Evaluating Structure from Motion (SfM) in 3D Printing Quality Control

Jesus Diaz[a]

[a]*Undergraduate Student, New Mexico State University, Department of Industrial Engineering,*

Abstract

In this study, stereo vision was utilized to generate point clouds representing edges during an in-process 3D printing pause. Images were captured in a grid pattern using a monocular nozzle camera attached to a 3D printer. The images were then converted into point clouds using RAFT Stereo for creating disparity maps, depth maps, camera intrinsic, 3D printer coordinates, and camera extrinsic.

Keywords: 3D printing, Structure from Motion

1. Introduction

3D printing, commonly referred to as Fused Filament Fabrication (FFF) and Fused Deposition Modeling (FDM), is a manufacturing process that creates parts by depositing melted material, primarily plastic polymers, layer by layer through a heated nozzle. The process aims to reduce waste compared to traditional manufacturing processes, making it an attractive method for producing high-quality products. From its conception, quality in 3D printing has been a topic of exploration for many organizations seeking to eliminate waste and utilize the freedom of utility that 3D printing provides. Defining and measuring quality can vary among organizations, ranging from a product's appearance to customer satisfaction. Amongst the different measurements, an organization may have for quality, quantifying quality with several tools provides insight for effective decision-making.

"Computer vision is a field of computer science that focuses on enabling computers to identify and understand objects and people in images and videos." (Microsoft, 2023) Within the umbrella of computer vision, stereo vision is the implementation of two or more cameras to describe a 3D scene. Concerning stereo vision, Structure from Motion (SfM) attempts to use camera positions and movement from a series of images to describe a 3D scene. In this paper, stereo vision techniques, primarily Structure from Motion, were used to achieve a 3D reconstruction model of a printed part by utilizing a nozzle camera attached to a 3D printer. Key characteristics such as the coordinate position of our nozzle and camera extrinsic aided in 3D reconstruction.

2. Tools

Within the evaluation, a Voron Switchwire 3D printer utilized a monocular nozzle camera, precisely a 3DO Nozzle Camera. The Voron Switchwire is a Core XZ printer, where the heated nozzle moves along the XZ plane, running the Klipper firmware. In addition, the printer deployed the user interface Mainsail for operation and configuration. The slicing software Ultimaker Cura aided in generating G-code instructions for the printer. The 3DO Nozzle Camera is a single-lens camera with a fixed focus lens, an 80° field of view, and utilizes a Sony IMX258 image sensor. The image sensor within the camera is capable of recording 30 frames per second at a 4K resolution and 60 frames per second at 1080P resolution with an aspect ratio of 3:4. (KB-3D, 2023) For experimentation, the camera operated at a resolution of 800 by 600 pixels.

In establishing the process for 3D reconstruction, several open-source tools were utilized such as: OpenCV (OpenCV Contributors, 2023b), mrcal (Kogan, 2023), RAFT Stereo(Princeton Vision and Learning Lab, 2023), Open3D (Open3D Contributors, 2023), and printerdata(Pyxis-ROC, 2023). At last, the open-source software CloudCompare (Miller, 2023) visualized the point cloud 3D reconstruction outcome.

3. Methodology

In order to perform 3D reconstruction, the nozzle camera's focal and world corrdinate system relationship is required. "The extrinsic matrix is a transformation matrix from the world coordinate system to the camera coordinate system, while the intrinsic matrix is a transformation matrix that converts points from the camera coordinate system to the pixel coordinate system."(Anwar, 2022) The camera intrinsic, illustrated in Equation (1), is a 3-by-3 matrix made up of the focal lengths (f_x and f_y) and the principal point (c_x, c_y) of the camera. Focal length describes "the distance between a cameras lens and physical sensor" (David, 2019) , as the principal point describe the "optical centers expressed in pixel coordinates."(OpenCV Contributors, 2023a) The camera intrinsic are obtainable via a camera calibration process consisting of taking photos of a chessboard pattern with a known spacing and vertices. Vertices are the corners of the chessboard which interact with four squares. For example, a six-by-seven patterned chessboard would have vertices of five-by-six. In obtaining the intrinsic of our nozzle camera, two samples of over 100 images of a six-by-six chessboard of

