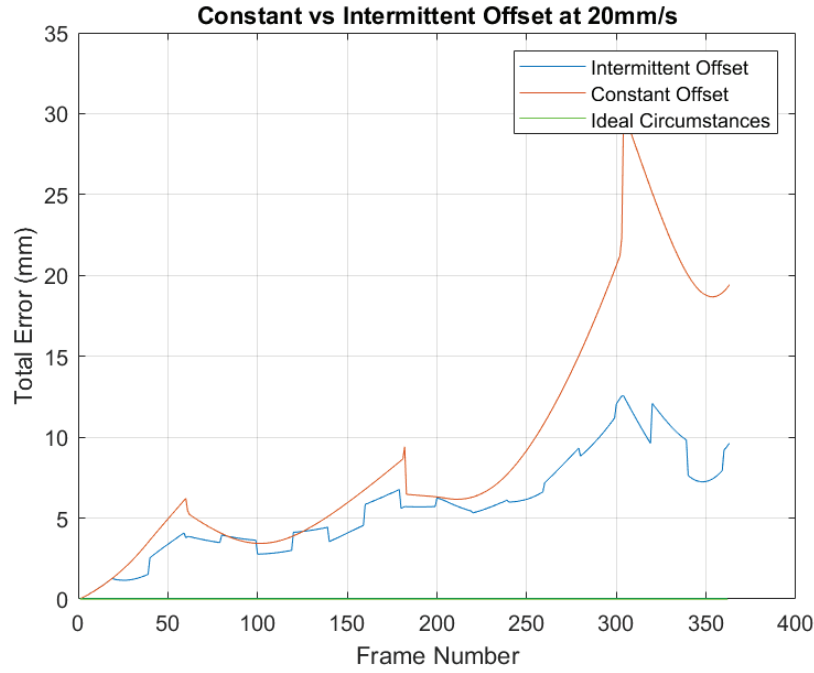


Figure 17: Comparison of estimated position error between constant and intermittent camera synchronization error.

In analysing the results of the simulation, we identify that camera synchronization offset is a source of growing error for trajectory reconstruction. While errors in camera synchronization yield reconstruction errors that are too large for this application, it does not nullify the feasibility of this approach. For future investigations into feature-based reconstruction, the existence of hardware synchronized cameras can alleviate the error shown above. Further, knowledge of the asynchronous nature of a camera pair can potentially be accounted for by mathematically analyzing the results with the assumption that they will include a growing error in position. Should error grow to a point where the extruder position cannot be estimated through an entire print, the extruder can be brought to a known position at regular intervals to reset drift by altering the G-code. The degree of accuracy for trajectory estimation is still yet to be determined on the TAZ6 testbed, so the frequency at which drift needs to be reset can be adjusted as needed.

In addition to studying the affects of asynchronous camera feeds, we sought to understand the impact of mismatched features in pose estimation. Feature detection and matching is an imperfect process and erroneous feature correspondences across the stereo image pair and across instances in time are a source of error. Accounting for this in simulation involves selecting a random feature from a frame at time step T , and incorrectly pairing it to another feature from the frame at time step $T-1$. Without a means of filtering faulty matches, the presence of incorrectly matched features was devastating for the VO algorithm, illustrated in Fig. 18.

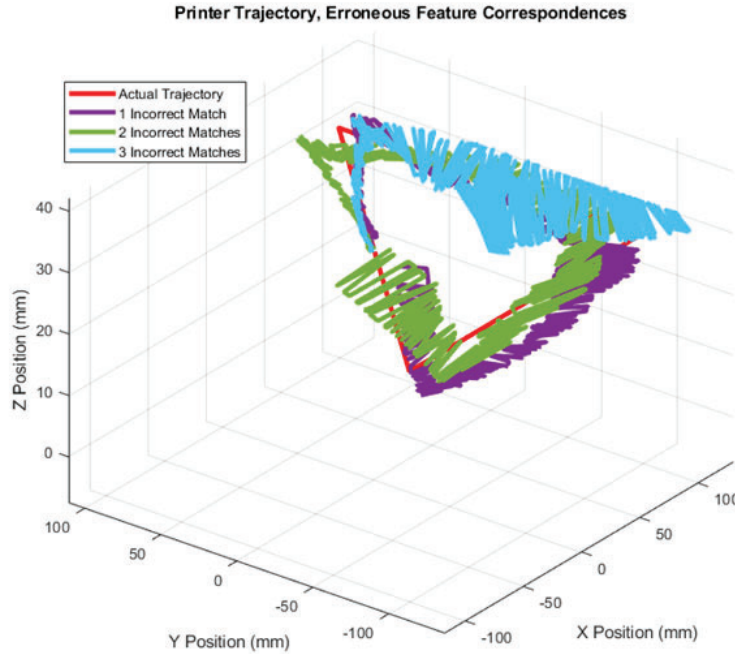


Figure 18: Estimated position of the extruder head in simulation with incorrectly matched features. Erroneous feature correspondences are severely damaging for the algorithm performance.

These results highlight the need for a robust feature correspondence verification process. The relative pose estimation algorithm is not a recurrent process and is highly affected by mismatched features. However, in moving forward this issue can be addressed; as identified in [7], only four feature correspondences are required to estimate the relative pose of the camera module in consecutive time steps. Strict, iterative methods of outlier removal, such as Random Sample Consensus (RANSAC), can be leveraged to minimize erroneous feature matches [8]. Furthermore, strict feature matching parameters can be selected to limit the matches to only those with the strongest correspondences. There are a variety of viable feature detection and description algorithms that can be investigated once we transition the feature-matching process to the physical 3D printer to include Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF) and Oriented FAST and Rotated BRIEF (ORB) [9, 12, 13].

3.1.3 Camouflage Pattern

While developing a feature-based visual odometry method, we recognized a lack of persistent features to track the motion of the camera module. Feature-based methods are highly dependent on abundant and unique features present in successive frames and across co-calibrated cameras. As discussed in Section 3.1 motion of the camera is calculated by identifying and tracking these features in successive frames through time. We discovered that the additive manufacturing environment creates limitations and opportunities for feature tracking that are otherwise not significant in other implementations. First, the material used on the TAZ6 is a constant color and changes consistency through the course of the print. Strong features are defined by sharp edges and unique formations. Thus, unique features are very difficult

to identify on the object itself during the printing process. The material is a constant color, thus the only features that can be identified are on the contour of the object. This is problematic as the contour of the object changes with each layer of the additive manufacturing process and the outline of the object is notably different for cameras of different perspectives.

While additive manufacturing presents limitations for a feature based approach, it also brings advantages. As compared to a traditional application for visual odometry such as autonomous navigation in unknown or unstructured environments, additive manufacturing is constrained to a controlled space. The TAZ6 operates in a known and controlled space, and thus, we can alter the surrounding environment to facilitate a feature based approach. For one such approach we developed a digital camouflage pattern for the print bed of the TAZ6. This camouflage pattern provides persistent and unique features that can be tracked through the course of the print so long as the print bed area stays visible from the cameras attached to the extruder. The final iteration of this pattern is visible in Fig. 19. We elected to use red, blue and white for the color scheme; these colors are distinct from the color of our printing material and are otherwise not present in the printing environment.

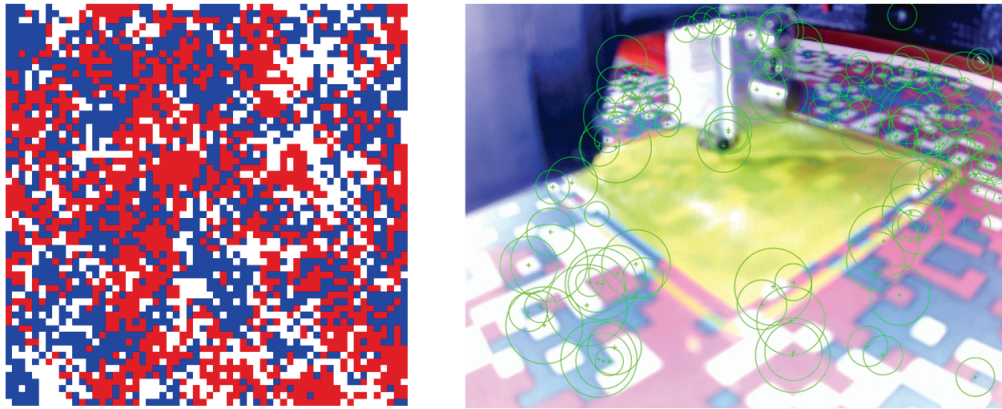


Figure 19: Left: Final iteration of the print bed camouflage pattern. Right: 100 strongest SURF features detected within a camera frame, represented by green points illustrating positions, and circles summarizing feature properties. Note the density of features detected on the printbed pattern vice the object itself.

3.1.4 Discussion

This research with the MATLAB simulation proves the feasibility of this feature-based method. However this approach is subject to limitations associated with camera synchronization, reliable feature detection, and reliable feature matching. As identified in Section 3.1.1 the feature-based visual odometry method is a *relative* estimate of pose. Because cumulative motion is used to estimate position, there will be an inherent growth in position error. While this error is manageable under ideal circumstances, error associated with the feature matching process is more troublesome. As identified in Section 3.1.2, a single mismatched feature causes substantial errors in pose estimation. These results motivate methods of obtaining absolute pose directly, discussed further in Section 4.

3.2 H.264 Motion Vectors

In investigating our feature based approach, we explored the novel idea of extracting motion vector information from the native H.264 encoding used by our USB cameras.

