Western Graduate&PostdoctoralStudies

Western University
Scholarship@Western

Electronic Thesis and Dissertation Repository

5-31-2024 10:00 AM

Improving Fused Filament Fabrication Additive Manufacturing through Computer Vision Analysis and Fabrication Optimization

Aliaksei Petsiuk, *Western University*

Supervisor: Pearce, Joshua M., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Doctor of Philosophy degree in Electrical and Computer Engineering

© Aliaksei Petsiuk 2024

Follow this and additional works at: https://ir.lib.uwo.ca/etd

Part of the Other Computer Engineering Commons, Other Electrical and Computer Engineering Commons, and the Signal Processing Commons

Recommended Citation

Petsiuk, Aliaksei, "Improving Fused Filament Fabrication Additive Manufacturing through Computer Vision Analysis and Fabrication Optimization" (2024). *Electronic Thesis and Dissertation Repository*. 10147.
https://ir.lib.uwo.ca/etd/10147

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlswadmin@uwo.ca.

Abstract

Additive manufacturing (AM), also known as 3-D printing, is one of the fundamental elements of Industry 4.0. According to ASTM standards, AM can be classified by production principles, types of raw materials, energy sources, and fabrication volumes. Fused filament fabrication (FFF) is one of the most accessible technologies that offers independent manufacturers great opportunities due to its simplicity, scalability, and low cost.

Modern 3-D printing is moving from single-material prototyping to complex multi-material product creation. It is firmly established in a wide range of applications, significantly expanding manufacturing horizons, providing innovative design capabilities, and improving product quality through the optimal combination of properties often impossible to achieve with traditional methods.

Despite the great potential and current exponential growth in production, AM, however, faces challenges that affect its adoption, efficiency, and product quality. An analysis of user databases shows the average failure rate is about 20 percent. The likelihood of manufacturing defects grows with the size of the object and the time required to print it, which can lead to increased material waste resulting from even a small failure rate. The ability to automatically detect anomalies in AM will greatly help reduce the wastage of material and time spent reproducing failed prints.

To strengthen the capabilities of AM technology, it is necessary to optimize the process of preparing a part for 3-D printing (slicing) and provide analysis systems that can detect and minimize the impact of emerging defects. The increasing complexity of geometric shapes and the number of materials used require optimization of fabrication processes and layer-by-layer monitoring of production processes for timely response.

This work presents several conceptually new approaches to FFF AM 3-D printer work volume monitoring and anomaly detection and localization based on monocular computer vision, machine learning, and synthetic data, as well as to increasing efficiency and reducing production waste in multi-color fabrication. Thus, a system for automatically creating a labeled G-code-based synthetic 3-D printing dataset was developed, providing layer-by-layer semantic segmentation of a printing part and its structural elements during the manufacturing process. A method has been developed for the procedural simulation of ideal fabrication by generating layer-wise photorealistic images of the manufactured part for further use as references for visual analysis at each manufacturing stage. To monitor the height, external contour, and internal structure of the manufactured object, a multi-stage approach based on computer vision has been developed, which allows analyzing images of each printed layer for compliance with the source 3-D model. A new fabrication method has been developed for multi-color printing to reduce energy and material costs for single-nozzle systems. The presented developments formed the basis for the concept of multifaceted visual analysis of 3-D printing processes. This will help improve FFF AM technology and reduce the amount of time, materials, and energy required to fabricate physical objects.

Keywords

3-D printing, additive manufacturing, anomaly detection, computer vision, fused filament fabrication, quality assurance

Summary for Lay Audience

Additive manufacturing (AM), or 3-D printing, is the cornerstone of the next-generation industry. Modern 3-D printing is moving beyond single-material prototyping to the creation of complex, multi-material products, expanding manufacturing capabilities across multiple fields, unlocking innovative design potential, and enhancing product quality through unique combinations of properties and geometries unachievable by traditional methods.

Exponential production growth, however, demonstrates a significant increase in plastic waste, which is also exacerbated by a high failure rate averaging 20%. This highlights the need for automatic failure detection systems in AM processes to reduce material, time, and energy losses.

This work demonstrates approaches to detecting and localizing 3-D printing defects based on computer vision, machine learning, and synthetic data, as well as to increasing efficiency and reducing production waste in multi-color fabrication. The presented methods will help improve additive manufacturing technology and reduce the amount of time, materials, and energy required to fabricate three-dimensional objects.

Acknowledgements

I am grateful to everyone who has been with me along this exciting journey.

Many thanks to my advisor, Dr. Joshua Pearce, for his invaluable support and driving force over these years and for the vision and knowledge he shared with me. This has strengthened my professional and academic skills and contributed significantly to my personal growth.

Thanks to my first academic supervisors at the Belarusian National Technical University and the National Academy of Sciences, Roman Novichikhin, Siarhey Hryshyn, and Yury Livshits, for igniting my passion for scientific research.

Thanks to my Belarusian colleagues, Maksim, Vladislav, Slava, Andrei, Nikolay, Yuri, Igor, and others, for constant friendly competition and discussions about physics, astronomy, and scientific instrumentation.

Thanks to my mentors at Michigan Tech, Dr. Jeremy Bos, Dr. Michael Roggemann, Dr. Jeffrey Burl, and Dr. Glen Archer, and others. I learned a lot from you.

Thanks to my colleagues at Michigan Open Sustainability Technology Lab, Shane, Adam, Jerry, Aubrey, Nupur, Nagendra, and others for your warm support.

Thanks to my teammates at Free Appropriate Sustainability Technology (FAST) Research Group at Western University, Koami, Rav, Michelle, Xiang, Jatin, Apoorv, Fangkai, Giorgio, Shafquat, Riya, Janik, Catalina, and others, for your support and motivation.

Thanks to my colleagues at Mosaic Manufacturing, Brandon Bloch, Derek Vogt, Kanstantsin Pachkouski, Mitchell Debora, and others, for your motivation and help in my multi-color 3-D printing endeavors.

Thanks to my parents, Halina and Leanid Petsiuk, to my brother Vitali, to my grandparents and the rest of the family, to my friends, Narendra, Niaz, Sheefa, Sadaf, Khalid, Ashfiq, Lota, Max, Anya, Mikola, and others, for supporting me during tough moments.

Thank you, my dear Ale, for your care, support, motivation, understanding, and love.

Thanks to my committee members, Dr. Ana Luisa Trejos, Dr. Harvey Shi, Dr. Kenneth McIsaac, and Dr. David Rosen, for your expertise, time, and attention.

Thanks to Jules Verne, Isaac Asimov, William Gibson, and many other authors for inspiration through literature.

Without you it would not be possible.

Co-Authorship Statement

The following thesis contains material from manuscripts that have been published or are in press. The work presented herein has been written by Aliaksei Petsiuk under the supervision of Dr. Joshua Pearce, and has been co-authored by Aliaksei Petsiuk, Harnoor Singh, Himanshu Dadhwal, Brandon Bloch, Derek Vogt, Mitchell Debora, and Dr. Joshua Pearce. Four articles were selected to form the body of the dissertation, and the extent of co-author collaboration is indicated below.

Chapter 2. Computer Vision-based Layer-wise 3-D Printing Analysis

Paper: Open Source Computer Vision-based Layer-wise 3-D Printing Analysis.

Current Status: Published in Additive Manufacturing (Elsevier), vol. 36, 101473, 2020, DOI:10.1016/j.addma.2020.101473.

Aliaksei Petsiuk: First author, corresponding author, developed an experimental setup and an algorithm for monocular vision-based assessment of both global and local printing defects analyzed data, and wrote the manuscript.

Dr. Joshua Pearce: Co-author, supervised the experimental setup and data collection, assisted in conceptualization, methodology, writing, and editing the manuscript.

Chapter 3. Interlayer Anomaly Detection Based on HOG-features and Synthetic Images

Paper: Towards Smart Monitored AM: Open Source In Situ Layer-wise 3-D Printing Image Anomaly Detection Using Histograms of Oriented Gradients and a Physics-based Rendering Engine.

Current Status: Published in Additive Manufacturing (Elsevier), vol. 52, 102690, 2022, DOI:10.1016/j.addma.2022.102690.

Aliaksei Petsiuk: First author, developed an experimental setup and algorithm for monocular vision-based assessment of internal 3-D printed layer deviations analyzed data, and wrote the manuscript.

Dr. Joshua Pearce: Corresponding author, supervised the experimental setup, conceptualization, methodology, simulation study, data collection, assisted in writing, and editing the manuscript.

Chapter 4. Synthetic-to-real Composite Semantic Segmentation in AM

Paper: Synthetic-to-real Composite Semantic Segmentation in Additive Manufacturing.

Current Status: Published in Journal of Manufacturing and Materials Processing (MDPI), vol. 8, issue 2, 66, 2024. DOI:10.3390/jmmp8020066.

Aliaksei Petsiuk: First author, analyzed 3-D printer users' data, developed an automated G-code-based synthetic data-generating system, trained neural networks, analyzed the capabilities of image-to-image style transfer, and wrote the manuscript.

Harnoor Singh: Co-author, assisted in developing a neural network model for semantic segmentation, analyzing data, and editing the manuscript.

Himanshu Dadhwal: Co-author, assisted in developing the G-code parsing and rendering pipelines, collecting data, and editing the manuscript.

Dr. Joshua Pearce: Corresponding author, provided data for the analysis, supervised the conceptualization, methodology, simulation study and data collection, assisted in editing the manuscript.

Chapter 5. Tool change Reduction in Multi-color 3-D Printing

Paper: Tool change Reduction for Multi-color Fused Filament Fabrication Through Interlayer Tool Clustering Implemented in PrusaSlicer.

Current Status: Accepted (in press) in Rapid Prototyping Journal (Emerald Publishing). Preprint is available at https://dx.doi.org/10.2139/ssrn.4655383.

Aliaksei Petsiuk: First author, developed a PrusaSlicer fork with an interlayer tool clustering algorithm, that reduces the number of tool changes in multi-color fabrication, performed the experiments, analyzed print statistics, and wrote the manuscript.

Brandon Bloch: Co-author, assisted with conceptualization, methodology, experimental setup, main algorithm development, print statistics analysis, and editing the manuscript.

Derek Vogt: Co-author, assisted with conceptualization, experimental setup, main algorithm development, print statistics analysis, and editing the manuscript.

Mitch Debora: Co-author, assisted with conceptualization, supervised the simulation study, data collection, experimental setup, and editing the manuscript.

Dr. Joshua Pearce: Corresponding author, supervised the conceptualization, simulation study data collection, assisted in writing and editing the manuscript.

Table of Contents

Abstract ii

Summary for Lay Audience iv

Acknowledgements v

Co-Authorship Statement vii

Table of Contents x

List of Tables xiv

List of Figures xv

Acronyms xxvi

Chapter 1 1

1 Introduction 1

- 1.1 FFF AM Technology 1
- 1.2 FFF AM Challenges 3
- 1.3 Research Objectives 5
- 1.4 Thesis Outline and Contributions 6
- 1.5 Bibliography 9

Chapter 2 13

2 Computer Vision-based Layer-wise 3-D Printing Analysis 13

- 2.1 Abstract 13
- 2.2 Introduction 14
- 2.3 Visual Platform Design 17
- 2.4 Algorithm Development 24
  - 2.4.1 Side View Height Validation 26
  - 2.4.2 Global Trajectory Correction 26
  - 2.4.3 Local Texture Analysis 29

2.4.4 Targeted Failures and Corrective Actions 32

2.5 Experimental Results 34

2.5.1 Height Validation Results 35

2.5.2 Results of the Global Trajectory Analysis 37

2.5.3 Results of the Local Texture Analysis 40

2.5.4 Runtime Analysis 43

2.5.5 Failures Database and Future Development 45

2.6 Conclusions 47

2.7 Bibliography 48

Chapter 3 57

3 Interlayer Anomaly Detection Based on HOG-features and Synthetic Images 57

3.1 Abstract 57

3.2 Introduction 58

3.3 Method 60

3.3.1 Experimental Apparatus 60

3.3.2 Creation of Synthetic Reference Images 62

3.3.3 Comparison of the Printed Layer with the Reference Image 65

3.4 Results 75

3.4.1 Test Print Modes for Selecting Optimal Similarity Measures 75

3.4.2 Comparing HOG-based Similarity Measures 76

3.5 Discussion 79

3.6 Conclusions 82

3.7 Bibliography 82

Chapter 4 90

4 Synthetic-to-real Composite Semantic Segmentation in Additive Manufacturing 90

4.1 Abstract 90

4.2 Introduction 91

4.3 Background 94

4.4 Methods 96

4.4.1 Creation of Synthetic Image Datasets 97

4.4.2 Semantic Image Segmentation 104

4.4.3 Image-to-image Translation 104

4.5 Results 106

4.6 Discussion and Conclusions 111

4.7 Bibliography 113

Chapter 5 120

5 Tool change Reduction in Multi-color 3-D Printing 120

5.1 Abstract 120

5.2 Introduction 121

5.3 Background: Traditional Slicing Approach and Related Works 123

5.4 Theory and Application: Proposed Tool Clustering 127

5.4.1 Material, Time, and Energy Savings 129

5.4.2 Implementation: PrusaSlicer Upgrade 133

5.5 Experimental Methods 138

5.6 Results and Discussion 139

5.7 Conclusions 145

5.8 Bibliography 146

Chapter 6 153

6 Conclusion 153

6.1 Summary 153

6.2 Contributions ... 153

6.3 Limitations and Future Work ... 155

Curriculum Vitae ... 157

List of Tables

Table 2.1: 3-D printing parameters that may cause failures. 18

Table 2.2: Target failures and corrective actions. 32

Table 3.1: Interlayer G-code commands. 61

Table 3.2: Similarity and distance measures. 72

Table 4.1: Segmentation results for synthetic test datasets (mIoU scores, %). 108

Table 5.1: Comparison of time and material allocation for different fabrication techniques of a selected set of 3-D models [71]. 141

List of Figures

Figure 1.1: Key steps to transform a digital design into a physical object: *a)* – tessellated mesh of an object model, *b)* – slicing the object before 3-D printing (green – support, red – internal infill, yellow – internal perimeter, orange – external perimeter, and purple – solid infill), *c)* – part of an object's tessellated mesh file, representing information about its vertices and normals, *d)* – part of an object's G-code file that represents information about the nozzle's trajectory and the amount of material being extruded. 2

Figure 1.2: Example of common 3-D printing failures a) – blocked nozzle, b) – under-extrusion, c) – over-extrusion, d) – adhesion problem, e) – displacement of individual layers of the printing part, f) – separation and shift of the printing part from the printing surface ("spaghetti problem"). 4

Figure 2.1: Visual platform: working area (left), printer assembly (right): *a)* – camera; *b)* – 3-D printer frame; *c)* – visual marker plate on top of the printing bed; *d)* – extruder; *e)* – movable lighting frame; *f)* –printed part. 19

Figure 2.2: Projective transformation of the G-code and STL model applied to the source image frame: *a)* – camera position relative to the STL model; *b)* – G-code trajectories projected on the source image frame. This and the following slides illustrate the printing analysis for a low polygonal fox model [67]. 20

Figure 2.3: Frames obtained from the monocular vision control system: *a)* – rectified source image frame; *b)* – unwrapped virtual top-view; *c)* – unwrapped pseudo-side-view. 21

Figure 2.4: Volumetric analysis of the printed part: *a)* – STL model with the orthogonal scan planes; *b)* – vertical XZ slice; *c)* – vertical YZ slice. 22

Figure 2.5: Layer-wise G-code segmentation: *a)* – an STL layer outline (black dashed line) overlay with the segmented layer of the source G-code with such trajectory categories as outer (blue) and inner walls (red), infill (green), and support (yellow); *b)* – possible inconsistency of STL and G-code; *c)* – unwrapped virtual top-view overlay with the outer wall (blue), infill (green), and support (yellow) extruder trajectories obtained from the G-code. 23

Figure 2.6: Image processing pipeline. During the analysis stage, two auxiliary images (a virtual top view and a pseudo-side-view) are created from the main captured frame. Next, the pseudo-side-view is used to validate the height of the layer, while the virtual top view is used to analyze the outline of the layer and its interior. These three processes occur independently of each other. 24

Figure 2.7: 3-D printing control algorithm. After completing each layer, a virtual top view and a pseudo side view are obtained from the source image frame and used to independently analyze the height of the layer, as well as its external and internal parameters. Depending on the deviations found, certain error elimination procedures are launched and the cycle is repeated. If the deviation cannot be corrected, the printing process pauses and awaits corrective actions from the user. 25

Figure 2.8: Defective layer recovery using the ICP algorithm. After the fabrication of the k-th layer, its contour is detected by visual analysis (red points) and compared with the reference contour obtained from the G-code (blue points). If there is a discrepancy, the transformation parameters (scale, translation, and rotation) are determined to return the layer to its initial position according to the G-code. These transformations are then applied to all of the subsequent layers to prevent geometric distortions caused by mechanical defects or part movement during manufacturing. 29

Figure 2.9: The Leung-Malik orientation and spatial-frequency selective filter bank: 36 Gaussian derivative filters (top three rows), 8 difference of Gaussians and 4 Gaussian filters (bottom row). 30

Figure 2.10: Example of failure correction. If defects are detected on the current layer, the G-code coordinates for the next layer can be updated. 34

Figure 2.11: Pseudo-side-view generation: *a)* – computed linear visibility delimiter (white dashed line), the edge for the visible side region (blue) and invisible side region (red) of the printed part; *b)* – designated visible area for unwrapping; *c)* – unwrapped region with the reference vertical level (yellow) and the maximum double-layer errors in both directions (red); *d)* – detected vertical edge error (blue) with the reference layer height (yellow) and maximum double-layer errors in both directions (red). 35

Figure 2.12: Results of vertical level validation. The maximum amount of detected layer edge displacement (total vertical level error) depends on the observed part's geometry at each stage of its completion. Under normal printing conditions, however, the median error of the detected edge for each layer does not exceed the maximum deviations for one layer. 36

Figure 2.13: Global displacement detection based on MTM algorithm: *a)* – contour-based binary template; *b)* – printed part shifted due to failure; *c)* – computed shift distance and direction. 38

Figure 2.14: Global G-code trajectory matching for a single layer based on ICP algorithm: *a)* – reference outline (green) mismatched with the detected contour points (red), the saturated region around the reference outline illustrates the restrictive mask obtained from the STL layer outline, which constrains the point cloud correspondence search for the ICP algorithm within the given area; *b)* – initial ICP iteration; *c)* – final ICP iteration. 38

Figure 2.15: Results of the global trajectory matching analysis. Under normal printing conditions, in this case, the maximum detected displacement does not exceed 1.8 mm, and the layer rotation is 2.5 degrees. Here the upper layers are smaller in size relative to the lower ones. This affects the number of detected contour points used as input data for the global trajectory analysis procedures and, in turn, can introduce errors in the detected rotation, shift, and scale parameters. 39

Figure 2.16: Partitioning an image into texture channels (textons): *a)* – test image with 18 various texture samples (modified image from https://all3dp.com/2/infill-3d-printing-what-it-means-and-how-to-use-it); *b)* – segmented textures; *c)* – obtained texture channels. 41

Figure 2.17: Defective layer segmentation results: *a)* – source virtual top view; *b)* – segmented image with the with infill mask (green), infill textures (blue regions), outside textures (red regions); *c)* – segmented failures (red and blue regions) inside the infill area (green).. 42

Figure 2.18: Results of the texture analysis during the regular printing. The method can falsely detect anomalies of up to 15 percent of the total layer area or higher in normal printing mode, indicating the need for additional measures to prevent false positives. ... 43

Figure 2.19: Runtime distribution: *a)* – time decomposition of layer-wise visual analysis; *b)* – distribution of the total analysis time for all of the layers. 44

Figure 2.20: Detected regions with abnormal texture: *(a–d)* – segmented textures; *(e–h)* – detected failures; *(i–l)* – cropped regions of interest with failures (not to scale). 46

Figure 3.1: Experimental apparatus: *a)* – 3-D printer schematic, *b)* – 3-D printing area. 1 – 3-D printer, 2 – extruder, 3 – printing bed, 4 – movable circular lighting platform, 5 – lighting platform drive system, 6 – camera, 7 – visual markers. 60

Figure 3.2: Virtual workspace: *a)* – main elements of the Delta printer modeled in Blender, *b)* – virtual camera view area. 1 – printing bed, 2 – movable lighting platform, 3 – camera, 4 – rendered G-code, 5 – visual markers. 62

Figure 3.3: Shader graph for procedural texture generation: *a)* – shader nodes, *b)* – procedural texture samples. The output material is a combination of Principled and Translucent BSDFs. The Principled node is responsible for surface parameters such as color, roughness, and reflection. Texture nodes create realistic surface irregularities. Adding Color ramp nodes is used to limit texture irregularities and truncate differences in surface height created by the Bump node. 63

Figure 3.4: Texture matching examples: *a)* – synthetic image of the centrifuge part (layer 18 of 24), *b)* – real image of the centrifuge part (layer 18 of 24), *c)* – synthetic image of the slot die (layer 24 out of 35), *d)* – real image of the slot die (layer 24 out of 35)........ 64

Figure 3.5: Printing object: *a)* – STL file of the whole part, *b)* – G-code of a layer cross-section, *c)* – STL-based layer cross-section mask. 64

Figure 3.6: Spatial position of the active printing area: *a)* – real image, *b)* – rendered image. 1 – active printing plane, 2 – print surface plane. 65

Figure 3.7: Virtual top view: *a)* – camera frame, *b)* – unwrapped virtual top view, *c)* – masked printing area, *d)* – Blender scene frame, *e)* – unwrapped synthetic image, *f)* – masked rendered region. 67

Figure 3.8: Consecutive set of unwrapped layers combined into a volumetric view. This property allows overlaying detected anomalies on a volumetric view for subsequent analysis. 68

Figure 3.9: Detection of the dominant gradient orientation in local image areas (top) using histograms of oriented gradients (bottom). Each bar of the histogram corresponds to the tilt angle of the image gradient in the range from 0° to 180°. The N parameter reflects the noise level in the source image. 69

Figure 3.10: Characteristic infill patterns *(a, c)* and their feature descriptors *(b, d).* The HOG method captures the underlying geometric structures representing different infill areas by identifying the directions of contrasting edges and creating a unique histogram pattern for each image region. 69

Figure 3.11: Stages of comparative image analysis: *a)* – splitting the original camera image into local areas, *b)* – separate regions of the source image, *c)* – converting image areas to feature vectors, *d)* – comparison of normalized feature vectors of the original and reference images, *e)* – resulting similarity map. 70

Figure 3.12: Normalized similarity measures, expressed as a percentage of coincidence, for the following cases: *a)* – complete match, *b)* – similar histograms with small deviations, *c)* – similar histograms with differences in levels (represents alike image areas with varying illumination parameters), *d)* – histograms with significant shifts, *e)* – non-overlapping histograms. 74

Figure 3.13: Test images of regular and failed printed layers: *a)* – local infill defects, *b)* –presence of a foreign body in the layer, *c)* – spaghetti problem, *d)* – separation and shift of the printing part from the working surface, *e)* – defects in thin walls, *f)* – layer shift. 75

Figure 3.14: Heat maps of the regular and failed prints for the example components. Defective areas of the image have a lower similarity value (green and blue colors), while areas with high similarity values (yellow, orange, and red colors) are most consistent with the “ideal” synthetic prints (reference synthetic images). 77

Figure 3.15: Discriminative power of the selected metrics after applying the 70% failure threshold. Vertical color bars represent the difference between the regular (bottoms) and failed (tops) prints: local infill defects (blue), presence of a foreign body in the layer (orange), spaghetti problem (green), separation and shift of the printing part from the working surface (red), defects in thin walls (purple), layer shift (brown). 78

Figure 3.16: Areas of a printed part with different print modes and overlaid similarity masks. Jaccard-based failure masks cover most of both prints, considering them as failed due to high sensitivity to minor texture deviations, while cosine-based masks allow to distinguish two different print modes by the size of the covered areas. 78

Figure 3.17: Example of failure detection based on HOG features and cosine similarity: *a)* – local infill defects, *b)* – presence of a foreign body in the layer, *c)* – spaghetti problem, *d)* – separation and shift of the printing part from the working surface, *e)* – defects in thin walls, *f)* – layer shift. 79

Figure 4.1: Analysis of 3-D printer users’ activity for 2.3 years. The runtime distribution shows a 24% failure rate for all of the 5.6 million printing tasks longer than 5 minutes. 92

Figure 4.2: Distribution of the 25 most frequently used words in file names for 3-D printing. A detailed analysis of the users’ print tasks database is given in the source file repository [56]. 97

Figure 4.3: Synthetic AM database creation pipeline. Each 3-D part in the form of an STL (green) file is converted into a set of printer tool head trajectories (G-code, blue), which is the input parameter of the automated scripted section (gray). Blender environment (textures, camera, lights) and compositing settings can also be automated in the future. The image-mask pairs (red) are the result of a frame-by-frame animation rendering for each individual G-code file. 98

Figure 4.4: Blender scene: user window (left) and the virtual camera viewport (right). 1 – printing bed/ground surface texture, 2 – background image plane simulating ambient environment, 3 – rendered manufactured part, 4 – light sources with variable locations, 5 – camera with variable location. 99

Figure 4.5: Texture samples for the printing bed/ground surface. More than 15 photographs of surfaces such as wood, metal, paper, stone, and others were superimposed onto the virtual working area. Variations in lighting, cropping, scaling, and image orientation during animation allow the creation of unique backgrounds. 100

Figure 4.6: The shading node network has been experimentally developed to achieve maximum realism of generated renders. The creation of all of the connections and node settings is fully automated in the code, which provides the flexibility to adjust the color, transparency, reflectivity, and other characteristics of the output material (red). 100

Figure 4.7: Detailed example of texture creation: (a) single Principled BSDF node; (b) Principled BSDF node mixed with Glossy, Diffuse, and Transparent BSDFs; (c) combined BSDF material with Noise and Voronoi textures; (d) final output with added color ramp nodes to truncate the Bump heights and create transmission anisotropy in the Principled BSDF. 101

Figure 4.8: The composite node network (for internal structure segmentation, in this example) assigns user-defined color labels to each pixel in the output image, depending on whether it belongs to a particular area (infill, shell, or support) of the rendered part. This creates a pixel-precise ground truth mask (red) for each output image frame (red) in the animation. 102

Figure 4.9: 3-D model slicing procedure. (a) Whole part in STL format. (b) Internal structure of sliced layers (red – outer shell, green – inner shell, yellow – infill, blue – support). (c) Side view illustrates current printing layer (top layer at each manufacturing stage). 103

Figure 4.10: Image-mask pair samples for each AM synthetic dataset: (a) whole part segmentation, (b) newly fabricated top layer segmentation, (c) internal layer segmentation. 103

Figure 4.11: Unpaired image-to-image translation using the cycle-consistent adversarial network. Handpicked images of real and virtual printed parts were loaded into CycleGAN, which learns to map real domain images to their synthetic counterparts and vice versa, minimizing the cycle consistency loss L_C. Here, the red and blue circles represent the same image presented in different domains. 105

Figure 4.12: Image-to-image style translation example. Translating a real image into its synthetic version reduces the contrast and saturation of the reflections and incidental filament strings........ 106

Figure 4.13: The results of semantic segmentation, presented using several real images. The neural network was trained on similar synthetic 3-D models. The color, printing surface texture, and slicing parameters, however, differ from those used in the training dataset. 107

Figure 4.14: Datasets for the style transfer influence analysis: (a) synthetic data, (b) real data, (c) real data after style transfer. The upper row shows sample images and the lower row illustrates the corresponding ground truth masks. 109

Figure 4.15: Domain comparison via t-SNE projections (a), and segmentation performance before and after style translation (b). 109

Figure 4.16: The results of image segmentation before and after style translation. Real-to-synthetic style transfer reduces the saturation of the incidental filament strings and reflections on the printing platform, which, in turn, affects the results of semantic segmentation. 110

Figure 5.1: Traditional slicing approach. The nozzle sequentially fills the inner and outer regions of the layer (kidney model [44]) with material and moves to a new layer only after the completion of the current one. If the layer consists of several colored regions – the nozzle performs additional material extrusion (purging) for each transition between colors. This removes the remaining melted plastic in the nozzle and ensures sharp color transitions and stable filament flow. 124

Figure 5.2: Color coding of different types of extruded segments. During the slicing process, each extruded segment is assigned a series of specific properties that determine line length and type, material color, print temperature, fan speed, etc.: *a)* – sliced multi-color model with support, where each color represents a different material, *b)* – color coded extruded segments (green – support, red – internal infill, yellow – internal perimeter, orange – external perimeter, and purple – solid infill). 125

Figure 5.3: Fabrication order of a multi-color part: *a)* – source model, consisting of three colors (color regions R0–R2) and four layers (L0–L3), *b)* – schematic side view of the source model, *c)* – traditional (default) fabrication order with 8 tool changes (marked as red arrows), *d)* – fabrication order produced by the proposed tool clustering method (4 tool changes, marked as red arrows). 127

Figure 5.4: Tool clustering algorithm. The developed algorithm in action on the example of a simple part in the shape of a rectangular parallelepiped. The part is shown in the side view and each of the four layers (L0...L3) consists of three different colored regions (R0...R2). Starting from the first region, the program analyzes the possibility of printing a candidate region (the top layer of the same color as the previously selected region) according to two criteria: the allowable safe height and the presence of intersections in the unfinished underlying regions of other colors. If it is possible to print a candidate region – the region is added to a linked list (print map) and marked as processed (red crosshair). If printing is not possible, the candidate is discarded, and the pointer moves to the next unprocessed region. 128

Figure 5.5: Converting the initial printing map into a clustered sequence. As a result of the analysis, the initial printing map in the form of a linked list, based on the default slicing parameters, is transformed into a tool clustered batched map. 129

Figure 5.6: Conventional *(a)* vs clustered *(b)* fabrication approaches. A cross section of a rectangular piece consisting of three layers, each containing three different materials (yellow, red, and blue colors) is shown. Gradient regions represent wiping layers with material transitions. The black arrows indicate travel nozzle moves without extruding material, while the white arrows within the layers illustrate nozzle movements with material extrusion... 130

Figure 5.7: Time *(a)* and material *(b)* losses. The colored surfaces are the default losses without layer batching depending on number of materials used and the fabrication part size, while the lines projected onto the orthogonal planes represent the losses after applying layer batching 132

Figure 5.8: Internal data flow of the slicing process in the PrusaSlicer "libslic3r" library. The process begins by loading a 3-D model and generating a printing object consisting of a set of resulting object and support layers. The data are further used to create a material transition tower and is converted into a set of corresponding layer-by-layer instructions for moving the extrusion tool... 133

Figure 5.9: High-level slicing pipeline of the developed PrusaSlicer upgrade. The main functionality of the developed ITC algorithm is integrated in the G-code generation process... 134

Figure 5.10: Tool clustering control panel in the PrusaSlicer GUI. The main control variables are the allowable height of a stack of layers processed within one material transition and the maximum acceptable intersection with underlying regions of other colors.. 135

Figure 5.11: Colormap visualization of the clustered sliced model in the PrusaSlicer GUI: *a)* – five-color source model, *b)* – sliced material regions grouped into layer clusters, *c)* – critical region intersections. 135

Figure 5.12: Nozzle system and its clearance. The extrusion system of each printer has a certain gap and filled space in the immediate vicinity of the nozzle, which prevents free movement in non-planar printing *(a)*. When printing batched layers, subsequent surrounding regions are damaged in most cases. The maximum damage volume *(b)* is equal to the volume of the intersection with the nozzle, which can be represented in equation (5-6). Here, the parameters P1, P2 and P3 define the physical constraints imposed by the dimensions of the print head system, $\boldsymbol{h}$ is the batch height, and $\boldsymbol{r}$ is the nozzle radius at a given height $\boldsymbol{h}$. 136

Figure 5.13: Multi-color experimental model. The height of the model was gradually increased from 10 mm (solid color parts) to 20 mm (semitransparent gray volumes) to analyze material, time and energy consumption depending on the number of fabricated layers. 138

Figure 5.14: Analysis of time *(a)* and material *(b)* consumption during the manufacturing of the experimental models depending on the layer batch size. 139

Figure 5.15: Dependence of energy consumption, fabrication time, and number of tool changes on the height of the parts manufactured using variable layer batch size. Batch size B=1 corresponds to conventional layer-wise manufacturing, and batch sizes B=4 and B=6 correspond to the developed ITC method with grouping of fabricated layers of the same material into batches of 4 and 6 layers, respectively. 140

Figure 5.16: The boundaries of the material of adjacent batches. The layer thickness is 0.2 mm, and the B-parameter indicates the number of layers in a single batch. 141

Figure 5.17: Comparative time consumption when fabricating a set of models. 142

Figure 5.18: Comparative material consumption when fabricating a random set of models. 143

Figure 5.19: Default *(a)* vs tool clustering *(b)* fabrication of the multi-color mandala model. With the default method, tool changes occur on each layer, while with the ITC method, tool changes occur after each batch is printed. 144

Acronyms

2-D	Two-Dimensional
3-D	Three-Dimensional
3MF	3-D Manufacturing Format
4-D	Four-Dimensional
AHC	Agglomerative Hierarchical Clustering
AI	Artificial Intelligence
AM	Additive Manufacturing
ASTM	American Society for Testing and Materials
BSDF	Bidirectional Scattering Distribution Function
CAD	Computer-Aided Design
CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Convolutional Neural Network
DA	Domain Adaptation
DRAM	Distributed Recycling for Additive Manufacturing
EM	Expectation-Maximization
FDM	Fused Deposition Modeling
FFF	Fused Filament Fabrication
FPF	Fused Particle Fabrication
GAN	Generative Adversarial Network

GMM	Gaussian Mixture Model
GUI	Graphical User Interface
HOG	Histograms of Oriented Gradients
ICP	Iterative Closest Point
IoU	Intersection Over Union
ISO	International Organization for Standardization
ITC	Interlayer Tool Clustering
LCA	Life Cycle Analysis
LM	Leung-Malik
LPBF	Laser Powder Bed Fusion
mIoU	Mean Intersection Over Union
MMU	Multi Material Unit
MTM	Multi-Template Matching
OSF	Open Science Framework
PLA	Polylactic Acid
RAMPS	RepRap Arduino Mega Polulu Shield
RepRap	Self-Replicating Rapid Prototype
RGB	Red, Green, Blue
RGBG	Red, Green, Blue, Green
ST	Style Transfer

STL	Standard Tessellation Language
t-SNE	T-distribution Stochastic Neighbor Embedding

Chapter 1

1 Introduction

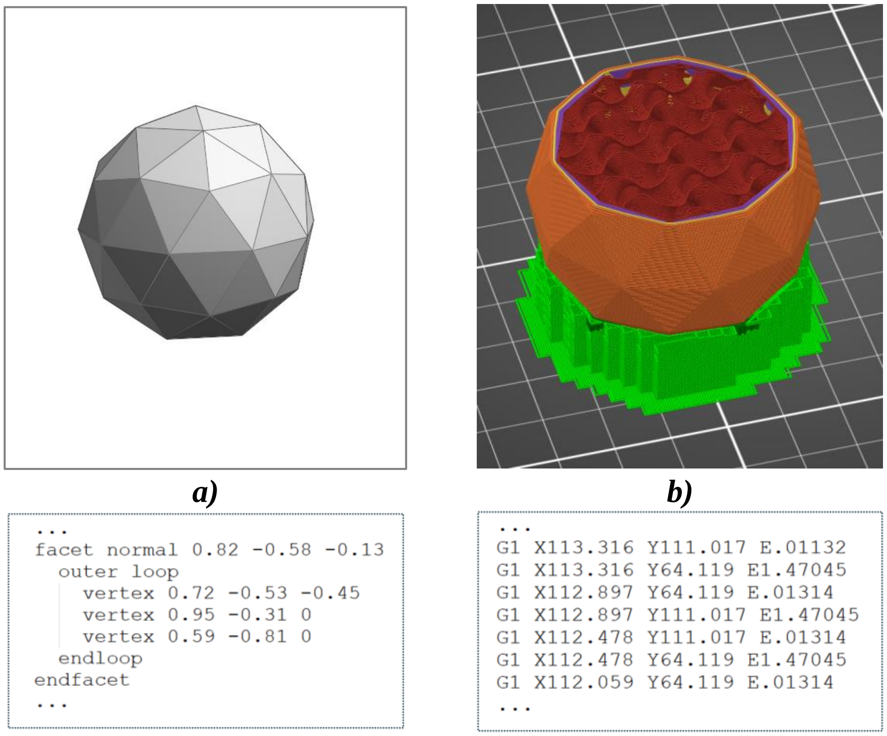
1.1 FFF AM Technology

Additive manufacturing (AM) [1], or 3-D printing, stands as one of the foundational elements within the framework of Industry 4.0 [2,3]. Since the original inventions in 1970-80s [4,5], the primary emphasis of the technology has revolved around single-material rapid prototyping, a generation of a draft part representation to find the optimal shape before its final release. Subsequent advances in AM technology and materials development, as well as market demand for enhanced functionality and high geometric complexity, however, have shifted the focus towards the direct production of final products and the simultaneous use of multiple materials. Modern additive manufacturing is firmly established in diverse areas such as medicine, food industry, robotics, automotive, aerospace, and many others [6-18].

According to the ISO/ASTM 529000-21 standard [21], AM can be categorized based on fabrication principles, feedstock types, energy sources, and fabrication volumes. Each of the various processing techniques has its own advantages and challenges [22,23]. Fused filament fabrication (FFF), however, is one of the most popular technologies and offers great opportunities for general users to become producing consumers (prosumers) [24–26] due to its simplicity and availability [27,28].

FFF AM involves a series of steps to convert a digital design file into a physical object: 1) creating a 3-D model, 2) exporting the design file, 3) preparing the model for printing, 4) slicing the model, 5) configuring the 3-D Printer, 6) printing the model, 7) post-processing. The key steps are illustrated in Figure 1.1.