varying chessboard square sizes, 5mm, and 3mm. In identifying the chessboard and its vertices, mrgingham, an open-source tool used to gather checkerboard locations, and OpenCV processed images within the two samples. The data from mrgingham transition to mrcal, an advanced camera calibration tool, to be represented to different lens models. However, the chessboard data from OpenCV are further processed by the OpenCV camera calibration feature.

$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The camera extrinsic describes the camera position and rotations within a real-world cartesian system through a 4-by-4 matrix consisting of the rotation matrix and translation vector. As the nozzle camera is fixed in position, the rotation matrix remains constant with the identity matrix. Within stereo vision, the accumulation of the difference between the location of two images amongst the world coordinate system axes defines the translation vector. Utilizing the tool printerdata, we can correlate a pair of nozzle camera positions with images to create a translation vector (t_x, t_y, t_z). The translation vecotr (t_x, t_y, t_z) can be obtained with Equation (2) using two coordinates with corresponding and consecutive image pairs, (x_1, y_1, z_1) and (x_2, y_2, z_2).

$$\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{bmatrix} \tag{2}$$

The camera extrinsic with the translation vector (t_x, t_y, t_z) is illustrated in Equation (3)

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

To calculate depth, RAFT Stereo, "a new deep architecture for rectified stereo…[utilizing] multi-level convolutional GRUs" (Lipson et al., 2021), runs consecutive image pairs to create a disparity map. RAFT-Stereo utilized the pre-trained Middlebury model. Afterward, the Euclidean distance of coordinates from printerdata correlated with images ran through RAFT Stereo is calculated. With the disparity map generated by RAFT Stereo and the Euclidean distance between image pairs, a depth map can be created by applying Equation (4) across the dispairty map.

$$\text{Depth} = \frac{f_x \times \text{Euclidean distance}}{\text{Disparity}} \tag{4}$$

A series of transformations are applied to our depth information to obtain point cloud data. The first transformation converts the depth map into the camera's coordinate system (X_c, Y_c, Z_c) by taking the location (u, v) within the depth map, the depth value (d) at (u, v), the principal point, c_x and c_y, and focal lengths, f_x and f_y from the camera intrinsic matrix. Equation (5) illustrates the transformation of the depth map into the camera's coordinate system.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} (b - c_x) \times (\frac{d}{f_x}) \\ (b - c_y) \times (\frac{d}{f_y}) \\ d \end{bmatrix} \tag{5}$$

With (X_c, Y_c, Z_c) representing coordinates in the camera's coordinate system, the coordinates are transformed into coordinates (X, Y, Z), a point cloud representation, in the world coordinate system through Equation (6) using the camera extrinsic.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \tag{6}$$