3.2.1 Background

H.264 encoding is a video compression standard based on block-oriented, motion-compensated coding that is used on board cameras to reduce the size of video files. In the compression process, video images are broken into small, 16x16 pixel, 8x16 pixel, or 8x8 pixel “macroblocks” [14]. To reduce the file size of the video, the “macroblocks” are tracked through successive images and given an associated motion vector at each step between frames. A visualization of the motion vectors associated with one frame in a recorded video are shown in Fig. 20. All motion vector calculations are done when the video images are recorded and saved. The motion vector information for an H.264 compressed video can be extracted using post-process open source software with a relatively light computational load [15].



Figure 20: H.264 motion vectors extracted from an example recorded video of a moving hand. The direction and magnitude of travel of the macroblocks are represented by blue arrows.

3.2.2 Investigation

After seeing the motion vector overlay in the video shown in Fig. 20, we recognized the possibility of using these motion vectors to estimate relative motion of the extruder and camera module. These motion vectors resemble what is considered an optical flow in the field of computer vision. The motion vectors contained within H.264 encoded video include both a direction and magnitude, thus, we believed that a method could be developed to aggregate the direction and magnitude of these motion vectors to return relative motion of the extruder and camera module. To evaluate the approach in a controlled environment, we created a simulation in MATLAB to record videos with controlled motion. The simulation was a video of a square moving across backgrounds of different colors. We then compressed the recorded videos with the H.264 codec to examine the associated motion vectors. The resulting motion vector output was

inconsistent. Some frames would include 20 motion vectors, while others would only include four. Further, the magnitude of the motion vectors varied drastically from frame to frame.

We concluded that the images created in simulation lacked sufficient texture, or difference in the environment for the codec to identify and track specific regions. The simulated videos were created with only one color for the square, and one color for the background. To solve this issue, used the same digital camouflage as described in Section 3.1.3. As shown in Fig. 21, this drastically improved the quantity of motion vectors returned, placing the simulated video on par with the recorded video shown in Fig. 20.

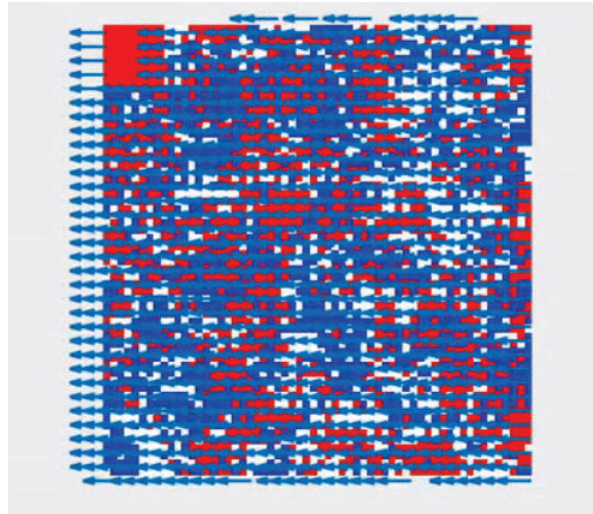


Figure 21: H.264 motion vectors extracted from a simulated video of the digital camouflage pattern moving right to left across the screen. The direction of travel and magnitude of the motion are represented by blue arrows.

3.2.3 Results and Discussion

While we were able to establish a baseline feasibility for this approach, the motion vectors returned from the simulated video yielded magnitudes that lacked sufficient consistency to establish a sound estimation of motion. To make use of this motion vector information would require further investigation in the H.264 calculation process, and does not lend itself to the primary research goals of this Trident Project. However, this is not to say that the investigation into this process was not worthwhile. A developed H.264 method could yield a supplemental source of information for other motion tracking algorithms with a negligible increase to computational load. Further, this H.264 investigation prompted the development of a refined camouflage pattern that has become a significant piece in our other machine vision methods, discussed in Sections 4.2 and 5. The pattern provides a “feature-rich” environment compared to the standard print-bed, with increased feature density and an abundance of unique features that can be used for absolute position estimation.

4 Absolute Pose Estimation

As compared to relative pose estimation, absolute pose estimation involves solving for the position of the extruder directly from a single image. The process is not a result of cumulative motion and as a result is not impacted by growing position error. We have identified two methods capable of absolute position estimation: Perspective-n-Point (PnP) algorithms and machine learning.

4.1 Background Patterns and PnP

Following our implementation of a camouflage pattern for VO, we sought to develop a pattern that enables the estimation of pose directly using a PnP approach. PnP is the process of estimating the 3D pose of a calibrated camera by identifying a set of known features in a camera image. The general process for absolute pose estimation from PnP is as follows: a set of key-points are placed at known locations in three dimensional space, the points are identified from a camera image, the position of the camera is estimated mathematically given the camera intrinsics and positions of the features in 3D space [16].

4.1.1 Checkerboard/Grid Pattern

Beyond adding features with a camouflage pattern, we sought to extract more information from a controlled background environment. As discussed in Section 3.1.2, synchronizing the two cameras across time proved to be a difficult task without additional hardware. Thus we explored other methods of motion estimation and tracking. A color grid with a known pattern was the next step in this development. An image of the pattern template and the pattern on the TAZ6 printbed are shown in Fig. 22. With a priori knowledge of the color grid pattern, features can be tracked in known world coordinates. The features are identified and referenced against the color grid template. With knowledge of the color grid position relative to the print bed frame and the location of a feature in the color grid frame, the position of a feature in the print bed frame can be resolved. As a result, motion estimation can be done successfully with one camera, proposing an alternative estimation method that uses the same hardware setup and addresses camera synchronization issues.

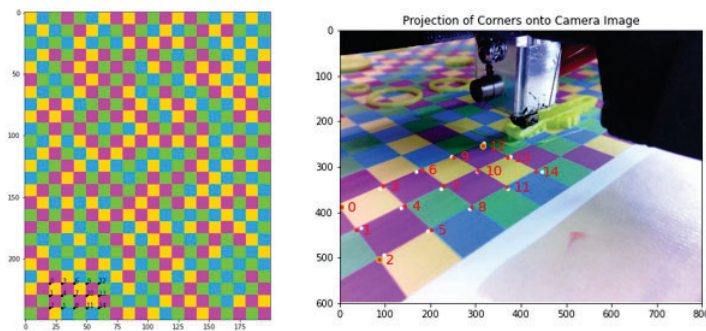


Figure 22: Left: Digital color grid template. Right: Color grid template on the printbed of the TAZ6.

4.1.2 AprilTags

Adopting AprilTags was the next step towards increasing the information density in the printing environment. AprilTags is visual fiducial system used in a wide variety of tasks including augmented reality, camera calibration and mobile navigation [17]. AprilTags is conceptually very similar to QR codes; they are a two dimensional bar code consisting of black and white squares. However, in comparison to QR codes, AprilTags are intended to encode far less information - typically less than 16 bits [17]. This enables an AprilTags system to be detectable at long ranges and run on low-grade processors at real-time speed. An example of an AprilTag pattern and the implementation of an AprilTag printbed are shown in Fig. 23. AprilTags provide even more information in comparison to the color grid pattern in application to the additive manufacturing motion estimation problem. Each AprilTag is unique, given a specific label and provide a coordinate frame of reference. Consequently, there is no need for comparison to a template grid to identify the position of a tag. This has several advantages. First, the computational resources required to identify the position of an AprilTag are notably lower in comparison to the color grid. Further, the AprilTag system is flexible and can be adapted for specific prints. It is not required to keep AprilTags in a grid formation and thus, a pattern can be designed for a specific .stl file to provide the greatest information density without loss to occlusion.

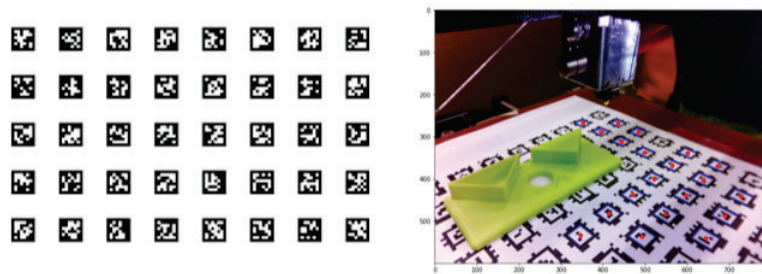


Figure 23: Left: Example AprilTag template grid. Right: AprilTags on the print bed of the TAZ6. Identified AprilTags are marked with a blue square and red dot.

4.1.3 Discussion

The three print bed pattern methods discussed Sections 4.1.1 and 4.1.2 all have their advantages and respective applications. The camouflage pattern is a simple approach to increasing the density of information in stereo images for a feature-based approach. In practice, this can be achieved by simply applying unique marking to the print bed. There is no need for a specific structure, so long as the marking are unique and identifiable. This pattern is best suited to assist in dual-camera approaches. The color pattern approach is the most complex implementation of the print bed patterns, but can be implemented with a single camera. The color pattern approach requires an additional algorithm to search the pattern template for a matching feature. While we explored this approach, a similar outcome could be achieved with AprilTags. Like the color pattern approach, AprilTags can yield features relative to the world frame with a single camera. In comparison to the digital camouflage and a visual odometry approach, AprilTags enables an estimate of extruder position instead of extruder motion. As a result, the primary advantage of AprilTags and estimating position directly (instead of summing the relative motion of the printer) is the removal of drift. As discussed in Section 3.1.2, small errors in each estimate of motion compounded to

grow error over time. The primary drawback of the April Tag approach is the in-depth set up and specific AprilTag code required.