During the creation step, a 3-D object model represents a Computer-aided design (CAD) hierarchy implemented in software (OpenSCAD, FreeCAD, Blender, etc.). Later, the model is converted into a tessellated mesh (STL, OBJ, or 3MF file format) containing its vertices and normals to represent the spatial volume of the object design. The tessellated mesh is then imported into slicing software (PrusaSlicer, Cura, Simplify3D, etc.), which allows scaling and positioning the developed virtual object onto a building plane, as well as setting up the required fabrication parameters. Next, the slicing represents a process where the spatial model mesh is converted into a G-code, a set of instructions for a 3-D printer. The post-processing step consists of removing the printed object from the building platform (print bed) and removing the support material.



a) *b)*

```
...
facet normal 0.82 -0.58 -0.13
  outer loop
    vertex 0.72 -0.53 -0.45
    vertex 0.95 -0.31 0
    vertex 0.59 -0.81 0
  endloop
endfacet
...
```

```
...
G1 X113.316 Y111.017 E.01132
G1 X113.316 Y64.119 E1.47045
G1 X112.897 Y64.119 E.01314
G1 X112.897 Y111.017 E1.47045
G1 X112.478 Y111.017 E.01314
G1 X112.478 Y64.119 E1.47045
G1 X112.059 Y64.119 E.01314
...
```

c) *d)*

Figure 1.1: Key steps to transform a digital design into a physical object: *a)* – tessellated mesh of an object model, *b)* – slicing the object before 3-D printing (green – support, red – internal infill, yellow – internal perimeter, orange – external perimeter, and purple – solid infill), *c)* – part of an object's tessellated mesh file, representing information about its vertices and normals, *d)* – part of an object's G-code file that represents information about the nozzle's trajectory and the amount of material being extruded.

The tessellated mesh of a model contains only information about the external boundaries of the object, while the G-code adds geometric structures both inside and outside the object to enable physical fabrication.

During slicing, each extruded segment (a set of G-code lines) for each fabrication layer has its own characteristic (Figure 1.1), such as type (external and internal perimeters, internal, solid, and bridge infills, support material and interface, skirt/brim, wipe tower, and others) and tool (material, color). Each segment type can be assigned a specific print speed and temperature mode. Further, during 3-D printing, the printer nozzle sequentially fills the inner and outer areas of the layer with material and moves on to a new layer only after the current one is completed.

Multi-color and multi-material FFF 3-D printing greatly expand the horizons of manufacturing, providing innovative design opportunities and enhancing the quality of products through the optimal combination of properties, which is often impossible to achieve with traditional methods [19]. The main fabrication materials are polymers, metals, ceramics, and biological substances. The integration of electrically functional elements and shape memory polymers, in turn, marked the beginning of 3-D printing of smart devices and 4-D printing, respectively [20].

The G-code is thus a complete set of instructions for the 3-D printer to create a physical object. The main disadvantage of the traditional approach to manufacturing, however, is the fact that after slicing, the G-code is loaded onto the printer’s memory card without the possibility of further modification in the event of abnormal conditions.

1.2 FFF AM Challenges

Despite its great potential, FFF AM faces technical, economic, and regulatory challenges that affect its adoption, efficiency, and product quality. The technical side includes material limitations, accuracy, fault tolerance, manufacturing speed, scalability, post-processing requirements, etc. Overcoming these challenges requires advances in materials science, improvements in printing technology, and the development of comprehensive standards and regulations. By addressing these problems, the industry can improve the reliability and efficiency of FFF AM technologies.

The focus of this work is on an approach to detecting defects in polymer 3-D printing to improve production reliability. With the current exponential manufacturing growth, the amount of plastic waste produced could reach 250 billion tons by 2050 [29], significant quantities of which cause pollution of the natural environment on land and in the ocean [30]. This is not only caused by additional plastic products but also by disturbing failure rates. Early work on self-built RepRaps estimated an average 20% failure rate [31] and more recent values of about 10% [32], which is in the range (1–20%) of polling of the Reddit "r/3Dprinting" community [33]. Although the cost of printing with an FFF-based 3-D printer is trivial compared to the other AM techniques, printing errors are substantial enough to impact the economic and environmental merits of the approach.

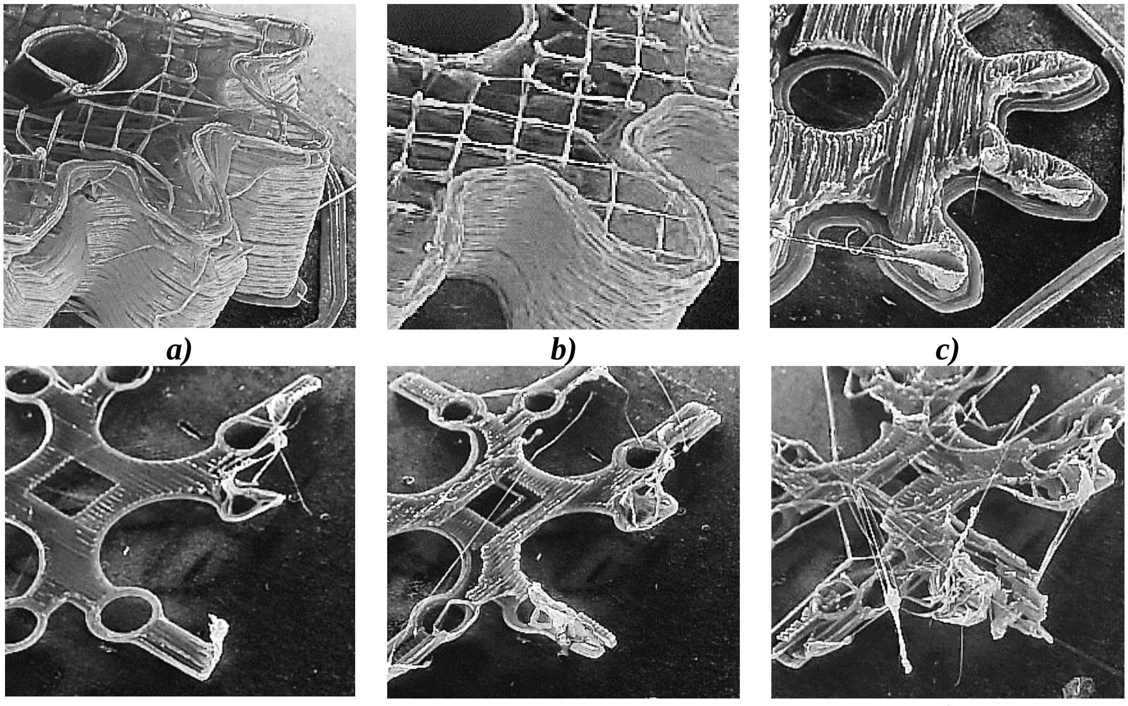


Figure 1.2: Example of common 3-D printing failures a) – blocked nozzle, b) – under-extrusion, c) – over-extrusion, d) – adhesion problem, e) – displacement of individual layers of the printing part, f) – separation and shift of the printing part from the printing surface ("spaghetti problem").

The probability of a manufacturing defect increases with the size and print time of the object (e.g., using large-scale fused filament printers [34] or products [35,36], or fused granule printers [37,38]), which can magnify the waste materials created from even a small percentage of failures. Despite evidence that AM in distributed manufacturing may reduce environmental impacts [39–43], both economic and environmental aftermaths largely depend on manufacturing success rate.

The most common failures in 3-D printing are nozzle clogging, filament run-out, under- and over-extrusion (reducing or increasing the flow of material, respectively, leading to object deformations), adhesion problems, displacement of individual layers or the entire part, etc. (Figure 1.2). All of these problems require the attention of the 3-D printer operator and when they occur, the manufactured object is usually reprinted with new settings.

Many defects can occur in combination and also influence the appearance of other failures. 3-D printing communities actively discuss manufacturing defects and possible ways to their elimination [44–46], based on previous accumulated experience. Due to the variety of available polymer materials and printing parameters, as well as object designs and 3-D printers produced, eliminating such defects is difficult to automate.

1.3 Research Objectives

To strengthen the position of AM and ensure the proper quality of the final product, developments are needed in the field of continuous intelligent monitoring of fabrication processes, as well as methods of adaptive control.

The objectives of this work are summarized as follows:

1. Creation of a basis for an integrated approach to layer-wise vision-based analysis of FFF AM processes and adaptive 3-D printing control that allows updating the G-code during manufacturing.
2. Development of a physics-based simulation system to utilize synthetic visual data and compare real images with photorealistic renders.
3. Creation of a system for semantic segmentation of a manufactured part and its structural elements to reduce the requirements for rigid linking of the camera to visual markers for monitoring tasks.
4. Development of a fabrication optimization approach for multi-color 3-D printing in order to reduce material and time costs.

Having a way to automatically detect critical errors will significantly reduce material waste, as well as energy and time spent on remanufacturing failed prints. In order to mature the FFF-based 3-D printing quality control, this study introduces various computer vision systems that enable layer-wise analysis of 3-D printing processes to segment structural elements and track manufacturing errors. Regarding multi-color printing, this study introduces an interlayer tool clustering (ITC) method that optimizes the slicing procedure to achieve significant savings in time and materials. This approach also reduces the likelihood of technical failures by minimizing the number of tool changes (material transitions).

The presented developments formed the basis for the concept of multifaceted visual analysis of 3-D printing processes. This will help improve FFF AM technology and reduce the amount of time, materials, and energy required to fabricate physical objects.

1.4 Thesis Outline and Contributions

This research develops a computer-vision-based algorithmic framework for layer-wise monitoring and analysis of 3-D printing processes. This dissertation presents conceptual approaches to semantic segmentation of structural elements of manufactured objects, layer-wise detection of printing errors, and fabrication optimization to reduce the likelihood of failure in multi-color 3-D printing. The main contributions of each chapter are summarized as follows:

Chapter 2. Computer Vision-based Layer-wise 3-D Printing Analysis: An open source computer vision-based hardware structure and software algorithm, which analyzes layer-wise the 3-D printing processes, tracks printing errors, and generates appropriate printer actions to improve reliability. This approach is built upon multiple stage monocular image examination, which allows monitoring both the external shape of the printed object and internal structure of its layers. Starting with the side-view height validation, the developed program analyzes the virtual top view for outer shell contour correspondence using the multi-template matching and iterative closest point algorithms, as well as inner layer texture quality clustering the spatial-frequency filter responses with Gaussian mixture models and segmenting structural anomalies with the agglomerative hierarchical clustering algorithm. This allows evaluation of both global and local parameters of the printing modes. The experimentally verified analysis time per layer is less than one minute, which can be considered a quasi-real-time process for large prints. The systems can work as an intelligent printing suspension tool designed to save time and material. The highlighted contributions are presented in the following list:

- Visual servoing platform with a monocular multi-stage image analysis;
- Control algorithm preventing critical failures during 3-D printing;
- Tracking printing errors on the interior and exterior parts of manufactured objects.

Chapter 3. Interlayer Anomaly Detection Based on HOG-features and Synthetic Images: A method for detecting 3-D printing anomalies by comparing images of printed layers from a stationary monocular camera with G-code-based reference images of ideal 3-D printing processes generated with a physics-based rendering engine. Recognition of visual deviations was accomplished by analyzing the similarity of histograms of oriented gradients (HOG) of local image areas. The developed technique requires preliminary modeling of the working environment to achieve the best match for orientation, color rendering, lighting, and other parameters of the printed part. The output parameter is a level of mismatch between printed and synthetic reference layers. Twelve similarity measures were implemented and compared for their effectiveness at detecting 3-D printing errors on six different representative failure types (local infill defects, presence of a foreign body in the layer, spaghetti problem, separation and shift of the printing part from the working surface, defects in thin walls, and layer shift) and their control error-free print images. The highlighted contributions are presented in the following list:

- Automated G-code-based system for generating layer-wise synthetic images;
- Anomaly detection in layer-wise 3-D printing based on synthetic references and similarity metrics of local histograms of oriented gradients;
- Detection of critical errors in the early stages of their occurrence.

Chapter 4. Synthetic-to-real Composite Semantic Segmentation in AM: A method of using physics-based rendering for automated labeled image dataset generation for real image segmentation in AM. Multi-class semantic segmentation experiments were carried out based on the U-Net model and the cycle generative adversarial network. The test results demonstrated the capacity of this method to detect such structural elements of 3-D-printed parts as a top (last printed) layer, infill, shell, and support. A basis for further segmentation system enhancement by utilizing image-to-image style transfer and domain adaptation techniques was also considered. The application of computer vision and machine learning for semantic segmentation of the structural elements of 3-D-printed products can improve real-time failure analysis systems and potentially reduce the number of defects by providing additional tools for in situ corrections. The highlighted contributions are presented in the following list:

- Analysis of 3-D printer users' activity over 2.3 years (5.6 million print jobs with a 24% failure rate);
- Automated G-code-based labeled synthetic dataset generation;
- Semantic segmentation of background, printed part, top (last printed) layer, infill, shell, and support categories;
- Analysis of image-to-image style transfer capabilities.

Chapter 5. Tool change Reduction in Multi-color 3-D Printing: An alternative fabrication approach based on interlayer tool clustering (ITC) is presented here for the first time, which is compatible with any commercial 3-D printer without the need for hardware modifications. The theoretical time, mass and energy savings are calculated and validated with a series of experiments to evaluate the proposed algorithm qualitatively and quantitatively. The results show the novel ITC method can significantly increase the efficiency of multi-material printing, with an average 1.7-fold reduction in material used, and an average 1.4-fold reduction in both time and 3-D printing energy use. In addition, this approach reduces the likelihood of technical failures in the manufacturing of the entire part by reducing the number of tool changes, or material transitions, on average by 2.4 times.

- Novel fabrication method (interlayer tool clustering) for multi-material 3-D printing implemented in the open source PrusaSlicer;

- Average 1.7-fold reduction in material use, 1.4-fold reduction in both time and energy use;
- Number of tool changes can be reduced by an average of 2.4 times.

1.5 Bibliography

[1] I. Gibson, D.W. Rosen, B. Stucker, M. Khorasani, Additive manufacturing technologies. Cham, Switzerland: Springer; 2021.

[2] B. Esmaeilian, S. Behdad, B. Wang, The evolution and future of manufacturing: A review, J. Manuf. Syst. 39 (2016) 79–100. https://doi.org/10.1016/j.jmsy.2016.03.001.

[3] M. Ghobakhloo, The future of manufacturing industry: a strategic roadmap toward Industry 4.0, J. Manuf. Technol. Manag. 29 (2018) 910–936. https://doi.org/10.1108/JMTM-02-2018-0057.

[4] Charles W. Hull, Apparatus for production of three-dimensional objects by stereolithography, 4,575,330, 1986. https://patents.google.com/patent/US4575330A/en (accessed August 31, 2023)

[5] Pierre Alfred Leon Ciraud, Method and device for manufacturing any articles from any meltable material, DE2263777A1, n.d. https://patents.google.com/patent/DE2263777A1/en (accessed September 8, 2023).

[6] S. Ford, T. Minshall, Invited review article: Where and how 3D printing is used in teaching and education, Addit. Manuf. 25 (2019) 131–150. https://doi.org/10.1016/j.addma.2018.10.028.

[7] C.-Y. Liaw, M. Guvendiren, Current and emerging applications of 3D printing in medicine, Biofabrication. 9 (2017) 024102. https://doi.org/10.1088/1758-5090/aa7279.

[8] Mark Zastrow, 3D printing gets bigger, faster and stronger, Nature. 578 (2020) 20–23.

[9] J. Jeong, H. Park, Y. Lee, J. Kang, J. Chun, Developing parametric design fashion products using 3D printing technology, Fash. Text. 8 (2021) 22. https://doi.org/10.1186/s40691-021-00247-8.

[10] I. Dankar, A. Haddarah, F.E.L. Omar, F. Sepulcre, M. Pujolà, 3D printing technology: The new era for food customization and elaboration, Trends Food Sci. Technol. 75 (2018) 231–242. https://doi.org/10.1016/j.tifs.2018.03.018.

[11] J.M. Pearce, Cut costs with open-source hardware, Nature. 505 (2014) 618–618. https://doi.org/10.1038/505618d.

[12] P. Wu, J. Wang, X. Wang, A critical review of the use of 3-D printing in the construction industry, Autom. Constr. 68 (2016) 21–31. https://doi.org/10.1016/j.autcon.2016.04.005.

[13] R. Orange Kedem, N. Opatovski, D. Xiao, B. Ferdman, O. Alalouf, S. Kumar Pal, Z. Wang, H. Von Der Emde, M. Weber, S.J. Sahl, A. Ponjavic, A. Arie, S.W. Hell, Y. Shechtman, Near index matching enables solid diffractive optical element fabrication via additive manufacturing, Light Sci. Appl. 12 (2023) 222. https://doi.org/10.1038/s41377-023-01277-1.

[14] T. Gissibl, S. Thiele, A. Herkommer, H. Giessen, Sub-micrometre accurate free-form optics by three-dimensional printing on single-mode fibres, Nat. Commun. 7 (2016) 11763. https://doi.org/10.1038/ncomms11763.

[15] T.J. Wallin, J. Pikul, R.F. Shepherd, 3D printing of soft robotic systems, Nat. Rev. Mater. 3 (2018) 84–100. https://doi.org/10.1038/s41578-018-0002-2.

[16] J.C. Vasco, Additive manufacturing for the automotive industry, in: Addit. Manuf., Elsevier, 2021: pp. 505–530. https://doi.org/10.1016/B978-0-12-818411-0.00010-0.

[17] J. Pierre, F. Iervolino, R.D. Farahani, N. Piccirelli, M. Lévesque, D. Therriault, Material extrusion additive manufacturing of multifunctional sandwich panels with load-bearing and acoustic capabilities for aerospace applications, Addit. Manuf. 61 (2023) 103344. https://doi.org/10.1016/j.addma.2022.103344.

[18] D. Wangpraseurt, S. You, F. Azam, G. Jacucci, O. Gaidarenko, M. Hildebrand, M. Kühl, A.G. Smith, M.P. Davey, A. Smith, D.D. Deheyn, S. Chen, S. Vignolini, Bionic 3D printed corals, Nat. Commun. 11 (2020) 1748. https://doi.org/10.1038/s41467-020-15486-4.

[19] M. Rafiee, R.D. Farahani, D. Therriault, Multi-Material 3D and 4D Printing: A Survey, Adv. Sci. 7 (2020) 1902307. https://doi.org/10.1002/advs.201902307.

[20] J. Lee, H.-C. Kim, J.-W. Choi, I.H. Lee, A review on 3D printed smart devices for 4D printing, Int. J. Precis. Eng. Manuf.-Green Technol. 4 (2017) 373–383. https://doi.org/10.1007/s40684-017-0042-x.

[21] F42 Committee, Terminology for Additive Manufacturing - General Principles - Terminology, ASTM International. https://doi.org/10.1520/F3177-21.

[22] D. Han, H. Lee, Recent advances in multi-material additive manufacturing: methods and applications, Curr. Opin. Chem. Eng. 28 (2020) 158–166. https://doi.org/10.1016/j.coche.2020.03.004.

[23] A. Bandyopadhyay, B. Heer, Additive manufacturing of multi-material structures, Mater. Sci. Eng. R Rep. 129 (2018) 1–16. https://doi.org/10.1016/j.mser.2018.04.001.

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

[25] E.E. Petersen, J. Pearce, Emergence of Home Manufacturing in the Developed World: Return on Investment for Open-Source 3-D Printers, Technologies. 5 (2017) 7. https://doi.org/10.3390/technologies5010007.

[26] J. Pearce, J.-Y. Qian, Economic Impact of DIY Home Manufacturing of Consumer Products with Low-cost 3D Printing from Free and Open Source Designs, Eur. J. Soc. Impact Circ. Econ. 3 (2022) 1–24. https://doi.org/10.13135/2704-9906/6508.

[27] Martin Placek, Most used 3D printing technologies worldwide 2021, (2023). https://www.statista.com/statistics/560304/worldwide-survey-3d-printing-top-technologies/ (accessed September 1, 2023).

[28] Sophian Beyerlein, M. Aboushama, Evaluation of Continuous Fiber Reinforcement Desktop 3D Printers Desktop 3D Printers Overview, (2020). https://doi.org/10.13140/RG.2.2.16640.87040.

[29] R. Geyer, J.R. Jambeck, and K.L. Law, Production, Use, and Fate of all Plastics Ever Made, Sci. Adv., 2017, 3(7), e1700782, doi: 10.1126/sciadv.1700782.

[30] J.R. Jambeck, R. Geyer, C. Wilcox, T.R. Siegler, M. Perryman, A. Andrady, R. Narayan, and K.L. Law, Plastic Waste Inputs From Land Into the Ocean, Science, 2015, 347(6223), pp. 768–771, doi: 10.1126/science.1260352.

[31] B.T. Wittbrodt, A.G. Glover, J. Laureto, G.C. Anzalone, D. Oppliger, J.L. Irwin and J.M. Pearce, 2013. Life-cycle economic analysis of distributed manufacturing with open-source 3-D printers, Mechatronics, 23(6), 713-726. https://doi.org/10.1016/j.mechatronics.2013.06.002.

[32] J. King, The True Cost of Running a Desktop 3D Printer. https://3dprinthq.com/costrunning-desktop-3d-printer, 2017 (accessed 01 March 2023).

[33] Reddit 3D Printing Community, https://www.reddit.com/r/3Dprinting/comments/57ycv0/poll_whats_your_average_failure_rate, (accessed 01 March 2020).

[34] H.D. Kang, Analysis of furniture design cases using 3D printing technique, The Journal of the Korea Contents Association, 2015, 15(2), 177-186.

[35] J.K. Bow, N. Gallup, S.A. Sadat, and J.M. Pearce, Open source surgical fracture table for digitally distributed manufacturing, PloS one, 2022, 17(7), e0270328.

[36] J.I. Novak and J. O’Neill, A design for additive manufacturing case study: fingerprint stool on a BigRep ONE, Rapid Prototyping Journal, 2019, 25(6), pp. 1069–1079.

[37] A. Petsiuk, B. Lavu, R. Dick, and J.M. Pearce, Waste Plastic Direct Extrusion Hangprinter, Inventions, 2022, 7(3), p.70.

[38] A.L. Woern, D.J. Byard, R.B. Oakley, M.J. Fiedler, S.L. Snabes, and J.M. Pearce, Fused particle fabrication 3-D printing: Recycled materials’ optimization and mechanical properties, Materials, 2018, 11(8), p.1413.

[39] M. Kreiger and J.M. Pearce, 2013. Environmental impacts of distributed manufacturing from 3-D printing of polymer components and products, MRS Online Proceedings Library Archive, 1492, 85-90. https://doi.org/10.1557/opl.2013.319.25

[40] M. Kreiger and J.M. Pearce, 2013. Environmental life cycle analysis of distributed threedimensional printing and conventional manufacturing of polymer products, ACS Sustainable Chemistry & Engineering, 1(12), 1511-1519. https://doi.org/10.1021/sc400093k.

[41] D. Chen, S. Heyer, S. Ibbotson, K. Salonitis, J.G. Steingrímsson and S. Thiede, 2015. Direct digital manufacturing: definition, evolution, and sustainability implications, Journal of Cleaner Production, 107, 615-625. https://doi.org/10.1016/j.jclepro.2015.05.009.

[42] C. Kohtala and S. Hyysalo, 2015. Anticipated environmental sustainability of personal fabrication, Journal of Cleaner Production, 99, 333-344. https://doi.org/10.1016/j.jclepro.2015.02.093.

[43] S. Zhong and J.M. Pearce, 2018. Tightening the loop on the circular economy: Coupled distributed recycling and manufacturing with recyclebot and RepRap 3-D printing, Resources, Conservation and Recycling, 128, 48-58. https://doi.org/10.1016/j.resconrec.2017.09.023.

[44] Prusa Research: Print Quality Troubleshooting. https://help.prusa3d.com/category/print-quality-troubleshooting_225 (accessed 02 July 2024).

[45] Troubleshooting Guide to Common 3D Printing Problems. https://all3dp.com/1/common-3d-printing-problems-troubleshooting-3d-printer-issues, 2020 (accessed 01 March 2020).

[46] MatterHackers: 3D Printer Troubleshooting Guide, 2016, https://www.matterhackers.com/articles/3d-printer-troubleshootingguide (accessed 01 March 2020).

Chapter 2

2 Computer Vision-based Layer-wise 3-D Printing Analysis

This chapter[1] is adapted from the “Open Source Computer Vision-based Layer-wise 3-D Printing Analysis” with minor modifications to the version published in Additive Manufacturing, vol. 36, 101473, 2020, DOI:10.1016/j.addma.2020.101473.

2.1 Abstract

The paper describes an open source computer vision-based hardware structure and software algorithm, which analyzes layer-wise 3-D printing processes, tracks printing errors, and generates appropriate printer actions to improve reliability. This approach is built upon multiple-stage monocular image examination, which allows monitoring both the external shape of the printed object and internal structure of its layers. Starting with the side-view height validation, the developed program analyzes the virtual top view for outer shell contour correspondence using the multi-template matching and iterative closest point algorithms, as well as inner layer texture quality clustering the spatial-frequency filter responses with Gaussian mixture models and segmenting structural anomalies with the agglomerative hierarchical clustering algorithm. This allows evaluation of both global and local parameters of the printing modes. The experimentally verified analysis time per layer is less than one minute, which can be considered a quasi-real-time process for large prints. The systems can work as an intelligent printing suspension tool designed to save time and material. However, the results show that the algorithm provides a means to systematize *in situ* printing data as a first step in a fully open source failure correction algorithm for additive manufacturing.

---

[1] A version of this chapter has been published in Additive Manufacturing journal. A. Petsiuk, J.M. Pearce, Open source computer vision-based layer-wise 3D printing analysis. Additive Manufacturing, vol. 36, no. 101473, 2020, doi:10.1016/j.addma.2020.101473.

2.2 Introduction

Despite a long evolution of additive manufacturing (AM), starting from the first patent in 1971 [1], 3-D printing technology has only recently exploded in popularity due to the radical decreases in costs brought on by the introduction of the self-replicating rapid prototyper (RepRap) 3-D printer [2-4]. With the generalized material extrusion printing process called fused filament fabrication (FFF) technology gaining prominence in the field with the expiration of fused deposition modeling (FDM) patents, FFF now dominates the 3-D printing market for printers in use [5]. Making AM accessible to the masses of consumers has enabled the emergence of a distributed manufacturing paradigm [6-14], where 3-D printing can be used to manufacture open source products for the consumer and by the consumer directly for less (in many cases more than an order of magnitude less) money than purchasing of mass-manufactured proprietary products [10,15-18]. The downloaded substitution values [19, 20] for digital manufacturing with AM of even sophisticated high-end products [21-24] provides a high return on investment [25]. In addition, there is some evidence that AM distributed manufacturing reduces the impact on the environment [26-30]. However, both the economics and environmental impact of distributed manufacturing is heavily impacted by success rate. Early work on self-built RepRaps estimated a 20% failure rate [9] and more recent values of about 10% [31], which is in the range (1–20%) of recent polling of the Reddit “r/3Dprinting” community [32].

Although the cost of printing with an FFF-based 3-D printer is trivial compared to the other AM techniques, printing errors are substantial enough to impact the economic and environmental merits of the approach. To this end, several studies and techniques have been attempted to reduce failure rates. Nuchitprasitchai et al. [33] were able to detect the “blocked nozzle” and “incomplete print” failures for six different objects in five colors. The printed objects have been tested in a single- and double-camera experiments to determine critical 5% deviations in shape and size at every 50th layer. In the subsequent work [34], the authors used three pairs of cameras 120 degrees apart to reconstruct the 3-D surface points of the printed objects at every 30th layer to detect critical 10% size deviations. Garanger et al. [35] implemented a closed-loop control system for a certain additive manufacturing process to reach the acceptable stiffness for leaf spring-like objects.

Delli and Chang [36] proposed a binary (failed/not failed) 3-D printing error classification approach based on a supervised machine learning technique, where the quality check is being performed at critical stages during the printing process. Fastowicz and Okarma [37] developed a quality assessment method based on texture analysis using the gray-level co-occurrence matrix [38-40] and selected Haralick features [41]. In [42], Cummings et al., developed a closed-loop control framework that detects and corrects filament bonding failures by using the ultrasonic sensor and manipulating the print bed temperature during the printing process. Rao et al. [43] developed a heterogeneous sensor framework for real-time surface roughness analysis based on such printing parameters as extruder temperature, layer thickness on build quality, and feed to flow ratio. Jin et al. [44] introduced a system of automatic correction of extrusion defects based on machine learning. The method, however, assumes the presence of a large image database (the order of hundreds of thousands) and performs the analysis of only rectangular print fragments for over- and under extrusion [44]. He et al. [45] developed a layer-wise vision-based system for monitoring the external geometric profile of FFF fabricated parts. This method covers simple geometric shapes and does not take into account the internal structure of the printed layers. Finally, the authors of “The Spaghetti Detective” project [46] utilize the webcam to detect critical “spaghetti” failures based on machine learning techniques. This method allows stopping the printing process and notifying the user in case of fatal errors such as model detachment and nozzle positioning failure. However, this does not cover a wide range of possible malfunctions and does not allow implementing any correction algorithms.

In the more mature areas of AM with higher-priced materials and 3-D printers, various methods of quality control have been instituted to minimize print failure. Scime and Beth [47] introduced an in-situ anomaly detection approach based on the unsupervised machine learning technique for laser powder bed fusion (LPBF) additive manufacturing. The developed method [47] determines the possible causes for partial fusion failures and provides potential strategies to build quality enhancement in the future. Xiong et al. [48] developed a camera-based system in gas metal arc welding to monitor and control the distance between the nozzle and the top surface by compensating such parameters as the deposition rate and working flat level. Nassar et al. [49] developed a temperature-based inner layer control strategy and analyzed its effects on the hardness and microstructure of the metal printed component. In both [48] and [49], the authors developed closed-loop control capabilities, but the methods, however, focus on microscopic properties without considering the global macrostructure of the object being built. Okaro et al. [50] introduced a semi-supervised machine learning algorithm based on large sets of photodiode data for automatic “faulty” and “acceptable” tensile strength assessment in laser power bed fusion additive manufacturing. Garanger et al. [51] suggested a number of semantic rules within 3-D printing files, which provide desired specifications and, based on material properties, real-time topology and finite element analysis, generate feedback laws for the control system. Yuan et al. [52] developed a two-step machine learning approach to real-time laser track welds assessment in LPBF processes. Sitthi-Amorn et al. [53] introduced a layer-wise correction technique for multi-material 3-D printing. Razaviarab et al. [54] proposed a closed-loop machine learning algorithm to detect layer defects in metal printing.

The key point of the previous works is addressing a limited number of specific cases of local defects without taking into account the global parameters of the printed parts (such as full-scale shift or deformation, deviation of the dimensions of the parts from the calculated ones, etc.). Most of the methods described also do not imply an on-the-fly algorithm for compensating, correcting or eliminating manufacturing failures.

In order to mature the FFF-based 3-D printing quality control to reach that of the more expensive AM technologies, this study presents a free and open source software algorithm and hardware structure based on computer vision, which allows layer-wise analysis of 3-D printing processes to segment and track manufacturing errors, and repairing procedures to generate appropriate printer actions during fabrication. These repair-based actions are designed to minimize the waste of printing material and printing time caused by erroneous printing attempts. The approach is based on monocular multiple-stage image processing that monitors the external shape of the printed object and internal structure of the layers. The developed framework analyzes both global (deformation of overall dimensions) and local (deformation of filling) deviations of print modes, it restores the level of scale and displacement of the deformed layer and introduces a potential opportunity of repairing internal defects in printed layers. The analysis time as a function of layers is quantified and the results are discussed.

2.3 Visual Platform Design

FFF/FDM 3-D printing is the simplest and most common form of additive manufacturing [55]. Increasing the resiliency and quality of FFF printing will thus provide a significant benefit to makers' communities around the world.

Table 2.1, created by the authors, summarizes FFF 3-D printing errors and methods for their elimination guided by makers’ experience with low-cost desktop 3-D printers [56-58]. As can be seen from Table 2.1, print options that affect the possibility of an error can be divided into three categories:

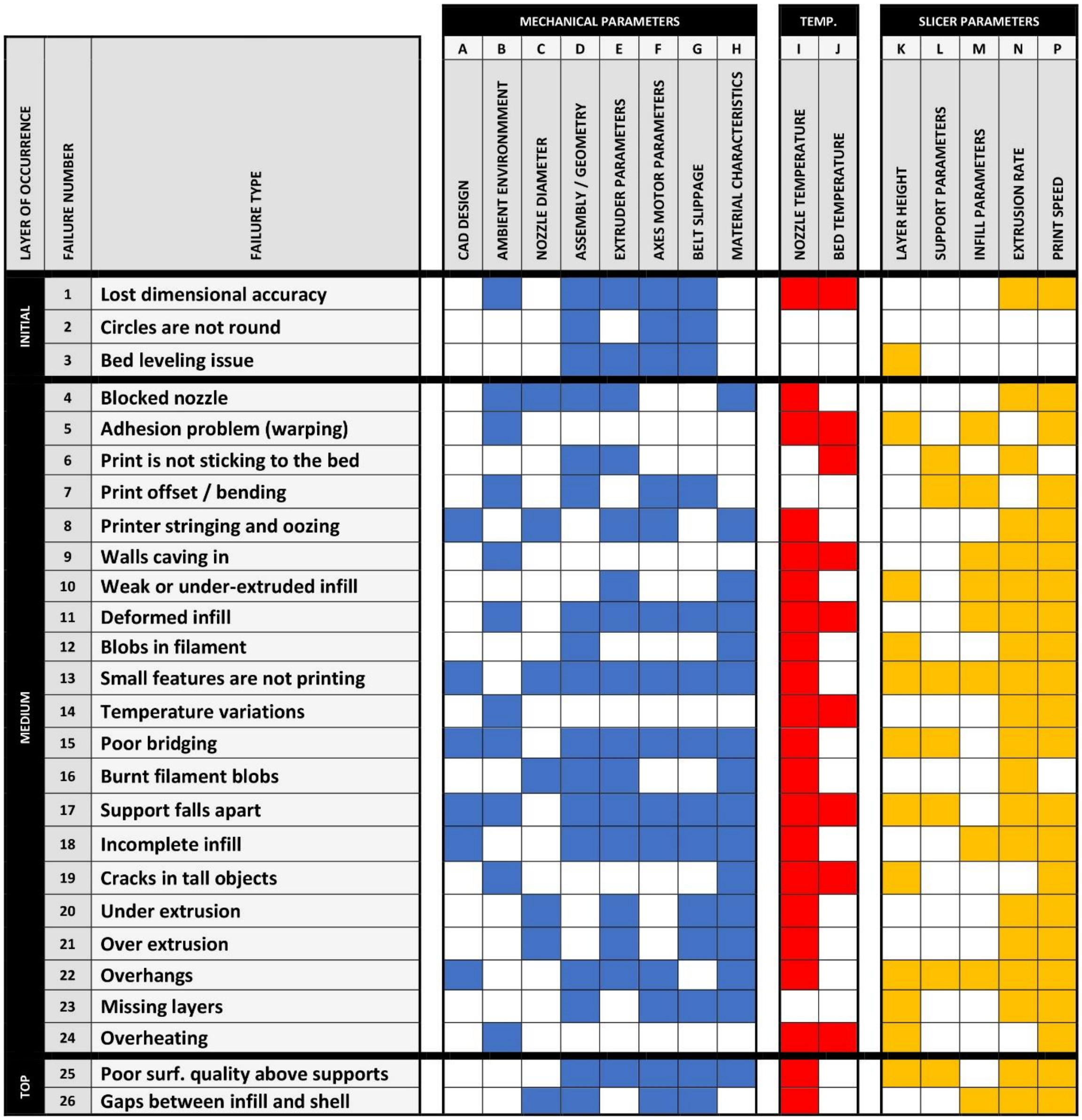
1. Mechanical parameters of the 3-D printer and its environment;
2. Temperature parameters of the fan/nozzle and the printing bed;
3. Algorithm for converting a standard tessellation language (STL) model to G-code instructions (i.e., slicer parameters).

The described three sources [56-58] are the most popular and frequently visited Internet platforms of the makers' community for 3-D printing. These sources already provide the most complete and detailed information about the causes and possible solutions for each printing defect based on previous cumulative experience.

The temperature conditions, feed rate, traveling speed of the extruder, as well as some parameters of the slicing algorithm (such as height of the printed layer, thickness of lines and percentage of their overlapping, etc.) can be controlled by the G-code commands. Thus, by adapting the G-code parameters during printing, it is possible to indirectly compensate for the shortcomings of the temperature regime and slicer parameters. The mechanical parameters of the printer (stability of assembly, the presence of grease in moving parts, belt tension, the electrical voltage of stepper motor drivers, etc.), as well as flaws of the model design, are practically impossible to compensate for during printing. However, having the ability to vary certain 3-D printing options, a computer algorithm can be written to eliminate or reduce the likelihood of failures from the former root causes, the latter need to be addressed by physically fixing the machine.

The existing market for FFF 3-D printers is very diverse and is represented by models of various shapes with both moving and stationary working surfaces (printing beds). To test the developed method, a delta-type RepRap printer was chosen [59], which is an open source development with a fixed working surface, which simplifies the synchronization of the camera with printing processes.

Table 2.1: 3-D printing parameters that may cause failures.