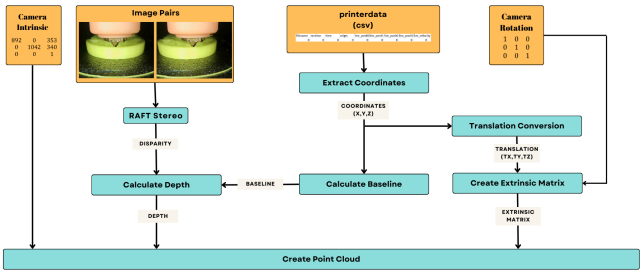


Figure 1: Flowchart of the process in creating point clouds from images

4. Results

The process of camera calibration, which commonly consists of taking images of a known measurement, is required to obtain camera intrinsic values such as the focal length and principal point. In gathering the camera intrinsic values, two image sets of a six-by-six chessboard pattern were captured. Each image set consisted of a chessboard pattern of different square sizes at 3mm and 5mm. An example of the images taken with the nozzle camera for camera calibration is shown in Figure 2 The image set with 3 mm-sized squares contained 237 images, while the other image set with 5 mm-sized squares contained 100 images. With the mrgingham tool, the checkerboard pattern was identified in 158 and 42 images of the 3mm and 5mm datasets respectively. The camera calibration tool mrcal tool processed results from mrgingham into several camera models to extract intrinsic parameters such as focal length and principal point. Results from running mrcal on both image sets are displayed in Table 1 and Table 2 with the 3mm and 5mm-sized squares respectively.

In an alternate route, the same samples would be processed under OpenCV's camera calibration Python API. With OpenCV, the checkerboard pattern was found in 14 and 76 images in the 5mm and 3mm image sets respectively. The results

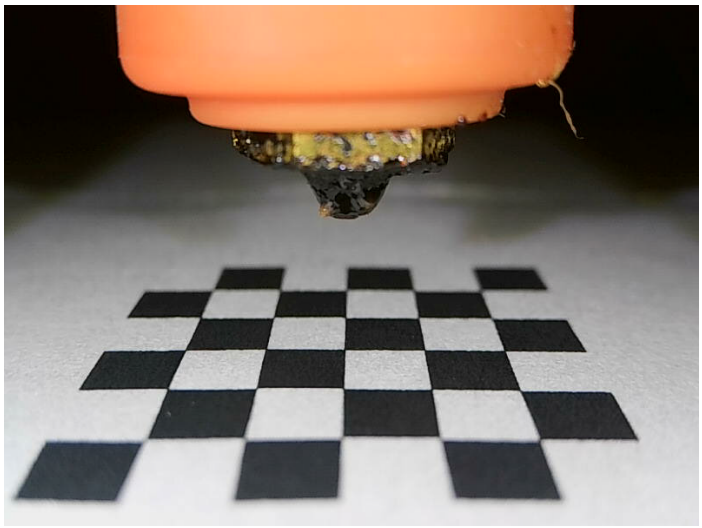


Figure 2: Camera calibration of the nozzle camera with a chessboard pattern

Lensmodel	f_x	f_y	c_x	c_y
Pinhole	953.1325	3191.237	396.8969	290.0021
OpenCV12*	1066.512	451.4552	495.9754	4.988957
OpenCV8*	845.0293	568.8355	410.3504	322.2841
OpenCV5*	844.7204	566.999	410.4842	321.728
OpenCV4*	834.1481	570.7482	416.8985	336.5039
Cahvor*	980.9348	567.0836	568.0304	314.7654

* Camera model had additional intrisict values which were the parameters of their respective model

Table 1: Camera intrinsic generated by different lensmodesl in mrcal from 3mm chessboard square spacing

Lensmodel	f_x	f_y	c_x	c_y
Pinhole	867.9742	999.9146	405.121	437.0037
OpenCV12 *	974.6048	1132.425	405.2198	-61.5075
OpenCV8 *	880.9118	1004.712	459.6673	374.1776
OpenCV5 *	876.3786	1007.699	458.1172	387.1971
OpenCV4 *	865.607	1007.484	434.3785	405.1796
Cahvor *	659.7097	753.0182	366.3163	479.9683

* Camera model had additional intrisict values which were the parameters of their respective model

Table 2: Camera intrinsic generated by different lensmodesl in mrcal from 5mm chessboard square spacing

from OpenCV occurred from utilizing different units for the object spacing parameter used in OpenCV. Results from OpenCV are illustrated in Tables 3 and 4 below for the 3mm and 5mm-sized squares respectively. Due to the repeatability of the camera intrinsic from OpenCV of the 5mm-sized squares, the process utilized the focal lengths and the principal point for the intrinsic matrix.