4.2 Machine Learning

In the original proposal for this project, we identified two vision based approaches towards estimating the trajectory of a AM system: feature-based visual odometry and machine learning. Informed by our initial research, we recognized that machine learning may offer alternative methods towards estimating the absolute position of the extruder from camera module images.

4.2.1 Background

Machine learning is a form of computational statistics where a task is accomplished by an algorithm that is not provided with explicit information about the task itself. Machine learning algorithms are developed with sets of data to accomplish tasks based on patterns and inferences learned in the training process. A machine learning approach for position estimation involves constructing a convolutional neural network that takes images as input, and returns the pose of the extruder as an output.

Artificial neural networks are composed of various connected nodes that send information to each other. These nodes are typically divided into layers, which can perform different transformations on their inputs. Artificial neural networks begin at the input layer, travel across intermediate hidden layers, then reach the output/final layer. The artificial neural network is built by training the nodes and layers on labeled truth data. The training process establishes weights that are applied to connections between nodes. These connections determine what information from the input has a high correspondence to the output, and thus, creates the artificial neural network. A convolutional neural network is a type of network ANN that is typically applied to analyze visual imagery. Convolutional neural networks involve abstracting the input images by passing them through layers that convolve the input by applying a kernel. Then, the dimensions of the convolutional layer output are reduced with a pooling layer. The basic structure of a convolutional ANN is shown in Fig. 24.

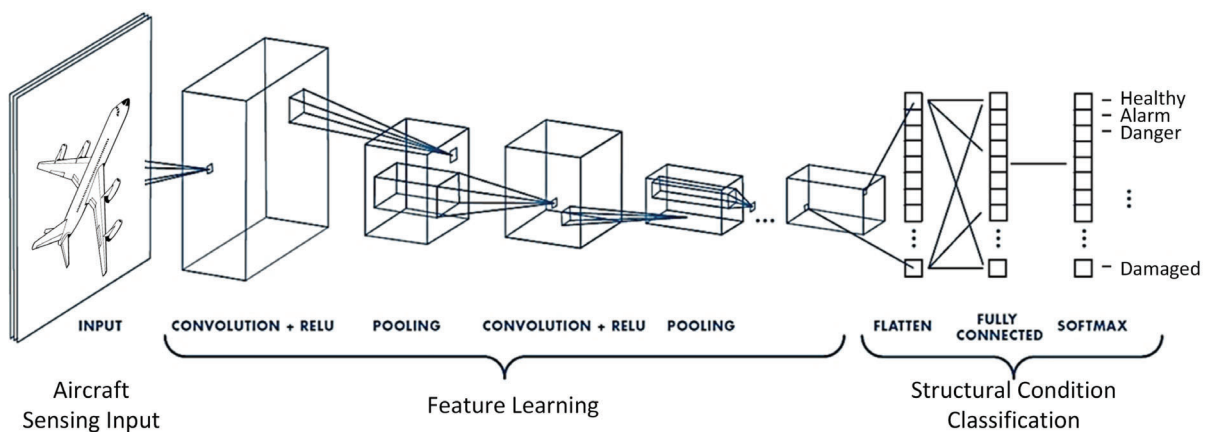


Figure 24: Structure of a simple artificial neural network [18].

A machine learning approach has several advantages. First, constructing an artificial neural network for visual odometry does not require any knowledge of the math associated with traditional, feature-

based visual odometry. In fact, the network is built with no implementation of the process outlined in Section 3.1.1. Because of this, a machine learning algorithm can establish powerful unrealized relationships without explicit knowledge of visual odometry. Additionally, the artificial neural network that we are constructing returns the absolute position of the printer instead of printer motion. This avoids the issue of growing error that was encountered when calculating position using an estimation of motion in the feature-based approach. When we originally proposed a machine learning approach, we anticipated the output to be an estimate of motion. However, the development of the digital camouflage background has enabled this change from motion estimation to absolute position estimation.

With these advantages come drawbacks. First, depending on the structure of the network the number of parameters can be extremely large, exceeding tens of thousands. This being said, general assumptions were made in the construction of the network. Instead of building from the ground up, we expect to use the framework of existing, successful networks and make modifications to increase performance from there. Further, it is next to impossible to determine what calculations are being made within the network itself. Because of this, it can be extremely difficult to systematically improve the performance of the algorithm without the knowledge of where it is making errors or incorrect interpretations. Finally, artificial neural networks are a very computationally heavy approach. A large amount of data is required to build the network and a significant amount of computational power is required to execute the process. To address this issue, all of our tests are executed on a high-end HP laptop with an NVIDIA Quadro RTX3000 Graphics Processing Unit (GPU) specifically designed to work with machine learning algorithms.

4.2.2 General Process

We created a convolutional neural network for AM pose estimation to test the feasibility of a machine learning approach. This machine learning algorithm is implemented in python using the TensorFlow open-source software library and the Keras software interface. The network is constructed with multiple convolution and pooling layers. This neural network is built with three 2D convolution layers with 32 filters and a kernel size of 3x3, 2 max pooling layers with “same” padding and a pool size of 3x3, a global average pooling layer, and a final dense layer to estimate x,y coordinates. The specifics for our developed network are given in Fig. 25.

Model: "functional_1"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 75, 150, 3)]	0
rescaling (Rescaling)	(None, 75, 150, 3)	0
conv2d (Conv2D)	(None, 73, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 25, 50, 32)	0
conv2d_1 (Conv2D)	(None, 23, 48, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 7, 16, 32)	0
conv2d_2 (Conv2D)	(None, 5, 14, 32)	9248
global_average_pooling2d (GlobalAveragePooling2D)	(None, 32)	0
dense (Dense)	(None, 2)	66

Total params: 19,458
 Trainable params: 19,458
 Non-trainable params: 0

Figure 25: Technical structure of the convolutional neural network used for preliminary testing.

The input to the system is a single image comprised of two separate images of the camouflage print pattern attached together. This preliminary network is trained on images from an entire print with encoder position information as truth data. The general implementation for a neural network approach goes as follows:

1. Capture a stereo image pair from the TAZ6 extruder camera module.
2. Crop the camera images to remove non-essential information and reduce input information size. Combine the cropped images into a single input image.
3. Feed the input image into the trained convolutional neural network. The network returns the estimated position of the extruder relative to the printbed.

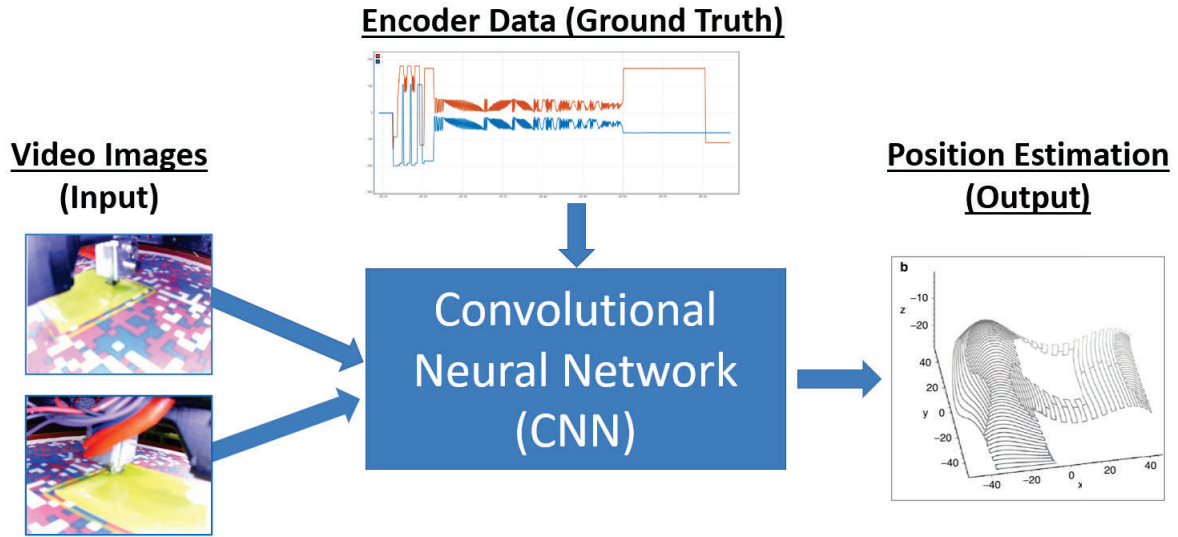


Figure 26: Visualization of the convolutional network used for testing.

4.2.3 Results and Discussion

Our preliminary tests show promising results. A convolutional neural network applied to one data set from a recorded print shows a strong correlation between the actual and estimated position of the extruder, shown in Fig. 27.