			MECHANICAL PARAMETERS								TEMP.		SLICER PARAMETERS				
			A	B	C	D	E	F	G	H	I	J	K	L	M	N	P
LAYER OF OCCURRENCE	FAILURE NUMBER	FAILURE TYPE	CAD DESIGN	AMBIENT ENVIRONMMENT	NOZZLE DIAMETER	ASSEMBLY / GEOMETRY	EXTRUDER PARAMETERS	AXES MOTOR PARAMETERS	BELT SLIPPAGE	MATERIAL CHARACTERISTICS	NOZZLE TEMPERATURE	BED TEMPERATURE	LAYER HEIGHT	SUPPORT PARAMETERS	INFILL PARAMETERS	EXTRUSION RATE	PRINT SPEED
INITIAL	1	Lost dimensional accuracy		🟦		🟦	🟦	🟦	🟦		🟥	🟥				🟨	🟨
	2	Circles are not round				🟦		🟦	🟦								
	3	Bed leveling issue				🟦	🟦	🟦	🟦				🟨				
MEDIUM	4	Blocked nozzle		🟦	🟦	🟦	🟦			🟦	🟥					🟨	🟨
	5	Adhesion problem (warping)		🟦							🟥	🟥	🟨		🟨		🟨
	6	Print is not sticking to the bed				🟦	🟦					🟥		🟨		🟨	
	7	Print offset / bending		🟦		🟦		🟦	🟦					🟨	🟨		🟨
	8	Printer stringing and oozing	🟦		🟦		🟦	🟦		🟦	🟥					🟨	🟨
	9	Walls caving in		🟦							🟥	🟥			🟨	🟨	🟨
	10	Weak or under-extruded infill					🟦			🟦	🟥		🟨		🟨	🟨	🟨
	11	Deformed infill		🟦		🟦	🟦	🟦	🟦	🟦	🟥	🟥			🟨	🟨	🟨
	12	Blobs in filament				🟦				🟦	🟥		🟨			🟨	🟨
	13	Small features are not printing	🟦		🟦	🟦	🟦	🟦	🟦	🟦	🟥		🟨	🟨	🟨	🟨	🟨
	14	Temperature variations		🟦							🟥	🟥				🟨	🟨
	15	Poor bridging	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟥		🟨	🟨		🟨	🟨
	16	Burnt filament blobs			🟦	🟦	🟦			🟦	🟥					🟨	
	17	Support falls apart	🟦	🟦		🟦	🟦	🟦	🟦	🟦	🟥	🟥	🟨	🟨		🟨	🟨
	18	Incomplete infill	🟦			🟦	🟦	🟦	🟦	🟦	🟥				🟨	🟨	🟨
	19	Cracks in tall objects		🟦						🟦	🟥	🟥	🟨				🟨
	20	Under extrusion			🟦		🟦		🟦	🟦	🟥					🟨	🟨
	21	Over extrusion			🟦		🟦		🟦	🟦	🟥					🟨	🟨
	22	Overhangs	🟦			🟦	🟦	🟦		🟦	🟥		🟨	🟨	🟨	🟨	🟨
	23	Missing layers				🟦		🟦	🟦	🟦			🟨			🟨	🟨
	24	Overheating		🟦							🟥	🟥	🟨				🟨
TOP	25	Poor surf. quality above supports				🟦	🟦	🟦	🟦	🟦	🟥		🟨	🟨		🟨	🟨
	26	Gaps between infill and shell			🟦	🟦		🟦	🟦		🟥				🟨	🟨	🟨

A Michigan Tech Open Sustainability Technology (MOST) Delta RepRap FFF-based 3-D printer [59] with a 250 mm diameter and 240 mm high cylindrical working volume was used (Figure 2.1). It fuses 1.75 mm polylactic acid (PLA) plastic filament under a temperature of 200 °C from a nozzle with a 0.4 mm diameter. The printer operates by RAMPS 1.4 printer controller with an integrated card reader. The MOST Delta operates with 12-tooth T5 belts at 53.33 steps/mm for a Z precision of about 19 microns. The resolution of the printer in an XY plane is a function of distance from apexes, so it changes with distance from the center of the build platform [60].

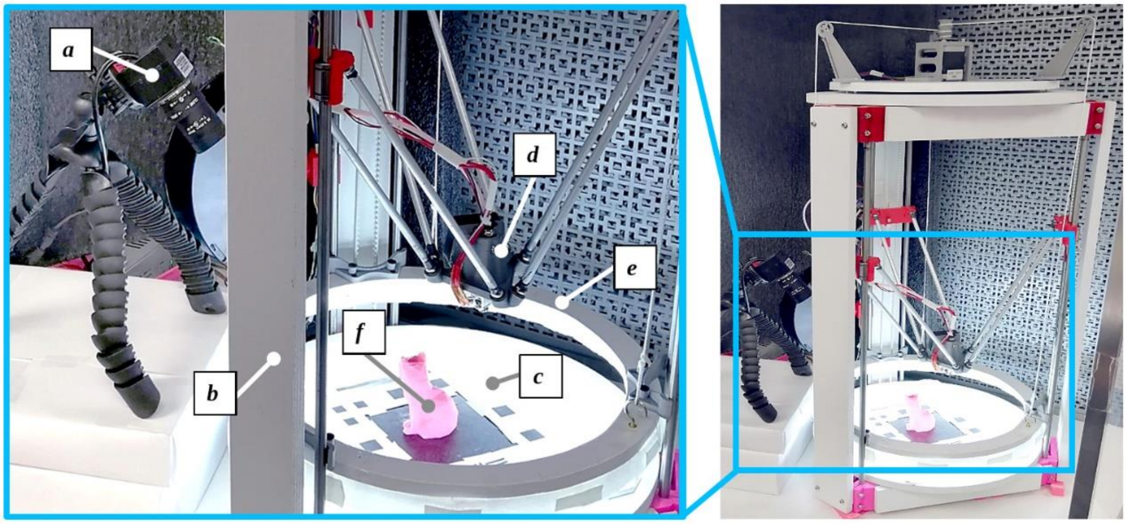


Figure 2.1: Visual platform: working area (left), printer assembly (right): *a)* – camera; *b)* – 3-D printer frame; *c)* – visual marker plate on top of the printing bed; *d)* – extruder; *e)* – movable lighting frame; *f)* –printed part.

The camera is based on 1/2.9 inch (6.23 mm in diagonal) Sony IMX322 CMOS Image Sensor [61]. This sensor consists of 2.24M square 2.8x2.8 μm pixels, 2000 pixels per horizontal line and 1121 pixels per vertical line. IMX322 has a Bayer RGBG color filter pattern (50% green, 25% red, and 25% blue) with 0.46÷0.61 red-to-green and 0.34÷0.49 blue-to-green sensitivity ratios. In operating mode, the camera captures 1280×720 pixel frames at a frequency of 30 Hz. The camera was calibrated on a widely used asymmetric circular grid pattern [62]. The circle grid pattern can provide more accuracy and stability since the calibration technique is based on the detection of the center of gravity of each circle [63, 64].

The printing area is under monocular surveillance (Figure 2.1), where the single camera provides a rectified top view and pseudo-side-view of the printed part (Figure 2.3).

A developed visual marker plate located on top of the printing bed (Figure 2.1) enables the determination of the spatial position of the working area relative to the camera. The plate has a 88×88 mm printing area where seven contrast square markers (15×15 mm and 10×10 mm) build a reference frame for the camera. A computed camera pose in homogeneous coordinates allows the computation of a one-to-one relationship between the Euclidean world points $\mathbb{R}^n$ recorded in the G-code or in the STL model and the planar image points captured by the camera by applying projective transformations (Figure 2.2). In computer vision problems, projective transformations are used as a convenient way of representing the real 3-D world by extending it to the three-dimensional projective space $\mathbb{P}^n$, where its points are homogeneous vectors [65,66].

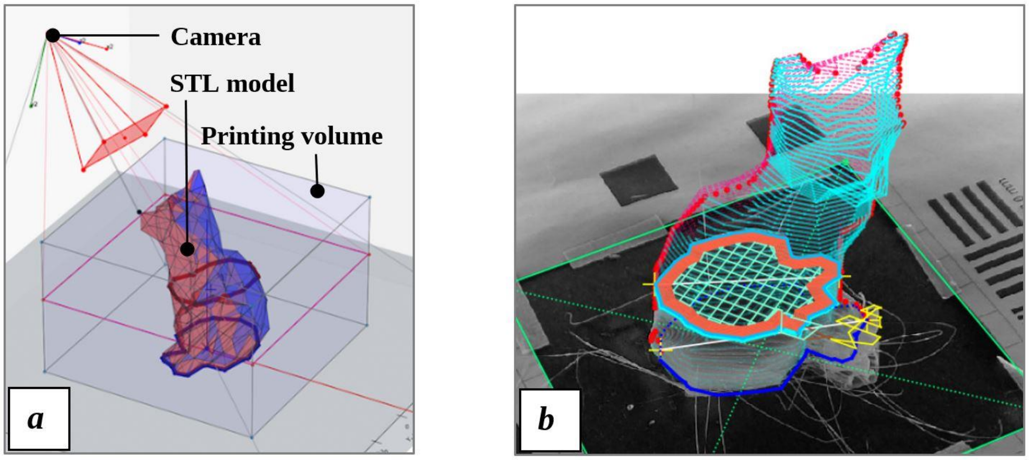


Figure 2.2: Projective transformation of the G-code and STL model applied to the source image frame: *a)* – camera position relative to the STL model; *b)* – G-code trajectories projected on the source image frame. This and the following slides illustrate the printing analysis for a low polygonal fox model [67].

The image pixel positions correspond to their three-dimensional spatial locations in accordance with the following equation (2-1), where the index *p* means “picture plane”, and the index *w* means “world space”:

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}, \qquad (2\text{-}1)$$

where $K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$ is the intrinsic camera parameters obtained during calibration, f_x and f_y are the focal lengths in image coordinates, c_x and c_y are the coordinates of the optical center in image coordinates (the principal point), $\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$ is the rotation matrix, and $\boldsymbol{t} = [t_x \quad t_y \quad t_z]^T$ is the translation vector. The camera position parameters $(\boldsymbol{R}, \boldsymbol{t})$ are determined using the known coordinates of visual markers. Thus, having the position of the camera and its intrinsic parameters, it is possible to determine the correspondence between the spatial coordinates of the working space (X_W, Y_W, Z_W) and image pixels (x_p, y_p).

Applying projective transformations to rectified frames of the observed printing area, it is possible to obtain a virtual top-view as if the camera is mounted parallel to the normal vector of the printing bed [47] and a pseudo-side-view as if the camera is mounted perpendicular to the normal vector of the printing bed (Figure 2.3). Observing the printed layer through the camera lens, a slice of the material is viewed as a set of pixels, or a tuple of numbers, characterizing the local areas of the layer. Therefore, analyzing a two-dimensional image provides an understanding of the nature of the texture of the object.

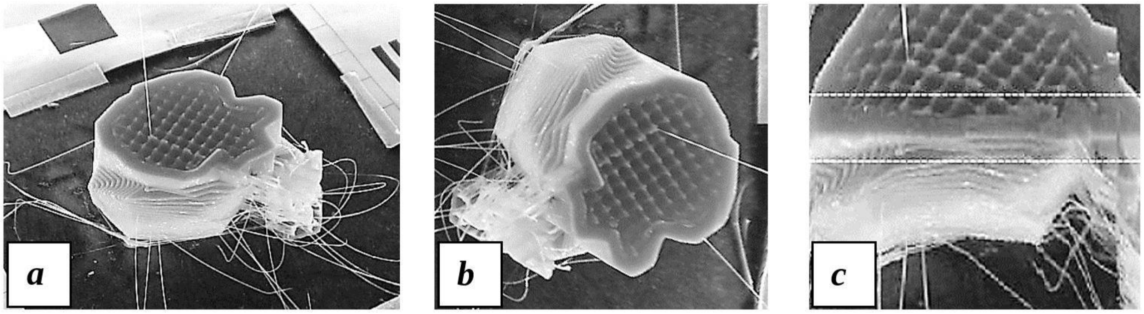


Figure 2.3: Frames obtained from the monocular vision control system: *a)* – rectified source image frame; *b)* – unwrapped virtual top-view; *c)* – unwrapped pseudo-side-view.

After each layer, based on the 3-D printed layer height, an analytical projective plane in the image shifts accordingly with the layer number, so the rectified image frame remains orthogonal to the optical axis of the virtual top-camera. Thus, by utilizing a rich group of image processing techniques, it becomes possible to segment meaningful contour and texture regions based on images and known parameters of the STL model and the G-code of the printing object. At the end of the printing process, a layered set of images provides additional data for volumetric analysis of the printed object in the future (Figure 2.4).

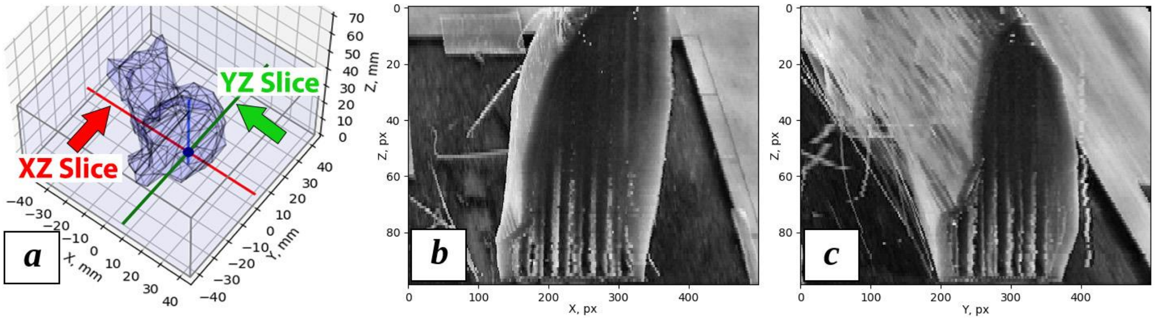


Figure 2.4: Volumetric analysis of the printed part: *a)* – STL model with the orthogonal scan planes; *b)* – vertical XZ slice; *c)* – vertical YZ slice.

A movable circle-shaped lighting frame was installed above the bed surface. The motor and mechanical system for tensioning the cables are mounted on top of the printer. The motor is connected to the stepper motor driver in the RAMPS printer control system and drives a lighting frame, which rises with each newly printed layer to a distance equal to this layer height, which ensures constant and uniform illumination of the printed part. The lighting frame, in turn, is a circular set of 56 2.8×3.5 mm light emitting diodes with a glowing temperature of 6000 K (cool white light spectrum), a total luminous flux of 280 lumens, a power of 18 watts, and a supply voltage of 12 volts [68].

The software developed in Python-language environment parses the source G-code, dividing it into layers and segmenting the extruder paths into categories such as a skirt, infill, outer and inner walls, support, etc. (Figure 2.5). The impact of environmental factors was minimized: applying blur and median filtration to grayscale images reduces the influence of color and print material while using a moving light frame minimizes the effect of extraneous illumination.

The categories of the G-code paths depend on the software algorithm used to slice the STL model. The developed program synchronized with the printer uses RAMPS 1.4 3-D printer control system and the open source firmware Marlin [69] as an intermediate driver. The source code is available: https://osf.io/8ubgn/ under the open source license GPLv3.

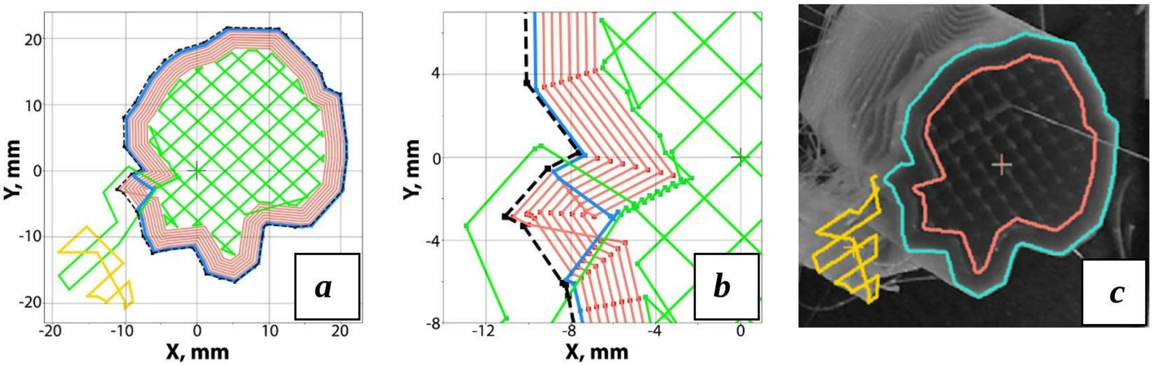


Figure 2.5: Layer-wise G-code segmentation: *a)* – an STL layer outline (black dashed line) overlay with the segmented layer of the source G-code with such trajectory categories as outer (blue) and inner walls (red), infill (green), and support (yellow); *b)* – possible inconsistency of STL and G-code; *c)* – unwrapped virtual top-view overlay with the outer wall (blue), infill (green), and support (yellow) extruder trajectories obtained from the G-code.

Using the reference layer-by-layer paths of the extruder obtained by the G-code analysis, in addition to the virtual top view, it is also possible to generate a pseudo-side-view. This approach does not allow for the creation of a full scan of the entire side surface of the model, however, this provides an opportunity to assess the divergence in the vertical size of the part with the reference height obtained from the G-code. The use of one camera instead of two (the main and a secondary camera for a side view) reduces the computational load and eliminates the need to synchronize the processing of two images.

The temperature parameters, the coordinates of the trajectories and the traveling speed of the extruder, the feed rate of the material, as well as thickness of the printed lines and the height of the layer are stored in the program memory for each layer. Print commands, pauses, image analysis, and decision-making algorithms are carried out by the developed software, giving a control over the state of the printing process. Therefore, in case of critical deviations, printing can be suspended, and if it is possible to repair the part, a sequence of corrective G-code commands will be launched.

2.4 Algorithm Development

The image processing pipeline could be divided into three branches and presented in Figure 2.6. Each branch of the pipeline will be described further in this paper.

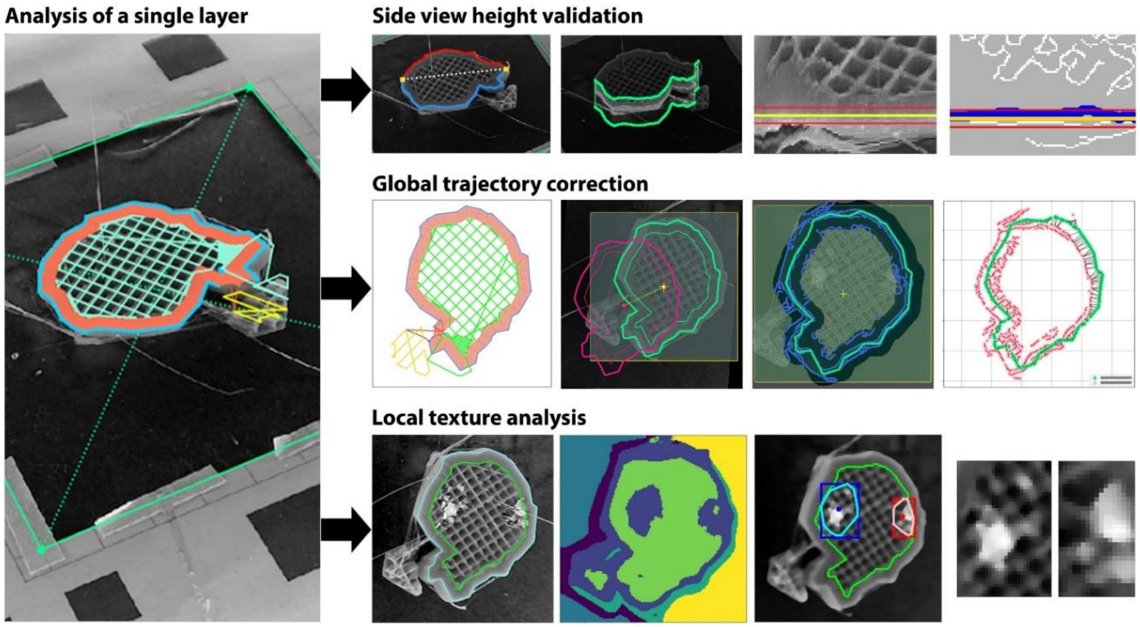

Analysis of a single layer
Side view height validation
Global trajectory correction
Local texture analysis

Figure 2.6: Image processing pipeline. During the analysis stage, two auxiliary images (a virtual top view and a pseudo-side-view) are created from the main captured frame. Next, the pseudo-side-view is used to validate the height of the layer, while the virtual top view is used to analyze the outline of the layer and its interior. These three processes occur independently of each other.

Starting with the side-view height validation, the algorithm analyzes the virtual top view for global trajectory matching and local texture examination. This allows taking into account both global and local parameters of printing processes.

The proposed algorithm (Figure 2.7) for detecting printing failures assumes the presence of one camera located at an angle to the working surface of the 3-D printer. An angled camera allows us to observe both the active printable layer and part of the printing model from the side. Thus, one source frame can be divided into a virtual top view from above and a pseudo-view from the side.

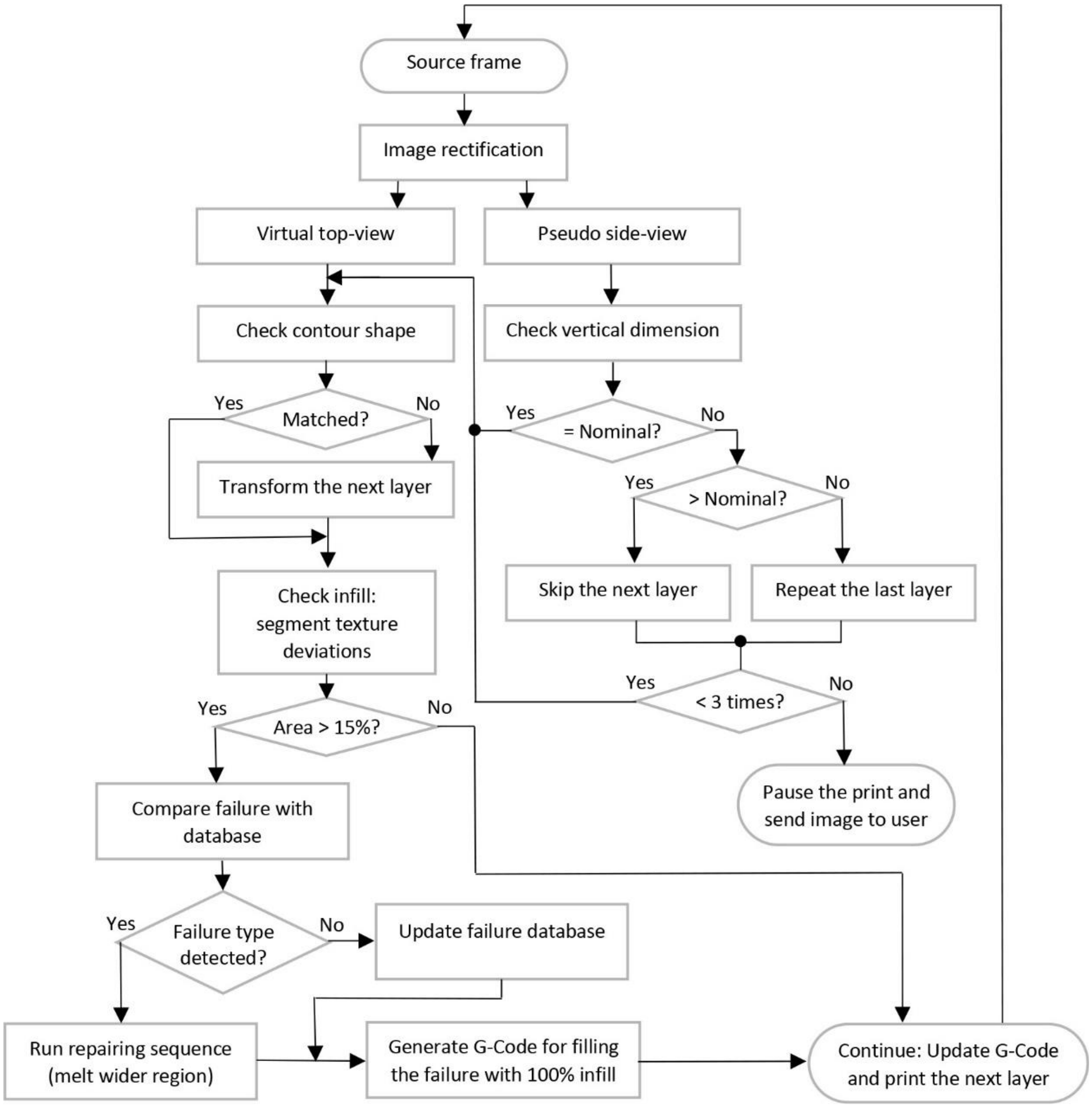


Figure 2.7: 3-D printing control algorithm. After completing each layer, a virtual top view and a pseudo side view are obtained from the source image frame and used to independently analyze the height of the layer, as well as its external and internal parameters. Depending on the deviations found, certain error elimination procedures are launched and the cycle is repeated. If the deviation cannot be corrected, the printing process pauses and awaits corrective actions from the user.

Such criteria as bed leveling, dimensionality lost, and non-circularity are dependent on a specific printer model and are manually calibrated by the user at the time of the first run. It is possible to create calibration tables to determine the correction factors for G-code trajectories. However, at this stage, the above parameters are checked only for compliance/non-compliance with the specified values. In case of non-compliance in bed leveling, dimensionality, and circularity, printing is suspended. This method does not eliminate these errors during the printing process, but it can save time and material.

2.4.1 Side View Height Validation

Knowing the camera position and G-code trajectories for a given layer, it is possible to analyze the visibility of the side area of the printed part by solving the system of equations (2-2) that provides slope and shift coefficients for a linear visibility delimiter.

$$\begin{cases} y_p^{(1)} = m \cdot x_p^{(min)} + b \\ y_p^{(2)} = m \cdot x_p^{(max)} + b \end{cases}, \tag{2-2}$$

where m and b – are the coefficients of the linear visibility delimiter, $x_p^{(min)}$ and $x_p^{(max)}$ obtained from the extreme contour points of the projection of the G-code outline on the picture plane, $y_p^{(1)}$ and $y_p^{(2)}$ – are the y coordinates of the corresponding extreme points on the picture plane.

The visibility analysis is necessary to determine the boundaries of the source frame, the area within which must be transformed to obtain a pseudo-side-view. Next, the height of the printed layer is determined by detecting its contrasting vertical edge in the pseudo-side-view image. A deviation of the height of the detected edge from the corresponding values in the g-code will indicate a mismatch between the printed model and its source g-code. The pseudo-side-view, therefore, allows monitoring the height of the printed part and detect critical failures such as blocked nozzle, lack of material, major deformations, etc.

2.4.2 Global Trajectory Correction

After checking the vertical size, a virtual top view is used for the subsequent two-stage analysis of the external contour and infill of the printed layer. Having data on the corresponding extruder trajectories from the G-code and the resulting contour obtained from image analysis, it is possible to determine if there is a mismatch between the real outline and the reference borders using the multi-template matching (MTM) [70] and the iterative closest point (ICP) [71,72] algorithms. MTM allows to track significant horizontal and vertical displacements of the printed part based on the binary layer template obtained from the G-code trajectories, and the ICP algorithm determines fine rotation and translation within the small deviation range. As a result, we obtain a transformation matrix, which multiplied by the spatial coordinates of the extruder trajectories, eliminates small linear shifts and scale distortions of the printed layer.

The MTM method computes a correlation map between the reference layer outline and binary edge image of the virtual top view based on the *"match Template"* OpenCV function [73] and predicts the template (G-code outline) position within the image. Since the algorithm performs the search by sliding the template over the image, it detects the object with a similar orientation as the template and may not be sensitive to rotations.

The ICP algorithm aimed at finding the transformation matrix between two point clouds by minimizing the squared error (2-3) between the corresponding surfaces using the gradient descent method. The iterative algorithm converges when the starting positions are close to each other.

Given two corresponding point sets $\{m_1, m_2, \dots, m_n\}$ and $\{p_1, p_2, \dots, p_n\}$ we can find translation $\boldsymbol{t}$, rotation $\boldsymbol{R}$**,** and scaling s that minimize the sum of the squared error:

$$E(\boldsymbol{R}, \boldsymbol{t}, s) = \frac{1}{N_p} \sum_{i=1}^{N_p} \| m_i - s\boldsymbol{R}(p_i) - \boldsymbol{t} \|^2 , \tag{2-3}$$

where $\boldsymbol{R}$, $\boldsymbol{t}$, and s – are rotation, translation, and scaling respectively. Since the scaling operation can also translate an object, the center of the G-code layer projection should be placed at the origin.

Based on rotation, scaling and translation obtained from the ICP algorithm and assuming that the z-level was found during the vertical size check, the G-code trajectories for the next $(k+1)^{th}$ layer $\begin{bmatrix} G_x^{(k+1)\,\prime} \\ G_y^{(k+1)\,\prime} \end{bmatrix} = \begin{bmatrix} x_1^{(k+1)\prime} & x_2^{(k+1)\prime} & \dots & x_n^{(k+1)\prime} \\ y_1^{(k+1)\prime} & y_2^{(k+1)\prime} & \dots & y_n^{(k+1)\prime} \end{bmatrix}$ will be transformed from the initial trajectories of the next layer $\begin{bmatrix} G_x^{(k+1)} \\ G_y^{(k+1)} \end{bmatrix} = \begin{bmatrix} x_1^{(k+1)} & x_2^{(k+1)} & \dots & x_n^{(k+1)} \\ y_1^{(k+1)} & y_2^{(k+1)} & \dots & y_n^{(k+1)} \end{bmatrix}$ in accordance with the following equation (2-4):

$$\begin{bmatrix} G_x^{(k+1)\,\prime} \\ G_y^{(k+1)\,\prime} \end{bmatrix} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} \cdot \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \left(\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \cdot \begin{bmatrix} G_x^{(k+1)} \\ G_y^{(k+1)} \\ 1 \end{bmatrix} \right) \tag{2-4}$$

Rotation and scaling operations must be performed after the defective layer path is shifted to the origin by $[t_x \quad t_y]^T$ calculated by the ICP algorithm. Thus, the new coordinates of the next, (k+1)th, layer can be written in the following way (2-5):

$$\begin{cases} x_n^{(k+1)\prime} = s\left[\left(x_n^{(k+1)} + t_x\right)\cos\theta - \left(y_n^{(k+1)} + t_y\right)\sin\theta\right] \\ y_n^{(k+1)\prime} = s\left[\left(x_n^{(k+1)} + t_x\right)\sin\theta + \left(y_n^{(k+1)} + t_y\right)\cos\theta\right] \end{cases} \tag{2-5}$$

Figure 2.8 shows the operation of the ICP algorithm for an example of a defective dumbbell-shaped layer. The reference layer path was rotated by $\theta = 20$ degrees, scaled by $s = 0.85$, and shifted by $t_x = 9.0\, mm$ horizontally and $t_y = 6.0\, mm$ vertically. After several iterations, the algorithm detected a geometric mismatch with the following parameters: rotation of 18 degrees, scale 0.88, horizontal offset of 8.1 mm, and vertical offset of 6.6 mm, which is close to the true values. Then the coordinates of the defective layer were transformed based on the parameters found.

It is worth noting that this method works with a “delay” of one layer. Thus, if a mismatch is found in the coordinates of the current printed layer, the adjustment can only be made for the next layer. Therefore, a 3-D model of a complex geometric shape with significant differences between successive layers may not be adjusted in all of cases.

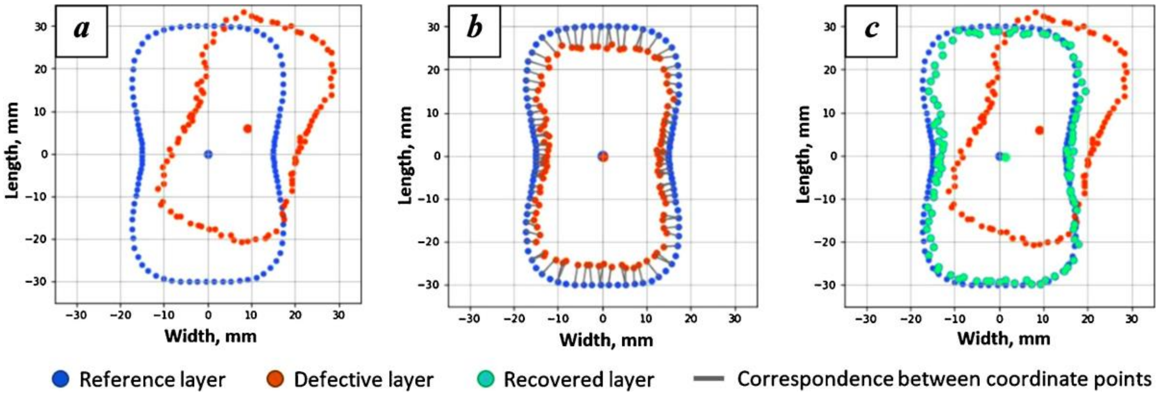


Figure 2.8: Defective layer recovery using the ICP algorithm. After the fabrication of the k-th layer, its contour is detected by visual analysis (red points) and compared with the reference contour obtained from the G-code (blue points). If there is a discrepancy, the transformation parameters (scale, translation, and rotation) are determined to return the layer to its initial position according to the G-code. These transformations are then applied to all of the subsequent layers to prevent geometric distortions caused by mechanical defects or part movement during manufacturing.

2.4.3 Local Texture Analysis

After analyzing the contour, a check is made to the layer filled with the material. The purpose of this step is to identify irregular sections of the texture within the layer infill. At this stage, it is assumed that the vertical dimension of the part corresponds to the specified one, and the correct location of the real boundaries of the part is determined. Thus, only irregularities of the texture in the region bounded by the outer shell of the layer are considered.

The textural features based on local probabilistic similarity measures are simple, have a low computational load, and could serve well for a small number of specific cases, but may not be efficient for a wide range of real-world problems [74-76]. Because of the complex surface topology, textural variations may not be explicitly expressed as a histogram comparison [77].

In this work, the texton-based approach to texture segmentation was implemented, since the given method has repeatedly demonstrated its effectiveness and scalability [77-81]. The texton-based segmentation utilizes Leung-Malik (LM) filters [77,82] (Figure 2.9) that work as visual cortex cells and allow the segmentation of an image into channels of coherent brightness and texture in a natural manner, where the texture originates from the spatial variation of surface normal and reflectance. The LM filter bank consists of 48 filter kernels, which is a mix of 36 elongated filters, 8 difference of Gaussians, and 4 low-pass Gaussian filters [77].

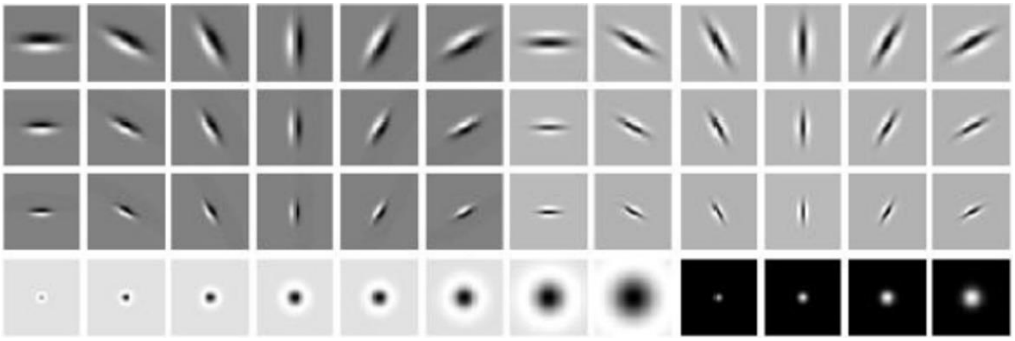


Figure 2.9: The Leung-Malik orientation and spatial-frequency selective filter bank: 36 Gaussian derivative filters (top three rows), 8 difference of Gaussians and 4 Gaussian filters (bottom row).

After the convolution of an image with the filter kernels, each pixel is transformed into a multidimensional vector of filter responses. After clustering, these vectors form a set of texton channels, or appearance vectors, that define the image partitioning.

In both plastic and metal additive manufacturing the lighting effects and mutual reflectance create a non-Lambertian environment, where similar surfaces may look significantly different under varying viewing angles [77,81,83], which narrows the set of possible image processing techniques. According to [78,84-86], filter responses encode appearance information over a broad-scale range and can serve as a preprocessing method that can be combined with dense descriptors for efficient texture classification with varying illumination conditions.

Without prior texture information, a suitable approach for clustering the obtained filter responses, and, therefore, for texture segmentation, is an unsupervised machine learning method, which is a widely used technique in object recognition. Unsupervised machine learning is used to draw derivations from data consisting of input information with no labeled responses.

Clustering, being the most common unsupervised learning method, aimed at finding hidden patterns in the source data set and grouping them accordingly to their salient features and the number of clusters k specified by the user. The majority of previous works in texture segmentation are based on k-means clustering [78], but the non-probabilistic nature, hard cluster boundaries and lack of flexibility in cluster shape in k-means clustering leads to practical challenges and may not perform well in real-world applications. The Gaussian mixture model (GMM) clustering, implemented in this work, considers both cluster centers and covariances describing the location and shape of clusters [78].

The GMM clustering is based on filter responses, and attempts to partition an unlabeled input layer texture as a mixture of multidimensional Gaussian probability distributions in regions that share common characteristics. The Gaussian mixture model for k unknown clusters can be written as a superposition of Gaussians [87]:

$$f_{GMM}(x) = \sum_{j=1}^{k} w_j f_{\mathcal{N}(\mu_j,\Sigma_j)}(x) \text{ , where } \sum_{j=1}^{k} w_j = 1, \qquad (2\text{-}5)$$

which is a combination of weighted w_j normal probability density functions $f_{\mathcal{N}(\mu_j,\Sigma_j)}$ with mean vector μ_j and covariance matrix Σ_j.

The GMM-based clustering determines the maximum likelihood for Gaussian models with latent variables by utilizing the expectation-maximization (EM) algorithm [88, 89], which iteratively estimates the means, covariances, and weighting coefficients in a way that each cluster is associated with a smooth Gaussian model. However, despite the effectiveness of the described method, the exact number of clusters k should be specified in advance, which is a critical decision and determines the success of texture segmentation.

Using the infill mask obtained from the G-code paths, GMM segmentation and failure analysis are performed only within the layer infill texture region. Since the result of segmentation is not completely predictable and one anomalous region can consist of several segments, the agglomerative hierarchical clustering (AHC) [90] is launched after the GMM partitioning. The AHC, being an unsupervised technique, recursively merges pairs of individual clusters based on their location within the segmented area, which provides a consistent failure map for the analyzed infill region.