Due to the position of the nozzle camera, the factor of noise was a major concern for generating point cloud data. The noise related to the position of the camera is attributed to the fixed location of the nozzle and heating chamber across all images. To

Chessboard Square Spacing	f_x	f_y	c_x	c_y
3mm	12864	1880	347	268
.003m	6900	1233	383	238

Table 3: Camera intrinsic generate by OpenCV from 3mm chessboard square spacing based on units used within the checkerboard spacing parameter.

Chessboard Square Spacing	f_x	f_y	c_x	c_y
5mm	892	1042	353	340
.005m	892	1042	353	340

Table 4: OpenCV camera intrinsic from 5mm chessboard square spacing based on units used within the checkerboard spacing parameter.

measure the influence of the permanent structures in images on RAFT Stereo, the output disparity of RAFT Stereo was compared amongst image pairs in three different conditions. The three conditions consisted of utilizing the raw original image, masking permanent structures(the nozzle and heating chamber), and cropping the first 257 rows of pixels. Under five image pairs, minor standard deviations amongst the three methods of generating depth information ranged from 0mm to 5mm in areas where plastic material was deposited as the background experienced larger standard deviations. Figure 3 is a visualization of the standard deviation of one of the five image pairs that showcase the minor standard deviation of a 3D printed part. From these trials, using the raw original images within RAFT Stereo became standard in the process of 3D reconstruction, and the elimination of noise from the nozzle would be achieved by cropping the first 257 rows of the depth map.

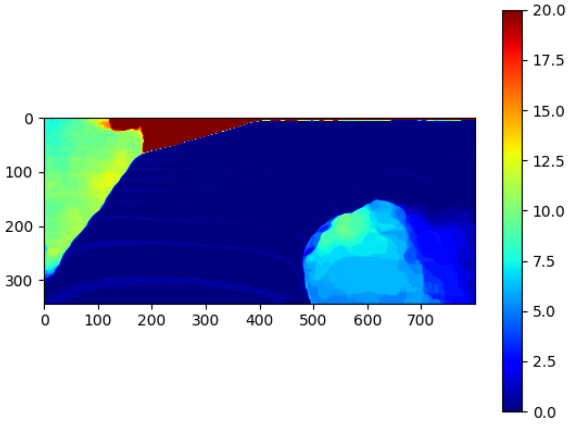
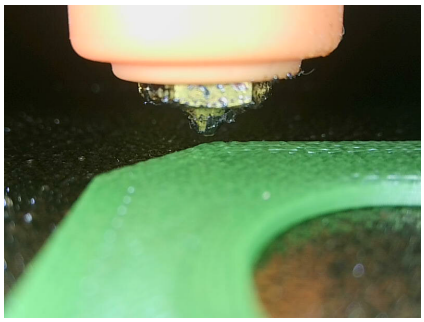
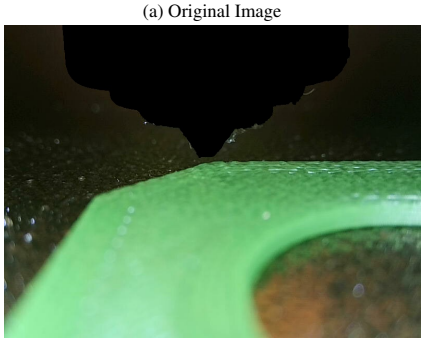


Figure 3: Visualizing the Standard Deviation Amongst the Original, Masked, and Cropped image

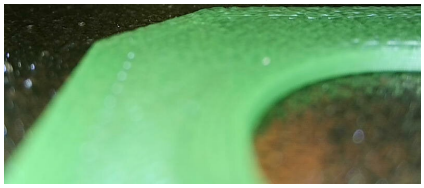
In evaluating the process, several sample images were captured under different conditions of completed and in-process parts. Image samples captured in a grid-like pattern experiencing slight movement in the x-axis and y-axis utilized the macros feature present in Klipper firmware. With the Klipper firmware, scripts, referred to as macros, containing G-code instructions