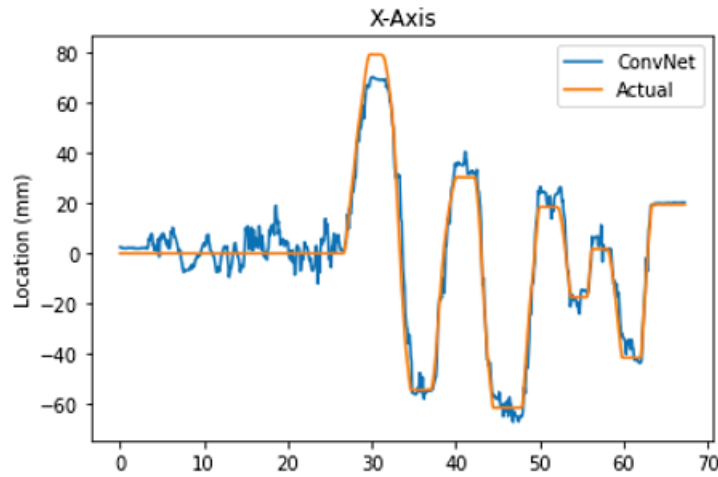


Figure 27: Preliminary performance of a convolutional neural network applied to the visual odometry problem

While the results appear promising, in developing this approach we recognized limitations that affect the implementation of machine learning to the AM pose estimation problem. First, as discussed in Section 4.2.1 the artificial neural network structure is non-deterministic – meaning that there is no way of knowing what information the network is using to calculate the output. While the network provided strong results in one test, we recognize that it would be extremely difficult to replicate the same outcomes

under a different set of environmental circumstances, such as a change in lighting environment or printing material. Consequently, a machine learning approach to pose estimation does not lend itself to one of our primary research objects in developing a means of AM product verification: a generalized approach that can be applied to a variety of AM machines and environments. This raises another concern, to develop a generalized machine learning approach to pose estimation would require an extensive amount of training data. For proper performance the training data would need to include images with as many different environmental variables as possible. To further complicate the matter, being that there is no way of determining what specific information the network is training on we can never be certain that all environmental considerations are accounted for. As a result, we do not see collecting large amounts of data for an CNN algorithm as a means to our research goals.

It is important to note that the investigation into a machine learning process was worthwhile. While a machine learning approach to pose estimation is not feasible for this project’s research goals, it is an area of potential future development for other research.

5 Image Comparison and Template Matching

In developing our feature-based visual odometry method, we recognized factors that may limit its application to physical hardware, as discussed in Section 3.1.2. This project began with the goal of verifying additive manufacturing products with machine vision. We recognize that there are multiple methods towards this end, and our research has been informed by the developments we have made over time. This being said, we explored image comparison and template matching as an alternative to a feature-based visual odometry approach. Template matching is a field of machine vision that is well established in the medical field for X-Ray and MRI analysis [19, 20]. Template matching is the process of finding parts of a captured image that match an expected template image. Applications include searching medical imaging for tumors or defects.

Instead of tracking the motion of the extruder over the course of the print and comparing it to the intended trajectory, as in the visual odometry method, a template matching approach involves taking a picture of the part and comparing it to what it should look like. In practice, this involves capturing an image at set intervals throughout the printing process and using a comparison/matching metric to compare the “real” image to a simulated render of the part. It is important to note that in order to create a simulated render of the part for comparison, the position of the extruder must be known. As a result, the image comparison and template matching method builds upon the research outlined in the previous sections.

This section will be using the following coordinate frames throughout. Frame m refers to the 2D camera matrix frames, in pixels. Frame c refers to 3D coordinates relative to the camera. Frame b refers to the print bed coordinates, referenced from the homing corner of the printer. Frame p refers to the coordinate frame relative to the printed part, referenced from the center of the bounding box that surrounds the part. Frame s refers to the 3D coordinate frame of the simulated part.

5.1 General Process

A template matching/image comparison method to product verification entails the following general steps:

1. Capture an image of the part at a specific instance in the printing process. This can be completed

at every layer, or at a specific interval dictated by a trade-off between increased printing time and frequency of verification.

2. Segment the the part image I_p from the background environment using a color threshold or pixel-wise posterior [21].
3. Convert the .gcode file of the part to an .stl file using Blender [22]. Blender is a free and open source 3D creation suite.
4. Estimate the pose of the part on the TAZ6 print bed relative to the camera H_p^c .
5. Use the pose estimate, .stl file, and MATLAB to create a simulated .stl image I_s .
6. Compare the part I_p and .stl I_s images to identify defects and recover H_p^s
7. (Optional) Recursively refine object pose estimate to reduce comparison error.

5.2 Image Capture and Segmentation

Images of the part are captured from the camera module attached to the extruder. However, unlike the feature-based visual odometry method only one camera is required for image comparison. To note, for future developments it may be beneficial to run image comparison methods in tandem across two cameras of different perspectives. This would enable a cross-comparison of error detection should one camera identify a defect that the other camera cannot see.

The image capture process includes filtering with Gaussian smoothing kernels and segmentation using a color threshold. Background noise is removed by applying another smoothing filter and labeling the connected components, keeping the blob of the largest area. An image of a part on the print bed pre and post segmentation is shown in Fig. 28.

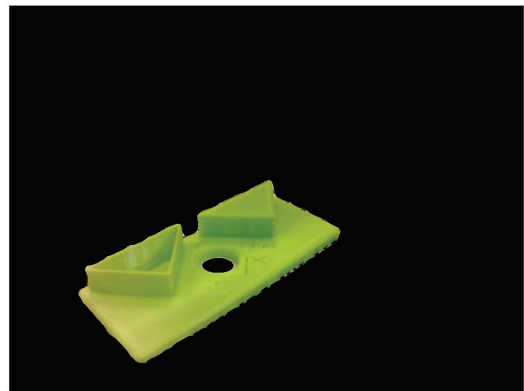
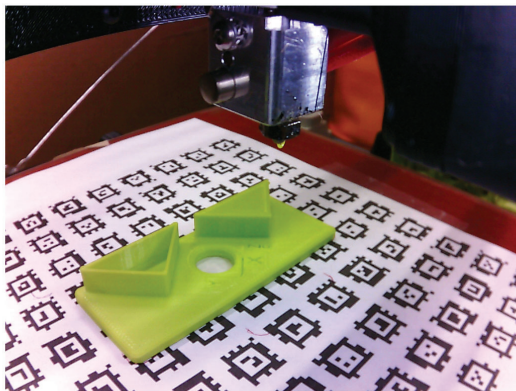


Figure 28: Left: Image of sample product on the TAZ6 printbed. Right: Image of sample product after color threshold and segmentation.

5.3 Initial Pose Estimation

Camera intrinsics A_c^m and an initial estimate of the object pose relative to the camera H_s^c is required to create a simulated render of the .stl file. The camera intrinsics A_c^m are established with a calibration grid. We investigated two means of establishing an initial estimate of pose H_s^c — a print bed pattern method and a user defined estimate.

A full implementation of our proposed verification algorithm would use an estimate of camera pose from one of the methods described in Section 3 or Section 4. This pose estimate would be used as the initial estimate for the object render.

We developed a simple Graphical User Interface (GUI) as a proof of concept and substitute for a complete pose estimation algorithm. In practice, an operator would visually align a simulated render of the part on the print bed. Once the “real” and simulated images are aligned, a comparison metric would be used to search for defects.

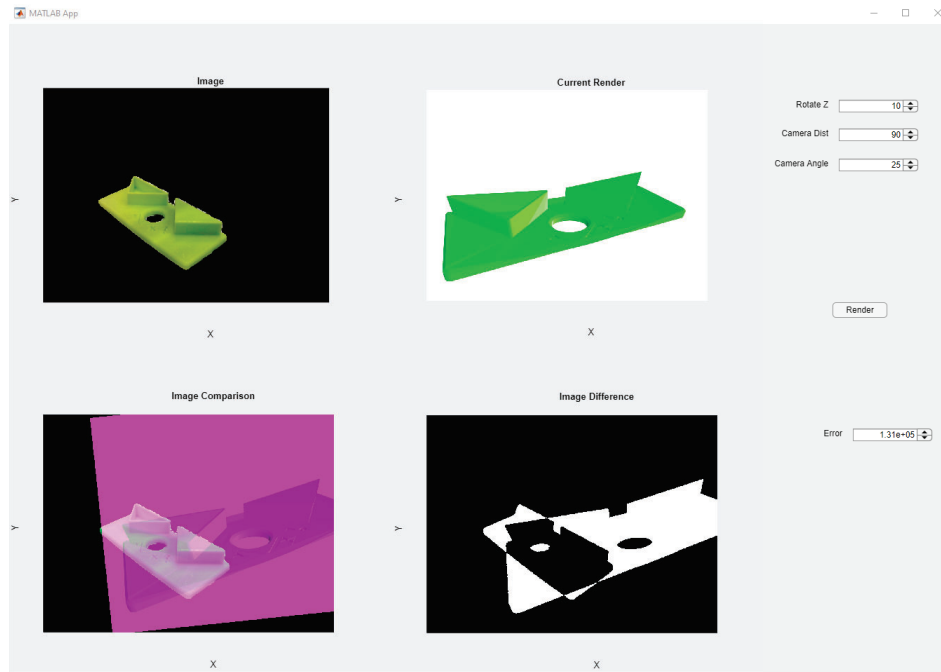


Figure 29: GUI for aligning a simulated render of a part to a captured image. The top left shows the segmented image from the test bed. The top right shows the current render of part. The bottom left shows the top two images overlaid and the bottom right shows the difference between the image contours.

5.4 Object Rendering and Visualization

With an understanding of the rigid body transformations that relate the different parts of the print bed, a 3D simulation of the part on the printer can be rendered to represent any layer in the printing process. Once a .gcode file is returned from the slicing software, it is uploaded to blender for conversion to a .stl file. Blender attempts to simulate the texture of PLA material and creates separate objects for each layer of the print. Thus, layers can selectively be deleted. This enables us to create a render of a specific object

reflective at a specific time in the printing process. Fig. 30 illustrates a simulated render of an object with and without the final layers of the print.