2.4.4 Targeted Failures and Corrective Actions

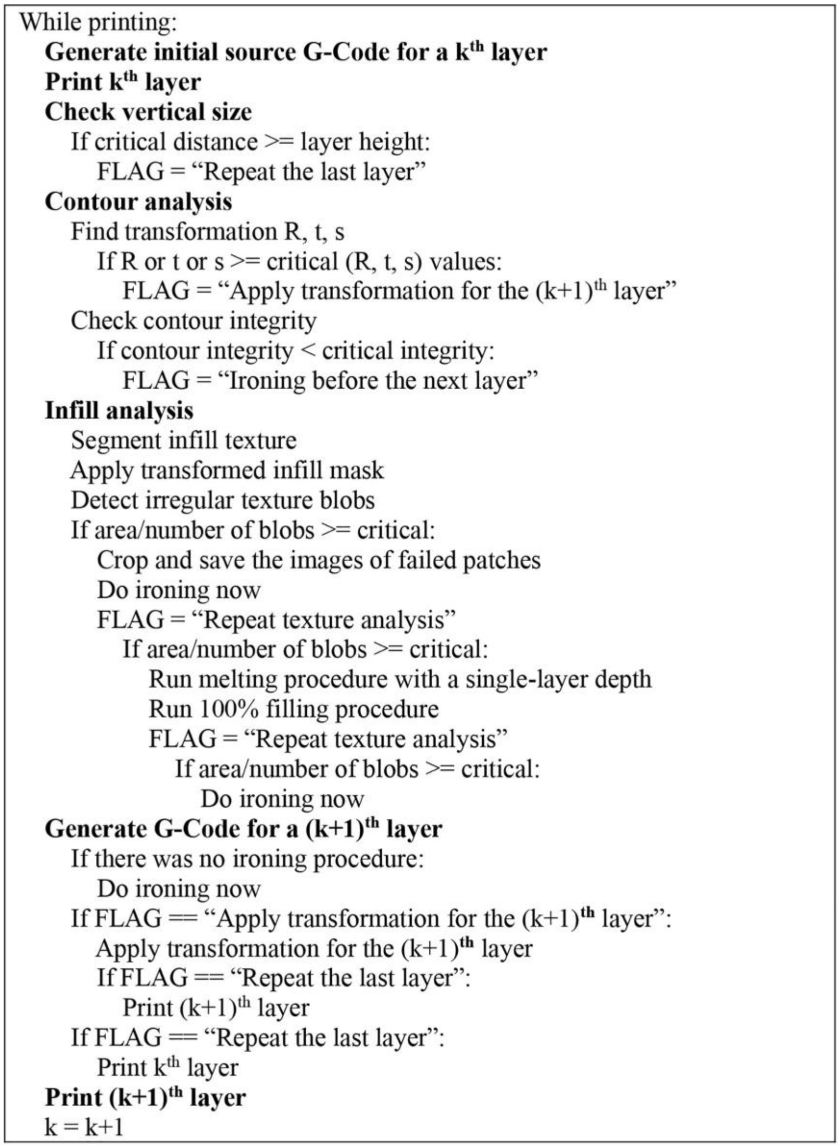
Using the above-mentioned techniques, it is possible to approach a considerable number of the failures listed in Table 2.1. The main target failures along with the proposed corrective actions are introduced in Table 2.2. These printing errors can be detected and/or eliminated by using the appropriate G-code commands without adjusting the mechanical parameters of the printer and slicing modes. This is an algorithmic concept for eliminating critical and most common failures, since there can be a multilevel branching causal relationship behind each printing failure, which makes identifying the causes of failure difficult even for a user. The developed method allows determining the initial vector for a global solution to the problems posed since it is hard to create a universal method for solving all of possible printing issues at once.

Table 2.2: Target failures and corrective actions.

	Failure type	**Detection strategy**	**Printer action**
1	Out of filament	Vertical level + MTM & ICP algorithms	Pause / Report
2	Blocked nozzle	Vertical level + MTM & ICP algorithms	Increase nozzle temperature; Repeat the previous layer a finite number of times
3	Missing layer	Vertical level + MTM & ICP algorithms	Repeat the previous layer a finite number of times
4	Lost dimensional accuracy	MTM & ICP algorithms	Update G-code coordinates

5	Bed leveling issue	Texture segmentation	Pause / Report; Manual level recalibration
6	Adhesion problem (warping)	Tracking vertical level of the initial layer	Increase bed temperature; Pause / Report in case of critical vertical deviation
7	Print is not sticking to the bed	Vertical level + MTM & ICP algorithms	Increase bed temperature; Pause / Report
8	Print offset / bending	Vertical level + MTM & ICP algorithms	Update G-code coordinates
9	Weak or under-extruded infill	Texture segmentation	Increase nozzle temperature and feed rate
10	Deformed infill	Texture segmentation	Change nozzle temperature and feed rate
11	Burnt filament blobs	Texture segmentation	Smooth out irregularities by moving the hot nozzle over the surface without material extrusion (ironing procedure)
12	Incomplete infill	Texture segmentation	Patch replacement procedure
13	Poor surface quality above supports	Texture segmentation	Change feed rate
14	Gaps between infill and shell	Texture segmentation	Change feed rate

A simplified pseudo-code in accordance with the basic algorithm (Figure 2.7) for a layer processing is shown in Figure 2.10.



```
While printing:
  Generate initial source G-Code for a $k^{th}$ layer
  Print $k^{th}$ layer
  Check vertical size
    If critical distance >= layer height:
      FLAG = “Repeat the last layer”
  Contour analysis
    Find transformation R, t, s
      If R or t or s >= critical (R, t, s) values:
        FLAG = “Apply transformation for the $(k+1)^{th}$ layer”
    Check contour integrity
      If contour integrity < critical integrity:
        FLAG = “Ironing before the next layer”
  Infill analysis
    Segment infill texture
    Apply transformed infill mask
    Detect irregular texture blobs
    If area/number of blobs >= critical:
      Crop and save the images of failed patches
      Do ironing now
      FLAG = “Repeat texture analysis”
        If area/number of blobs >= critical:
          Run melting procedure with a single-layer depth
          Run 100% filling procedure
          FLAG = “Repeat texture analysis”
            If area/number of blobs >= critical:
              Do ironing now
  Generate G-Code for a $(k+1)^{th}$ layer
    If there was no ironing procedure:
      Do ironing now
    If FLAG == “Apply transformation for the $(k+1)^{th}$ layer”:
      Apply transformation for the $(k+1)^{th}$ layer
      If FLAG == “Repeat the last layer”:
        Print $(k+1)^{th}$ layer
    If FLAG == “Repeat the last layer”:
      Print $k^{th}$ layer
  Print $(k+1)^{th}$ layer
  k = k+1
```

Figure 2.10: Example of failure correction. If defects are detected on the current layer, the G-code coordinates for the next layer can be updated.

2.5 Experimental Results

The algorithm was tested during regular printing without failures of the 42×51×70 mm low-polygonal fox model [67] with the following printing parameters: 1.75 mm PLA, 0.4 mm layer height, 0.4 mm line width, 30% grid infill, and 3.2 mm wall thickness. The entire model consists of 175 layers, but the tests were carried out for the first 96 layers since part of the model was located outside of the visible area.

Bed leveling and dimensionality checks were calibrated in advance before printing. The visual analysis of the vertical level, deformation of the outer shell, and infill texture was performed for each printing layer on a 2.5 GHz processor with 8 GB of RAM. These experiments for the case of a normal printing mode without deviations allow determining the accuracy and tolerance of the adaptive algorithm.

2.5.1 Height Validation Results

Starting with the vertical level validation (Figure 2.6), the algorithm analyzes the virtual top view for global trajectory matching and local texture examination. This allows taking into account both global and local parameters of printing processes. Based on the G-code reference contour and the system of equations (2-2), the calculated visual separator is used to generate a pseudo-side-view projection where the curved part of the model is shown as a straight line (Figure 2.11).

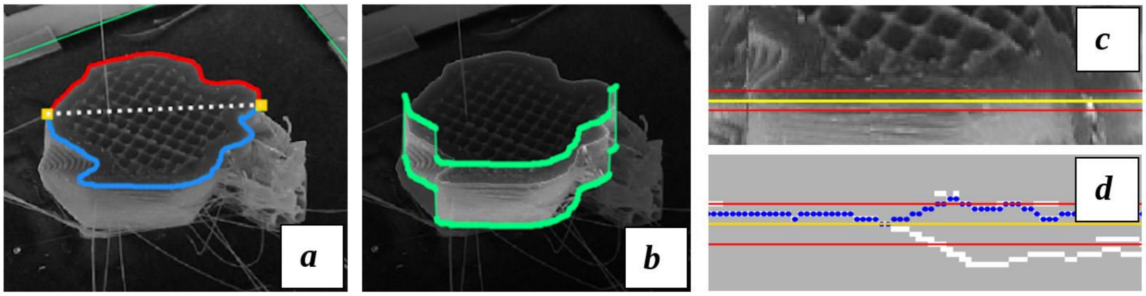


Figure 2.11: Pseudo-side-view generation: *a)* – computed linear visibility delimiter (white dashed line), the edge for the visible side region (blue) and invisible side region (red) of the printed part; *b)* – designated visible area for unwrapping; *c)* – unwrapped region with the reference vertical level (yellow) and the maximum double-layer errors in both directions (red); *d)* – detected vertical edge error (blue) with the reference layer height (yellow) and maximum double-layer errors in both directions (red).

Due to uniform all-round lighting, the contrast between adjacent faces of the part is enough to find a clear edge. Thus, comparing the pixel-wise deviation of the detected edge from the reference contour becomes possible to obtain the distribution and total vertical level error for each layer (Figure 2.12).

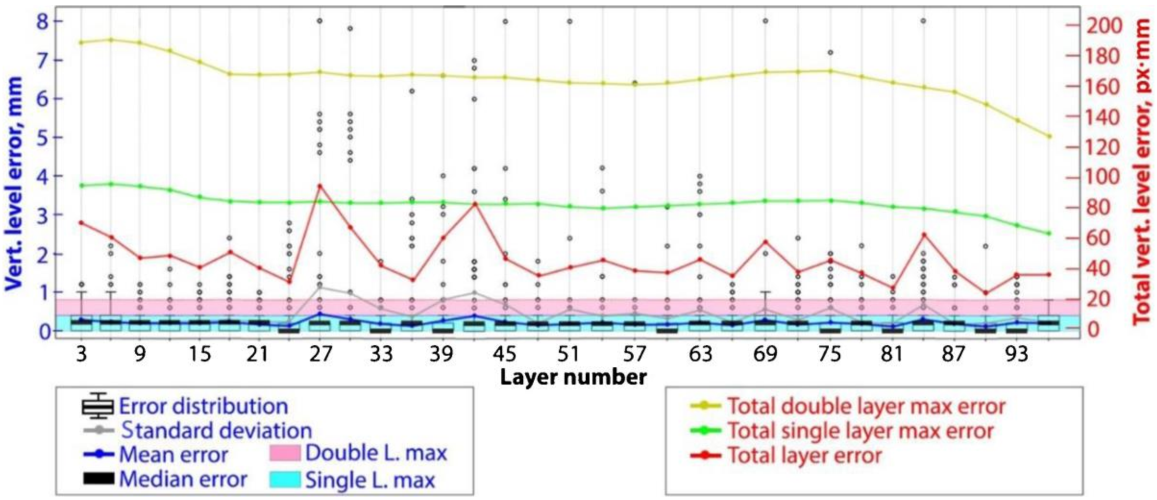


Figure 2.12: Results of vertical level validation. The maximum amount of detected layer edge displacement (total vertical level error) depends on the observed part's geometry at each stage of its completion. Under normal printing conditions, however, the median error of the detected edge for each layer does not exceed the maximum deviations for one layer.

Based on the camera resolution, the relationship between the linear dimensions of the part and the number of pixels in the image, which is 5 pixels per 1 mm, was calculated. Thus, knowing the height of one layer (0.4 mm) and the length of the visible side region (that may have a different value for each layer), it is possible to calculate the maximum single- and double-layer mismatch between the detected contour and the calculated one, which is equal to the height of two layers:

$$E_{total} = \frac{L \cdot h}{m}, \qquad (2\text{-}7)$$

where E_{total} – is the maximum total layer error in pixels·mm, L – is the length of the visible side region in pixels, $h = 2$ – is the layer height in pixels, and $m = 5$ – is the scaling factor for the side view in pixels per millimeter. For the 7th layer, for example, having the length of the visible side region of $L = 235$ pixels, the maximum total double-layer error would be $E_{total} = 235 \cdot 4/5 = 188$ pixels·mm (Figure 2.12).

As can be seen from the experimental results, in the normal printing mode, both the average and total error values do not exceed the maximum deviation equal to the height of the two layers. For each individual layer, however, the vertical level error may exceed the maximum error value equal to the value of one layer. The discriminative power of the failure detection depends on the resolution of the camera, the distance to the print area, the size of the part, and can be taken into account in the algorithm in such a way that a one-time deviation exceeding the height of one layer will be ignored, while multiple consecutive excesses of the level of one layer will be taken as a true error.

At this stage of development, a defect on the backside of the part will be unnoticed during the side view height validation, however, it can be recognized in the virtual top view if this failure affects the inner portion of the layer. It should be noted that the optimal location of the part on the printing bed, where the most critical areas of the part are oriented in the direction of the camera, reduces the likelihood of skipping printing defects.

2.5.2 Results of the Global Trajectory Analysis

The global outline analysis uses a virtual top view and a combination of MTM and ICP algorithms. It is currently assumed that the manufactured part does not contain numerous through holes, and, also, that the grayscale gradient on the faces of the printed object is sufficient to identify the outline. In addition, an STL-based restrictive mask is used to limit the search for the layer contour within the expected region. At the first stage, the MTM algorithm detects significant horizontal and vertical shifts (Figure 2.13), after which the ICP algorithm provides more accurate information about the rotation and displacement in a small range within the area detected by the MTM algorithm (Figure 2.14).

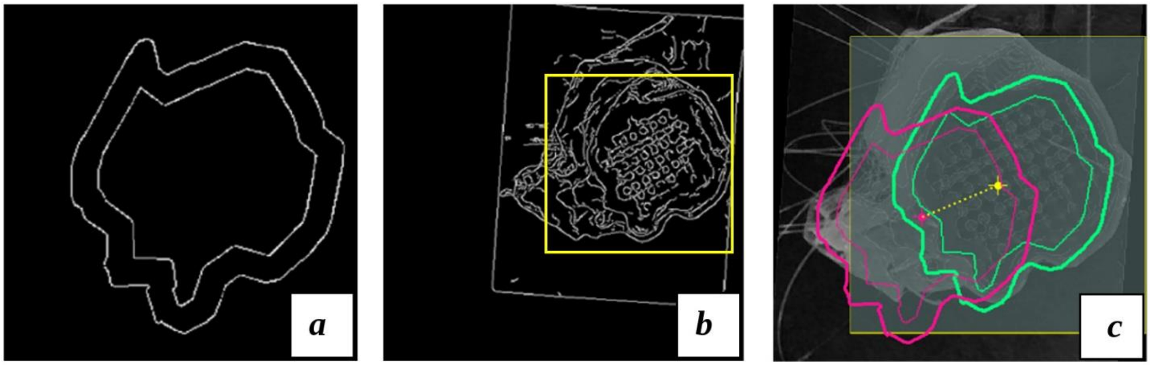


Figure 2.13: Global displacement detection based on MTM algorithm: *a)* – contour-based binary template; *b)* – printed part shifted due to failure; *c)* – computed shift distance and direction.

To ensure the reliability of the ICP algorithm, a restrictive mask based on the STL layer outline is used that limits the number of detected edge points used in the calculations of displacement and rotation relative to the G-code reference contour (Figure 2.14, a).

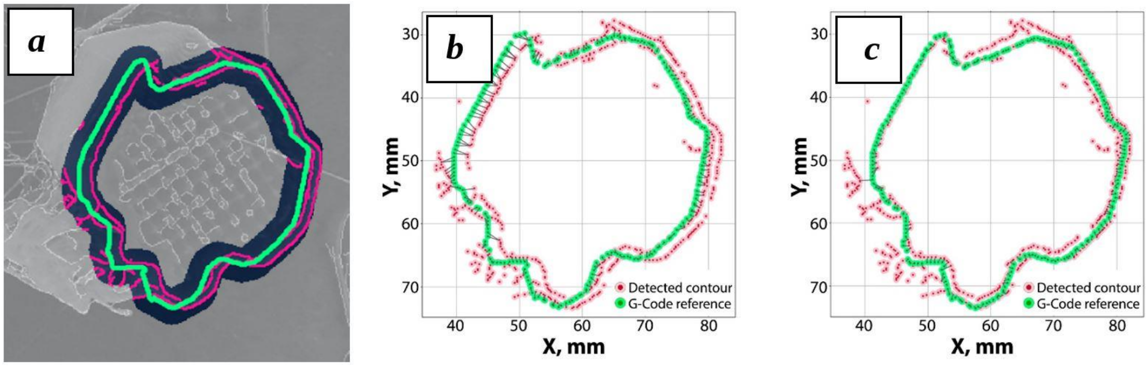


Figure 2.14: Global G-code trajectory matching for a single layer based on ICP algorithm: *a)* – reference outline (green) mismatched with the detected contour points (red), the saturated region around the reference outline illustrates the restrictive mask obtained from the STL layer outline, which constrains the point cloud correspondence search for the ICP algorithm within the given area; *b)* – initial ICP iteration; *c)* – final ICP iteration.

The figures above show an instance of an exemplary error in positioning a printed layer. A large initial displacement of 15 mm horizontally and 7 mm vertically (Figure 2.13) may not be detected using the ICP algorithm, since this requires an initial guess about the real coordinates of the layer. However, without information on the nature and magnitude of the displacement, the MTM algorithm can be used as an intermediate step to obtain an initial assumption of the shifted coordinates. Then, using the initial guess about real layer location, the ICP algorithm allows detecting minor displacement and rotation of the printed layer within a few millimeters and degrees, respectively. In this example (Figure 2.14), a layer rotation of 0.2 degrees was determined, as well as an offset of 3.2 mm horizontally and 0.9 mm vertically.

Figure 2.15 shows the results of the ICP algorithm. It was revealed during the experiments that the restrictive mask with a width of 30 pixels (this corresponds to 5.7 mm for the given setup, or about 10% of the horizontal size of the printed part) allows to obtain a stable result and to identify the maximum displacement and rotation of 8 mm and 10 degrees, respectively.

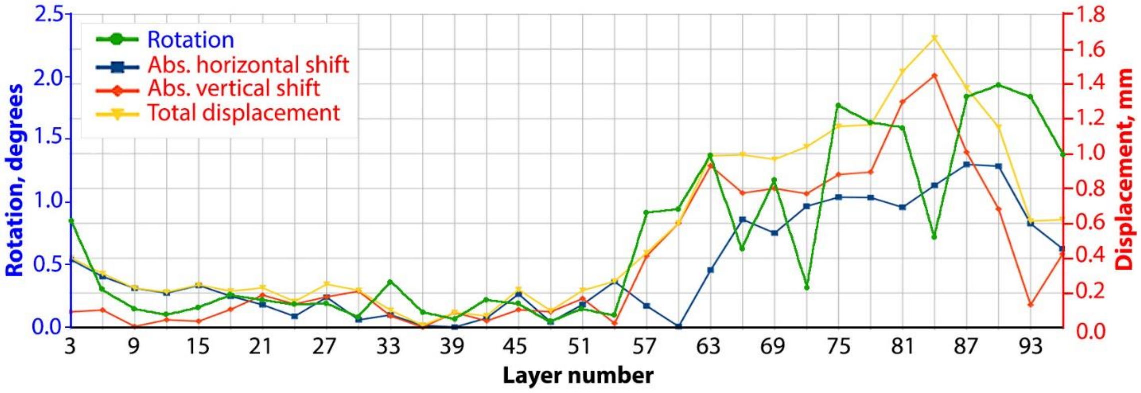


Figure 2.15: Results of the global trajectory matching analysis. Under normal printing conditions, in this case, the maximum detected displacement does not exceed 1.8 mm, and the layer rotation is 2.5 degrees. Here the upper layers are smaller in size relative to the lower ones. This affects the number of detected contour points used as input data for the global trajectory analysis procedures and, in turn, can introduce errors in the detected rotation, shift, and scale parameters.

In the nominal printing mode, rotation errors of up to 2 degrees and displacement errors of up to 1.7 mm were observed.

2.5.3 Results of the Local Texture Analysis

According to [91], the dimensions of the filter should correlate with the size of prevailing image structures to reduce the noise effect during image processing. Varma and Zisserman in [92,93] also presented a detailed analysis of the effect of filter size on classification efficiency for a number of filter banks and filter sizes.

Taking into account the facts that to accelerate the clustering operation, the original image can be reduced in size, and that a larger filter can better suppress visual noise, it was experimentally determined that the filter dimension of 1/3 of the input image allows to effectively segment the layer texture with high speed. Thus, the dimensions of the input image and the Leung-Malik filters are 150×150 pixels and 49×49 pixels, respectively.

In addition to the filter size, the number of expected texture clusters is also an equally important parameter. During the experiments was found that six clusters (six expected textures in the input image) can provide an effective ratio of speed and quality of segmentation.

Figure 2.16 shows the result of the GMM segmentation of the test image with 18 various texture samples, where the size of each texture patch is equal to the size of the filter and is 49×49 pixels.

The textons, appearance vectors, or the corresponding filter response vectors, capture characteristic shapes of different materials and features at various viewing angles and lighting conditions. In Figure 2.16 the 18 clusters correspond to the prevalent features in the image [77], where each of the cluster centers visualized by a pseudo-inverse that codes geometric features such as grooves, bumps, ridges, and hollows into an image icon [94].

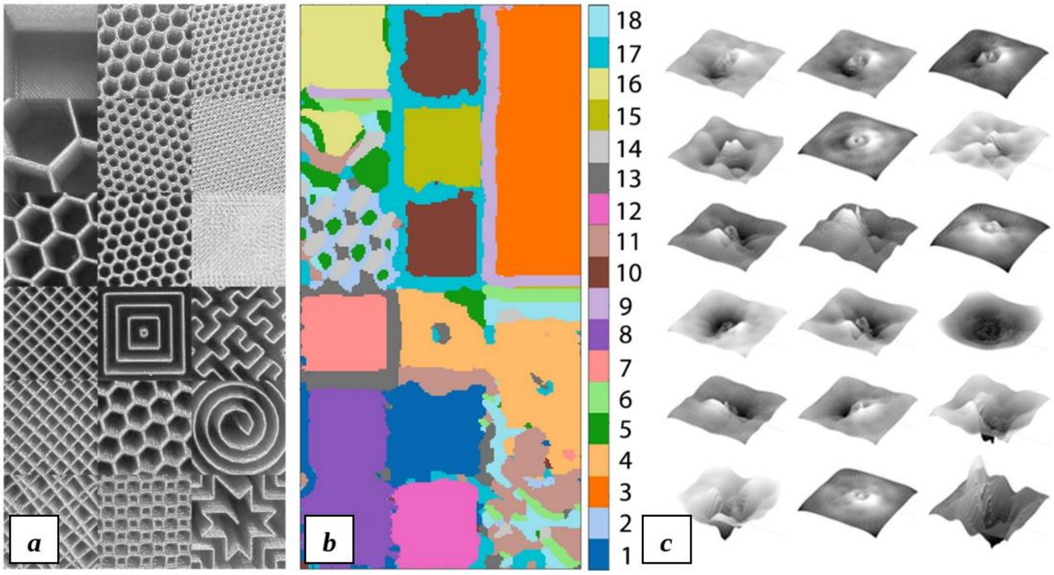


Figure 2.16: Partitioning an image into texture channels (textons): *a)* – test image with 18 various texture samples (modified image from https://all3dp.com/2/infill-3d-printing-what-it-means-and-how-to-use-it); *b)* – segmented textures; *c)* – obtained texture channels.

The texture areas shown in Figure 2.16 represent common types of infill patterns with different filling densities. As can be seen, not all of the areas are labeled correctly. This may be due to the dimensions of the source frame, the level of image processing required to suppress visual noise, as well as the ratio of the LM filter size to the image dimensions, and the proportions of the characteristic infill areas relative to the surrounding texture. For example, three fill areas in the upper right corner have the same label, which may be due to the similarity of these areas after scaling and processing of the input image, as well as the large size of the filters in relation to the characteristic areas of the fill pattern. Other segmentation errors in the upper left and lower right corners may be caused by significant geometric variability in the infill pattern and its large size relative to the LM filters. The developed method for analyzing the texture of infill areas therefore requires further research and adaptation for a wide range of possible texture patterns.

After segmentation of the input image, labeling of the textures regions inside the infill mask occurs. Figure 2.17 shows an example of a failed layer with two defective regions injected to demonstrate the texture segmentation and clustering processes. Texture regions other than the main infill texture are considered abnormal and, subject to a certain shape and size, are taken into account in further analysis. If the total area of the considered anomalous regions exceeds a critical value of 15% of the total area of the layer infill region, the layer is considered defective, and further unsupervised agglomerative hierarchical clustering occurs.

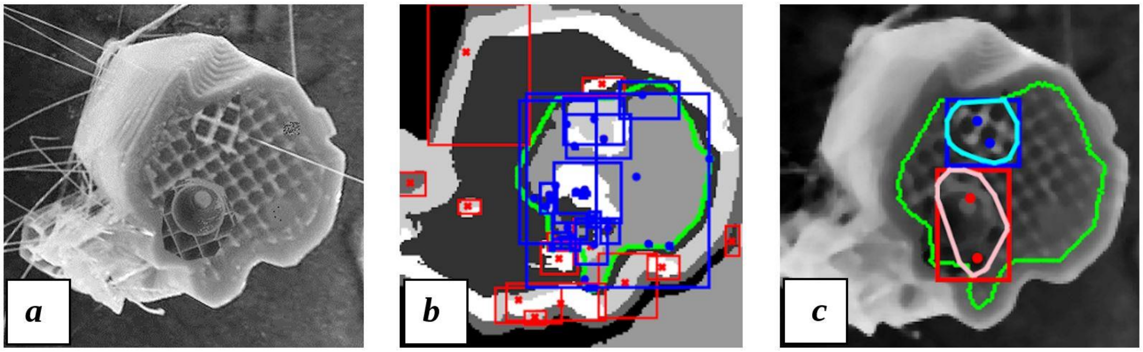


Figure 2.17: Defective layer segmentation results: *a)* – source virtual top view; *b)* – segmented image with the with infill mask (green), infill textures (blue regions), outside textures (red regions); *c)* – segmented failures (red and blue regions) inside the infill area (green).

Since after a GMM segmentation a defective part of a texture can consist of several segments (blue regions in Figure 2.17), it is necessary to determine their belonging to one or another defective group. For the AHC algorithm, the clustering parameters were experimentally selected based on the centroid locations of the anomalous regions and the distance between them. The algorithm assumes one or two defective infill sections and determines whether the segmented anomalies belong to one or the other section.

Figure 2.18 shows the results of the infill texture analysis for the specified printed layers. During the experiments, the entire total area of anomalous regions was considered without taking into account their shapes.

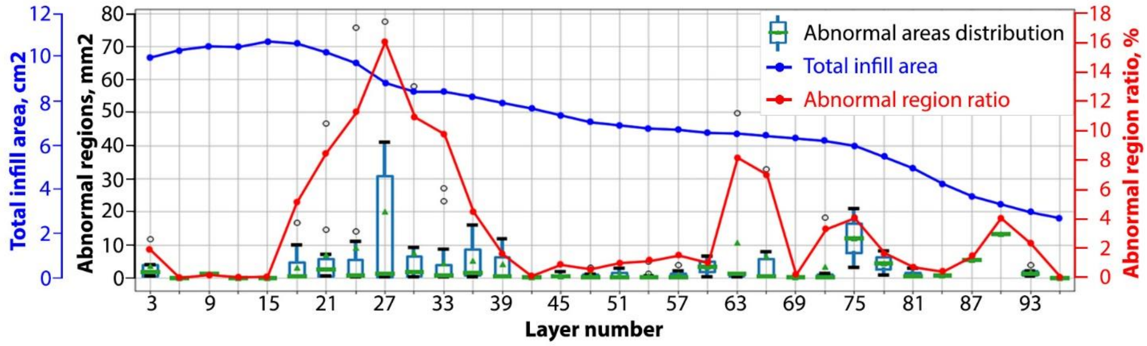


Figure 2.18: Results of the texture analysis during the regular printing. The method can falsely detect anomalies of up to 15 percent of the total layer area or higher in normal printing mode, indicating the need for additional measures to prevent false positives.

As can be seen from the figure above, the total area of the anomalous regions can reach 10 or more percent of the infill region. This fact is due to the ingress of the outer wall texture inside the infill mask, which is a false alarm and can be eliminated by analyzing its shape.

2.5.4 Runtime Analysis

The total analysis time can be divided into the following components (Figure 2.18):

1. Side view height validation;
2. Global outline correction;
3. Local texture analysis.

The side view height validation algorithm consists of G-Code parsing and image processing parts. The computational complexity of the parsing stage is linear (*O(n),* where n represents the size of the input data) with respect to the number of lines of the G-Code file. The image processing stage consists of OpenCV functions such as *“projectPoints”, “polylines”, “findContours”,* and *“drawContours”,* as well as Canny edge detection and median filtering, applied sequentially to a fixed-size image. Since the number of G-Code lines for all of the layers is approximately the same, and image processing operations are applied to image frames of the same size, the runtime remains practically unchanged throughout the print cycle.

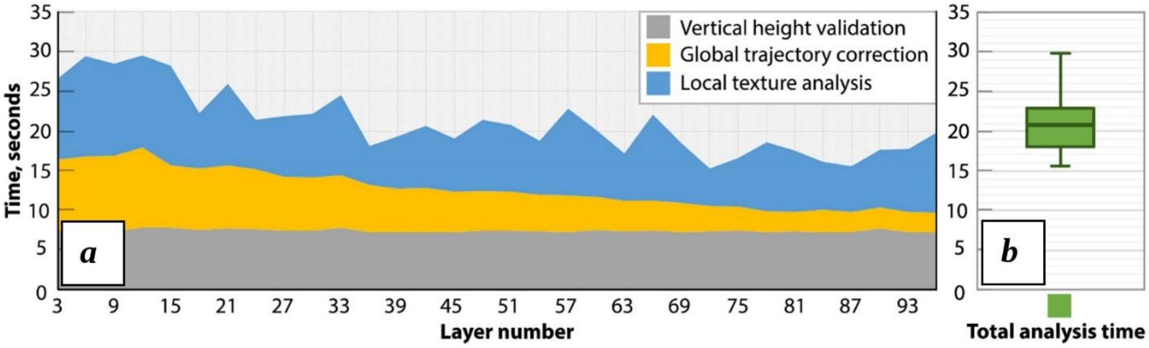


Figure 2.19: Runtime distribution: *a)* – time decomposition of layer-wise visual analysis; *b)* – distribution of the total analysis time for all of the layers.

The global texture correction step consists of MTM and ICP algorithms. The MTM algorithm is based on the OpenCV *"matchTemplate"* function, which, in turn, is based on the Fourier transform with the computational complexity $O(n\cdot log(n))$ [64,95,96]. The ICP algorithm has $O(n^2)$ complexity [97,98] which may imply long processing time when using high-resolution images. Thus, a significant reduction in the analysis time is observed due to a decrease in the geometrical dimensions of the part when approaching the upper printing layers, which, in turn, leads to a reduction in the number of data points involved in the contour transformation computations.

The final stage, local texture analysis, includes GMM clustering based on the EM algorithm with the computational complexity of $O(n)$ [99,100], and failure segmentation based on the AHC algorithm with $O(n^3)$ complexity [90,101], which makes it relatively slow. The AHC algorithm, however, does not introduce a significant time delay due to the small number of texture centroids that make up the hierarchy. Due to the sporadic initialization of the center clusters during the GMM segmentation, however, the execution time of the expectation-maximization part may vary over a wide range.

Regular printing time (without using the visual analysis) of the model [67] is 2 hours and 14 minutes (8,040 seconds). The total analysis time varies between 15 and 30 seconds with an average of 21.4 seconds. Thus, introducing the visual analysis in the printing process of the given part increases the total time of the production process by an amount of the order of 50% (2-6):

$$\frac{L_t \cdot N}{P_r} \cdot 100\% = \frac{21.4 \cdot 175}{8040} \cdot 100\% = 46.6\% \,, \tag{2-8}$$

where L_t – is the average layer analysis time, N – is the total number of layers, P_r – is the regular printing time.

Manufacturing one layer of a large part may take several minutes, which is considerably longer than the processing time, and thus, the total production time may increase insignificantly. In a real-world scenario, an immediate suspension of the printing process in the event of a fatal error can be possible only in the case of constant monitoring by a human operator, which is difficult due to the time duration of the printing process (hours, or even days, per one part). Thus, despite the latency, delayed failure detection has an advantage over printing without error detection. Additionally, the printing process can be aborted immediately after the detection of a critical side-view level mismatch (since large deformations most probably will lead to significant distortions on the side view), which, in this case, takes several seconds.

Applying the above analysis to selective layers reflecting pivotal changes in the geometry of the part or in the temperature conditions of printing can offset the time costs and bring the analytical cycle closer to the real-time mode.

2.5.5 Failures Database and Future Development

Examples of segmentation of artificially created print defects are presented in Figure 2.20. One of the proposed methods for eliminating defects is the operation of ironing (smoothing out irregularities by moving the hot nozzle over the surface without material extrusion), followed by repeated segmentation and texture analysis. If the defective area is preserved, this area must be melted and filled with 100% material. A protocol for this procedure is under development.

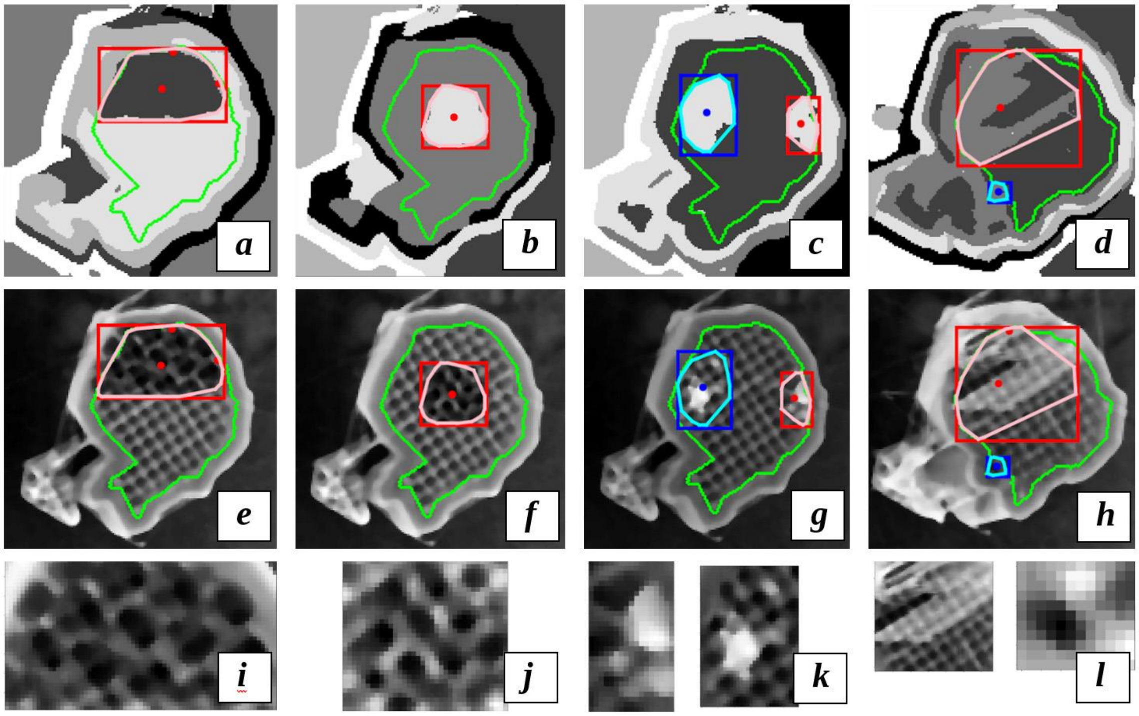


Figure 2.20: Detected regions with abnormal texture: *(a–d)* – segmented textures; *(e–h)* – detected failures; *(i–l)* – cropped regions of interest with failures (not to scale).

At this stage of texture analysis, in addition to direct segmentation, images of defective areas are saved in the database for subsequent labeling and classification. In the future, appropriate correction procedures will be developed individually for each type of defect.

It should also be noted some of the limitations present in this paper, which could be resolved in future works:

1. The considered algorithm can ambiguously interpret the layer contours of parts with complex geometric shapes. In the future, experiments will be conducted to determine effects of local geometry on the accuracy of determining the contours of the printed layers.
2. At this stage of development, a defect on the backside of the part will be unnoticed during the side view height validation. In the future, it is necessary to analyze the possibilities of solving this problem (e.g., the use of mirrors, moving the printed part with respect to the camera, or the use of multiple cameras).

3. The impact of environmental factors was minimized by utilizing the movable light frame and applying image processing techniques to the source images. However, further tests will be carried out under various environmental conditions in order to obtain quantitative characteristics of the reliability of the algorithm.
4. It is also necessary to conduct a series of tests to take into account the influence of the size of the printing part, since a large-sized model may be out of focus of the camera, and linear projection distortions should be considered.
5. Additional detailed experiments are needed on a large number of 3-D models to fully analyze the effectiveness of the developed method and accurately determine the boundaries between nominal and defective printing.
6. Mechanical tests on manufactured parts are required to analyze the correlation of pauses between each layer and mechanical strength.

2.6 Conclusions

The development of an adaptive 3-D printing control that allows updating the G-code during the fabrication process is a comprehensive and complex problem, because it is challenging to (1) uniquely visually determine the type of error, (2) establish a direct causal relationship between the type of error and the printing parameter involved, and (3) declare in advance what parameter value (scaling coefficients, feed rate, temperature, traveling speed, etc.) should be used to correct the failure.

This work introduces a conceptually new approach to comprehensive monocular layer-wise analysis of extrusion-based FFF AM processes. The developed method was tested on one part in normal printing mode without manufacturing defects. Several individual case studies were studied to demonstrate the detection of contour deviations and anomalies within the printed layer. However, additional studies are required for a detailed analysis of the performance of the developed method in the production of various parts under different 3-D printing conditions.

The experiments above are based on the assumption that the mechanical parameters (stability of assembly, the presence of grease in moving parts, belt tension, the electrical voltage of stepper motor drivers, etc.) of the printer are configured and calibrated optimally. The experimental results obtained for the case of the nominal printing mode without deviations allow determining the accuracy and tolerance of the adaptive algorithm.

Thus, at this stage of the research, the presented work is more an intelligent printing suspension tool designed to save time and material rather than a full failure correction algorithm for printing enhancement. However, this work will allow users to systematize knowledge about failure mechanisms and will serve as a starting point for deep study in the future and a full failure correction system for open source additive manufacturing.

2.7 Bibliography

[1] P. Ciraud. Pierre Alfred Leon Ciraud, A method and apparatus for manufacturing objects made of any arbitrary material meltable. German patent application DE2263777A1. December 28, 1971, https://patents.google.com/patent/DE2263777A1/en.

[2] E. Sells, S. Bailard, Z. Smith, A. Bowyer, V. Olliver, RepRap: the replicating rapid prototype – maximizing customizability by breeding the means of production, in F.T. Piller, M.M. Tseng (Eds.), Handbook of Research in Mass Customization and Personalization, World Scientific, Singapore, 2009, Vol. 1, pp. 568–580.