(a) Original Image



(b) Masked Image



(c) Cropped Image

Figure 4: An example of processed images used in comparing the output of RAFT Stereo and depth formulation

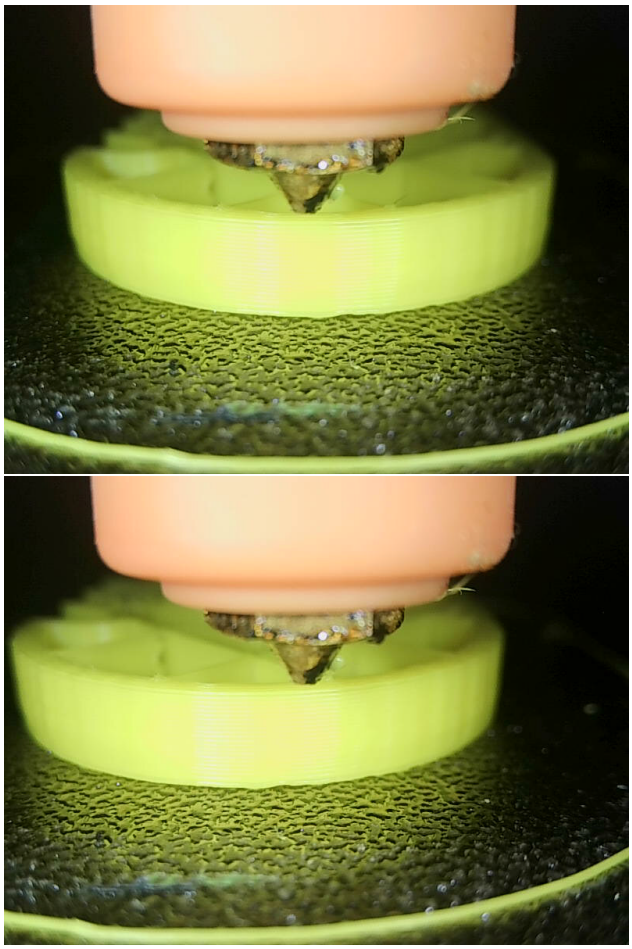


Figure 5: Input images that were taken 1mm apart from each other

can be utilized in the G-code instructions of a part. To capture photos of a part during printing, a macro was utilized to move in a grid-like pattern in increments of 3mm in the x and 5mm in the y while hovering 5mm above and 1mm in both the x and y hovering 0.2 mm above. Some preliminary results from the alignment of two image pairs, 5 , taken 1 mm apart, RAFT Stereo disparity, Figure 7, and point cloud generation, Figure 6 , are showcased below . On another note, manual movement of the nozzle through the Mainsail interface was used to capture images of a part in 10 mm increments in both the x and y of a 3D printed part on the print bed. From preliminary results, the process is capable of capturing straight and curved edges along the x-axis. To reduce the amount of noise, primarily the nozzle, the point cloud calculation is reduced to calculating the bottom 343 pixels of an image.