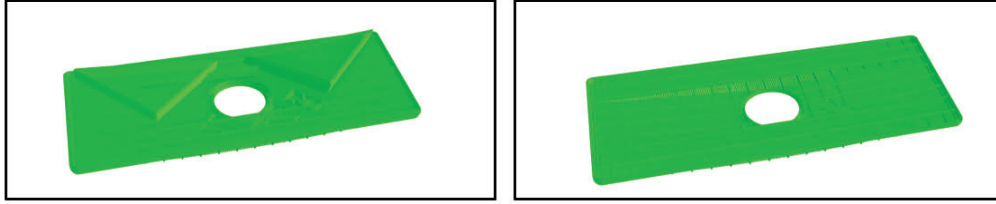


Figure 30: Left: Simulated render of complete product. Right: Render of incomplete product. Note the missing triangles on the surface of the object.

5.5 Image Comparison Metrics

We identified two primary metrics for image comparison and error detection. One is a contour comparison between the real and simulated image and the other is a comparison using the energy function defined in [23].

5.5.1 Contour Mask Method

The contour mask method involves comparing the contours of the real and simulated product to search for defects. The contour is extracted from both images using a color thresholding process. The contour binary of the real image is already found as described in Section 5.2. The same process is applied to the simulated image. An example captured image and its associated contour binary are shown in Fig. 31. An example captured image and its associated contour binary are shown in Fig. 32.

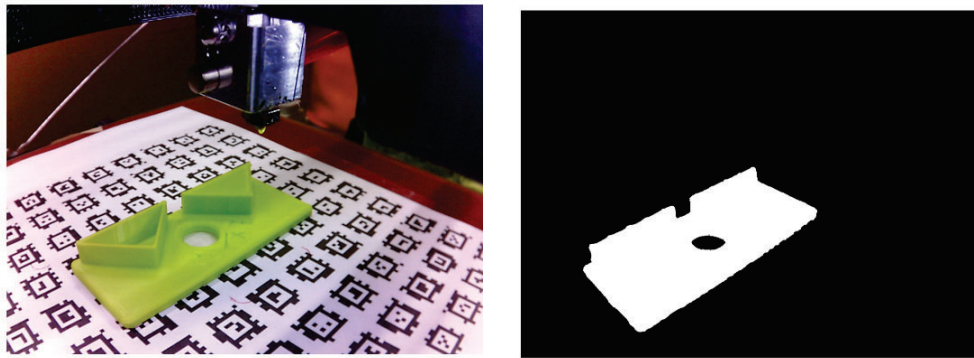


Figure 31: Left: Captured image of an AM product on the TAZ6 printbed. Right: The associated contour binary for the captured image.



Figure 32: Left: Simulated image render of an AM product on the TAZ6 printbed. Right: The associated contour binary for the simulated image.

To compute the error metric, these two binary contours are subtracted. The two images are the same dimensions and under ideal conditions should overlap exactly. Pixel Error is calculated as the sum of the positive values from this subtraction. The difference between the contours shown in Fig. 31 and Fig. 32 are shown in Fig. 33.



Figure 33: Difference between contour binaries for the physical part in Fig. 31 and the simulated render in Fig. 32.

We tested this method on a data set of 21 AM products of varying quality. Some parts printed successfully with no defects, some had minor anomalies, and others had large deformities. Results are shown and annotated in Fig. 34.

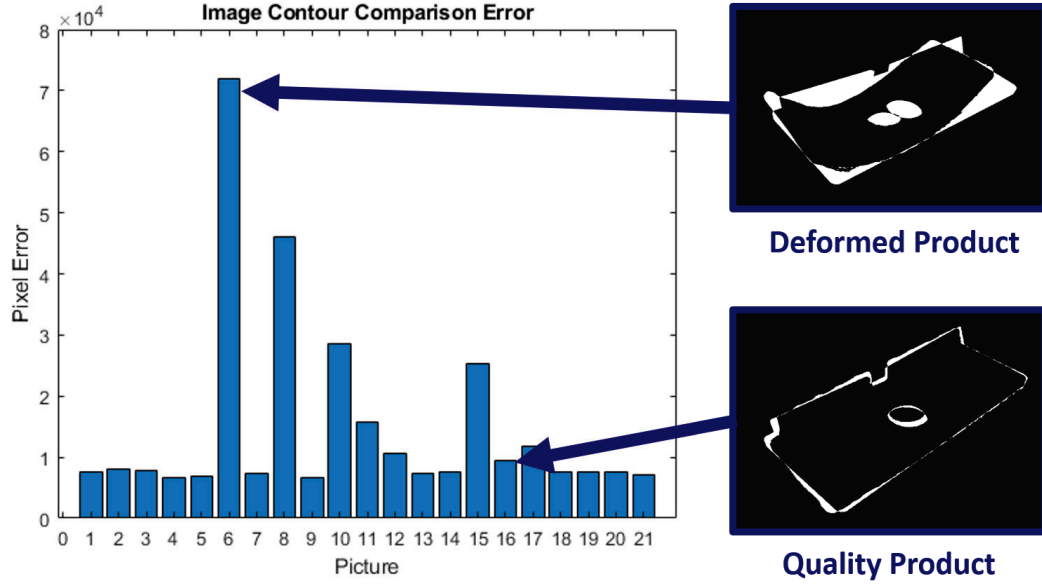


Figure 34: Binary contour results of 21 AM products of varying quality. Note that products with the high levels of pixel error can be flagged as deformed. This threshold will need to be user determined specific to the AM application.

The binary contour comparison method was successful in identifying products with large structural defects. It is important to note that error tolerance will be subjective to the manufacturing task. The user will be required to set an acceptable tolerance for their desired outcome.

While this method was successful, there are limitations that must be noted. First, the contour binary method only considers defects that alter the contour of the object. As a result, any defect that does not alter the external outline of the object cannot be accounted for because all information contained within the contour is lost. Further, this experiment was only conducted from a single camera in the stereo module. Anomalies may be difficult to detect on the backside of large objects that occlude large areas of the camera's field of view. Thus, this process can be largely improved by conducting binary comparison from two cameras at different perspectives simultaneously. This method does not require extensive computational resources and can be run in parallel. Finally, the results of the binary comparison are affected by the quality of the pose estimation. As described in Section 5.3, we used a use defined pose estimate to render an image the product. As a result, the estimate is not exact and error is present for even high quality prints. This discovery is a motivation for the implementation of recursive pose estimation, as discussed in Section 5.6.

5.5.2 Energy Function Method

An alternative to the binary contour method is to use the energy function described in [24]. This energy function is outlined and intended to be used for pose estimation. The form of the energy function is given in Eq. 3, where r_f and r_b are functions measuring the matching quality of the image pixels with the foreground and background models of the image. H_e represents the Heaviside function defined in Eq. 4, where s determines the pitch of the smooth transition (chosen as $s = 1.2$). The signed distance

function $\Phi(\mathbf{x})$ is found by taking the euclidean distance of every pixel from the contour of the image; with negative values in the foreground and positive values outside (in the background).

$$E(\Phi) = \int_{\Omega} H_e(\Phi)r_f(\mathbf{x}) + (1 - H_e(\Phi))r_b(\mathbf{x})d\Omega \quad (3)$$

$$H_e(\Phi) = \frac{1}{\pi}(-\arctan(s \cdot \Phi)) + \frac{\pi}{2} \quad (4)$$

While the authors of [24] use the energy function for pose estimation, the problem we are trying to solve is very similar. For pose estimation, [24] attempts to minimize the error returned from the energy function by continuously changing the pose of the simulated render to match the actual object. We utilize the energy function to alternatively return an error value that reflects how well the expected render of the product matches what is captured on the print bed.

5.6 Recursive Pose Estimation

As discussed in section 5.5.1, the accuracy of an image comparison method is heavily dependent upon the estimate of the product's pose on the print bed. In testing our template matching methods, we used a GUI to align the test part. While this was sufficient to run experiments and return results, the implementation of a recursive pose estimation process is the next step in the development of a complete verification algorithm.

5.6.1 General Process

From the literature we identified several possible methods to address this issue. While methods such as [24] and [23] are not specially designed for additive manufacturing processes, we are successfully able to adapt methods from these algorithms to implement an AM pose estimation process. For the sake of testing recursive pose estimation we elected to work entirely in simulation to remove errors induced by the noise in a captured image. Furthermore, we selected a pyramid .stl for simplicity and run time efficiency. A visualization of the pyramid is shown in Fig. 35.

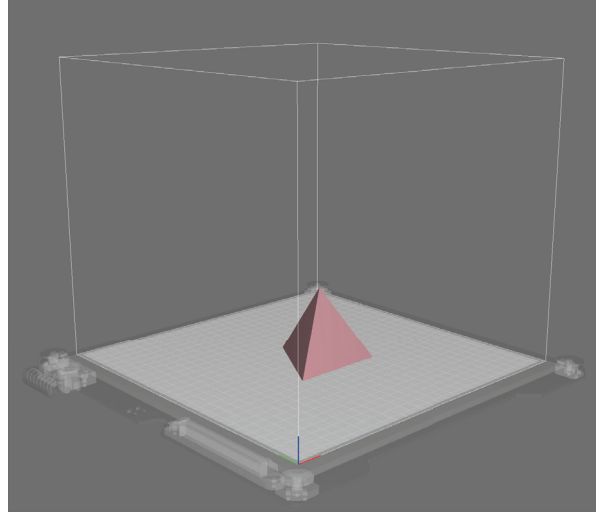


Figure 35: Pyramid .stl file used for recursive motion estimation experimentation.

The general process of our developed method, based on methods from [24] and [23], goes as follows:

1. A simulated image of the product at the current state of the print is rendered. The foreground mask is calculated using a color threshold function, and the contour of the product is found by detecting the edge of the foreground mask. For this explanation the product will be a pyramid, as depicted in Fig. 36.