[3] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer and A. Bowyer, 2011. RepRap – the replicating rapid prototype, Robotica. 29(1), 177-191. https://doi.org/10.1017/S026357471000069X.

[4] A. Bowyer, 2014. 3D Printing and Humanity’s First Imperfect Replicator. 3D Printing and Additive Manufacturing, 1(1), 4–5. https://doi.org/10.1089/3dp.2013.0003.

[5] T. Wohlers, I. Campbell et al., Wohlers Report: 3D Printing and Additive Manufacturing State of the Industry, Wohlers Associates, Inc., 2018.

[6] N. Gershenfeld, Fab: the coming revolution on your desktop – from personal computers to personal fabrication, Basic Books, New York, 2005.

[7] J.M. Pearce, C.M. Blair, K.J. Laciak, R. Andrews, A. Nosrat and I. Zelenika-Zovko, 2010. 3-D printing of open source appropriate technologies for self-directed sustainable development, Journal of Sustainable Development. 3(4), 17-29. https://doi.org/10.5539/jsd.v3n4p17.

[8] C. Mota, The rise of personal fabrication. C&C 2011, In Proceedings of the 8th ACM conference on Creativity and cognition, pp. 279–288. https://doi.org/10.1145/2069618.2069665.

[9] B.T. Wittbrodt, A.G. Glover, J. Laureto, G.C. Anzalone, D. Oppliger, J.L. Irwin and J.M. Pearce, 2013. Life-cycle economic analysis of distributed manufacturing

with open-source 3-D printers, Mechatronics, 23(6), 713-726. https://doi.org/10.1016/j.mechatronics.2013.06.002.

[10] J. Gwamuri, B.T. Wittbrodt, N.C. Anzalone and J.M. Pearce, 2014. Reversing the trend of large scale and centralization in manufacturing: The case of distributed manufacturing of customizable 3-D-printable self-adjustable glasses, Challenges in Sustainability, 2(1), 30-40. https://doi.org/10.12924/cis2014.02010030.24

[11] J. Kietzmann, L. Pitt and P. Berthon, 2015. Disruptions, decisions, and destinations: Enter the age of 3-D printing and additive manufacturing, Business Horizons, 58(2), 209-215. https://doi.org/10.1016/j.bushor.2014.11.005.

[12] G. Dafermos, 2015. Transforming the productive base of the economy through the open design commons and distributed manufacturing, Journal of Peer Production, Issue 7.

[13] J. Mai, L. Zhang, F. Tao and L. Ren, 2016. Customized production based on distributed 3D printing services in cloud manufacturing, International Journal of Advanced Manufacturing Technology, 84(1-4), 71-83. https://doi.org/10.1007/s00170-015-7871-y.

[14] J.S. Srai, M. Kumar, G. Graham, W. Phillips, J. Tooze, S. Ford, P. Beecher, B. Raj, M. Gregory, M.K. Tiwari and B. Ravi, 2016. Distributed manufacturing: scope, challenges and opportunities, International Journal of Production Research, 54(23), 6917-6935. https://doi.org/10.1080/00207543.2016.1192302.

[15] B. Wittbrodt, J. Laureto, B. Tymrak and J.M. Pearce, 2015. Distributed manufacturing with 3-D printing: a case study of recreational vehicle solar photovoltaic mounting systems, Journal of Frugal Innovation, 1(1), 1. https://doi.org/10.1186/s40669-014-0001-z.

[16] A.L. Woern and J.M. Pearce, 2017. Distributed manufacturing of flexible products: Technical feasibility and economic viability, Technologies, 5(4), 71. https://doi.org/10.3390/technologies5040071.

[17] E.E. Petersen and J.M. Pearce, 2017. Emergence of home manufacturing in the developed world: Return on investment for open-source 3-D printers, Technologies, 5(1), 7. https://doi.org/10.3390/technologies5010007.

[18] E.E. Petersen, R.W. Kidd and J.M. Pearce, 2017. Impact of DIY home manufacturing with 3D printing on the toy and game market, Technologies, 5(3), 45. https://doi.org/10.3390/technologies5030045.

[19] J.M. Pearce, 2015. Quantifying the Value of Open Source Hardware Development, Modern Economy, 6, 1-11. http://dx.doi.org/10.4236/me.2015.61001.

[20] M. Moritz, T. Redlich, S. Günyar, L. Winter and J.P. Wulfsberg, 2019. On the Economic Value of Open Source Hardware – Case Study of an Open Source Magnetic Resonance Imaging Scanner, Journal of Open Hardware, 3(1), 2. http://doi.org/10.5334/joh.14.

[21] J.M. Pearce, 2012. Building research equipment with free, open-source hardware, Science, 337(6100), 1303-1304. https://doi.org/10.1126/science.1228183.

[22] J.M. Pearce, Open-Source Lab: How to Build Your Own Hardware and Reduce Research Costs, first ed., Elsevier, Waltham, MA, USA, 2014.

[23] T. Baden, A. Chagas, T. Marzullo, L. Prieto-Godino, T. Euler, 2015. Open Labware: 3-D Printing Your Own Lab Equipment, PLoS Biol. 13, e1002175. https://doi.org/10.1371/journal.pbio.1002086.

[24] M. Coakley, D.E. Hurt, 2016. 3D Printing in the Laboratory: Maximize Time and Funds with Customized and Open-Source Labware, J. Lab. Autom. 21(4), 489–495. https://doi.org/10.1177/2211068216649578.

[25] J.M. Pearce, 2016. Return on investment for open source scientific hardware development, Science and public policy, 43(2), 192–195. https://doi.org/10.1093/scipol/scv034.

[26] M. Kreiger and J.M. Pearce, 2013. Environmental impacts of distributed manufacturing from 3-D printing of polymer components and products, MRS Online Proceedings Library Archive, 1492, 85-90. https://doi.org/10.1557/opl.2013.319.25

[27] M. Kreiger and J.M. Pearce, 2013. Environmental life cycle analysis of distributed threedimensional printing and conventional manufacturing of polymer products, ACS Sustainable Chemistry & Engineering, 1(12), 1511-1519. https://doi.org/10.1021/sc400093k.

[28] D. Chen, S. Heyer, S. Ibbotson, K. Salonitis, J.G. Steingrímsson and S. Thiede, 2015. Direct digital manufacturing: definition, evolution, and sustainability implications, Journal of Cleaner Production, 107, 615-625. https://doi.org/10.1016/j.jclepro.2015.05.009.

[29] C. Kohtala and S. Hyysalo, 2015. Anticipated environmental sustainability of personal fabrication, Journal of Cleaner Production, 99, 333-344. https://doi.org/10.1016/j.jclepro.2015.02.093.

[30] S. Zhong and J.M. Pearce, 2018. Tightening the loop on the circular economy: Coupled distributed recycling and manufacturing with recyclebot and RepRap 3-D printing, Resources, Conservation and Recycling, 128, 48-58. https://doi.org/10.1016/j.resconrec.2017.09.023.

[31] J. King, The True Cost of Running a Desktop 3D Printer. https://3dprinthq.com/costrunning-desktop-3d-printer, 2017 (accessed 01 March 2020).

[32] Reddit 3D Printing Community, https://www.reddit.com/r/3Dprinting/comments/57ycv0/poll_whats_your_average_failure_rate (accessed 01 March 2020).

[33] S. Nuchitprasitchai, M.C. Roggemann & J.M. Pearce, 2017. Factors effecting real-time optical monitoring of fused filament 3D printing, Progress in Additive

Manufacturing Journal, 2017, 2(3), 133-149. https://doi.org/10.1007/s40964-017-0027-x.

[34] S. Nuchitprasitchai, M.C. Roggemann & J.M. Pearce, 2017. Three Hundred and Sixty Degree Real-time Monitoring of 3-D Printing Using Computer Analysis of Two Camera Views, J. Manuf. Mater. Process. 2017, 1(1), 2. https://doi.org/10.3390/jmmp1010002.

[35] K. Garanger, T. Khamvilai, E. Feron, E. 3D Printing of a Leaf Spring: A Demonstration of Closed-Loop Control in Additive Manufacturing, IEEE Conference on Control Technology and Applications (CCTA), 2018, pp. 465–470. https://doi.org/10.1109/CCTA.2018.8511509.

[36] U. Delli, S. Chang, Automated processes monitoring in 3D printing using supervised machine learning, Procedia Manufacturing, 26 (2018), 865-870. https://doi.org/10.1016/j.promfg.2018.07.111.

[37] J. Fastowicz, K. Okarma. Texture based quality assessment of 3D prints for different lighting conditions. In Proceedings of the International Conference on Computer Vision and Graphics, ICCVG 2016. https://doi.org/10.1007/978-3-319-46418-3_2.

[38] C. Caetano, J.A. dos Santos, W.R. Schwartz, Optical Flow Co-occurrence Matrices: A novel spatiotemporal feature descriptor, 23rd International Conference on Pattern Recognition, ICPR 2016. https://doi.org/10.1109/ICPR.2016.7899921.

[39] L. Nanni, S. Brahnam, S. Ghidoni, E. Menegatti, T. Barrier, 2013. Different Approaches for Extracting Information from the Co-Occurrence Matrix. PLoS ONE 8(12): e83554. https://doi.org/10.1371/journal.pone.0083554.

[40] L. Setia, A. Teynor, A. Halawani, H. Burkhardt, Image classification using cluster cooccurrence matrices of local relational features. Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, MIR 2006. https://doi.org/10.1145/1178677.1178703.26

[41] R.M. Haralick, K. Shanmugam, I. Dinstein, 1973. Textural Features for Image Classification, IEEE Transactions on Systems, Man, and Cybernetics, 3(6). https://doi.org/10.1109/TSMC.1973.4309314.

[42] I.T. Cummings, M.E. Bax, I.J. Fuller, A.J. Wachtor, J.D. Bernardin, 2017. A Framework for Additive Manufacturing Process Monitoring & Control. In: Mains M., Blough J. (eds) Topics in Modal Analysis & Testing, Volume 10. Conference Proceedings of the Society for Experimental Mechanics Series. Springer, Cham. 10, 137-146. https://doi.org/10.1007/978-3-319-54810-4_14.

[43] P.K. Rao, J. Liu, D. Roberson, Z. Kong, C. Williams, 2015. Online Real-Time Quality Monitoring in Additive Manufacturing Processes Using Heterogeneous Sensors. J. Manuf. Sci. Eng. 2015, 137(6): 061007. https://doi.org/10.1115/1.4029823.

[44] Z. Jin, Z. Zhang, G.X. Gu, Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning, Manuf. Lett. 22 (2019) 11–15, https://doi.org/10.1016/j.mfglet.2019.09.005.

[45] K. He, Q. Zhang, Y. Hong, Profile monitoring based quality control method for fused deposition modeling process, J. Intell. Manuf. 30 (2019) 947–958, https://doi.org/10.1007/s10845-018-1424-9.

[46] The Spaghetti Detective Project, (2020) (accessed 01 July 2020), https://www.thespaghettidetective.com.

[47] L. Scime, J. Beuth, Anomaly Detection and Classification in a Laser Powder Bed Additive Manufacturing Process Using a Trained Computer Vision Algorithm, Additive Manufacturing 2018, (19), 114–126, https://doi.org/10.1016/j.addma.2017.11.009.

[48] J. Xiong, G. Zhang, 2014. Adaptive control of deposited height in GMAW-based layer additive manufacturing, Journal of Materials Processing Technology, 214(4), 962-968. https://doi.org/10.1016/j.jmatprotec.2013.11.014.

[49] A.R. Nassar, J.S. Keist, E.W. Reutzel, T.J. Spurgeon, 2015. Intra-layer closed-loop control of build plan during directed energy additive manufacturing of Ti-6Al-4V, Additive Manufacturing, 6, 39-52. https://doi.org/10.1016/j.addma.2015.03.005.

[50] I.A. Okaro, S. Jayasinghe, C. Sutcliffe, K. Black, P. Paoletti, P.L. Green, 2019. Automatic Fault Detection for Laser Powder-Bed Fusion Using Semi-Supervised Machine Learning. Additive Manufacturing, 27, 42-53. https://doi.org/10.1016/j.addma.2019.01.006.

[51] K. Garanger, E. Feron, P-L. Garoche, et al. Foundations of Intelligent Additive Manufacturing. ArXiv 2017, arXiv:1705.00960 [cs.LO].

[52] B. Yuan, G.M. Guss, A.C. Wilson et al., 2018. Machine-Learning-Based Monitoring of Laser Powder Bed Fusion, Wiley Online Library, Advanced Materials Technologies, 3(12). https://doi.org/10.1002/admt.201800136.

[53] P. Sitthi-Amorn, J.E. Ramos, Y. Wang, et al. MultiFab: a Machine Vision Assisted Platform for Multi-Material 3D Printing, ACM Transactions on Graphics, TOG 2015. 34(4), article no. 129, https://doi.org/10.1145/2766962.

[54] N. Razaviarab, S. Sharifi, Y.M. Banadaki, 2019. Smart additive manufacturing empowered by a closed-loop machine learning algorithm. Proc. SPIE 10969, Nano-, Bio-, InfoTech Sensors and 3D Systems III, 109690H, 2019, https://doi.org/10.1117/12.2513816.

[55] W. Xiong, M. Bedewy, H. Kuhn, A. Elwany, Z. Pei, Accelerating NSF Research in Additive Manufacturing toward Industrial Applications. National Science Foundation Workshop, Pittsburgh, PA, USA, 2018.

[56] Prusa Research: Print Quality Troubleshooting. https://help.prusa3d.com/category/print-quality-troubleshooting_225 (accessed 02 July 2024).

[57] Troubleshooting Guide to Common 3D Printing Problems. https://all3dp.com/1/common-3d-printing-problems-troubleshooting-3d-printer-issues, 2020 (accessed 01 March 2020).

[58] MatterHackers: 3D Printer Troubleshooting Guide, (2016) (accessed 01 March 2020), https://www.matterhackers.com/articles/3d-printer-troubleshootingguide.

[59] G.C. Anzalone, B. Wijnen and J.M. Pearce, 2015. Multi-material additive and subtractive prosumer digital fabrication with a free and open-source convertible delta RepRap 3-D printer. Rapid Prototyping Journal, 21(5), 506-519. https://doi.org/10.1108/RPJ-09-2014-0113.27

[60] S. Graves, Johann C. Rocholl (Rostock) Style Delta Robot Kinematics. https://reprap.org/wiki/File:Rostock_Delta_Kinematics_3.pdf, 2015 (accessed 01 March 2020).

[61] SONY IMX322 Datasheet, https://dashcamtalk.com/cams/lk-7950-wd/Sony_IMX322.pdf (accessed 01 March 2020).

[62] Camera calibration with OpenCV: asymmetrical circle calibration pattern. https://docs.opencv.org/2.4/_downloads/acircles_pattern.png (accessed 01 March 2020).

[63] J. Heikkila, 2000. Geometric camera calibration using circular control points. IEEE Transactions on pattern analysis and machine intelligence, 22(10), 1066-1077. https://doi.org/10.1109/34.879788.

[64] G. Bradski, A. Kaehler, Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library, first ed., O'Reilly Media, Sebastopol, CA, USA, 2016.

[65] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, Cambridge, United Kingdom, 2004.

[66] D. Fletcher, I. Parberry, 3D Math Primer for Graphics and Game Development, Wordware Publishing, Inc. Sadbury, MA, USA, 2002.

[67] Low Polygonal STL Fox model. https://www.thingiverse.com/thing:937740, CC BY-NCSA 3.0 license (accessed 01 March 2020).

[68] LE 12V LED Flexible SMD 2835 16.4ft Strip Light. https://www.lepro.com/12v-ledstrip-light-3528-4100057-dw.html (accessed 01 March 2020).

[69] Marlin Firmware: Open Source RepRap Driver. https://marlinfw.org (accessed 01 March 2020).

[70] L.S.V. Thomas, J. Gehrig, 2020. Multi-template matching: a versatile tool for objectlocalization in microscopy images. BMC Bioinformatics 21, 44. https://doi.org/10.1186/s12859-020-3363-7.

[71] Y. Chen, G. Medioni, 1992. Object modelling by registration of multiple range images, Image and Vision Computing, 10(3), 145-155. https://doi.org/10.1016/0262-8856(92)90066-C.

[72] P.J. Besl, N.D. McKay, 1992. A Method for Registration of 3-D Shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence. 14(2), 239-256. https://doi.org/10.1109/34.121791.

[73] OpenCV: Template Matching. https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html (accessed 01 March 2020).

[74] E. Hernandez-Rivera, S.P. Coleman, M.A. Tschopp, 2016. Using similarity metrics to quantify differences in high-throughput datasets: application to X-ray diffraction patterns. ACS Comb Sci. 19(1), 25–36. https://doi.org/10.1021/acscombsci.6b00142.

[75] A.A. Goshtasby, Similarity and dissimilarity measures, in: S.Singh, S.B. Kang (Eds.), Image Registration. Advances in Computer Vision and Pattern Recognition, Ch. 2, pp. 7–66. Springer, London, United Kingdom 2012. https://doi.org/10.1007/978-1-4471-2458-0_2.

[76] S.H. Cha, 2007. Comprehensive survey on distance/similarity measures between probability density functions. Int. J. Math. Model. Meth. Appl. Sci., 1.28

[77] T. Leung, J. Malik, 2001. Representing and recognizing the visual appearance of materials using three-dimensional textons, International Journal of computer vision 43, 29-44. https://doi.org/10.1023/A:1011126920638.

[78] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, M. Pietikäinen, 2019. From BoW to CNN: Two Decades of Texture Representation for Texture Classification, International Journal of Computer Vision. 127(1), 74-109. https://doi.org/10.1007/s11263-018-1125-z.

[79] J. Malik, S. Belongie, T. Leung, J. Shi, 2001. Contour and Texture Analysis for Image Segmentation, International Journal of Computer Vision. 43(1), 7-27. https://doi.org/10.1023/A:1011174803800.

[80] U.R. Acharya, K.M. Meiburger, J.E. Koh et al., 2019. A Novel Algorithm for Breast Lesion Detection Using Textons and Local Configuration Pattern Features With Ultrasound Imagery, IEEE Access. 7, 22829-22842. https://doi.org/10.1109/ACCESS.2019.2898121.

[81] L. Zhang, G. Yang, X. Ye, 2019. Automatic skin lesion segmentation by coupling deep fully convolutional networks and shallow network with textons, J. of Medical Imaging. 6(2), 024001. https://doi.org/10.1117/1.JMI.6.2.024001.

[82] T. Joseph, Python implementation of the Leung-Malik filter bank. https://github.com/CVDLBOT/LM_filter_bank_python_code, 2016 (accessed 01 March 2020).

[83] A. Crivellaro, V. Lepetit, 2014. Robust 3D Tracking with Descriptor Fields, IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, 3414-3421. https://doi.org/10.1109/CVPR.2014.436.

[84] M. Pietikäinen, A. Hadid, G. Zhao, T. Ahonen, Computer vision using local binary patterns, Springer, London, United Kingdom, 2011.

[85] U. Kandaswamy, S. Schuckers, D. Adjeroh, 2011. Comparison of texture analysis schemes under nonideal conditions, IEEE Trans Image Processing. 20(8), 2260-2275. https://doi.org/10.1109/TIP.2010.2101612.

[86] W. Zhang, S. Shan, W. Gao, X. Chen, H. Zhang, 2005. Local Gabor binary pattern histogram sequence (LGBPHS): A novel nonstatistical model for face representation and recognition, Tenth IEEE International Conference on Computer Vision, ICCV 2005. 1, 786–791. https://doi.org/10.1109/ICCV.2005.147.

[87] C. Bishop, Pattern Recognition and Machine Learning, Springer, New York, USA, 2006.

[88] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Statist. Soc. Ser. B, 39 (1977), 1-38.

[89] G.J. McLachlan, T. Krishnan, S.K. Ng, The EM Algorithm, Papers, No. 2004, 24, Humboldt-Universität zu Berlin, Center for Applied Statistics and Economics (CASE), Berlin, 2004.

[90] F. Nielsen, Hierarchical Clustering, in: Introduction to HPC with MPI for Data Science, Undergraduate topics in computer science, Springer, Basel, Switzerland, 2016, pp. 195–211. https://doi.org/10.1007/978-3-319-21903-5_8.

[91] E. Somoza, G.O. Cula, C. Correa, J.B. Hirsch, Automatic Localization of Skin Layers in Reflectance Confocal Microscopy, in: A. Campilho, M. Kamel (Eds.), Image Analysis and Recognition. ICIAR 2014. Lecture Notes in Computer Science, vol 8815. Springer, Cham, pp. 141–150.29

[92] M. Varma, A. Zisserman, 2003. Texture classification: are filter banks necessary?, In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003, Madison, WI, USA, pp. II–691. https://doi.org/10.1109/CVPR.2003.1211534.

[93] M. Varma, A. Zisserman, 2005. A Statistical Approach to Texture Classification from Single Images, Int J Comput Vision. 62, 61-81. https://doi.org/10.1023/B:VISI.0000046589.39864.ee.

[94] S.-C. Zhu, C.-E. Guo, Y. Wang, Z. Xu, 2005. What are Textons? International Journal of Computer Vision 62(1/2), 121-143. https://doi.org/10.1007/s11263-005-4638-1.

[95] Mathias Lohne, The computational complexity of the fast fourier transform, Tech. Rep., Tech. Rep. 2017 (2017) [Online]. Available: https://folk.uio.no/mathialo/texts/fftcomplexity.pdf.

[96] D.-M. Tsai, C.-T. Lin, Fast normalized cross correlation for defect detection, Pattern Recognit. Lett. 24 (15) (2003) 2625–2631, https://doi.org/10.1016/S0167-8655(03)00106-5.

[97] T. Jost, H. Hügli, Fast ICP algorithms for shape registration, in: L. Van Gool (Ed.), Pattern Recognition. DAGM 2002. Lecture Notes in Computer Science,

2449 Springer, Berlin, Heidelberg, 2002, , https://doi.org/10.1007/3-540-45783-6_12.

[98] D. Arthur, S. Vassilvitskii, Worst-case and smoothed analysis of the ICP algorithm, with an application to the k-means method, 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06), Berkeley, CA, 2006, 2006, pp. 153–164, , https://doi.org/10.1109/FOCS.2006.79.

[99] R.C. Pinto, P.M. Engel, Correction: a fast incremental gaussian mixture model, PLoS One 10 (10) (2015) e0141942, https://doi.org/10.1371/journal.pone.0139931.

[100] J.J. Verbeek, N. Vlassis, B. Kröse, Efficient greedy learning of Gaussian mixture models, Neural Comput. 15 (2) (2003) 469–485, https://doi.org/10.1162/089976603762553004.

[101] W.H.E. Day, H. Edelsbrunner, Efficient algorithms for agglomerative hierarchical clustering methods, J. Classif. 1 (1984) 7–24, https://doi.org/10.1007/BF01890115.

Chapter 3

3 Interlayer Anomaly Detection Based on HOG-features and Synthetic Images

This chapter[2] is adapted from the "Towards Smart Monitored AM: Open Source In Situ Layer-wise 3-D Printing Image Anomaly Detection Using Histograms of Oriented Gradients and a Physics-based Rendering Engine" with minor modifications to the version published in Additive Manufacturing, vol. 52, 102690, 2022, DOI:10.1016/j.addma.2022.102690.

3.1 Abstract

This study presents an open source method for detecting 3-D printing anomalies by comparing images of printed layers from a stationary monocular camera with synthetic G-code-based reference images of an ideal process generated with Blender, an open source free physics-based rendering engine. Recognition of visual deviations was accomplished by analyzing the similarity of histograms of oriented gradients (HOG) of local image areas. The developed technique requires preliminary modeling of the working environment to achieve the best match for orientation, color rendering, lighting, and other parameters of the printed part. The output parameter of the developed visual analysis method is a similarity measure for each area of the printed layer image with an analogous area of its synthetic render, representing an ideal print. Twelve similarity and distance measures were implemented and compared for their effectiveness at detecting 3-D printing errors on six different representative failure types (local infill defects, presence of a foreign body in the layer, spaghetti problem, separation and shift of the printing part from the working surface, defects in thin walls, and layer shift) and their control error-free print images. The results show that although Kendall's tau, Jaccard, and Sorensen similarities are the most sensitive,

---

[2] A version of this chapter has been published in Additive Manufacturing journal. A. Petsiuk, J.M. Pearce, Towards smart monitored AM: open source *in situ* layer-wise 3D printing image anomaly detection using histograms of oriented gradients and a physics-based rendering engine. Additive Manufacturing, vol. 52, no. 102690, 2022, doi:10.1016/j.addma.2022.102690.

Pearson's r, Spearman's rho, cosine, and Dice similarities produce the more distinguishable results. This open source method allows the program to notice critical errors in the early stages of their occurrence and either pause manufacturing processes for further investigation by an operator or in the future intelligent automatic error correction. The implementation of this novel method does not require preliminary data for training, and the greatest efficiency can be achieved with the mass production of parts by either additive or subtractive manufacturing of the same geometric shape. This open source method has the potential means of enabling smart distributed recycling for additive manufacturing in challenging environments.

3.2 Introduction

Over the past decades, additive manufacturing (AM) has become a widespread technology that has found application in various fields of science and technology. AM allows the fabrication of high-performance components with complex geometries and continues to attract research interest. Extrusion-based 3-D printing, widely spread with the open source release of the self-replicating rapid prototyper (RepRap) [1-3], remains prevailing manufacturing technology due to its low cost [4], availability of components, and a wide variety of printing materials [5,6] including waste plastics [7-10]. Despite its affordability and relative ease of use, however, this technology is not free from fabrication failures, which reduces economic impact [11,12], environmental merits [13], and limits the prospects for industrialization [14,15].

According to a recent comprehensive state-of-the-art review of monitoring techniques for material extrusion AM [16], the number of publications in the field of anomaly analysis grows steadily as this is a major impediment to widespread deployment. This phenomenon can be explained by the fact that the fused filament fabrication (FFF) technology dominates the 3-D printing market for printers in use [17].

Analysis of extrusion-based AM processes can consist of examining parameters such as temperature [18,19], vibration [20,21], acoustic emissions [22,23], electrical characteristics [24,25], and others [26-29]. The main source of information, however, remains 2-D and 3-D image data obtained from single or multiple camera systems [16]. Since 3-D printed parts are mostly fabricated in layers, most of the developed failure detection methods analyze manufacturing processes after a certain number of layers have been printed. Nuchitprasitchai et al. [30], Johnson et al. [31], and Hurd [32] proposed the concepts of failure analysis based on comparison with Standard Tessellation Language (STL) files. Jeong et al. [33] and Wasserfall et al. [34] employed information obtained from G-code files of printing parts. Ceruti et al. [35] utilized data from computer-aided design (CAD) files. Researchers also use comparison with reference data [36,37] or ideal printing processes [38,39]. Malik et al. [40] presented a 3-D reconstruction-based scanning method for real-time monitoring of AM processes.

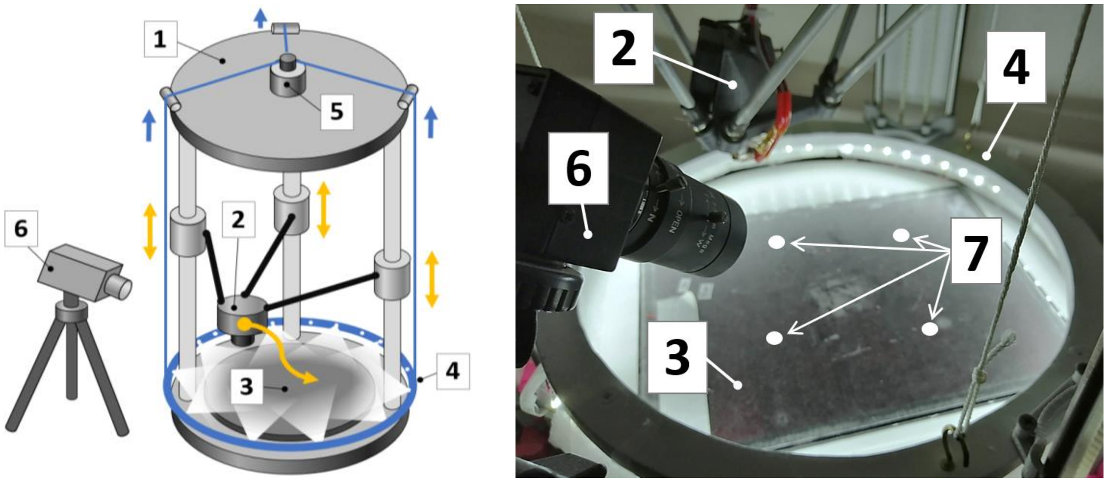
Having a way to automatically detect critical errors will significantly reduce material waste and time spent on failed prints. In order to reach this goal, this study reports on a developed monocular system for the analysis of plastic FFF processes that monitors contour deviations and infill distortions for each layer. This work expands on previous developments of the authors [41] by using an open source physics rendering engine to generate G-code-based synthetic reference images for each printing stage. With certain rendering parameters, a synthetic image can represent a real captured layer under ideal printing conditions. It is hypothesized that further comparative texture analysis based on image processing techniques can reveal the location and the degree of structural deviations. To this end a material extrusion-based 3-D printer was monitored with a stationary monocular camera. Synthetic reference images for the setup were created with Blender. Images were compared based on the similarity degree of the feature descriptors, represented by histograms of oriented gradients. Twelve similarity and distance measures were implemented and compared for their effectiveness at detecting 3-D printing errors on six different representative failure types (local infill defects, presence of a foreign body in the layer, spaghetti problem, separation and shift of the printing part from the working surface, defects in thin walls, and layer shift) and their control error-free print images. The sensitivities of the measures are quantified and the results are discussed in the context of creating intelligent manufacturing devices.

3.3 Method

3.3.1 Experimental Apparatus

For experimental tests, an open source delta-style FFF-based 3-D printer [42] was used, which represents a derivative of the self-replicating rapid prototype (RepRap) printer [1-3]. The device operates in Cartesian coordinates under the control of a RepRap Arduino Mega Polulu Shield (RAMPS) system [43]. It has a cylindrical volume of Ø240×250 mm and an extruder with a 0.4 mm nozzle diameter. The main feedstock material is 1.75 mm polylactic acid (PLA) plastic filament.

A stationary monocular camera was mounted on a tripod near the printing bed at an angle of ~45 degrees. The camera is based on a 2-megapixel 1/2.9-inch Sony IMX322 CMOS sensor [44] and has manual focus and aperture control. As shown in Figure 3.1, the four white dots on the printing surface are visual markers for the camera position and orientation determination. The markers indicate the active 90×90 mm observation area with the origin in the center of the square.



a) *b)*

Figure 3.1: Experimental apparatus: *a)* – 3-D printer schematic, *b)* – 3-D printing area. 1 – 3-D printer, 2 – extruder, 3 – printing bed, 4 – movable circular lighting platform, 5 – lighting platform drive system, 6 – camera, 7 – visual markers.

A movable circular lighting platform [41], controlled through G-code, is located above the working surface. The light frame consists of 56 light-emitting diodes with a color temperature of 6000 K (cool white light spectrum) and a total power of 18 watts. The stepper motor driving the mechanical structure is located on top of the printer and is connected to the RAMPS controller as an additional extruder.

Table 3.1 shows a set of G-code commands that are placed after the printing instructions for each layer. This pauses the fabrication process, moves the extruder out of the video surveillance area, and raises the lighting platform to a height equal to the thickness of the printed layer. Thus, the active print area is evenly lit around the perimeter, regardless of the current layer and the working level of the extruder nozzle. This allows capturing 2-D images of each completed layer with uniform illumination and applying unified image processing techniques to each image frame.

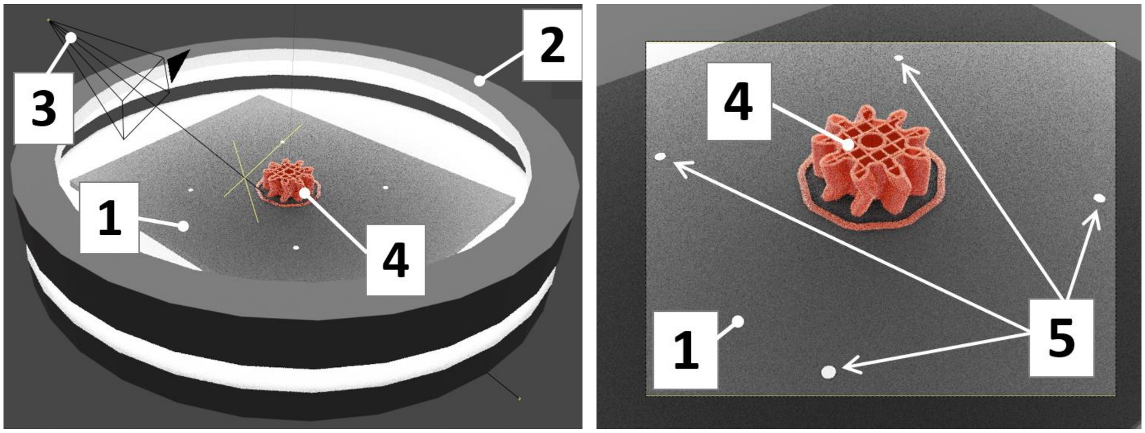
Table 3.1: Interlayer G-code commands.

G-code command	Description
M400	Wait for moves to finish
G91	Switch to relative coordinates
G1 E-20 F1000	Retract the filament 10 mm before lifting the nozzle
G1 Z80	Move the nozzle 80 mm up
G1 X20 Y20	Move the nozzle 20 mm aside
T1	Set the active extruder to 1 (lighting platform)
G1 E-0.25 F600	Move the lighting platform one layer height up
M400	Wait for moves to finish
M42 P57 S200	Indicator ON (optional)
Create layer snapshot	
G1 X-20 Y-20	Move the nozzle 20 mm back
G1 Z-80	Move the nozzle 80 mm down
G4 P500	Wait 500ms for the nozzle vibration to stabilize
T0	Set the active extruder to 0 (extruder nozzle)
G90	Switch to absolute coordinates
M42 P57 S0	Indicator OFF (optional)

3.3.2 Creation of Synthetic Reference Images

Synthetic reference images represent the ideal 3-D printed model fabricated in optimal conditions. The Blender [45] software was used to create images for each layer during the printing process. It is a multifunctional software environment for 3-D graphics. The set of Blender tools includes 3-D modeling, lighting and animation control, texture editing, and photorealistic rendering. There is also a Python scripting interface for customizing and automating the entire production pipeline.

Figure 3.2 depicts a virtual model of the key parts of the Delta printer in use.

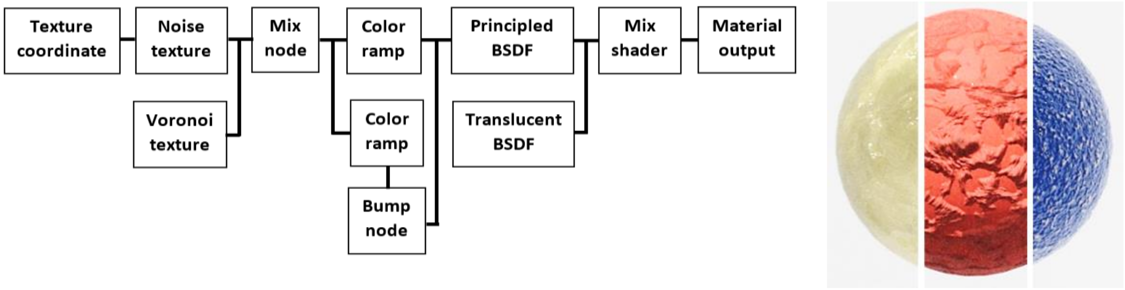


a) ***b)***

Figure 3.2: Virtual workspace: *a)* – main elements of the Delta printer modeled in Blender, *b)* – virtual camera view area. 1 – printing bed, 2 – movable lighting platform, 3 – camera, 4 – rendered G-code, 5 – visual markers.

Previous research has repeatedly shown that Blender can be used as a reliable and flexible physics simulating environment for solving scientific and engineering problems. Kent [46] utilized Blender to visualize astronomical data, Gschwandtner et al. [47] and Romulo Fernandes et al. [48] performed range sensor testing and radar simulations, respectively. Flaischlen and Wehinger [49] performed particle-resolved computational fluid dynamics modelling for chemical industry, Ilba [50] estimated solar irradiation on buildings, Rohe [51] created an optical test simulator, and, finally, Reitmann et al. [52] developed an add-on to generate semantically labeled depth-sensing data in Blender.

To create realistic images of the ideal printing process, the main components of the Delta printer have been modeled while maintaining relative proportions (Figure 3.2). Based on the authors' experiments and experience of professional 3-D computer graphics communities [53,54], a shader graph was developed for the procedural generation of realistic plastic textures (Figure 3.3).

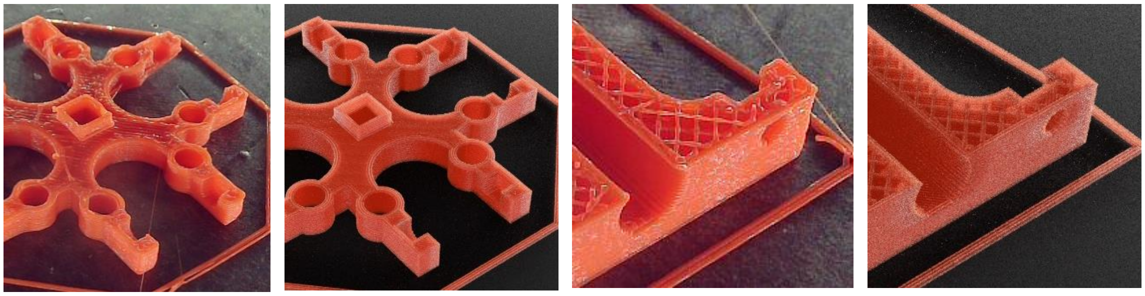


a) ***b)***

Figure 3.3: Shader graph for procedural texture generation: *a)* – shader nodes, *b)* – procedural texture samples. The output material is a combination of Principled and Translucent BSDFs. The Principled node is responsible for surface parameters such as color, roughness, and reflection. Texture nodes create realistic surface irregularities. Adding Color ramp nodes is used to limit texture irregularities and truncate differences in surface height created by the Bump node.