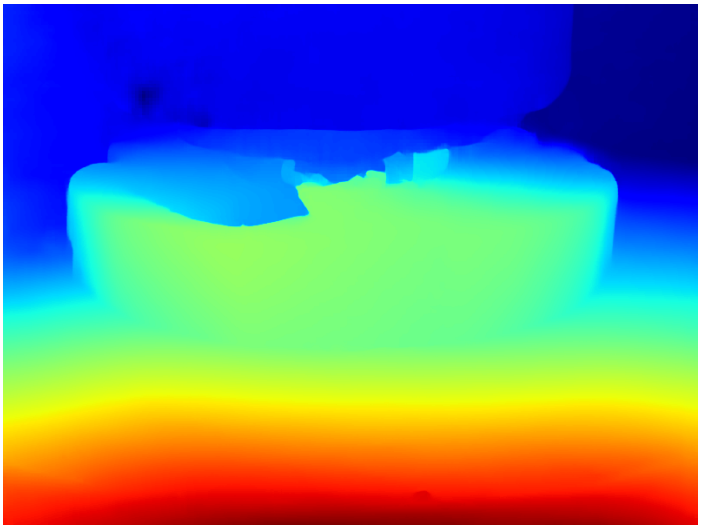


Figure 6: Disparity map generated with RAFT Stereo from image pair in Figure 5

5. Discussion

From mrcal and OpenCV, the variation of camera intrinsics can be attributed to three major factors: the spacing of the chessboard squares, dataset size, and spread of the chessboard pattern across the camera's field of view. Using a chessboard spacing of 3mm may have possibly led to errors in computing the camera intrinsic due to the amount of space the pattern covered in the camera's perspective in comparison to the 5mm spaced chessboard. On another hand, the possibility of utilizing too many images in computing the camera intrinsic may have contributed to a variation of results as the trial which uti-

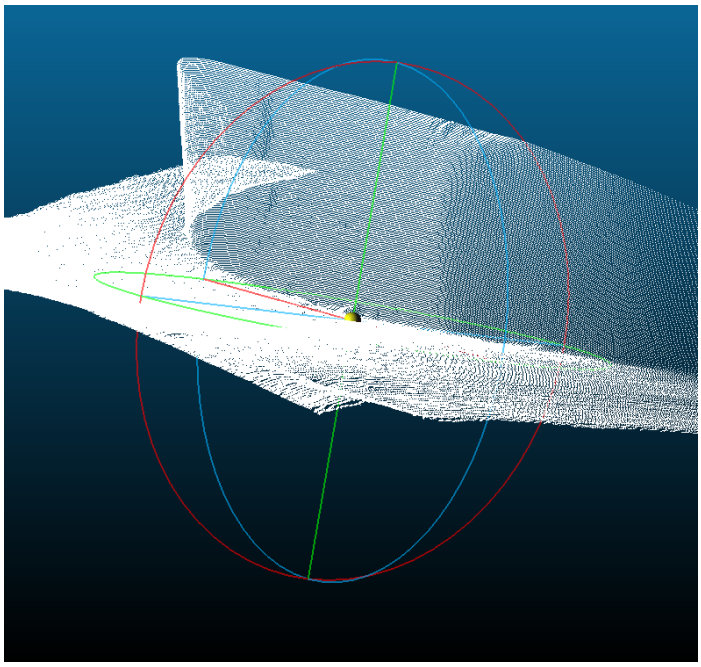


Figure 7: Point Cloud Generated from Disparity Map in Figure 6

lized the least number of images produced repeatable figures. Finally, the location of the chessboard pattern in the lower portions of the camera's perspective may have contributed the most to the variation of camera intrinsic due to the lack of distribution throughout the image pixel locations as illustrated in Figure 9 showcasing the spread of the chessboard pattern vertices from the 3mm dataset images.