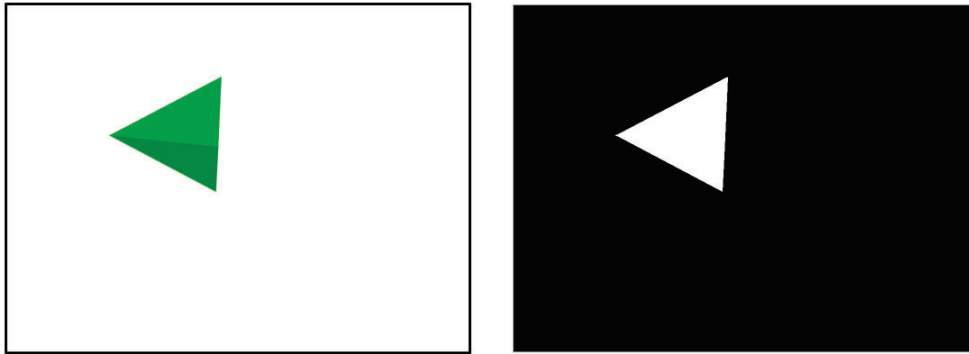


Figure 36: Left: Simulated render of an pyramid AM product Right: The associated foreground binary of the pyramid render. White pixels denote the foreground and black pixels denote the background.

2. The shape kernel, or the signed distance function $\Phi(\mathbf{x})$ is found by taking the euclidean distance of every pixel from the contour of the image; with negative values in the foreground and positive values in the background, as given in Eq. 5. C being the product contour line and x representing a pixel's position.

$$\Phi(\mathbf{x}) = d(\mathbf{x}, C) = \min_{c \in C} \|\mathbf{c} - \mathbf{x}\|_2 \quad (5)$$

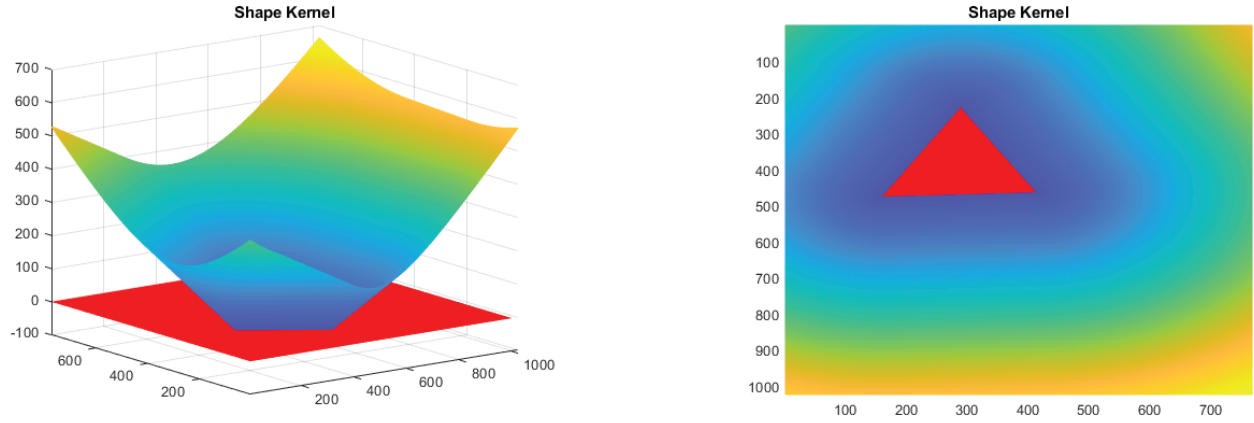


Figure 37: Visual representation of the product shape kernel $\Phi(\mathbf{x})$ from two different perspectives. The red plane serves to represent the x-y plane. Note that all red area in the right image correspond to the object foreground mask.

3. The shape kernel is passed through a smoothed Heaviside function H_e of the form given in Eq. 6, where s determines the pitch of the smooth transition (chosen as $s = 1.2$). A visual representation of the Heaviside function is shown in Fig. 38

$$H_e(\Phi(\mathbf{x})) = \frac{1}{\pi}(-(s \cdot \Phi(\mathbf{x})) + \frac{\pi}{2}) \quad (6)$$

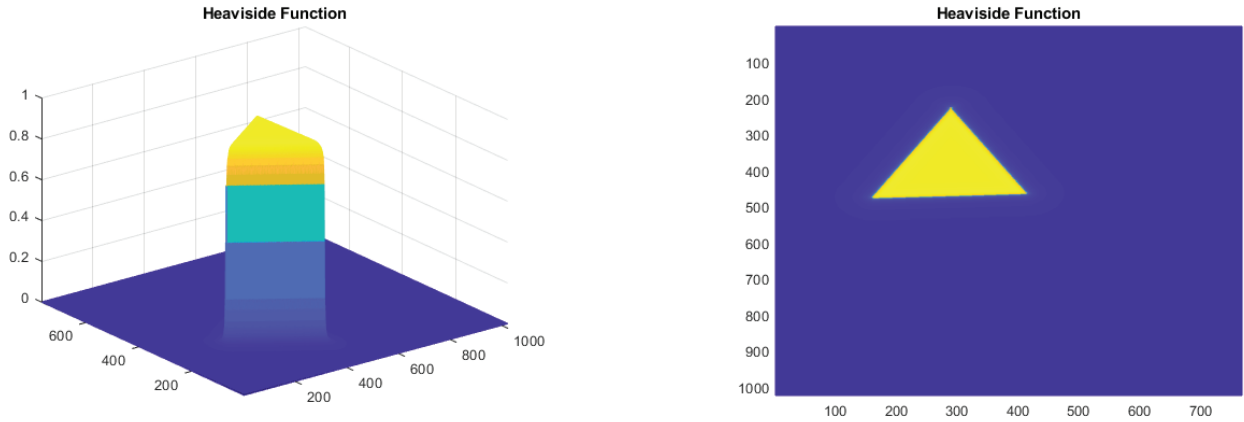


Figure 38: Visual representation of the product Heaviside function result $H_e(\Phi(\mathbf{x}))$ from two different perspectives. Note that the yellow area in the right image corresponds to the object foreground mask.

4. Global appearance models for the foreground $P_f(\mathbf{x})$ and background $P_b(\mathbf{x})$ of the captured image are created from the pixel color distributions for these respective regions Ω_f and Ω_b , as given in Eq. 7 and Eq. 8. η_f and η_b are summations of the Heaviside function, given in Eq. 9 and Eq. 10.

$\mathbf{y}$ represents the RGB values for a respective pixel and M represents the respective foreground or background membership probability. Visual representations of $P_f(\mathbf{x})$ and $P_b(\mathbf{x})$ are shown in Fig. 39. Because we chose to validate the pose estimation in simulation, $P_f(\mathbf{x})$ and $P_b(\mathbf{x})$ return homogeneous values for pixels in the same region. An image of a product captured on the printed bed would likely not have yielded this clean result.

$$P_f(\mathbf{x}) = P(M_f|\mathbf{y}) = \frac{P(\mathbf{y}|M_f)}{\eta_f P(\mathbf{y}|M_f) + \eta_b P(\mathbf{y}|M_b)} \quad (7)$$

$$P_b(\mathbf{x}) = P(M_b|\mathbf{y}) = \frac{P(\mathbf{y}|M_b)}{\eta_f P(\mathbf{y}|M_f) + \eta_b P(\mathbf{y}|M_b)} \quad (8)$$

$$\eta_f = \sum_{\mathbf{x} \in \Omega} H_e(\Phi(\mathbf{x})) \quad (9)$$

$$\eta_b = \sum_{\mathbf{x} \in \Omega} 1 - H_e(\Phi(\mathbf{x})) \quad (10)$$

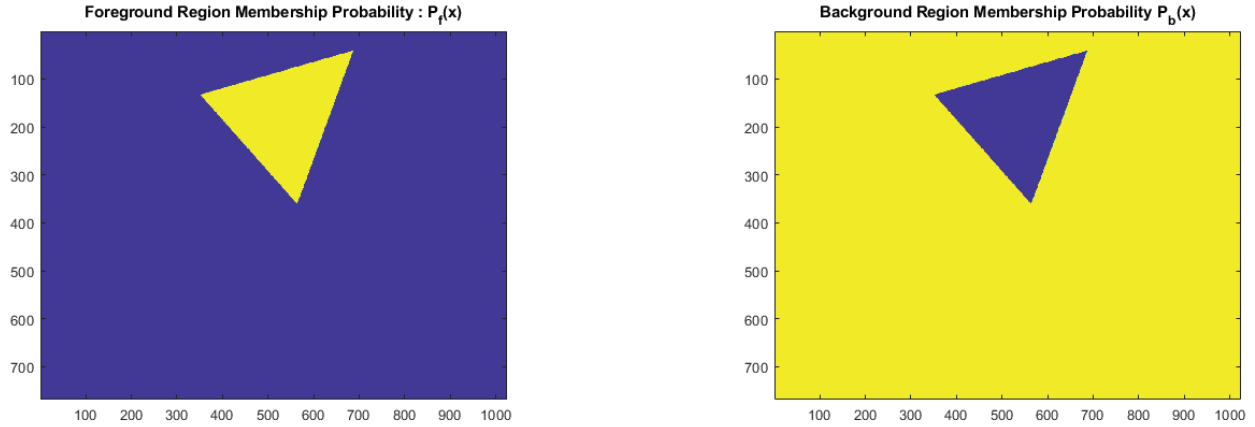


Figure 39: Probability that a given pixel value lies in the foreground or background of the image. Given that this process is run in simulation ideal, homogeneous outputs are returned.