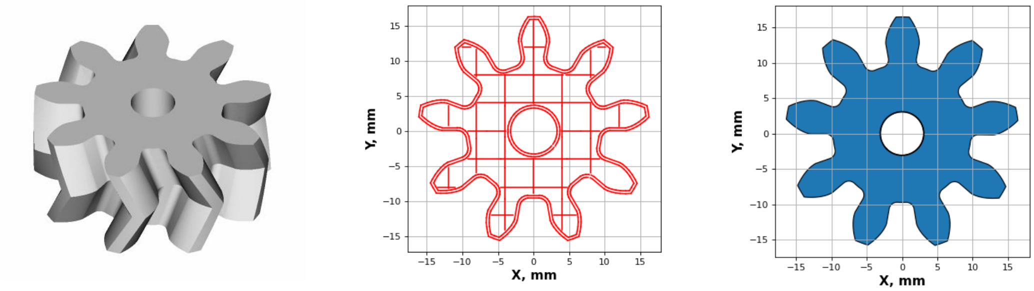
The main nodes are the Principled and Translucent bidirectional scattering distribution functions (BSDFs). The Principled BSDF shader includes multiple material properties (roughness, reflection, transmission, sheen, etc.) as layers to create a wide variety of materials, and the Translucent BSDF adds Lambertian diffuse transmission [55]. The texture shaders, in turn, add natural surface irregularities. Changing the parameters of the nodes allows maximizing similarity with the real printed parts. The given graph (Figure 3.3, a) was used to visualize the photorealistic textures of the printed parts (Figure 3.4). In addition, a single material node with emissive characteristics was used to model the lighting frame.

Several G-code exporters [56-59] were used as references in this work. An open source software toolchain has the G-code of the printing part to be loaded into the Blender programming interface and parsed layer by layer, where the extruder path is converted into a set of curves with an adjustable thickness parameter and preset material settings [60]. Therefore, each cross-section of the object can be represented as a G-code-based extruder path and an STL-based mask of the filled regions (Figure 3.5).



a) ***b)*** ***c)*** ***d)***

Figure 3.4: Texture matching examples: *a)* – synthetic image of the centrifuge part (layer 18 of 24), *b)* – real image of the centrifuge part (layer 18 of 24), *c)* – synthetic image of the slot die (layer 24 out of 35), *d)* – real image of the slot die (layer 24 out of 35).



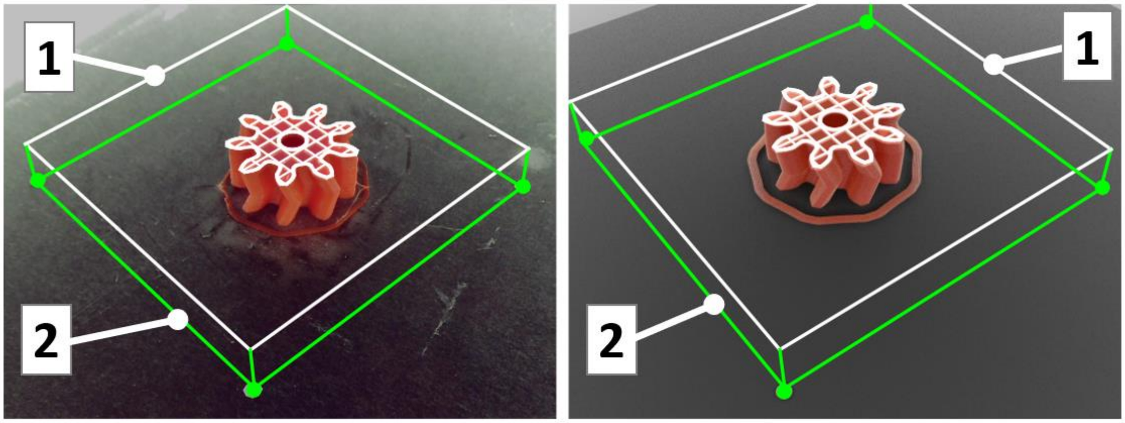
a) ***b)*** ***c)***

Figure 3.5: Printing object: *a)* – STL file of the whole part, *b)* – G-code of a layer cross-section, *c)* – STL-based layer cross-section mask.

In the programmed 3-D printing animation, a new printing layer is added with each consecutive frame, and the lighting platform is raised to the corresponding height until the synthetic print in Blender is complete. Each frame is rendered with Blender Cycles [61], a physics-based path tracer, and saved as a separate image – a "quality standard" for comparison with the actual camera image of the printed layer. This is shown in a supplemental video found in [60].

3.3.3 Comparison of the Printed Layer with the Reference Image

To compare real and synthetic data, images must be captured at a similar angle to match the printed objects. Both the real and virtual cameras in this work were located at an angle to the working area, which, however, made it possible to visually rotate the active printing surface (Figure 3.6) using a perspective projection (3-1) [62,63].



a) **b)**

Figure 3.6: Spatial position of the active printing area: *a)* – real image, *b)* – rendered image. 1 – active printing plane, 2 – print surface plane.

Thus, despite minor variations in the position and orientation of both cameras, virtual top views are used to analyze AM processes, as if both cameras were mounted directly above the print bed (Figure 3.7).

$$\begin{bmatrix} tx' \\ ty' \\ t \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \cdot \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} \tag{3-1}$$

where $[x_p \quad y_p \quad 1]^T$ is the active area of the printed layer, $\boldsymbol{M}$ is a projective transformation matrix, and $[tx' \quad ty' \quad t]^T$ is the virtual top view. Each pixel of the virtual top view can be calculated based on the following equation (3-2):

$$(tx', ty') = \left(\frac{m_{11}x_p + m_{12}y_p + m_{13}}{m_{31}x_p + m_{32}y_p + m_{33}}, \frac{m_{21}x_p + m_{22}y_p + m_{23}}{m_{31}x_p + m_{32}y_p + m_{33}} \right) \tag{3-2}$$

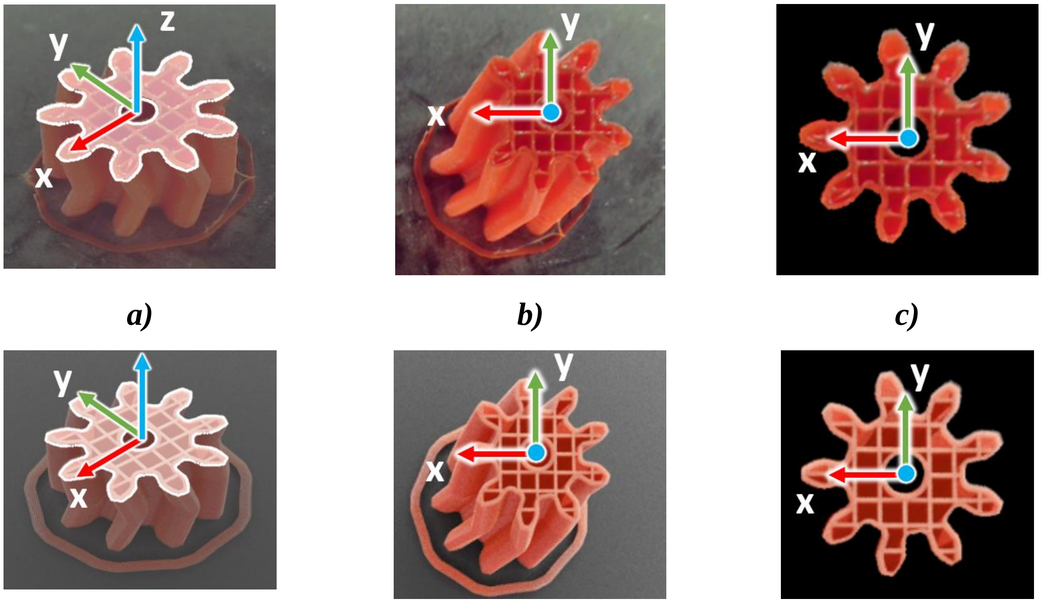
It should be noted, however, that the video surveillance area is shifting upwards by the corresponding height with the printing of each new layer, so the unwrapped top view will remain orthogonal to the optical axis of the virtual top camera. Thus, after calculating the vertical shift, the 3-D coordinates of the active printing plane $[X \quad Y \quad Z \quad 1]^T$ are projected onto the image frame to define the 2-D boundaries $[x_p \quad y_p \quad 1]^T$ for unwrapping (3-3):

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \boldsymbol{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{3x3} & \boldsymbol{t}_{3x1} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3-3}$$

where $[x_p \quad y_p \quad 1]^T$ is the active area projection onto the image plane, $\boldsymbol{K}$ is the intrinsic camera parameters obtained during calibration, $\boldsymbol{R}$ is the rotation matrix, $\boldsymbol{t}$ is the translation vector, and $[X \quad Y \quad Z \quad 1]^T$ is the 3-D coordinates of the active printing area. The camera position parameters, $\boldsymbol{R}_{3x3}$ and $\boldsymbol{t}_{3x1}$, are determined using the known coordinates of visual markers. Thus, having the position of the camera and its intrinsic parameters, it is possible to determine the correspondence between the spatial coordinates of the working space $[X \quad Y \quad Z \quad 1]^T$ and image pixels $[x_p \quad y_p \quad 1]^T$.

A G-code-based layer mask is necessary to segment the object within the layer and remove the background portion of the image from further analysis. Thus, knowing the camera location and the G-code coordinates, it is possible to rotate the printing area perpendicular to the camera axis, maintaining its origin in the center of the image (Figure 3.8).

After virtual rotation of the active printing plane, the real image is compared with the reference “ideal” one to analyze its texture and detect any possible defects inside the printed region. Image comparison is based on the similarity degree of the feature descriptors, represented by Histograms of Oriented Gradients (HOG) [64]. The descriptor analyzes local image areas, determines the orientation of the shaded gradients, and expresses this information as a histogram of direction channels.



a) ***b)*** ***c)***

d) ***e)*** ***f)***

Figure 3.7: Virtual top view: *a)* – camera frame, *b)* – unwrapped virtual top view, *c)* – masked printing area, *d)* – Blender scene frame, *e)* – unwrapped synthetic image, *f)* – masked rendered region.

HOG-based image analysis is widely applied in areas such as pattern recognition, template matching, and similarity determination. For example, Firuzi et al. [65] employed HOG features to recognize defects in electrical transformers, Malik et al. [66] presented a HOG-based landscape similarity analysis, Banerji et al. [67] enhanced HOG features with Fisher Model to extract geo-localization information from large-scale image datasets. Akila and Pavithra [68] developed an object detection algorithm based on scale invariant HOG descriptors, Joshi et al. [69] developed a sign language recognition system, and Supeng et al. [70] presented a HOG-based template matching algorithm.

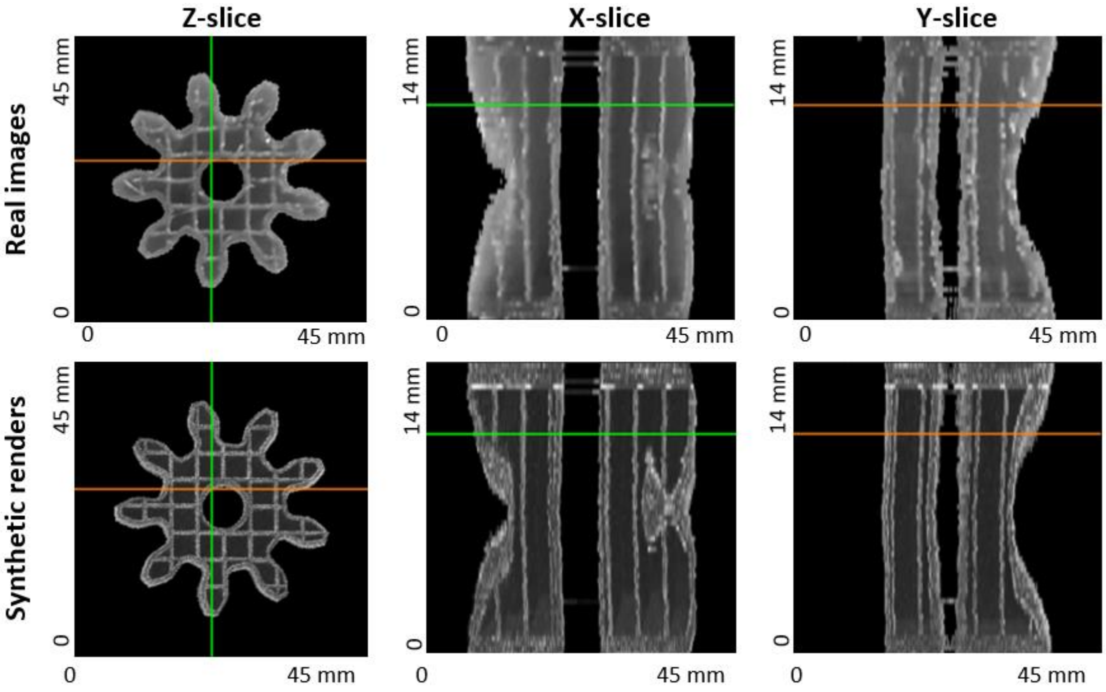


Figure 3.8: Consecutive set of unwrapped layers combined into a volumetric view. This property allows overlaying detected anomalies on a volumetric view for subsequent analysis.

Figures 3.9 and 3.10 illustrate the HOG-based detection of the dominant gradient orientation in local image areas and possible characteristic infill patterns with their feature descriptors, respectively. Each bar of the histogram (horizontal axis) corresponds to the tilt angle of the image gradient in the range from 0° to 180°. The vertical axis represents the normalized magnitude of the grayscale gradient and ranges from 0 to 100%. Thus, each individual gradient contributes to the overall histogram, where the peaks represent the prevailing tilt angles of the contrasting edges within the image area. This allows capturing basic geometric structures by detecting the directions of contrasting edges and creating a unique histogram pattern for each image area. As can be seen from Figure 3.9, the dominant orientation of the gradient continues to be determined in the presence of visual noise.

As can be seen in Figure 3.10, changing the direction of light and shadow does not affect the determination of the dominant gradient orientation. The presence of noise, however, lowers the contrast, thereby limiting the capabilities of this method.

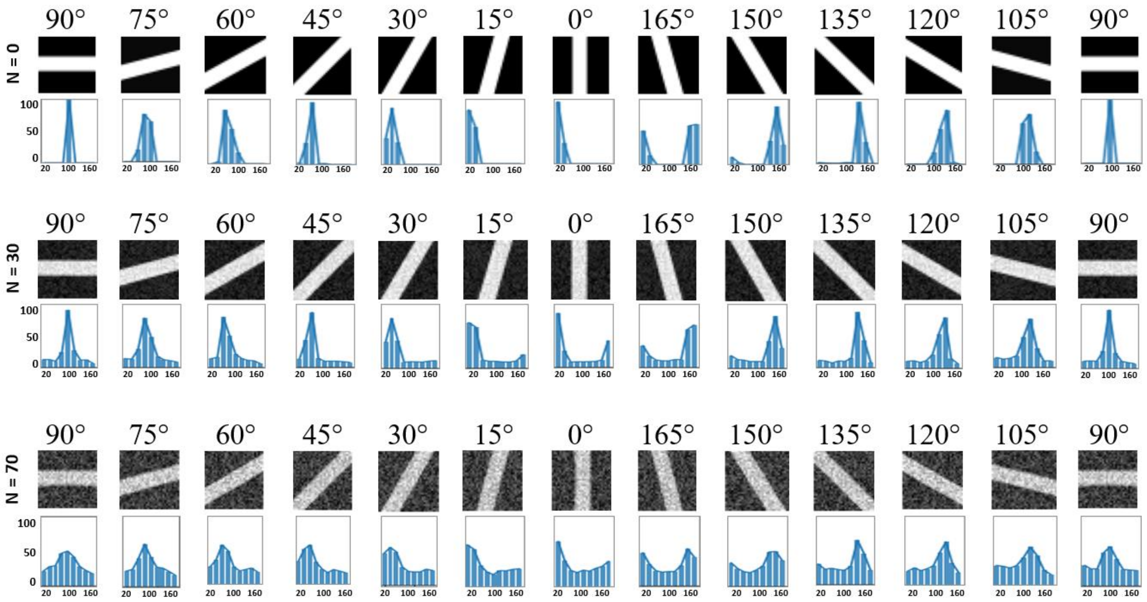


Figure 3.9: Detection of the dominant gradient orientation in local image areas (top) using histograms of oriented gradients (bottom). Each bar of the histogram corresponds to the tilt angle of the image gradient in the range from 0° to 180°. The N parameter reflects the noise level in the source image.

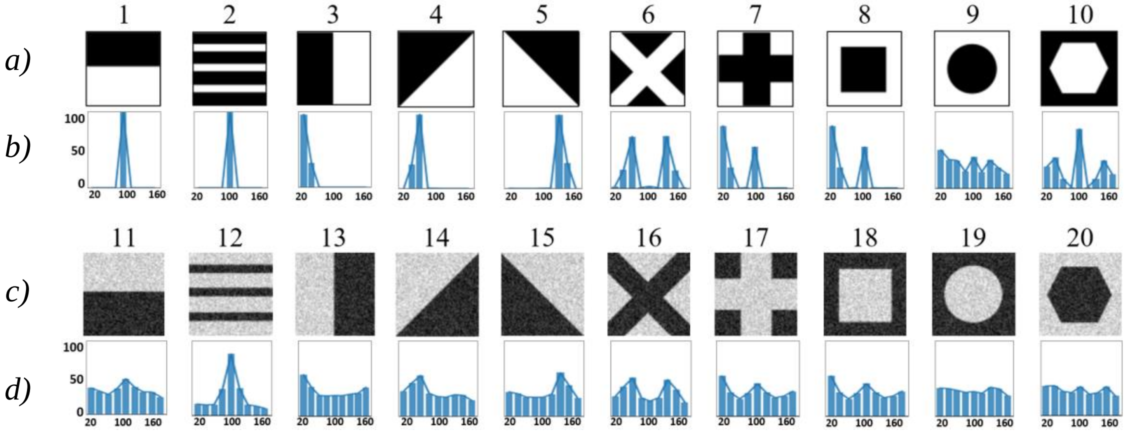


Figure 3.10: Characteristic infill patterns *(a, c)* and their feature descriptors *(b, d).* The HOG method captures the underlying geometric structures representing different infill areas by identifying the directions of contrasting edges and creating a unique histogram pattern for each image region.

Figure 3.11 shows the stages of comparative image analysis. The image of the printed layer is divided into small sections, square cells, each of which is converted into a feature vector, represented by a nine-channel histogram of oriented gradients, ranging from 0° to 180° with 20-degree intervals. The feature vectors are then combined into normalized 2×2 blocks (sliding window) in such a way that each feature vector simultaneously contributes to several adjacent blocks, which increases the robustness of texture analysis. The same procedure is carried out for the reference synthetic image, after that the obtained histograms are compared. After comparing a 2×2 block of cells, the sliding window is shifted by the size of one cell, and a similar comparison is repeated. The sliding window moves from left to right and top to bottom until the entire image is covered. This method assumes that possible alignment deviations are negligible and can be ignored. The result of this comparison is expressed in the form of a similarity map, where each section of the original camera image is assigned a numerical value of the degree of proximity to the "ideal" printing process. These values are then color coded to indicate ideal and non-ideal 3-D printing.

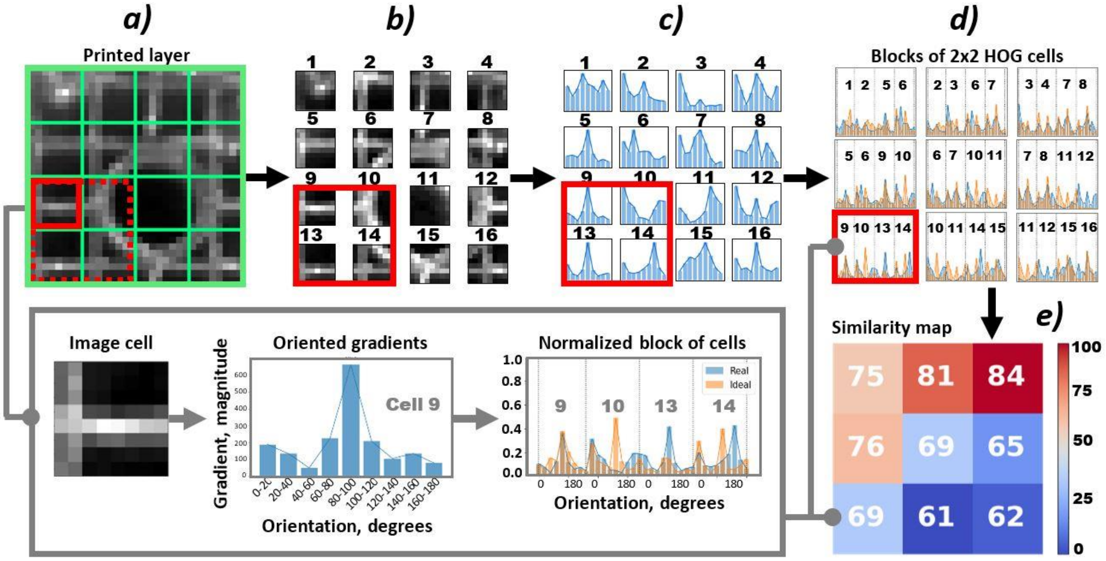


Figure 3.11: Stages of comparative image analysis: *a)* – splitting the original camera image into local areas, *b)* – separate regions of the source image, *c)* – converting image areas to feature vectors, *d)* – comparison of normalized feature vectors of the original and reference images, *e)* – resulting similarity map.

Thus, every unit section of the real image stores unique information from the corresponding area of the printed layer as a collection of features $\boldsymbol{p} = [p_1, p_2, \dots, p_k]$. A single printed layer, in this way, can be represented as a tensor $\boldsymbol{F}$ with dimensions $\boldsymbol{N} \times \boldsymbol{M} \times \boldsymbol{k}$, where $\boldsymbol{N} \times \boldsymbol{M}$ is the total number of image blocks, and $\boldsymbol{k}$ is the number of orientation channels of adjacent block cells. The tensor $\boldsymbol{F}$ is then transformed into an $\boldsymbol{N} \times \boldsymbol{M}$ similarity matrix $\boldsymbol{D}$, each element d_{ij} of which represents a similarity function $f(\boldsymbol{p}, \boldsymbol{q})$, where $\boldsymbol{q} = [q_1, q_2, \dots, q_k]$ is a collection of features of the corresponding reference image block.

Considering the camera parameters and the size of the working area, the scale of the captured images is 6.67 pixels per millimeter. The size of the minimum area of similarity analysis (2×2 block of 8-pixel cells) is therefore 4.8×4.8 mm, which lies in the range of 5–10% of the entire area of observation. This parameter can potentially be improved by using a high-definition camera.

Comparison of histograms is done by determining the similarity $s(\boldsymbol{p}, \boldsymbol{q})$ between the corresponding vectors, $\boldsymbol{p}$ and $\boldsymbol{q}$. A number of similarity metrics were selected based on the comprehensive survey on similarity measures [71], a high-throughput X-ray diffraction pattern analysis [72], and in-depth quantitative analysis in the context of two real problems of image comparison and pattern location [73].

Hernandez–Rivera et al. [72] utilized a set of 49 similarity metrics to analyze and quantify similarities between different Gaussian-based peak responses, as a surrogate for different characteristics in X-ray diffraction patterns. Research has not found universal metrics for all of vector features. It was also found that the behavior of the metric response is not uniform for members of a given similarity family. It was determined, however, that the Clark metric yields a good balance between sensitivity and smooth changes.

Goshtasby et al. [73] found that Pearson correlation coefficient, Spearman's rho, Kendall's tau, Jaccard measure, L_1 norm, and squared L_2 norm overall perform better than other measures. Cosine similarity is also widely used in conjunction with HOG features in various pattern recognition tasks [70,74,75].

In this work, the twelve metrics shown in Table 3.2 were implemented.

Table 3.2: Similarity and distance measures.

Metric	Equation	Initial output range	Normalized output range
Cosine similarity	$s(\boldsymbol{p}, \boldsymbol{q}) = \frac{\sum p_i q_i}{\sqrt{\sum p_i^2 \sum q_i^2}}$	[1, 0]	[1, 0]
Squared L$_2$ norm	$d(\boldsymbol{p}, \boldsymbol{q}) = \sum (p_i - q_i)^2$	[0, 2]	[1, 0]
Pearson's r	$r(\boldsymbol{p}, \boldsymbol{q}) = \frac{\sum(p_i - \mu_p)(q_i - \mu_q)}{\sqrt{\sum(p_i - \mu_p)^2} \cdot \sqrt{\sum(q_i - \mu_q)^2}}$ Where $\mu_p = \frac{1}{n}\sum_{i=1}^{n} p_i$ and $\mu_q = \frac{1}{n}\sum_{i=1}^{n} q_i$	[-1, 1]	[1, 0]
Spearman's rho	$\rho(\boldsymbol{p}, \boldsymbol{q}) = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$ Where d_i is the difference between the two ranks of each observation, n is the number of vector elements	[-1, 1]	[1, 0]
Kendall's tau	$\tau(\boldsymbol{p}, \boldsymbol{q}) = \frac{N_c - N_d}{n(n-1)/2}$ Where N_c and N_d are the numbers of concordant and discordant pairs of vector elements, respectively	[-1, 1]	[1, 0]
Jaccard	$s(\boldsymbol{p}, \boldsymbol{q}) = \frac{\sum p_i q_i}{\sqrt{\sum(p_i^2 + q_i^2 - p_i q_i)}}$	[1, 0]	[1, 0]

Dice	$s(\boldsymbol{p},\boldsymbol{q}) = \frac{2 \cdot \sum p_i q_i}{\sqrt{\sum(p_i^2 + q_i^2)}}$	[1, 0]	[1, 0]
L_1 norm	$d(\boldsymbol{p},\boldsymbol{q}) = \sum \lvert p_i - q_i \rvert$	[0, 2]	[1, 0]
Euclidean distance	$d(\boldsymbol{p},\boldsymbol{q}) = \sqrt{\sum (p_i - q_i)^2}$	$[0, \sqrt{2}]$	[1, 0]
Hellinger distance	$d(\boldsymbol{p},\boldsymbol{q}) = \sqrt{2 \sum (p_i - q_i)^2}$	[0, 2]	[1, 0]
Sorensen distance	$d(\boldsymbol{p},\boldsymbol{q}) = \frac{\sum \lvert p_i - q_i \rvert}{\sum (p_i + q_i)}$	[0, 1]	[1, 0]
Clark distance	$d(\boldsymbol{p},\boldsymbol{q}) = \sqrt{\sum \left(\left\lvert \frac{\lvert p_i - q_i \rvert}{p_i + q_i} \right\rvert^2 \right)}$	$[0, \sqrt{2}]$	[1, 0]

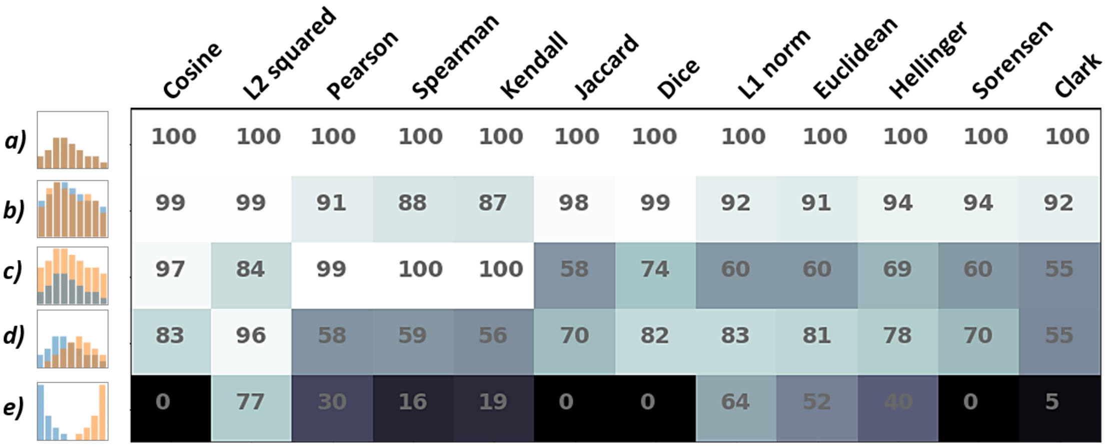
Similarity metrics $s(\boldsymbol{p},\boldsymbol{q})$ are expected to satisfy the following properties [73,76]: limited range, $s \leq s_0$; reflexivity, $s(\boldsymbol{p},\boldsymbol{q}) = s_0$ if $\boldsymbol{p} = \boldsymbol{q}$; symmetry, $s(\boldsymbol{p},\boldsymbol{q}) = s(\boldsymbol{q},\boldsymbol{p})$, triangle inequality, $s(\boldsymbol{p},\boldsymbol{q}) \leq s(\boldsymbol{p},\boldsymbol{z}) + s(\boldsymbol{q},\boldsymbol{z})$, where s_0 is the largest measure between all of possible vector inputs.

The input parameters $\boldsymbol{p}$ and $\boldsymbol{q}$ of the metrics shown in Table 3.2 are normalized vectors. The output range of similarity values is reduced to [1, 0], where “1” means the identity (overlap) of the input vectors, and “0” means a complete mismatch (no overlap). If a measure represents a distance between two inputs, $d(\boldsymbol{p},\boldsymbol{q})$, then the corresponding similarity is determined according to the following equation (3-4) [72]:

$$s(\boldsymbol{p},\boldsymbol{q}) = 1 - \frac{d(\boldsymbol{p},\boldsymbol{q})}{d^{max}(\boldsymbol{p},\boldsymbol{q})} \tag{3-4}$$

Where $d^{max}(\boldsymbol{p},\boldsymbol{q})$ is the absolute maximum possible distance between two input vectors for a particular measure. This method allows comparing all of the metrics, reducing them to the same scale.

To evaluate the given metrics, the authors created five test cases in the form of 9-bin histogram pairs, reflecting the possible deviations of printed layer sections relative to the reference ones. Figure 3.12 shows the degrees of similarity of the test inputs expressed as a percentage of coincidence. To achieve maximum efficiency, it is necessary to obtain high similarity values for the first three cases, *a), b),* and *c),* and low similarity values for the latter cases, *d)* and *e).* This allows visualizing the efficacy of various similarity measures for specific analytical cases but does not reveal the capabilities of the metrics as applied to real-world problems. From Figure 3.12, however, it can be concluded that L_1 and L_2^2 norms, as well as Euclidean and Hellinger distances, produce results that are far from expected.



	Cosine	L2 squared	Pearson	Spearman	Kendall	Jaccard	Dice	L1 norm	Euclidean	Hellinger	Sorensen	Clark
a)	100	100	100	100	100	100	100	100	100	100	100	100
b)	99	99	91	88	87	98	99	92	91	94	94	92
c)	97	84	99	100	100	58	74	60	60	69	60	55
d)	83	96	58	59	56	70	82	83	81	78	70	55
e)	0	77	30	16	19	0	0	64	52	40	0	5

Figure 3.12: Normalized similarity measures, expressed as a percentage of coincidence, for the following cases: *a)* – complete match, *b)* – similar histograms with small deviations, *c)* – similar histograms with differences in levels (represents alike image areas with varying illumination parameters), *d)* – histograms with significant shifts, *e)* – non-overlapping histograms.

After the initial assessment of the effectiveness of the selected metrics, Pearson's r, Spearman's rho, Kendall's tau, as well as cosine, Jaccard, Dice, and Sorensen similarities were chosen to test the method for detecting printing errors.

3.4 Results

3.4.1 Test Print Modes for Selecting Optimal Similarity Measures

To select the optimal similarity metrics, real-life test images were used, reflecting the typical problems of 3-D printing. Figure 3.13 shows the selected test images of erroneous layers for analyzing print mode abnormalities including a) local infill defects, b) presence of a foreign body in the layer, c) spaghetti problem, d) separation and shift of the printing part from the working surface, e) defects in thin walls, and f) layer shift. A regular printed layer is provided for each failed case, which allows comparing the outputs for various printing regimes and calculating the discriminative power for the selected metrics. In addition to different types of defects, the selected parts have geometries of varying degrees of complexity.

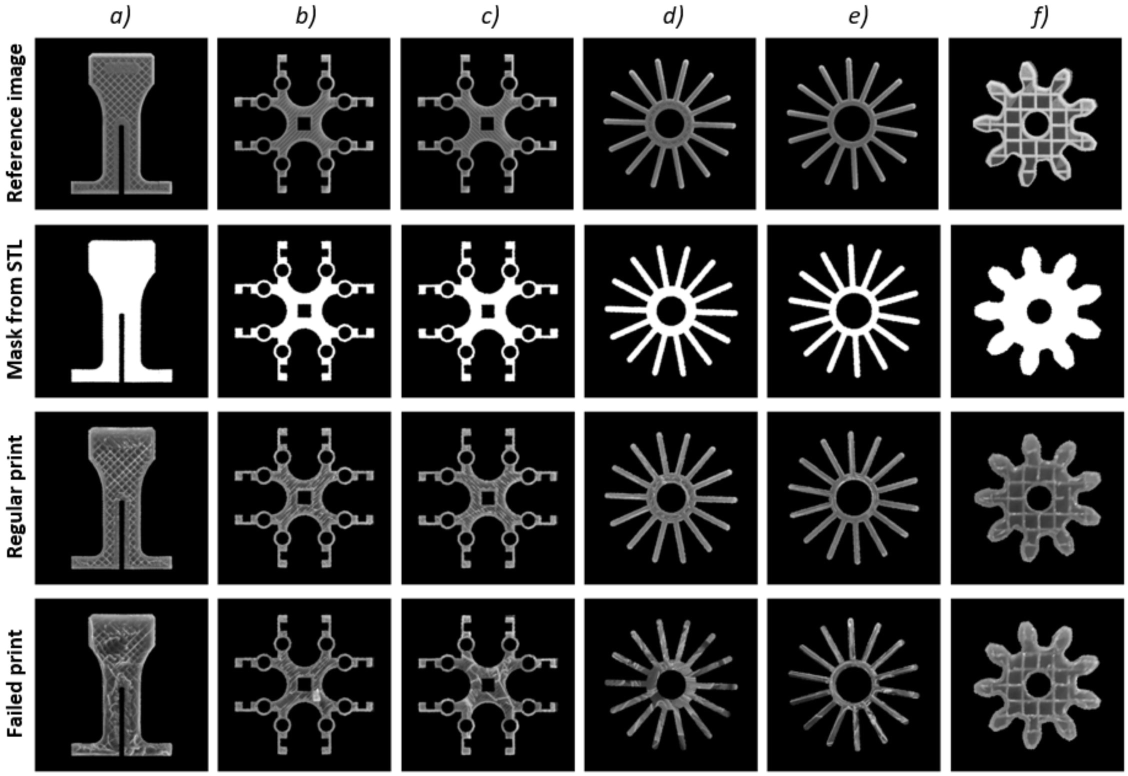


Figure 3.13: Test images of regular and failed printed layers: *a)* – local infill defects, *b)* –presence of a foreign body in the layer, *c)* – spaghetti problem, *d)* – separation and shift of the printing part from the working surface, *e)* – defects in thin walls, *f)* – layer shift.

In addition to continuous similarity for each local area of the image, an experimental failure threshold was also introduced. This is the main criterion for manufacturing defects, which allows varying sensitivity of the selected metrics and unequivocally segmenting the erroneous regions within the printed layers. Thus, an image area is considered defective if its match with the corresponding reference image is less than the chosen threshold.

3.4.2 Comparing HOG-based Similarity Measures

Figure 3.14 depicts the similarities of test images in the form of heatmaps within the unit range. Each of the metrics can be used to analyze additive manufacturing processes. As can be seen from the heatmaps, Kendall's tau, Jaccard, and Sorensen similarities are more sensitive to minor deviations in texture (have noticeably more green and blue areas on the heatmaps), in contrast to Pearson's r, Spearman's rho, cosine, and Dice similarities.

To determine the discriminative power of the selected metrics (Figure 3.15), an arbitrary 70% failure threshold (T_S) was applied to the calculated heatmaps (*H*). A layer region is considered normal if its similarity index is greater than or equal to T_S. Thus, the overall ratio of anomalous areas ($a_{\%}$) for each layer is calculated using the following expression (3-5):

$$a_{\%} = \frac{(H \leq T_S)}{S_P} \cdot 100\% \tag{3-5}$$

where S_P is the area of the entire printed layer, H is the layer similarity, and T_S is the 70% failure threshold.

The bottom portions of each bar in Figure 3.15 represent the similarity for each metric between the real image and the synthetic reference during normal printing, while the top portions of the bars represent the similarities during the introduced failures. It is difficult to print an ideal layer in real conditions, and almost every regular printing stage will deviate from a similar virtual one, which represents the ideal printing process. This is the reason for the high rates of defective areas for regular printing modes. Discriminatory power is a characteristic of each metric that allows one to distinguish between different states of objects during their production. Metrics with high discriminatory power (longer bars in

Figure 3.15) allow more accurate classification of the defect levels when fabricating layers of a part.