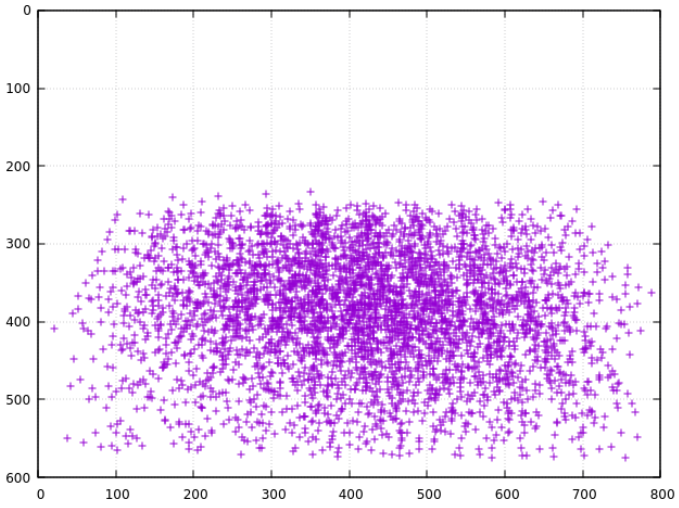


Figure 8: mrcal visualization of the spread of chessboard vertices

Transitioning the current pipeline into a real-time application will have difficulties in computational time from RAFT-Stereo alone. Figure 5 illustrates the vast computational time from the different models which are used within a series of sample images. The hardware processing RAFT Stereo within the process currently is a major attribution to the problem of computational time as RAFT Stereo is intended to run on NVIDIA GPUs over standard Intel CPUs. Applying the fastest model within RAFT Stereo would be a subject for future work as it utilizes multiple parameters which require experimentation to optimize the quality of disparity and speed.

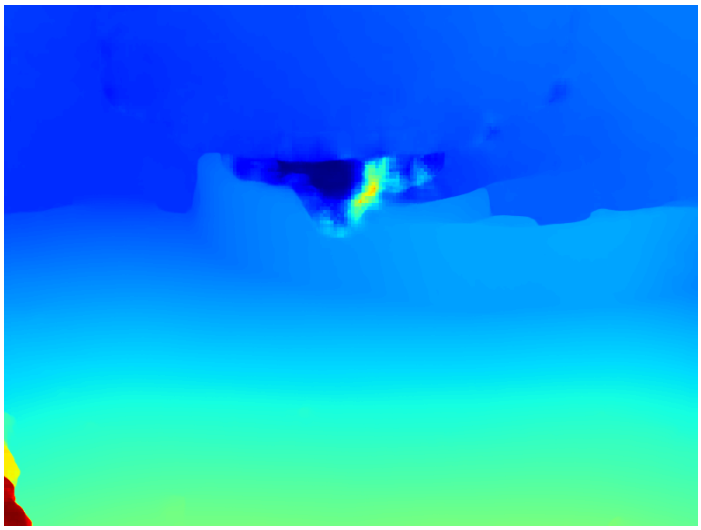
Training Model	Image Pair	Processing Times (s)
middlebury	1	136
middlebury	2	136
middlebury	3	137
middlebury	4	131
middlebury	5	137
sceneflow	1	141
sceneflow	2	133
sceneflow	3	133
sceneflow	4	132
sceneflow	5	134
eth3d	1	99
eth3d	2	105
eth3d	3	105
eth3d	4	103
eth3d	5	128
realtime0-full-image	1	7
realtime0-full-image	2	7
realtime0-full-image	3	7
realtime0-full-image	4	7
realtime0-full-image	5	7

Table 5: Computational Times of the Different Models Utilized by RAFT Stereo

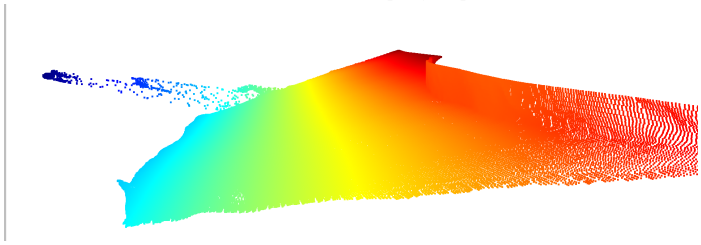
Out-of-place point clouds are a noticeable issue within the 3D reconstruction process. For example, Figure 9b showcases portions of the point cloud sticking out sorely beyond the range of our scene. The main cause of out-of-place can be attributed to the disparity maps generated by RAFT Stereo. Within the complementary figure, Figure 9a, the disparity map is shown to be distinct in its position in comparison to its surrounding as we expected it to share similar characteristics from looking at raw images. The bottom left corner of the RAFT disparity map showcases a red region which emphasizes a heavy disparity. This outcome is highly unexpected as the difference between the two images used is one millimeter in the x-axis.