5. An energy function $E(\xi)$ of the form given in Eq. 11 is applied. The energy function E represents the matching quality between the captured image and the simulated image (where ξ represents the pose estimate used to create the simulated image). A visual representation of the energy function is presented in Fig. 40.

$$E(\xi) = - \sum_{x \in \Omega} \log(H_e(\Phi(x(\xi)))P_f(x) + (1 - H_e(\Phi(x(\xi))))P_b(x)) \quad (11)$$

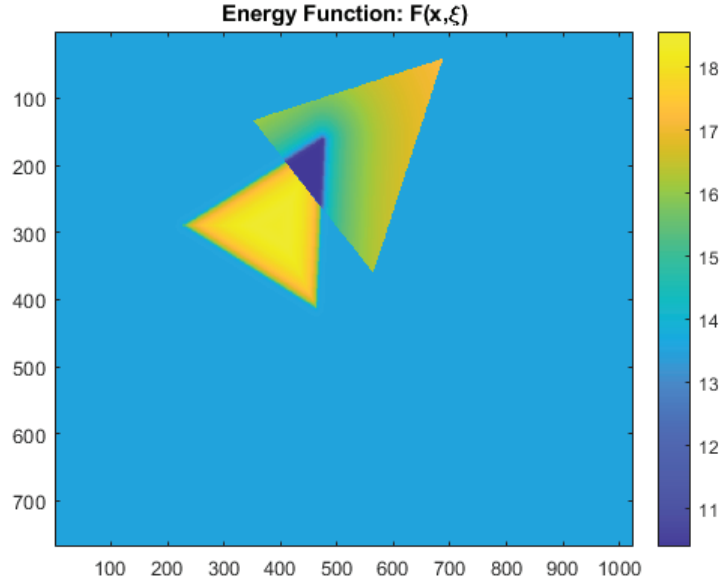


Figure 40: Energy Function

6. The simulated product's pose ξ is updated recursively to minimize the value returned from the energy function. As detailed in [23], a per-pixel Jacobian term J can be used to solve for an optimal update step to the pose parameters $\Delta\xi$:

$$\Delta\xi = -\left(\sum_{x \in \Omega} \psi(\mathbf{x}) J^\top J\right)^{-1} \sum_{x \in \Omega} J^\top \quad (12)$$

Where the Jacobian term J is defined as:

$$J = \frac{\bar{P}_b(x) - \bar{P}_f(x)}{H_e(\Phi(x))(\bar{P}_f(x) - \bar{P}_b(x)) + \bar{P}_b(x)} \delta_e \frac{\partial \Phi(x(\xi))}{\partial \xi} \quad (13)$$

Where $\delta_e(\Phi(\mathbf{x}))$ represents the Dirac delta function:

$$\delta_e(\Phi(\mathbf{x})) = \frac{s}{\pi \Phi(\mathbf{x})^2 s^2 + \pi} \quad (14)$$

To solve Eq. 13 the derivative of the signed distance transform $\frac{\partial \Phi(x(\xi))}{\partial \xi}$ must be computed [21]:

$$\frac{\partial \Phi(x(\xi))}{\partial \xi} = \begin{bmatrix} \frac{\partial \Phi}{\partial \mathbf{x}} & \frac{\partial \Phi}{\partial \mathbf{y}} \end{bmatrix} \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{X' f_x}{(Z')^2} \\ 0 & \frac{f_y}{Z'} & -\frac{Y' f_y}{(Z')^2} \end{bmatrix} \begin{bmatrix} 0 & Z' & -Y' & 1 & 0 & 0 \\ -Z' & 0 & X' & 0 & 1 & 0 \\ Y & -X' & 0 & 0 & 0 & 1 \end{bmatrix} \quad (15)$$

Where $[X', Y', Z']$ are the 3D coordinates of each pixel relative to the camera frame. $[X', Y', Z']$ is found by computing a depth map for the rendered product to capture the position of each pixel in 3D coordinates. A visualization of the depth map for the pyramid object is shown in Fig. 41.

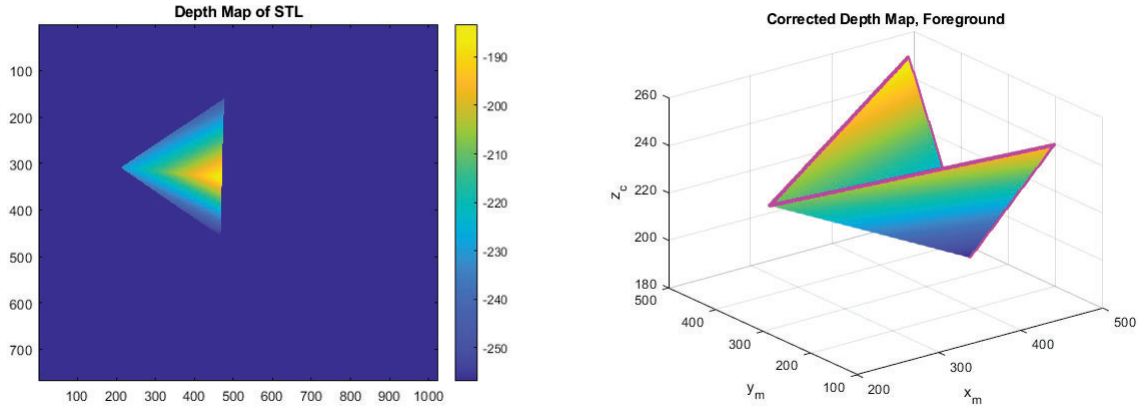


Figure 41: Left: 2D representation of the product depth map. Right: 3D representation of the product depth map.

Because there is no value for depth for pixels in the background of the image (not on the object) they must be assigned directly. The depth for pixels in the background of the image is selected as the depth value for the closest foreground pixel. A visualization for the “corrected” depth map D_c is shown in Fig. 42.

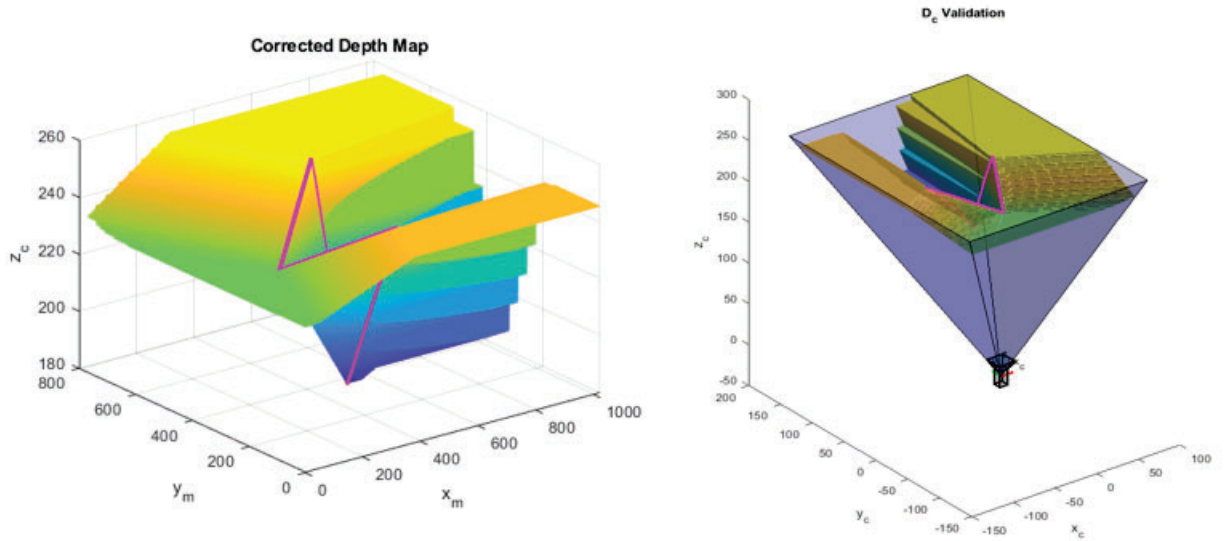


Figure 42: Left: 3D representation of the product depth map with background depths assigned to the value of the closest foreground pixel. Right: Reprojection of the depth map from the camera's perspective.

Finally, with this information Eq. 12 is solved for the update step. This process is run continuously until the energy function is driven to a minimized value. As discussed in Section 5, this pose estimation

algorithm will lower image comparison error by providing a more accurate render of the product. This will increase the accuracy to which the two images can be compared, and lower the threshold for defect detection.

6 Conclusions

This project successfully investigated a variety of computer vision-based methods of AM product verification. AM is an emerging technology that has the potential to forever change how the Department of Defense approaches supply chain management. With this technology comes the need for cost-effective and robust means of product verification.

Motion estimation has been demonstrated as a viable means to monitor the AM process for errors and defects. In this research, pose estimation from VO was investigated and implemented in simulation and preliminary experimentation. We ran a variety of experiments from the MATLAB simulation to understand the impact of a variety of factors on algorithm performance, ran extensive experimentation verifying the time synchronization limitations of USB cameras, and experimentally verified feature recognition methods on our system testbed. From this research, we arrived at the conclusion that relative position estimation (position derived from the cumulative motion of the extruder) for AM verification is possible, but less desirable when compared to absolute methods of pose estimation. The simulation results point to the issue of compounding error; because the pose of the extruder is estimated as a culmination of prior estimates, small errors in the VO process compound over time. In this work, we identified two sources of error in the VO process: errors associated with asynchronous cameras, and errors associated with mismatched features. The issue of mismatched features issue can potentially be addressed with more strict methods of feature matching, and time synchronization issues may be addressed in the future using methods associated with pixel velocities similar to the results derived by our H.264 analysis. Further, mathematical analysis of the results with the assumption that error is compounding may yield methods to account for anticipated error in part fidelity analysis.