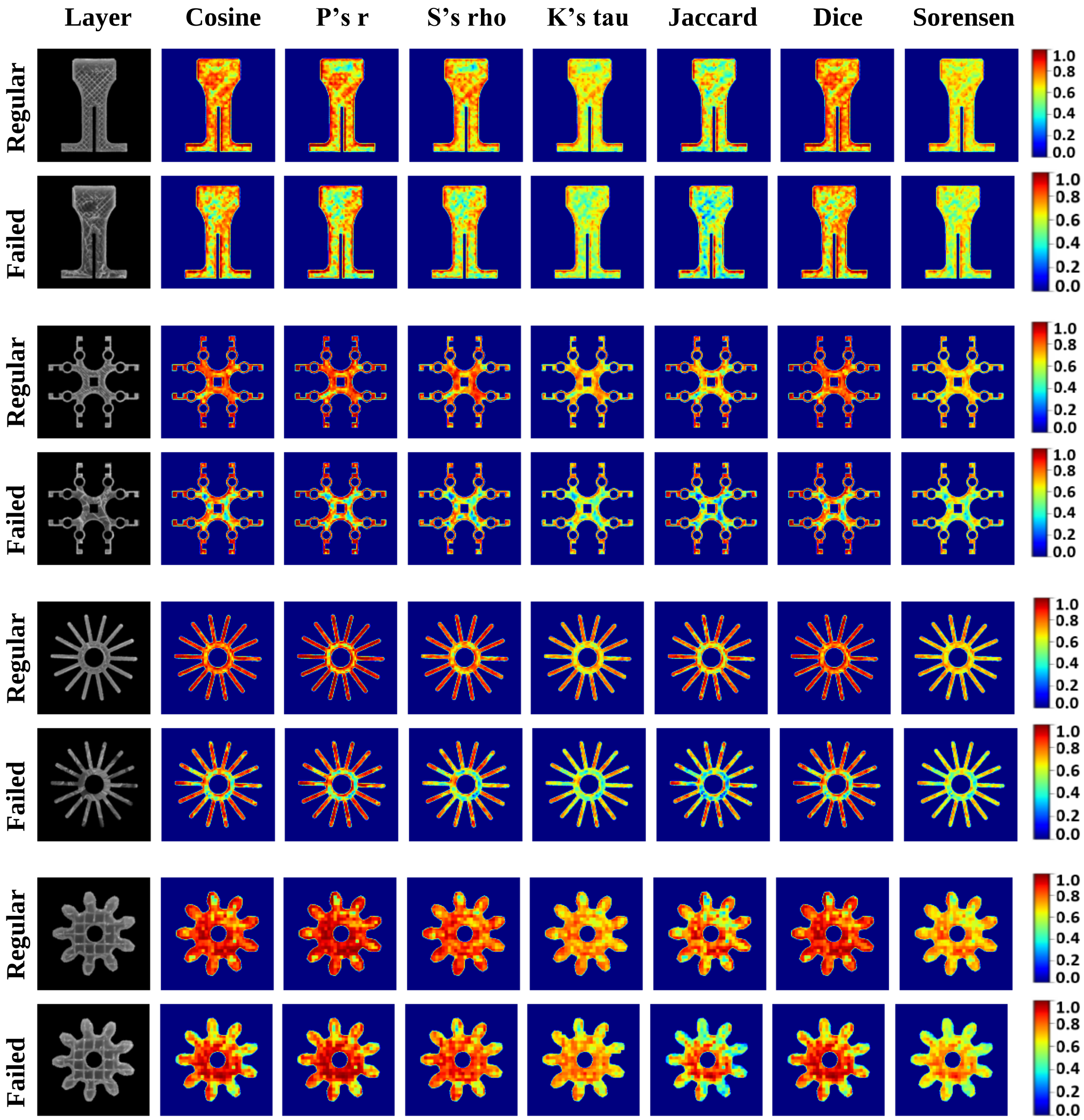


Figure 3.14: Heat maps of the regular and failed prints for the example components. Defective areas of the image have a lower similarity value (green and blue colors), while areas with high similarity values (yellow, orange, and red colors) are most consistent with the “ideal” synthetic prints (reference synthetic images).

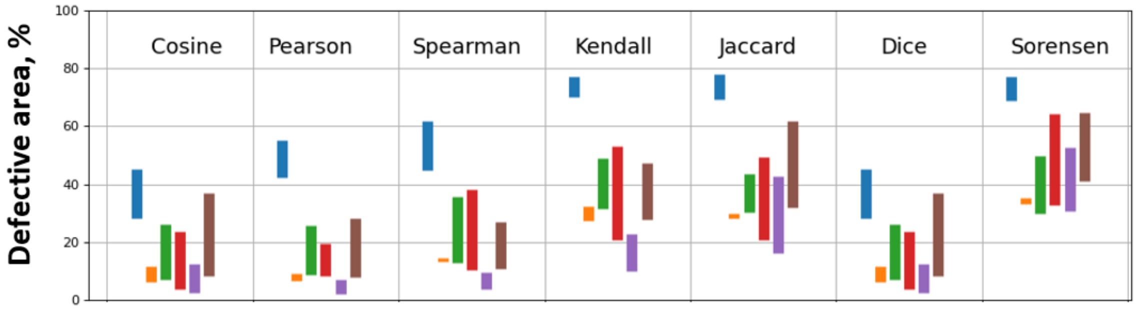


Figure 3.15: Discriminative power of the selected metrics after applying the 70% failure threshold. Vertical color bars represent the difference between the regular (bottoms) and failed (tops) prints: local infill defects (blue), presence of a foreign body in the layer (orange), spaghetti problem (green), separation and shift of the printing part from the working surface (red), defects in thin walls (purple), layer shift (brown).

Figure 3.16 illustrates the same area of the printed part with different print modes and overlaid similarity masks.

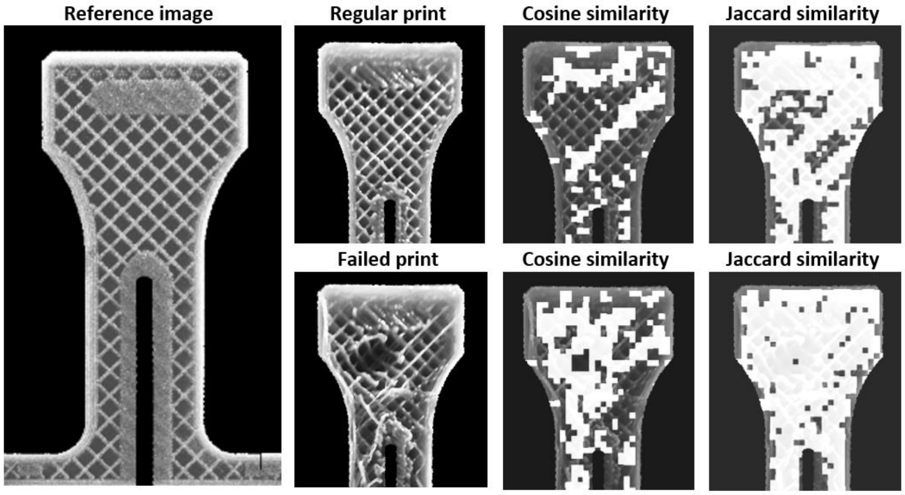


Figure 3.16: Areas of a printed part with different print modes and overlaid similarity masks. Jaccard-based failure masks cover most of both prints, considering them as failed due to high sensitivity to minor texture deviations, while cosine-based masks allow to distinguish two different print modes by the size of the covered areas.

In this example, Jaccard-based failure masks cover most of both prints, considering them as failed due to high sensitivity to minor texture deviations, while cosine-based masks distinguish two different print modes by the size of the covered areas. Introducing an experimental sensitivity threshold individually for each printing task can improve the performance of this method. This feature is especially relevant in the serial production of parts with similar geometry.

Figure 3.17 shows an example of failure detection and segmentation for the case of cosine similarity.

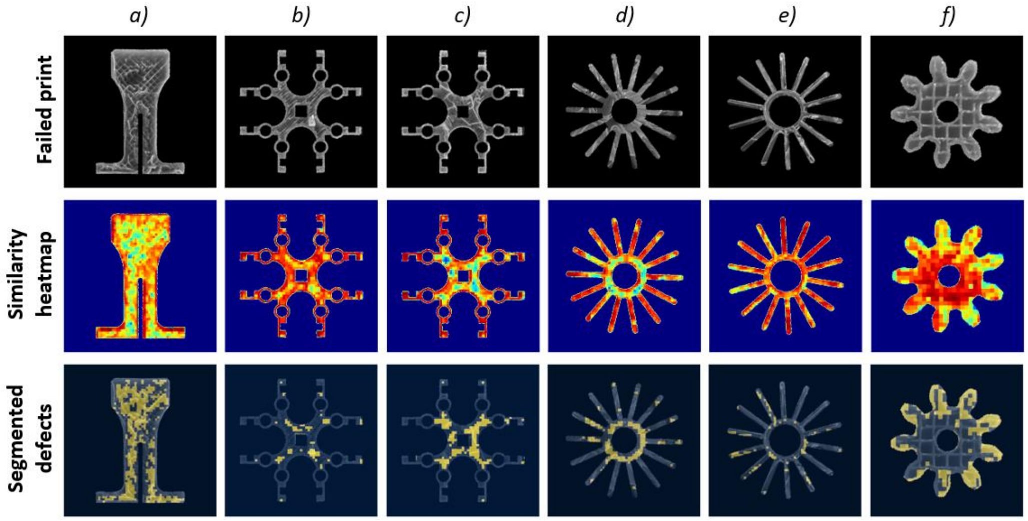


Figure 3.17: Example of failure detection based on HOG features and cosine similarity: *a)* – local infill defects, *b)* – presence of a foreign body in the layer, *c)* – spaghetti problem, *d)* – separation and shift of the printing part from the working surface, *e)* – defects in thin walls, *f)* – layer shift.

As can be seen from Figure 3.17, the blue and green shades on the heatmaps reflect defective areas discovered when comparing real images with synthetic renders. When the entire layer is displaced (Figure 3.17, f), defects appear mainly along the contour of the part. In this case, this is due to the grid infill pattern used, where the pattern inside the layer are subject to less visual distortion compared to areas along the contour.

3.5 Discussion

The proposed method makes it possible to analyze 3-D printed parts for each layer and segment of anomalous regions with a size of 5 mm or more. The main limiting factor for the developed technique is the time-consuming preliminary rendering. Generating a single image with optimized render settings can take up to several minutes on an Intel Core i7

2.60 GHz system with a dedicated NVIDIA GeForce RTX 2060 GPU. The large size and complex geometric shape of the part increase visualization time. Since the image acquisition and analysis take less than 5 seconds, most of the time spent on *in situ* monitoring can be attributed to the pause and nozzle retraction during the image capturing, which can consume more than 20 seconds of time.

Given the variety of possible printed parts and 3-D printing settings, it is difficult to create a generalized algorithm that can uniquely identify defective regions, so the choice of metric and similarity threshold used depends on the user.

There are several previously used approaches to analyze additive manufacturing processes based on information from 3-D models (G-code, CAD, or STL) and reference images. Johnson et al. [31] proposed a method using an STL file to generate a binary cross-sectional image of a part corresponding to the layer being inspected. Malik et al. [40] implemented a method for real-time layer-by-layer monitoring of AM processes using G-code-based masking, 3-D reconstruction, and augmented reality. Lyngby et al. [77] introduced a real-time vision system for non-nominal AM operation detection employing references from CAD models and color-based image segmentation. Wasserfall et al. [34] proposed a dual camera setup for in-situ layer-wise verification of 3-D printed electronics based on known G-code geometry. Hurd et al. [32] presented an approach using a phone camera to validate layers quality by comparing AM process images with 2-D images obtained from an STL file. Nuchitprasitchai et al. [30] developed single- and double-camera systems for detecting 5% deviations at every 50th layer utilizing shape and size data from STL files. Delli and Chang [78] proposed a binary 3-D printing error classification technique based on supervised machine learning, where the quality check is being performed at critical printing stages based on available images of an ideal printing process. The methods listed above, however, do not employ a physics-based rendering engine and, therefore, do not allow using the rich set of image processing techniques. This prevents using all of the information provided by the G-code and limits the process of comparing the physical creation of an object with its original model.

The presented method is not limited to the use only in the field of 3-D printing. This technique can be applied to compare parts produced by subtractive manufacturing and to reject defective printed circuit boards. The considered rendering system can be suitable for non-expert users to select parameters such as percentage and type of infill, wall thickness, and color of the material at the preprinting stage. Another application is 3-D printing in a humanitarian context [79-83] where the work environment has unreliable power supplies or air quality issues such as pollution/debris in the processing environment. This also includes distributed recycling for additive manufacturing (DRAM) [7-10] with non-uniform and heavily contaminated filament. In addition, these methods can also be applied to pellet extrusion 3-D printing [84-88] or fused particle fabrication (FPF) [89-93], where particles are direct 3-D printed (sometimes as simply shredded waste), as such conditions are more likely to cause critical errors.

The described technique can be applied to FFF 3-D printers of all of sizes that can be imaged. The additional movable lighting platform used in this work is not necessary equipment since the fundamental factor is the coincidence of physical and virtual printing systems, regardless of lighting conditions. The original 3-D model of the physical equipment can be easily customized to match the actual printing conditions for any type of 3-D printer. In the future, the failure detection method can be improved by integrating a physics-based rendering system into the printing pipeline to enable an intelligent monitored manufacturing process capable of correction anomalies from digital designs.

To assess the limits of the possibilities, however, additional tests are needed to verify the operation of the method under industrial conditions, as well as under various printing modes and materials. To fully quantify the considered metrics, the Intersection over Union *(IoU = Area of overlap / Area of the union)* index must be introduced, which, in turn, requires ground-truth hand-labeled masks. Unlike other segmentation tasks, creating a valid ground-truth failure mask for a 3-D printed layer is not a trivial process, as multiple human operators developing pixel-precise ground-truth failure masks may have different results based on their subjective perspectives. Thus, a thorough analysis of quantification approaches for a given task may constitute a separate study.

3.6 Conclusions

This study describes the conceptual capabilities of monocular layer-wise *in situ* monitoring and analysis of additive manufacturing processes using projective transformations and image processing techniques. The developed method can be applied to material extrusion 3-D printers of any size with a resolution of detecting anomalous regions of 5–10% of the overall observation area. The minimum area of similarity analysis can potentially be decreased by using a high-definition camera.

The results show that HOG-based similarity comparison does not introduce significant delays in the monitoring process. The processing time is a matter of seconds, which is negligible compared to the time required to fabricate a layer. The described method, however, requires time resources when preparing the virtual environment and rendering the reference images. Using different similarity metrics and failure thresholds provides flexibility and allows for varying sensitivity in printing anomalies segmentation. Of the twelve similarity and distance measures implemented and compared for their effectiveness at detecting 3-D printing errors, the results show that although Kendall’s tau, Jaccard, and Sorensen similarities are the most sensitive, Pearson's r, Spearman's rho, cosine, and Dice similarities produce the more distinguishable results. Thus, combining image cell size for HOG comparison and multi-stage application of various metrics can provide effective tools for more accurate detection and localization of production anomalies. The greatest efficiency of the given technique can be achieved with the mass production of parts of the same geometric shape as the open source Blender rendering only needs to occur once. Although this technique was tested here for additive manufacturing, it can be applied to compare parts produced by subtractive manufacturing and printed circuit boards.

3.7 Bibliography

[1] E. Sells, S. Bailard, Z. Smith, A. Bowyer, V. Olliver, RepRap: the replicating rapid prototype – maximizing customizability by breeding the means of production, in: F.T. Piller, M.M. Tseng (Eds.), Handbook of Research in Mass Customization and Personalization, 1 World Scientific, Singapore, 2009, pp. 568–580.

[2] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, A. Bowyer, RepRap – the replicating rapid prototype, Robotica 29 (1) (2011) 177–191, https://doi.org/10.1017/S026357471000069X.

[3] A. Bowyer, 3D printing and humanity's first imperfect replicator, 3D Print. Addit. Manuf. 1 (1) (2014) 4–5, https://doi.org/10.1089/3dp.2013.0003.

[4] T.D. Ngo, A. Kashani, G. Imbalzano, K.T.Q. Nguyen, D. Hui, Additive manufacturing (3D printing): A review of materials, methods, applications and challenges, Composites Part B: Engineering, 143 (2018), 172-196. https://doi.org/10.1016/j.compositesb.2018.02.012.

[5] N. Mohan, P. Senthil, S. Vinodh & N. Jayanth, A review on composite materials and process parameters optimisation for the fused deposition modelling process, Virtual and Physical Prototyping, 12:1 (2017) 47-59. https://doi.org/10.1080/17452759.2016.1274490.

[6] R. Anandkumar, S.R. Babu, FDM filaments with unique segmentation since evolution: a critical review. Prog Addit Manuf 4 (2019) 185–193. https://doi.org/10.1007/s40964-018-0069-8.

[7] C. Baechler, M. DeVuono, and J.M. Pearce, Distributed recycling of waste polymer into RepRap feedstock, Rapid Prototyping Journal, 19:2 (2013) 118-125. https://doi.org/10.1108/13552541311302978.

[8] F.A.C. Sanchez, H. Boudaoud, M. Camargo, J.M. Pearce, Plastic recycling in additive manufacturing: A systematic literature review and opportunities for the circular economy. Journal of Cleaner Production, 264 (2020), 121602.

[9] F.A.C. Sanchez, H. Boudaoud, S. Hoppe, M. Camargo, Polymer recycling in an open-source additive manufacturing context: Mechanical issues. Additive Manufacturing, 17 (2017) 87-105.

[10] S.C. Dertinger, N. Gallup, N.G. Tanikella, M. Grasso, S. Vahid, P.J. Foot, J.M. Pearce, Technical pathways for distributed recycling of polymer composites for distributed manufacturing: Windshield wiper blades. Resources, Conservation and Recycling, 157 (2020), 104810

[11] B.T. Wittbrodt, A.G. Glover, J. Laureto, G.C. Anzalone, D. Oppliger, J.L. Irwin, J.M. Pearce, Life-cycle economic analysis of distributed manufacturing with open-source 3-D printers. Mechatronics, 23:6 (2013) 713-726.

[12] E.E. Petersen, J.M. Pearce, Emergence of home manufacturing in the developed world: Return on investment for open-source 3-D printers. Technologies, 5:1 (2017), 7.

[13] M. Kreiger, J.M. Pearce, Environmental life cycle analysis of distributed three-dimensional printing and conventional manufacturing of polymer products. ACS Sustainable Chemistry & Engineering, 1:12 (2013) 1511-1519.

[14] J. Go, S.N. Schiffres, A.G. Stevens, A.J. Hart, Rate limits of additive manufacturing by fused filament fabrication and guidelines for high-throughput system design. Additive Manufacturing, 16 (2017) 1-11.

[15] H. Wu, Z. Yu, Y. Wang, Experimental study of the process failure diagnosis in additive manufacturing based on acoustic emission. Measurement, 136 (2019) 445-453.

[16] A. Oleff, B. Küster, M. Stonis, L. Overmeyer, Process monitoring for material extrusion additive manufacturing: a state-of-the-art review. Prog Addit Manuf (2021). https://doi.org/10.1007/s40964-021-00192-4.

[17] T. Wohlers, I. Campbell, O. Diegel et al. (2018) Wohlers Report 2018. 3D printing and additive manufacturing state of the industry: Annual Worldwide Progress Report. Wohlers Associates Inc, Fort Collins.

[18] M. Borish, B.K. Post, A. Roschli et al., In-situ thermal imaging for single layer build time alteration in large-scale polymer additive manufacturing. Procedia Manuf 34 (2019) 482–488. https://doi.org/10.1016/j.promfg.2019.06.202.

[19] E. Ferraris, J. Zhang, B. van Hooreweder, Thermography based in-process monitoring of Fused Filament Fabrication of polymeric parts. CIRP Ann Manuf Technol 68 (2019) 213–216. https://doi.org/10.1016/j.cirp.2019.04.123.

[20] Y. Tlegenov, G.S. Hong, W.F. Lu, Nozzle condition monitoring in 3D printing. Robot Comput Integr Manuf 54 (2018) 45–55. https://doi.org/10.1016/j.rcim.2018.05.010.

[21] Y. Li, W. Zhao, Q. Li et al., In-situ monitoring and diagnosing for fused filament fabrication process based on vibration sensors. Sensors (2019). https://doi.org/10.3390/s19112589.

[22] P. Becker, C. Roth, A. Roennau et al., Acoustic Anomaly Detection in Additive Manufacturing with Long Short-Term Memory Neural Networks. IEEE 7th Int Conf Ind Eng Appl (ICIEA) (2020) 921–926. https://doi.org/10.1109/ICIEA49774.2020.9102002.

[23] F. Li, Z. Yu, Z. Yang et al., Real-time distortion monitoring during fused deposition modeling via acoustic emission. Struct Health Monit 19 (2019) 412–423. https://doi.org/10.1177/1475921719849700.

[24] D.D.L. Chung, S. Somaratna, Laboratory simulation of capacitance-based layer-by-layer monitoring of three-dimensional printing. Sens Actuators A Phys 268 (2017) 101–109. https://doi.org/10.1016/j.sna.2017.10.061.

[25] Y. Tlegenov Y, W.F. Lu, G.S. Hong, A dynamic model for current-based nozzle condition monitoring in fused deposition modelling. Prog Addit Manuf 4 (2019) 211–223. https://doi.org/10.1007/s40964-019-00089-3.

[26] R.K. Rao, J. Liu, D. Roberson et al., Online Real-Time Quality Monitoring in Additive Manufacturing Processes Using Heterogeneous Sensors. J Manuf Sci Eng (2015). https://doi.org/10.1115/1.4029823.

[27] E.C. Koskelo, E.B. Flynn, P.J. Shull, A.L. Gyekenyesi, T. Yu, H.F. Wu, Scanning laser ultrasound and wavenumber spectroscopy for in-process inspection of additively manufactured parts. Proc SPIE Nondestruct Charact Monit Adv Mater Aerosp Civ Infrastruct (2016). https://doi.org/10.1117/12.2222130.

[28] T.J. Coogan, D.O. Kazmer, In-line rheological monitoring of fused deposition modeling. J Rheol 63 (2019) 141–155. https://doi.org/10.1122/1.5054648.

[29] J. Li, H. Xie, K. Ma, In-situ monitoring of the deformation during Fused Deposition Modeling process using CGS method. Polym Test 76 (2019) 166–172. https://doi.org/10.1016/j.polymertesting.2019.03.030.

[30] S. Nuchitprasitchai, M.C. Roggemann, J.M. Pearce, Factors effecting real-time optical monitoring of fused filament 3D printing, Prog. Addit. Manuf. J. 2 (3) (2017) 133–149, https://doi.org/10.1007/s40964-017-0027-x 2017.

[31] A. Johnson, H. Zarezadeh, X. Han, R. Bibb, R. Harris, Establishing in-process inspection requirements for material extrusion additive manufacturing. Loughborough University. Conference contribution (2016). https://hdl.handle.net/2134/20696.

[32] S. Hurd, C. Camp, J. White, Quality assurance in additive manufacturing through mobile computing. Int Conf Mob Comput Appl Serv (2015) 203–220. https://doi.org/10.1007/978-3-319-29003-4_12.

[33] H. Jeong, M. Kim, B. Park et al., Vision-Based Real-Time Layer Error Quantification for Additive Manufacturing. Proc ASME 2017 12th Int Manuf Sci Eng Conf. (2017). https://doi.org/10.1115/MSEC2017-2991.

[34] F. Wasserfall, D. Ahlers, N. Hendrich, Optical in-situ verification of 3D-printed electronic circuits. IEEE 15th Int Conf Autom Sci and Eng (CASE) (2019). https://doi.org/10.1109/COASE.2019.8842835.

[35] A. Ceruti, A. Liverani, T. Bombardi, Augmented vision and interactive monitoring in 3D printing process. Int J Interact Des Manuf 11 (2017) 385–395. https://doi.org/10.1007/s12008-016-0347-y.

[36] J. Straub, 3D printing cybersecurity: Detecting and preventing attacks that seek to weaken a printed object by changing fill level. Proc SPIE Dimens Opt Metrol Insp Pract Appl VI (2017). https://doi.org/10.1117/12.2264575.

[37] M.D. Kutzer, L.D. DeVries, C.D. Blas, Part monitoring and quality assessment of conformal additive manufacturing using image reconstruction. Proc ASME 2018 Int Des Eng Tech Conf Comput Inf Eng Conf 5B (2018). https://doi.org/10.1115/DETC2018-85370

[38] Z. Chen, R. Horowitz, Vision-assisted arm motion planning for freeform 3D Printing. 2019 Am Control Conf (ACC) (2019) 4204–4209. https://doi.org/10.23919/ACC.2019.8814699.

[39] H. Shen, W. Du, W. Sun et al., Visual detection of surface defects based on self-feature comparison in robot 3-D printing. Appl Sci. (2020). https://doi.org/10.3390/app10010235.

[40] A. Malik, H. Lhachemi, J. Ploennigs et al., An application of 3D model reconstruction and augmented reality for real-time monitoring of additive manufacturing. Procedia CIRP 81 (2019) 346–351. https://doi.org/10.1016/j.procir.2019.03.060.

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

[42] G.C. Anzalone, B. Wijnen, J.M. Pearce, Multi-material additive and subtractive prosumer digital fabrication with a free and open-source convertible delta RepRap 3-D printer, Rapid Prototyp. J. 21 (5) (2015) 506–519, https://doi.org/10.1108/RPJ-09-2014-0113.

[43] RAMPS 1.4. https://reprap.org/wiki/RAMPS_1.4, 2021 (accessed 03 November 2021).

[44] SONY IMX322 Datasheet. https://dashcamtalk.com/cams/lk-7950-wd/Sony_IMX322.pdf, 2021 (accessed 03 November 2021),

[45] Blender, free and open source 3D creation suite. https://www.blender.org/, 2021 (accessed 03 November 2021).

[46] B.R. Kent. Visualizing Astronomical Data with Blender. PASP 125:928 (2013) 731. https://iopscience.iop.org/article/10.1086/671412/meta.

[47] M. Gschwandtner, R. Kwitt, A. Uhl, W. Pree, BlenSor: Blender Sensor Simulation Toolbox. In: Bebis G. et al. (eds) Advances in Visual Computing. ISVC 2011. Lecture Notes in Computer Science, 6939 (2011). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24031-7_20.

[48] R.F. da Costa, D.S. de Medeiros, R. Andrade, O. Saotome, and R. Machado. General Purpose Radar Simulator based on Blender Cycles Path Tracer. (2020). DOI: 10.14209/SBRT.2020.1570649487.

[49] S. Flaischlen, G.D. Wehinger, Synthetic Packed-Bed Generation for CFD Simulations: Blender vs. STAR-CCM+. ChemEngineering, 3(2) (2019) 52. https://doi.org/10.3390/chemengineering3020052.

[50] M. Ilba, Estimating the daily solar irradiation on building roofs and facades using Blender Cycles path tracing algorithm. E3S Web of Conferences 10, 00027 (2016) https://doi.org/10.1051/e3sconf/20161000027.

[51] D.P. Rohe, An Optical Test Simulator Based on the Open-Source Blender Software. United States: N. p., 2019. Web. https://www.osti.gov/biblio/1642949.

[52] S. Reitmann, L. Neumann, B. Jung, BLAINDER—A Blender AI Add-On for Generation of Semantically Labeled Depth-Sensing Data. Sensors. 2021; 21(6):2144. https://doi.org/10.3390/s21062144.

[53] Blender: Procedural Textures. https://docs.blender.org/manual/en/2.79/render/blender_render/textures/types/procedural/index.html, 2021 (accessed 03 November 2021).

[54] Chris Bailey. Top 6 Blender Nodes To Make Any Material in Blender 2021. https://www.youtube.com/watch?v=yffWd4kI51Q, 2021 (accessed 03 November 2021).

[55] Blender Shader Nodes. https://docs.blender.org/manual/en/2.79/render/cycles/nodes/types/shaders/index.html, 2021 (accessed 03 November 2021).

[56] Heinz Löpmeier, Blender-Gcode-Importer. https://github.com/Heinz-Loepmeier/Blender-Gcode-Import, 2021 (accessed 03 November 2021).

[57] Jonathan Winterflood, YAGV - Yet Another Gcode Viewer, v0.4. https://github.com/jonathanwin/yagv, 2021 (accessed 03 November 2021).

[58] Alessandro Zomparelli, Gcode Exporter. https://github.com/alessandro-zomparelli/gcode-exporter, 2021 (accessed 03 November 2021).

[59] Blender for Science, Gcode Exporter. https://github.com/blender-for-science/import-G-code, 2021 (accessed 03 November 2021).

[60] Petsiuk, A., & Pearce, J. M. (2021, October 25). Layer-wise HOG-based anomaly detection in plastic FFF 3D printing. Retrieved from https://osf.io/q8ebj/, 2021 (accessed 03 November 2021).

[61] Blender Cycles, https://docs.blender.org/manual/en/latest/render/cycles/index.html, 2021 (accessed 03 November 2021).

[62] OpenCV: Camera Calibration and 3D Reconstruction. https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html, 2021 (accessed 03 November 2021).

[63] OpenCV: Geometric Image Transformations. https://docs.opencv.org/4.5.2/da/d54/group__imgproc__transform.html, 2021 (accessed 03 November 2021).

[64] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886–893 vol. 1, doi: 10.1109/CVPR.2005.177.

[65] Firuzi, M. Vakilian, B. T. Phung and T. R. Blackburn, Partial Discharges Pattern Recognition of Transformer Defect Model by LBP & HOG Features, in IEEE Transactions on Power Delivery, vol. 34, no. 2, pp. 542–550, April 2019, doi:10.1109/TPWRD.2018.2872820.

[66] K. Malik, C. Robertson, Landscape Similarity Analysis Using Texture Encoded Deep-Learning Features on Unclassified Remote Sensing Imagery. Remote Sensing, 13:3 (2021) 492. https://doi.org/10.3390/rs13030492.

[67] S. Banerji et al. Finding Better Matches: Improving Image Retrieval with EFM-HOG, International Journal on Advances in Software, 13:3-4 (2020). http://math.lakeforest.edu/banerji/research_files/IJAS.pdf,

[68] K. Akila and P. Pavithra. Optimized Scale Invariant HOG Descriptors for Object and Human Detection, IOP Conf. Ser.: Mater. Sci. Eng. (2021) 1119 012002.

[69] G. Joshi, S. Singh, R. Vig, Taguchi-TOPSIS based HOG parameter selection for complex background sign language recognition, Journal of Visual

Communication and Image Representation, 71 (2020), 102834, https://doi.org/10.1016/j.jvcir.2020.102834.

[70] J. Supeng, X. Wei, Y. Liu, Robust template matching algorithm with multi-feature using best-buddies similarity. Proceedings, v.11427 (2020), Second Target Recognition and Artificial Intelligence Summit Forum; 114271Y (2020). https://doi.org/10.1117/12.2552038.

[71] S.H. Cha, Comprehensive survey on distance/similarity measures between probability density functions, Int. J. Math. Model. Meth. Appl. Sci. (2007) 1.

[72] E. Hernandez-Rivera, S.P. Coleman, M.A. Tschopp, Using similarity metrics to quantify differences in high-throughput datasets: application to X-ray diffraction patterns, ACS Comb. Sci. 19 (1) (2016) 25–36, https://doi.org/10.1021/acscombsci.6b00142.

[73] A.A. Goshtasby, Similarity and dissimilarity measures, in: S. Singh, S.B. Kang (Eds.), Image Registration. Advances in Computer Vision and Pattern Recognition, Ch. 2, Springer, London, United Kingdom, 2012, pp. 7–66, , https://doi.org/10.1007/978-1-4471-2458-0_2.

[74] R. K. Sharma, N. Tater and R. Kumar, Genuinity Detection of People: A Comparative Analysis on HOG and One Shot Learning, 2020 6th International Conference on Signal Processing and Communication (ICSC), 2020, pp. 167–172, doi: 10.1109/ICSC48311.2020.9182738.

[75] J. Chen, Z. Guo, J. Hu, Ring-Regularized Cosine Similarity Learning for Fine-Grained Face Verification, Pattern Recognition Letters, 148 (2021) 68-74. https://doi.org/10.1016/j.patrec.2021.04.029.

[76] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th edn. Academic Press, New York (2009), pp. 602, 605, 606.

[77] R.A. Lyngby, J. Wilm, E.R. Eiríksson et al., In-line 3D print failure detection using computer vision. Joint Special Interest Group meeting between euspen and ASPE: Dimensional Accuracy and Surface Finish in Additive Manufacturing (2017). https://www.euspen.eu/knowledge-base/AM17133.pdf.

[78] U. Delli, S. Chang, Automated processes monitoring in 3D printing using supervised machine learning, Procedia Manuf. 26 (2018) 865–870, https://doi.org/10.1016/j.promfg.2018.07.111.

[79] L. Corsini, C.B. Aranda-Jan, J. Moultrie, The impact of 3D printing on the humanitarian supply chain. Production Planning & Control (2020) 1-13.

[80] O. Rodríguez-Espíndola, S. Chowdhury, A. Beltagui, and P. Albores, The potential of emergent disruptive technologies for humanitarian supply chains: the integration of blockchain, Artificial Intelligence and 3D printing. International Journal of Production Research, 58:15 (2020) 4610-4630.

[81] B.L. Savonen, T.J. Mahan, M.W. Curtis, J.W. Schreier, J.K. Gershenson, and J.M. Pearce, Development of a resilient 3-D printer for humanitarian crisis response. Technologies, 6:1 (2018) 30.

[82] J. Loy, P. Tatham, R. Healey, and C.L. Tapper, 2016. 3D Printing Meets Humanitarian Design Research: Creative Technologies in Remote Regions. In Creative Technologies for Multidisciplinary Applications (pp. 54–75). IGI Global.

[83] S. Lipsky, A. Przyjemski, M. Velasquez, and J. Gershenson, 2019, October. 3D Printing for Humanitarian Relief: The Printer Problem. In 2019 IEEE Global Humanitarian Technology Conference (GHTC) (pp. 1–7). IEEE.

[84] S. Whyman, K.M. Arif, and J. Potgieter, 2018. Design and development of an extrusion system for 3D printing biopolymer pellets. The International Journal of Advanced Manufacturing Technology, 96(9-12), pp.3417–3428.

[85] S. Singamneni, D. Smith, M.J. LeGuen, and D. Truong, Extrusion 3D printing of polybutyrate-adipate-terephthalate-polymer composites in the pellet form. Polymers, 10:8 (2018) 922.

[86] T. Cersoli, A. Cresanto, C. Herberger, E. MacDonald, and P. Cortes, 3D printed shape memory polymers produced via direct pellet extrusion. Micromachines, 12:1 (2021) 87. S. Singamneni, A. Warnakula, D.A. Smith, and M.J. Le Guen, Biopolymer alternatives in pellet form for 3D printing by extrusion. 3D Printing and Additive Manufacturing, 6:4 (2019) 217-226.

[88] D.M. Nieto, V.C. López, and S.I. Molina, Large-format polymeric pellet-based additive manufacturing for the naval industry. Additive Manufacturing, 23 (2018) 79-85.

[89] A.L. Woern, D.J. Byard, R.B. Oakley, M.J. Fiedler, S.L. Snabes, and J.M. Pearce, Fused particle fabrication 3-D printing: Recycled materials' optimization and mechanical properties. Materials, 11:8 (2018) 1413.

[90] M.J. Reich, A.L. Woern, N.G. Tanikella, and J.M. Pearce, Mechanical properties and applications of recycled polycarbonate particle material extrusion-based additive manufacturing. Materials, 12:10 (2019) 1642.

[91] A. Alexandre, F.A. Cruz Sanchez, H. Boudaoud, M. Camargo, and J.M. Pearce, Mechanical properties of direct waste printing of polylactic acid with universal pellets extruder: comparison to fused filament fabrication on open-source desktop three-dimensional printers. 3D Printing and Additive Manufacturing, 7:5 (2020) 237-247.

[92] D.J. Byard, A.L. Woern, R.B. Oakley, M.J. Fiedler, S.L. Snabes, and J.M. Pearce, Green fab lab applications of large-area waste polymer-based additive manufacturing. Additive Manufacturing, 27 (2019) 515-525.

[93] H.A. Little, N.G. Tanikella, M.J. Reich, M.J. Fiedler, S.L. Snabes, and J.M. Pearce, Towards distributed recycling with additive manufacturing of PET flake feedstocks. Materials, 13:19 (2020) 4273.

Chapter 4

4 Synthetic-to-real Composite Semantic Segmentation in Additive Manufacturing

This chapter[3] is adapted from the "Synthetic-to-Real Composite Semantic Segmentation in Additive Manufacturing" with minor modifications to the version published in Journal of Manufacturing and Materials Processing (MDPI), vol. 8, issue 2, 66, 2024, DOI:10.3390/jmmp8020066.

4.1 Abstract

The application of computer vision and machine learning methods for semantic segmentation of the structural elements of 3-D-printed products in the field of additive manufacturing (AM) can improve real-time failure analysis systems and potentially reduce the number of defects by providing additional tools for in situ corrections. This work demonstrates the possibilities of using physics-based rendering for labeled image dataset generation, as well as image-to-image style transfer capabilities to improve the accuracy of real image segmentation for AM systems. Multi-class semantic segmentation experiments were carried out based on the U-Net model and the cycle generative adversarial network. The test results demonstrated the capacity of this method to detect such structural elements of 3-D-printed parts as a top (last printed) layer, infill, shell, and support. A basis for further segmentation system enhancement by utilizing image-to-image style transfer and domain adaptation technologies was also considered. The results indicate that using style transfer as a precursor to domain adaptation can improve real 3-D printing image segmentation in situations where a model trained on synthetic data is the only tool available. The mean intersection over union (mIoU) scores for synthetic test datasets included 94.90% for the entire 3-D-printed part, 73.33% for the top layer, 78.93% for the infill, 55.31% for the shell, and 69.45% for supports.

---

[3] A version of this chapter has been published in Journal of Manufacturing and Materials Processing.
A. Petsiuk, H. Singh, H. Dadhwal, J.M. Pearce, Synthetic-to-real composite semantic segmentation in additive manufacturing. J. Manuf. Mater. Process., vol. 8(2), no. 66, 2024, doi:10.3390/jmmp8020066.

4.2 Introduction

With its current exponential growth, the amount of plastic waste produced could reach 250 billion tons by 2050 [1], vast quantities of which cause pollution of the natural environment on land and in the ocean [2]. Distributed manufacturing using additive manufacturing (AM) is reforming global value chains as its usage increases rapidly [3], because there are millions of free 3-D printable consumer product designs and 3-D printing them results in substantial cost savings compared to conventionally manufactured commercial products [4,5].

The growing popularity of 3-D printing is playing a notable role in the problem of recycling as 3-D-printed products rarely have recycling symbols [6], use uncommon polymers [7], and are increasing the overall market of plastic materials [8]. This is not only caused by additional plastic products, but also from disturbing failure rates. Inexperienced 3-D printer users are estimated to have failure rates of 20% [9]. Even experienced professionals working in 3-D print farms, however, have failure rates of at least 2% [10]. The probability of a manufacturing defect increases with the size and print time of the object (e.g., using large-scale fused filament printers [11] or products [12,13], or fused granule printers [14,15]), which can magnify the waste materials created from even a small percentage of failures. It is clear that the ability to automatically detect deviations in AM will significantly help to reduce material waste and the time spent on reproducing failed prints.