6. Future Work

Stress testing the capabilities and ability of the process to properly represent the part is an area of future work of main interest as ultimately unit measurements would like to be obtained within a 3D reconstruction model. The test images that have been utilized within this evaluation were of flat and simplistic parts which lacked complex shapes. In evaluating the process with complex shapes, the process becomes one step closer to a real-time application as a realistic complex part will stress the current process. On another hand, if the process continues experiencing issues in representing simplistic parts, the process will require a re-evaluation of how it processes data and generates



(a) RAFT Stereo Disparity Map



(b) Point cloud with an error displayed in Open3D

Figure 9: Flaws of the process within RAFT-Stereo generated disparity map and Point Cloud

point clouds. However, another subject of future work can enhance the ability of the process in representing a part accurately. Currently, the process can generate a multitude of point clouds from depth maps gathered from disparity maps generated from consecutive image pairs. However, there is no feature matching within the current process amongst multiple point clouds. Implementing feature matching within the process will reduce the number of point clouds generated by correlating corresponding points and overlaps across point clouds. In addition, feature matching will help in representing a part more accurately and aid in capturing infill patterns. The primary methods of feature matching to implement in the process in future work would focus on feature matching in images through algorithms like SIFT and ORB and in point cloud geometry across voxels.

On another note, the correlation between coordinates generated from printerdata and images was processed manually in extracting points that did not correlate to images. Future work would entail automating and enhancing the correlation between the output of printerdata with images. Automating this process would allow for an easier transition into processing videos as we could assign a coordinate value to a frame appropriately.

Acknowledgements

I would like to take this opportunity to acknowledge my lab partner Sophia Cao of the University of Michigan and my P.I. Dr. Sreepathi Pai. In addition, I would like to acknowledge Ph.D. student Rongcui Dong and Dr. Yu Feng for their contributions and aid in the project. At last, I would like to acknowledge the Goergen Institute for Data Science and The David T. Kearns Center for supporting the Computational Methods for Music, Media, and Minds REU, an NSF-funded REU.

References

Aqeel Anwar. What are Intrinsic and Extrinsic Camera Parameters in Computer Vision?, 2022. URL https://towardsdatascience.com/what-are-intrinsic-and-extrinsic-camera-parameters-in-computer-vi

Biga David. Focal Length Conversion to Pixels for Distance Calculation (Android), 2019. URL https://medium.com/@biga.david/focal-length-conversion-to-pixels-for-distance-calculation-androi

KB-3D. Nozzle Camera Kit, 2023. URL https://kb-3d.com/store/electronics/779-3do-nozzle-camera-kit.html.

Dima Kogan. mrcal: geometric camera calibration, dense stereo, and structure-from-motion. https://github.com/dkogan/mrcal, 2023.

Lahav Lipson, Zachary Teed, and Jia Deng. RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. *2021 International Conference on 3D Vision*, pages 218–227, 2021.

Microsoft. What is computer vision?, 2023. URL https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-is-computer-vision/#:~:text=Computer%20vision%20is%20a%20field,tasks%20that%20replicate%20human%20capabilities.

Daniel G. Miller. Creative Commons License Chooser. https://www.danielgm.net/cc/, 2023.

Open3D Contributors. Open3D: A Modern Library for 3D Data Processing. lhttps://github.com/isl-org/Open3D, 2023.

OpenCV Contributors. Camera calibration With OpenCV, 2023a. URL https://docs.opencv.org/4.x/d4/d94/tutorial_camera_calibration.html.

OpenCV Contributors. OpenCV: Open Source Computer Vision Library, 2023b. URL https://github.com/opencv/opencv.

Princeton Vision and Learning Lab. RAFT-Stereo: Code for "RAFT: Recurrent All Pairs Field Transforms for Optical Flow". https://github.com/princeton-vl/RAFT-Stereo, 2023.

Pyxis-ROC. printerdata: A Python Library for Printer Data Analysis. https://github.com/pyxis-roc/printerdata, 2023.