The highlighted issue of growing error in pose estimation can be addressed with absolute estimation. This research demonstrated alternatives to the feature-based VO approach to include a variety of printbed patterns and a machine learning approach. The development of printbed patterns ultimately yielded the implementation of an AprilTag system, where the absolute pose of the extruder is estimated from a single camera image, with redundancy in each image. Each AprilTag pattern provides a unique frame fixed to the printbed frame. Knowing the relative location of each AprilTag frame provides two complimentary approaches to recovering the pose of the extruder: using the origin of each AprilTag frame, a PnP algorithm can be used to solve for the extruder pose [25] [26]; or the pose can be recovered using the mean estimated camera pose recovered from each individual AprilTag [27].

The machine learning approach explored in this work yielded promising results, but is a potentially risky method for part verification. A convolutional neural network approach is non-deterministic, meaning that there is no way of knowing what information the network is using to produce results. With this in mind, it is extremely difficult to confidently generalize the approach across a wide variety of environments. This makes the neural network approach to product verification undesirable for our stated goal of generalized AM product verification.

In addition to pose estimation, this research investigated model-based registration approaches for product verification. These approaches entail rendering a simulated image of the desired product and comparing it to an image captured from the TAZ6 printbed. To render an image of the product requires an

estimate of extruder pose – either from manual user alignment or a pose estimation algorithm. As such, this method can utilize and build-upon our AprilTag absolute pose estimation algorithm. Further, we note that a recursive pose estimation algorithm can increase the accuracy of the product's pose estimate and yield a more accurate comparison between the rendered and physical images. Results yield successful identification of structural defects for AM products and highlight the need for recursive pose estimation.

Additive manufacturing is becoming more prolific by the day. For the sake of security and efficiency, the US Department of Defense and other industrial sectors must be prepared to implement robust means of AM product verification. From this research, we identify a combination of an AprilTag pose estimation algorithm and a recursive template matching technique as the viable approach to AM product verification. The AprilTag pattern system provides the greatest information density of the printed patterns and provides an absolute estimate of extruder pose. As a result, the AprilTag system is not hindered by the growing position error problem of VO. This pose estimation can be used to verify the trajectory of the extruder throughout the print as well as to initialize a pose estimate for object rendering. While the trajectory of the extruder is being verified, image comparison and template matching can be conducted continuously throughout the printing process or immediately following a print. The result is a holistic and low-cost visual verification system achieved by combining these two methods.

While this project has investigated a variety of visual verification methods, it has also identified a variety of areas for future research to further enhance our proposed solution. This project investigated some methods, such as H.264 and machine learning, which can be expanded in future research. Furthermore, pose estimation establishes a foundation for a variety of other methods of AM product verification to include deposition material tracking or thermal analysis. We believe that this project successfully establishes a foundation for visual inspection methods for AM product verification and is a pivotal step towards a more robust manufacturing process.

7 Publications

7.1 Symposium on Additive Manufacturing and Applications 2020

On October 10th we compiled our investigation on feature-based VO to a five page paper and submitted it to the peer-reviewed Institute of Electrical and Electronics Engineers (IEEE) International Symposium on Additive Manufacturing and Applications. October 31th, we received notification that the paper had been accepted to the conference and will be published as in the proceedings for the Symposium. Further, I attended the virtual conference from November 9th to November 13th and gave a presentation on our research on November 11th.

The process of writing the paper, receiving feedback, and presenting our research to an international audience was an incredibly rewarding experience. Under the guidance of Professor Kutzer and Professor Donnal, this was my first attempt at writing a paper for publication. Before our final acceptance, the reviewers provided comments and recommendations on our paper and provided an opportunity to resubmit with minor revisions. The reviewers provided insightful feedback that I will use in targeting future publications and noted that the paper "is well written and well referenced".

On the morning of November 11th I presented our research virtually at the conference to an international audience. While it was daunting being the only undergraduate student for the session, I found it valuable to present our research in front of professionals who are well established in the field of AM. I was asked several questions concerning our approach and I had the opportunity to explain some of our

visual methods to the audience, most of whom were more familiar with the field of AM than machine vision. Additionally, neural networks and machine learning were a hot topic in other presentations. I would not be surprised to see more publications in the field of additive manufacturing that are focused on the integration of machine learning techniques into part verification and path planning.

7.2 International Mechanical Engineering Congress and Exposition 2021

On March 9th, 2021 we submitted an abstract to the International Mechanical Engineering Congress Exposition 2021 conference (IMECE 2021). Our submission includes our work on image registration and template matching for AM product verification. On March 17th, we received notification that our abstract has been accepted and that we are invited to submit a 5-page paper for the conference. We are currently finalizing our submission and I am looking forward to the opportunity to receive feedback and to present our continued research.

References

- [1] M. J. Louis, T. Seymour, and J. Joyce, “3d opportunity for the department of defense: Additive manufacturing fires up,” *A Deloitte Series on Additive Manufacturing*, pp. 4–5, 2014.
- [2] A. B. Badiru, V. V. Valencia, and D. Liu, *Additive Manufacturing Handbook: Product Development for the Defense Industry*. CRC Press, 2017.
- [3] N. T. Aboulkhair, M. Simonelli, L. Parry, I. Ashcroft, C. Tuck, and R. Hague, “3d printing of aluminium alloys: Additive manufacturing of aluminium alloys using selective laser melting,” *Progress in Materials Science*, vol. 106, p. 100578, 2019.
- [4] L. D. Sturm, C. B. Williams, J. A. Camelio, J. White, and R. Parker, “Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the. stl file with human subjects,” *Journal of Manufacturing Systems*, vol. 44, pp. 154–164, 2017.
- [5] S. Belikovetsky, M. Yampolskiy, J. Toh, J. Gatlin, and Y. Elovici, “dr0wned–cyber-physical attack with additive manufacturing,” in *11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17)*, 2017.
- [6] *3D Printing Infill: The Basics*. Craftcloud, Jan 2021.
- [7] M. Kutzer, J. Donnal, G. Sinsley, and R. McDowell, “Toward detecting cyber-physical attacks in additive manufacturing using multi-view visual odometry,” in *ASME 2020 Manufacturing Science and Engineering Conference*, 2020.
- [8] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [9] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*, pp. 404–417, Springer, 2006.

- [10] D. W. Eggert, A. Lorusso, and R. B. Fisher, “Estimating 3-d rigid body transformations: A comparison of four major algorithms,” *Machine vision and applications*, vol. 9, no. 5-6, pp. 272–290, 1997.
- [11] G. S. Chirikjian, A. B. Kyatkin, and A. Buckingham, “Engineering applications of noncommutative harmonic analysis: With emphasis on rotation and motion groups,” *Appl. Mech. Rev.*, vol. 54, no. 6, pp. B97–B98, 2001.
- [12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *2011 International conference on computer vision*, pp. 2564–2571, IEEE, 2011.
- [13] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2, pp. 1150–1157, IEEE, 1999.
- [14] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the h. 264/avc video coding standard,” *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 560–576, 2003.
- [15] J. P, “Mv-tractus: A simple and fast tool to extract motion vectors from h264 encoded video streams.,” Oct. 2018.
- [16] S. Li, C. Xu, and M. Xie, “A robust o (n) solution to the perspective-n-point problem,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1444–1450, 2012.
- [17] E. Olson, “Apriltag: A robust and flexible visual fiducial system,” in *2011 IEEE International Conference on Robotics and Automation*, pp. 3400–3407, IEEE, 2011.
- [18] I. Tabian, H. Fu, and Z. Sharif Khodaei, “A convolutional neural network for impact detection and characterization of complex composite structures,” *Sensors*, vol. 19, no. 22, 2019.
- [19] G. Adluru, E. V. DiBella, and M. C. Schabel, “Model-based registration for dynamic cardiac perfusion mri,” *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 24, no. 5, pp. 1062–1070, 2006.
- [20] N. V. Ruiter, R. Stotzka, T.-O. Muller, H. Gemmeke, J. R. Reichenbach, and W. A. Kaiser, “Model-based registration of x-ray mammograms and mr images of the female breast,” *IEEE Transactions on Nuclear Science*, vol. 53, no. 1, pp. 204–211, 2006.
- [21] C. Bibby and I. Reid, “Robust real-time visual tracking using pixel-wise posteriors,” in *European Conference on Computer Vision*, pp. 831–844, Springer, 2008.
- [22] B. O. Community, *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.
- [23] H. Tjaden, U. Schwanecke, E. Schömer, and D. Cremers, “A region-based gauss-newton approach to real-time monocular multiple object tracking,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1797–1812, 2018.
- [24] V. A. Prisacariu and I. D. Reid, “Pwp3d: Real-time segmentation and tracking of 3d objects,” *International journal of computer vision*, vol. 98, no. 3, pp. 335–354, 2012.

- [25] G. Schweighofer and A. Pinz, “Robust pose estimation from a planar target,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2024–2030, 2006.
- [26] G. Schweighofer and A. Pinz, “Globally optimal $O(n)$ solution to the pnp problem for general camera models,” in *BMVC*, pp. 1–10, 2008.
- [27] A. W. Long, K. C. Wolfe, M. J. Mashner, and G. S. Chirikjian, “The banana distribution is gaussian: A localization study with exponential coordinates,” *Robotics: Science and Systems VIII*, vol. 265, 2013.