As recent studies [16] show, computer vision is becoming increasingly popular in analyzing AM and extrusion-based 3-D printing processes. For example, Ceruti et al. [17] utilized data from computer-aided design (CAD) files that are used in the first step of the design of a 3-D-printed component. Then, further down the software toolchain, Nuchitprasitchai et al. [18], Johnson et al. [19], and Hurd [20] developed failure analysis based on comparisons with the Standard Tessellation Language (STL) files used at the slicing step in most 3-D printing processes. Further still, both Jeong et al. [21] and Wasserfall et al. [22] used, instead, the G-code files that provide the 3-D printer with spatial toolpath instructions for printing parts. The 3-D printing software toolchain does not need to be used at all, as several approaches use comparisons with reference data [23,24] or ideal 3-D printing processes [25,26]. In addition, a 3-D reconstruction-based scanning method for real-time monitoring of AM processes is also possible [27]. In previous works, the authors considered the possibilities of detecting critical manufacturing errors using classical image-processing methods [28], as well as employing synthetic reference images rendered with a physics-based graphics engine [29]. The proposed methods, however, do not fully utilize the available information and are limited in determining the location-based categories of production deviations.

The popular open source Spaghetti Detective application [30,31] is also a direct confirmation of the effectiveness of visual monitoring. An analysis of Spaghetti Detective's [30] user performance database, collected over 2.3 years, showed that 24% of all of the 5.6 million print jobs were canceled, which can be represented as 456 wasted hours of continuous printing compared to 5232 h of printing where all of the print jobs were finished (Figure 4.1).

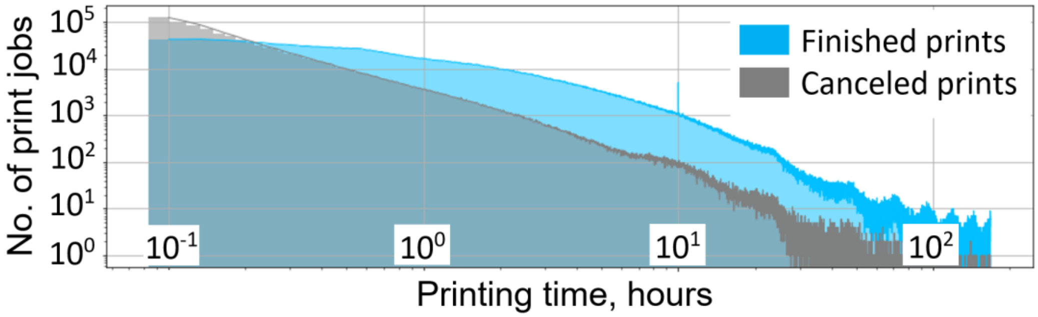


Figure 4.1: Analysis of 3-D printer users' activity for 2.3 years. The runtime distribution shows a 24% failure rate for all of the 5.6 million printing tasks longer than 5 minutes.

This statistic, however, does not include over a million canceled print jobs less than 5 min long, which are assumed to be due to initial bed-leveling issues and cannot, therefore, indicate manufacturing failures. It also does not consider the working time of human operators spent on starting later-canceled printing tasks.

Semantic segmentation [32] of both the entire manufactured part and its separate structural regions at the stage of production of each layer will expand the capacity of the visual analysis of AM processes and can make it possible to more accurately determine the nature of individual production errors depending on their localization. This can provide advanced tools for correcting printing defects in situ, where each successive layer can be modified depending on the deviations found in the previous stage, thus improving both the mechanical and aesthetic characteristics of the entire object. It may also reduce the requirements for camera positioning accuracy and calibration, eliminating the need for visual markers and rigid holders.

In the previous work [29], the authors demonstrated the ability of Blender [33], a free and open source physics rendering engine, to generate photorealistic images of ideal 3-D printing processes based on existing G-code files. This work served as a milestone in the development of a deep learning-based approach, presented in this paper, to the semantic segmentation of structural elements in 3-D printing environments.

Using a synthetic dataset, however, comes at the cost of a domain shift, which is often strongly associated with appearance changes [34]. When the source (synthetic images) and target (real images) domains are semantically related, but are different in visual representation, direct propagation of learned knowledge about one domain to another can adversely affect segmentation performance in the latter domain. Therefore, domain adaptation (DA) is needed in order to learn generalized segmentation rules in the presence of a gap between the source and target dataset distributions [34,35].

There are examples in the literature of successful synthetic-to-real (sim-to-real) DA applications. Imbusch, Schwarz, and Behnke [36] proposed an unsupervised Generative Adversarial Network (GAN)-based DA approach to a robotics environment image dataset that provides a performance close to those trained on real data and does not require annotations of real robotic setups. Li et al. [37] presented a semantically aware GAN-based neural network model for virtual-to-real urban scene adaptation with the ability to store important semantic information. Lee et al. [38] introduced a sim-to-real vehicle identification technique consisting of DA and semi-supervised learning methods.

Domain adaptation, however, is a separate area of research and is not covered in this article. The possibility of applying a cycle-consistent adversarial network (CycleGAN) [39] – an image-to-image style transfer method—was considered for segmentation improvement, as generative adversarial networks can perform a significant role in domain adaptation techniques and be used in future research.

The proposed method is a novel approach to segmenting key regions of manufactured parts during their fabrication using G-code information and synthetic data. Revealing this system to end users will allow constant expansion of the synthetic image database for subsequent neural network training. The presented contributions, therefore, can be summarized as follows:

- A technique for generating synthetic image-mask pairs of layer-by-layer ideal 3-D printing processes has been developed for subsequent neural network training;
- Three independent labeled synthetic image datasets for (a) the entire part, (b) the top (last printed) layer, and (c) the infill, shell, and supports for 3-D-printed objects have been created;
- A neural network was trained for the semantic segmentation of the entire printed part, as well as its last printed top layer and internal structure;
- Image-to-image style transfer approaches to improve segmentation results have been explored.

All of the above steps are sequentially described in this article after first reviewing related works in detail. The Results section will discuss the potential for the localization of 3-D-printed parts in the image frame, as well as the application of image processing methods to the parts' structural elements for subsequent detection of manufacturing deviations.

4.3 Background

Semantic image segmentation problems represent an actively developing area of research in deep machine learning [32,40]. The main limiting factor, however, is the difficulty of obtaining annotated databases for the training of machine learning architectures. This approach requires thousands of images with labeled masked regions, which is a difficult and time-consuming task – manual annotation of a single image with pixel-by-pixel semantic labels can take more than 1.5 h [41].

The use of synthetic images, in turn, allows the procurement of a segmented training database conditionally ”free of charge”, since masked regions of interest can be automatically annotated when creating virtual physics-based renders. In addition, advances in computer graphics make it possible to generate an almost unlimited amount of labeled data by varying environmental parameters in ranges that are difficult to obtain in real conditions [34]. The success of simulated labeled data is clearly illustrated in the already classic GTA5 [42] and Synthia [43] image sets.

There are many examples of applying synthetic datasets to solve real-world practical problems. Nikolenko [44] presented an up-to-date technological slice of the use of synthetic data in a wide variety of deep learning tasks. Melo et al. [45] outlined the most promising approaches to integrating synthesized data into deep learning pipelines. Ward, Moghadam, and Hudson [46] used a real plant leaf dataset augmented with rendered images—for instance, leaf segmentation—to measure complex phenotypic traits in agricultural sustainability problems. Boikov et al. [47] presented a methodology for steel defect recognition in automated production control systems based on synthesized image data.

Several researchers introduced artificial intelligence (AI)-based methods into the AM field to classify the quality of manufacturing regions, as well as to segment failed areas in 3-D printing processes. Valizadeh and Wolff [48] provided a comprehensive overview of neural network applications to several aspects of AM processes. Banadaki et al. [49] proposed a convolutional neural network (CNN)-based automated system for assessing surface quality and internal defects in AM processes. The model is trained on captured images during material layering at various speeds and temperatures, and demonstrates 94% accuracy in five failure gradations in real time. Saluja et al. [50] utilized deep learning algorithms to develop a warping detection system. Their method extracts the layered corners of printed components and identifies warpage with 99.3% accuracy. Jin et al. [51] presented a novel CNN-based method incorporating strain to measure and predict four levels of delamination conditions. These works solve a set of specific production problems. The developed algorithms, however, are difficult to generalize and scale. Brion and Pattison [52] introduced an error detection and correction system based on visual and neural network analyses of extruded segments. This study demonstrates promising results; however, it is limited in providing general information about the whole working volume.

Analysis based on semantic segmentation, in turn, has significant potential for detecting and evaluating a wide range of manufacturing defects. Wong et al. [53] have demonstrated U-Net CNN 3-D volumetric segmentation in AM using medical imaging techniques to automatically detect defects in X-ray-computed tomography images of specimens. Cannizzaro et al. [54] proposed an AI in-situ emerging defects monitoring system utilizing automatic GAN-based synthetic image generation to augment the training dataset. These functions are built into a holistic distributed AM platform that allows storage and integration of data at all of manufacturing stages. Davtalab et al. [55] presented a neural network-automated system of semantic pixel-wise segmentation, based on one million images, to detect defects in 3-D-printed layers.

Combining various analysis techniques with the segmentation of characteristic areas of fabricated parts will make a significant contribution to the field of AM. Having an open-structure annotated database for additive manufacturing will create considerable opportunities for the development of failure detection systems in the future. Segmentation and localization of the individual structural elements of manufactured objects can make it easier to detect and track erroneous regions when they occur.

4.4 Methods

Preparation for 3-D printing involves layer-by-layer slicing of the model, where each extruded segment corresponds to a certain set of characteristics, such as fan speed, temperature, plastic flow rate, line type (internal, external, and overhang perimeters; support and its interface; solid and internal infill; etc.), reflected in the G-code. By using this information as input to the developed visual processing pipeline, it is possible to create an individual pixel-perfect mask for each section of the manufactured part.

Based on the most common words in 3-D print filenames stored in the Spaghetti Detective database [30], sets of labeled images of printed products at various stages of their production were generated in the physics-based graphics engine [29]. These image–mask pairs were further used to train neural networks for the tasks of visual segmentation of manufactured parts and their structural elements. Additionally, the possibilities of image-to-image style translation were also explored, to reduce the domain gap and increase the segmentation precision. The segmentation efficiency was tested both on synthetic renders outside of training sets and on real images. Data and source code for this project can be obtained from the Open Science Framework (OSF) repository [56].

4.4.1 Creation of Synthetic Image Datasets

1) Selecting CAD designs for rendering: More than 5.6 million filenames were partitioned into meaningful lexical parts and processed in Spaghetti Detective's user performance database [30] analysis to create a dictionary of the most frequently used words (Figure 4.2). These print jobs were performed by 49,000 unique users on 57,000 different 3-D printers. The average print time was 3.6 hours.

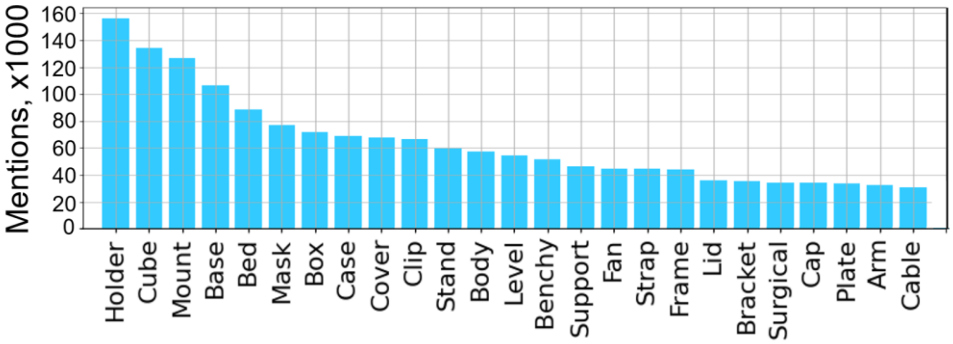


Figure 4.2: Distribution of the 25 most frequently used words in file names for 3-D printing. A detailed analysis of the users' print tasks database is given in the source file repository [56].

Based on the compiled dictionary, a set of random Standard Tessellation Language (STL) files was collected from Thingiverse [57]—an open catalog of widely used computer-aided designs (CADs) for 3-D printing—for further processing. These files formed the basis for generating a database of synthetic images. A complete list of the used CAD designs is in the OSF repository [56]. Knowing the most frequently used words in file names for 3-D printing does not provide an accurate idea of the size and complexity of the shapes of the produced parts. This, however, limits the initial selection of available models to the topic of household and technical items and excludes categories such as toys, jewelry, organic elements, multi-color designs, etc.

(2) Graphics rendering pipeline: All of the selected STL files were converted into G-codes in free MatterControl software [58], maintaining the same slicing parameters: 0.3 mm layer height, 0.4 mm nozzle diameter, 4 perimeters, and 30% grid infill. The resulting G-codes were further parsed layer by layer in the Blender [33] programming interface, where the extruder trajectory is converted into a set of curves with a controllable thickness parameter and preset material settings. Each G-code layer is thus transformed into an independent 3-D object. The whole rendering process is illustrated in the following diagram (Figure 4.3).

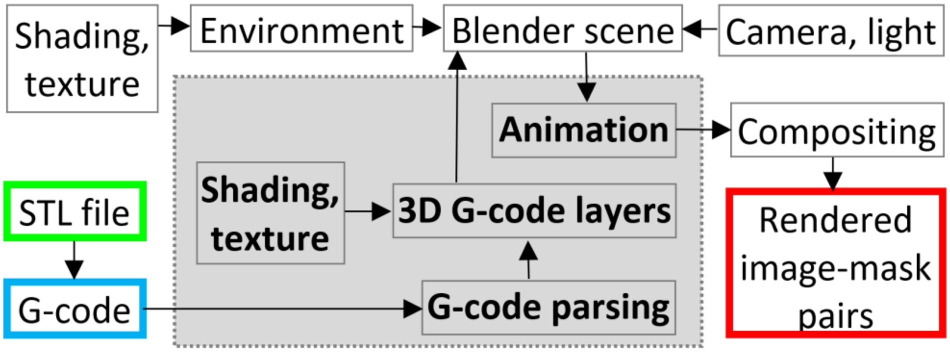


Figure 4.3: Synthetic AM database creation pipeline. Each 3-D part in the form of an STL (green) file is converted into a set of printer tool head trajectories (G-code, blue), which is the input parameter of the automated scripted section (gray). Blender environment (textures, camera, lights) and compositing settings can also be automated in the future. The image-mask pairs (red) are the result of a frame-by-frame animation rendering for each individual G-code file.

The functional component of the repository [59] was used as a basis for importing G-code files into the graphics engine. To create photorealistic renders, scenes similar to real physical environments were created in Blender. The position of the camera, as well as the degree of illumination and the location of light sources, were chosen to closely match the actual workspace. (Figure 4.4).

The whole scene, in addition to the printed part, includes components such as point light sources to create diverse heterogeneous all-round illumination; the “Sun”, to create uniform background lighting; a flat printing surface with realistic texture and reflectivity; and a plane with a superimposed blurred background image to create an illusion of a defocused ambient environment. Figure 4.5 illustrates several examples of realistic textures for the printing bed/ground surface plane.

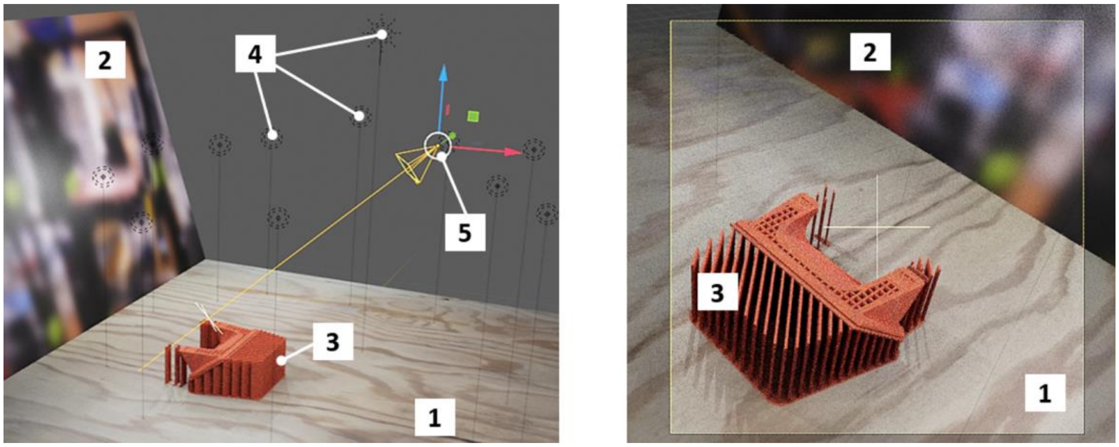


Figure 4.4: Blender scene: user window (left) and the virtual camera viewport (right). 1 – printing bed/ground surface texture, 2 – background image plane simulating ambient environment, 3 – rendered manufactured part, 4 – light sources with variable locations, 5 – camera with variable location.

The color of the plastic material and the surface characteristics of the printed part were created and adjusted empirically using a rich library of Blender shaders [60]. When simulating surface irregularities, the Noise Texture [61] and Voronoi Texture [62] nodes were used to add Perlin and Worley noises, respectively, while the "Bump" node was added to adjust the overall roughness. The Mix node was used to balance the Voronoi and Noise textures to create the desired roughness characteristics. Photorealistic color, transparency, and reflection parameters were obtained through the combination of Principled [63] (adds multiple layers to vary color, reflection, sheen, transmission, and other parameters), Glossy [64] (adds reflection with microfacet distribution), Diffuse [65] (adds Lambertian and Oren–Nayar diffuse reflections), and Transparent [66] (adds transparency without refraction) Bidirectional Scattering Distribution Functions (BSDFs) (Figure 4.6). Mix shaders 1 to 3 were used to adjust the strength of each BSDF component in the material output.

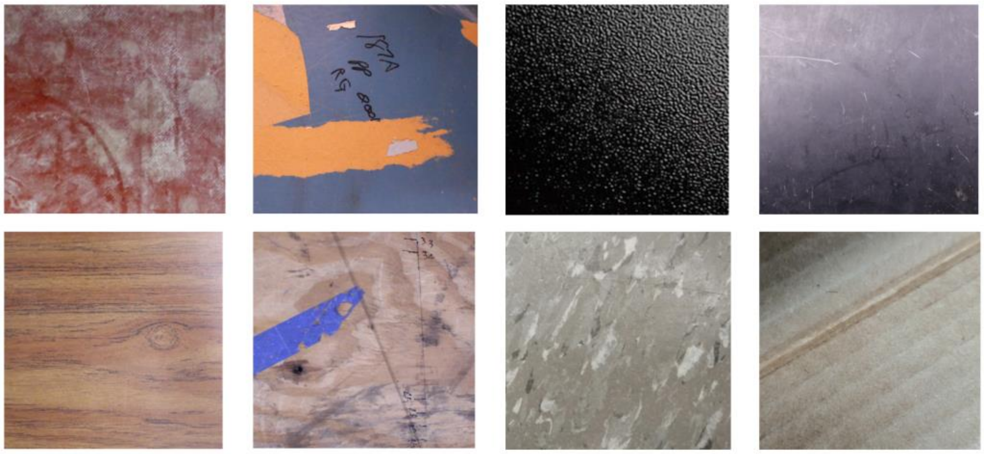


Figure 4.5: Texture samples for the printing bed/ground surface. More than 15 photographs of surfaces such as wood, metal, paper, stone, and others were superimposed onto the virtual working area. Variations in lighting, cropping, scaling, and image orientation during animation allow the creation of unique backgrounds.

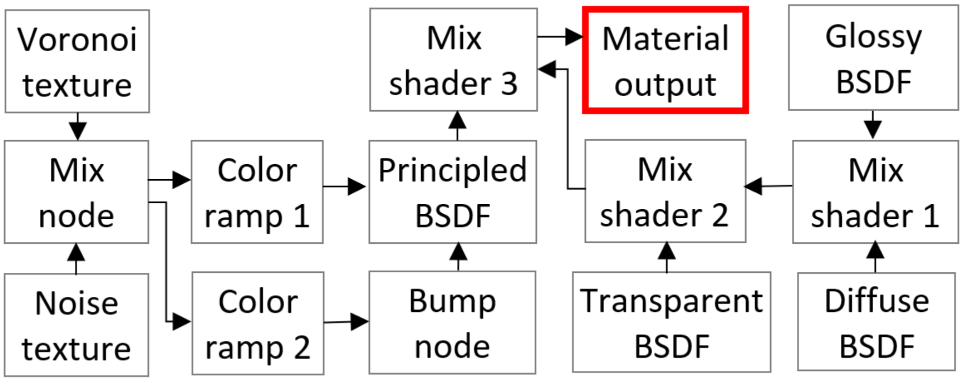


Figure 4.6: The shading node network has been experimentally developed to achieve maximum realism of generated renders. The creation of all of the connections and node settings is fully automated in the code, which provides the flexibility to adjust the color, transparency, reflectivity, and other characteristics of the output material (red).

A detailed example of the creation of such a texture is shown in Figure 4.7. The Principled BSDF node represents an elementary material. Adding Diffuse, Glossy, and Transparent shaders allows material variations to create a desirable effect. Adding different types of noise can simulate realistic unevenness and deviations in elevation in a surface map. The provided selection and hierarchy of BSDF shaders were chosen experimentally, and the desired result may be achieved in multiple alternative ways.

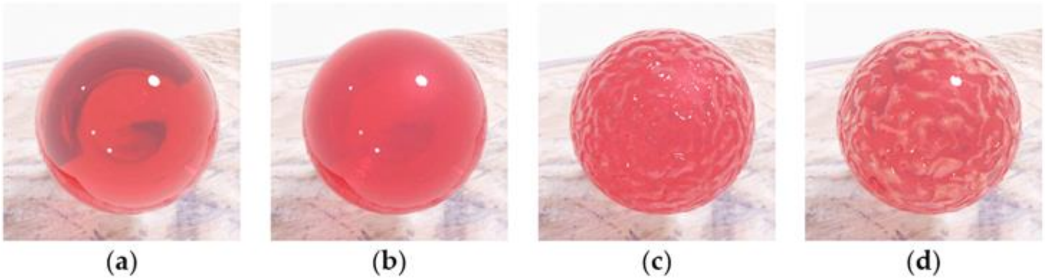


Figure 4.7: Detailed example of texture creation: (a) single Principled BSDF node; (b) Principled BSDF node mixed with Glossy, Diffuse, and Transparent BSDFs; (c) combined BSDF material with Noise and Voronoi textures; (d) final output with added color ramp nodes to truncate the Bump heights and create transmission anisotropy in the Principled BSDF.

The developed shading node network does not reflect all of possible and constantly expanding varieties of available 3-D printing materials, but provides end users with an initial set of tools to change color, texture, and transparency parameters to achieve the required effects. The developed materials are available in the open source file repository [56].

The G-code parsing procedure utilizes the functionality of the Blender application programming interface [67], which provides access to the properties of all of the shader nodes used in the scene. The entire animation process is scripted with randomized locations of the camera, light sources, and printing bed/ground surface plane in timeline keyframes, while the graphics engine adds intermediate frames by interpolation. Most of the G-codes were used twice with different levels of part completion, material color, print surface texture, light source locations, and camera orientations.

The built-in compositing interface [68] was used to create pixel-perfect ground truth masks for each frame (Figure 4.8). During the slicing procedure, each extruder path acquires its own type, which can be visualized in pseudo colors in the slicing environment (Figure 4.9). In this work, the outer and inner walls were combined into one structural element “shell”.

For visual segregation (masking) of individual scene elements (background, top layer at each manufacturing stage, infill, shell, and support), different values of the object pass index parameter [69] were set at the G-code parsing stage. This allows each selected element to be rendered as a region filled with a certain grayscale level.

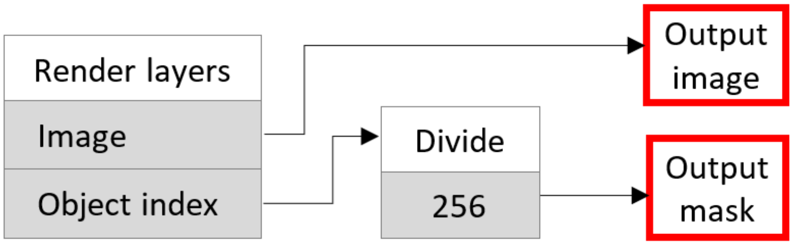


Figure 4.8: The composite node network (for internal structure segmentation, in this example) assigns user-defined color labels to each pixel in the output image, depending on whether it belongs to a particular area (infill, shell, or support) of the rendered part. This creates a pixel-precise ground truth mask (red) for each output image frame (red) in the animation.

The internal physics-based path tracer Cycles [70] was used to render each frame of the animation. To reduce rendering time, the number of samples was set to 64, the total number of light path reflections was reduced to 8, and the Reflective and Refractive Caustics features were disabled. This rendering optimization may restrict the quality of the images produced but can greatly reduce the computational load. Cycles' performance depends on the system's computational power. An 8 GB GPU setup with a 256×256 render tile size and an output image size of 1024×1024 pixels takes up to one minute to process a single frame, depending on the scale and geometric complexity of the scene within the camera viewport. Rendering an entire 50-frame animation this way can take up to one hour.

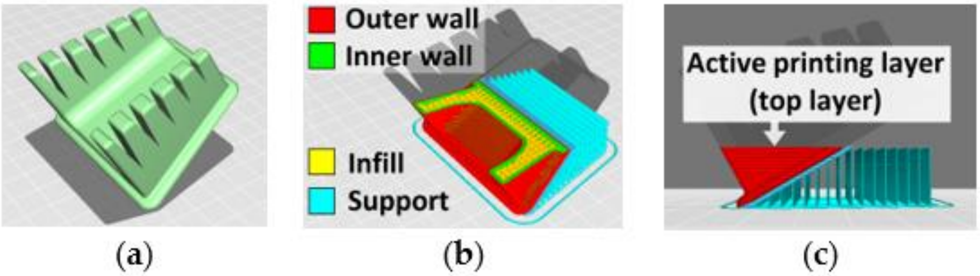


Figure 4.9: 3-D model slicing procedure. (a) Whole part in STL format. (b) Internal structure of sliced layers (red – outer shell, green – inner shell, yellow – infill, blue – support). (c) Side view illustrates current printing layer (top layer at each manufacturing stage).

3) Synthetic image datasets: For the further task of semantic segmentation, three separate datasets were created. Examples of image-mask pairs included in the created datasets are presented in Figure 4.10.. A total of 5763 1024×1024 pixel image–mask pairs were generated for the segmentation of the entire 3-D-printed part; 3570 – for the top layer segmentation; and 1140 – for the infill, shell, and support (internal layer structure) segmentation.

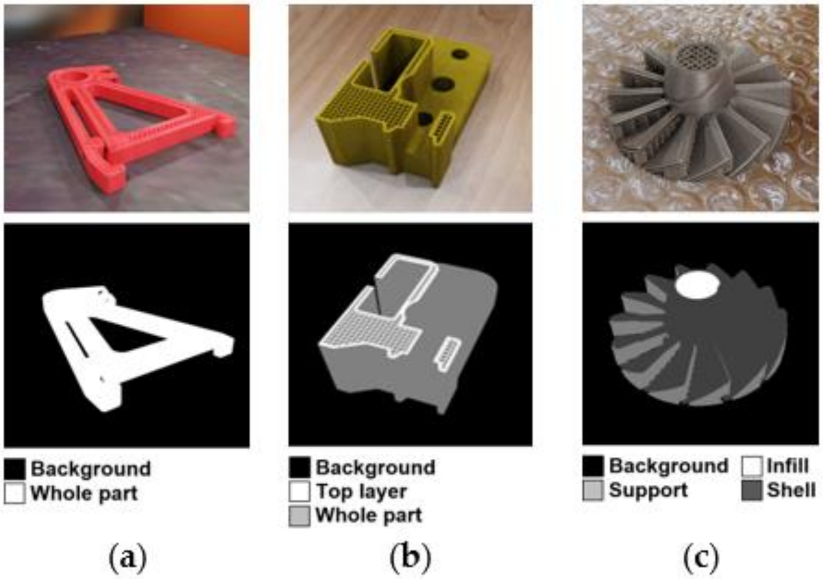


Figure 4.10: Image-mask pair samples for each AM synthetic dataset: (a) whole part segmentation, (b) newly fabricated top layer segmentation, (c) internal layer segmentation.

The training database for segmenting the whole part and image background included 21 models containing words in the names such as “holder”, “mount”, “base”, “clamp”, “stand”, “chassis”, “bracket”, “block”, etc. After slicing, 3-5 scenes were created for each model with various lighting parameters, material color, and background texture in the rendering engine. Next, in each scene, 50 to 80 keyframes were created showing the part at various stages of its fabrication. Each keyframe was then rendered to produce a unique image-mask pair. Similarly, 34 unique models were selected to create a training database for segmenting the top layer, and 11 models – for segmenting the internal structure of the parts.

4.4.2 Semantic Image Segmentation

Minaee et al. [32], as well as Ulku and Akagunduz [40], presented a comprehensive overview of the modern research state in the field of semantic segmentation. As can be seen from the works [71,72,73], the U-Net family of neural network architectures has demonstrated high segmentation efficiency with small amounts of training data. The DeepLab architecture, in turn, is one of the basic architectures for subsequent domain adaptation [74,75,76]. This work employs the U-Net architecture [77] and its multi-class adaptation [78] due to its efficiency and simplicity.

The intersection over union (IoU) quantifies the degree of overlap (from 0 to 100%) between the ground truth mask (a pixel-perfect grayscale image created in Blender as a mask in image-mask pairs) and the segmented pixel area of its predicted version, where a larger value indicates a more accurate segmentation, and the mIoU is the mean IoU value across the correspondent classes in the dataset. The calculation of mIoU scores for the real images was carried out only for the segmentation of the entire part, since obtaining manually-labeled ground truth masks for the top layer and the internal structure of the part is a nontrivial task, considering the geometric complexity of the filling elements.

4.4.3 Image-to-image Translation

To potentially improve the efficiency of semantic segmentation, the application of the unpaired image-to-image translation method based on the CycleGAN network [39] was considered. The given DA method learns the mapping between the source domain (real images) and the target domain (synthetic images) by minimizing the cycle consistency loss L_C (Figure 4.11) in the absence of paired data samples. This minimizes the domain gap by changing the appearance of the images, making the test data more similar to the training data.

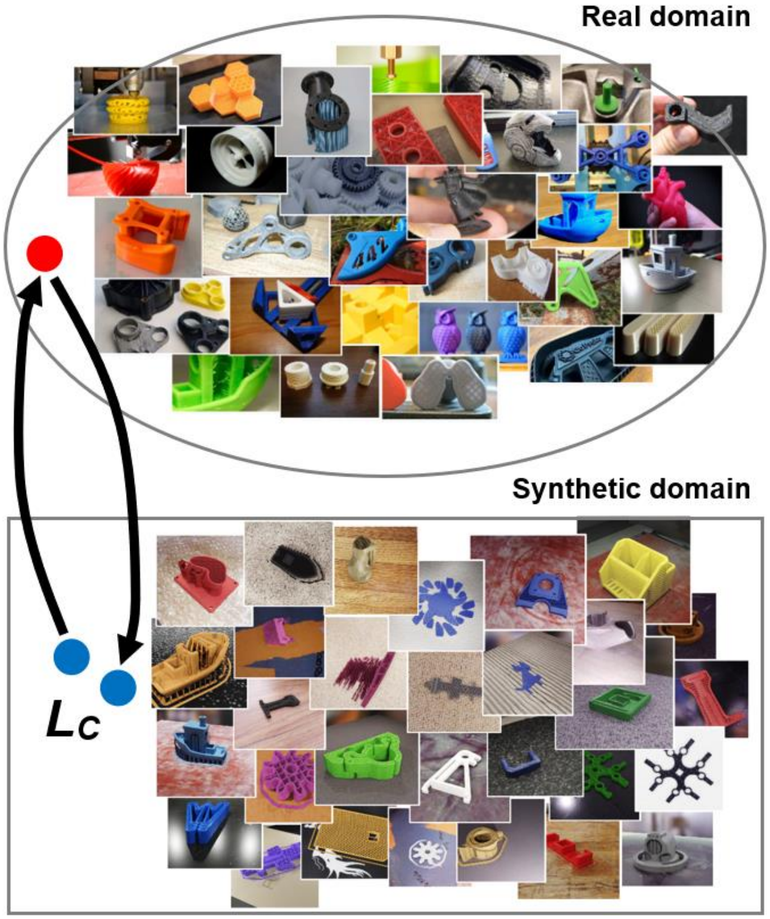


Figure 4.11: Unpaired image-to-image translation using the cycle-consistent adversarial network. Handpicked images of real and virtual printed parts were loaded into CycleGAN, which learns to map real domain images to their synthetic counterparts and vice versa, minimizing the cycle consistency loss L_C. Here, the red and blue circles represent the same image presented in different domains.

For this task, we manually selected 589 synthetic renders and 794 real images of 3-D-printed parts. The learning result is two generators that convert the original images of the real domain into their synthetic counterparts, and vice versa (Figure 4.12).

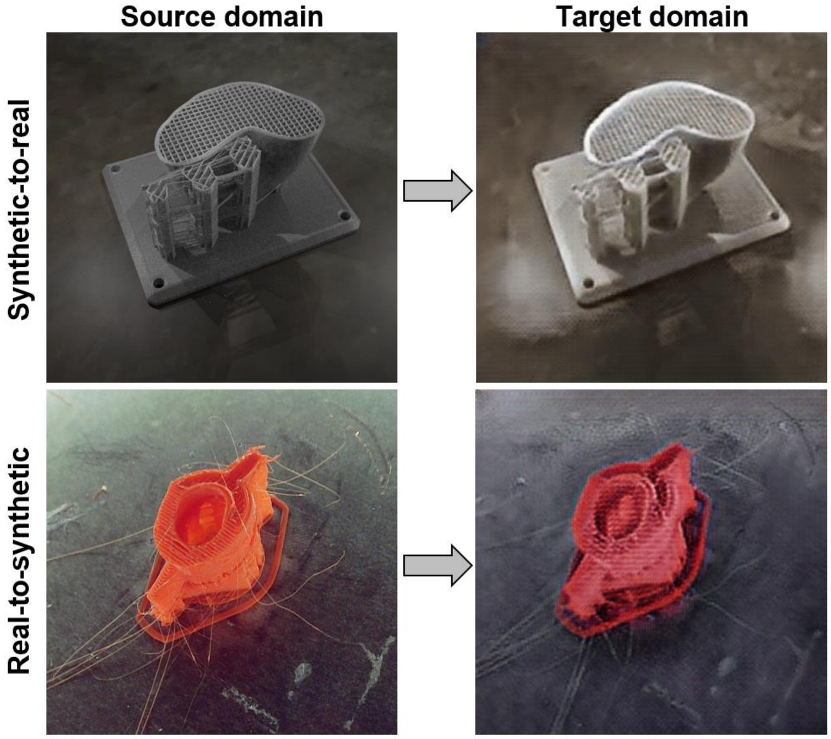


Figure 4.12: Image-to-image style translation example. Translating a real image into its synthetic version reduces the contrast and saturation of the reflections and incidental filament strings.

As can be seen from Figure 4.12, translating a synthetic render into a real image makes colors more natural, while translating a real image into a synthetic one also reduces the contrast and saturation of both reflections on the printing bed/ground surface and incidental filament strings. This characteristic can improve segmentation in mediocre images.

4.5 Results

The results of the semantic segmentation are presented using several real images, presented in Figure 4.13. The training of the neural network was carried out on synthetic renders without using the style translation technique.

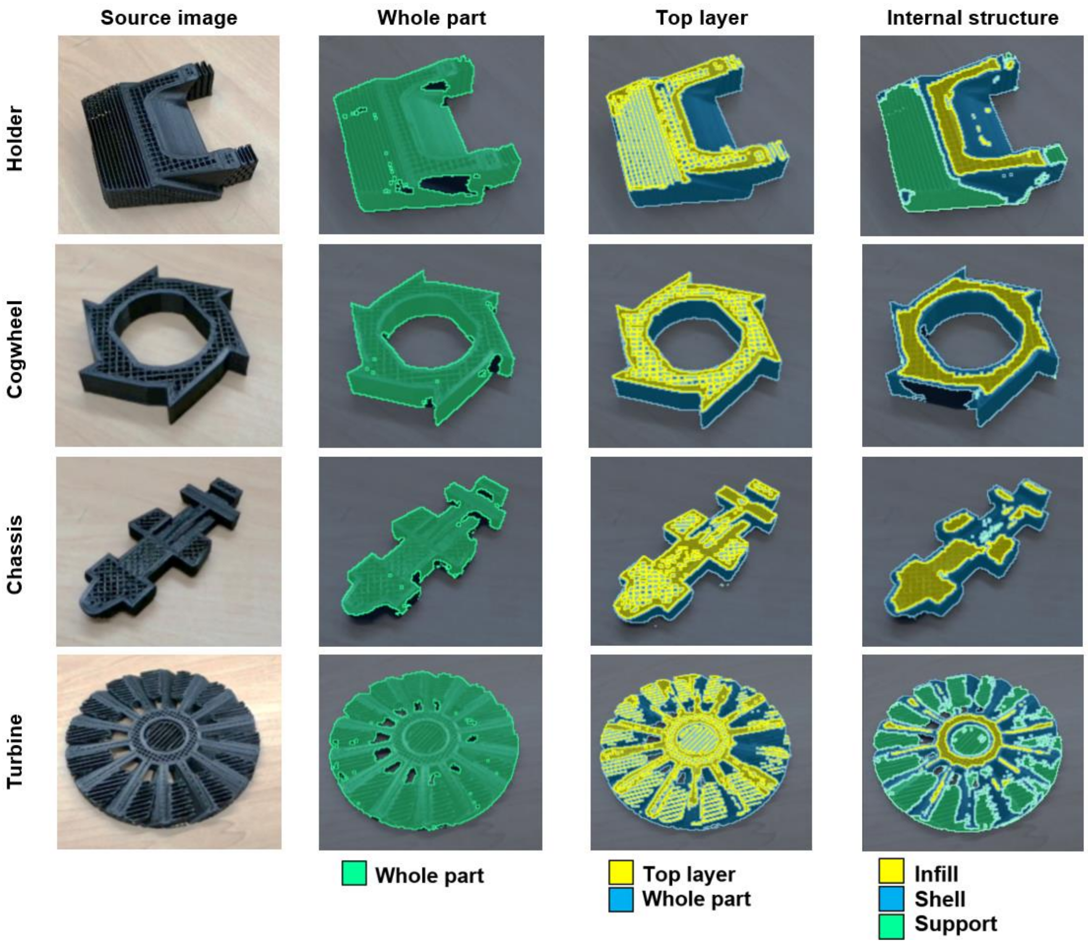


Figure 4.13: The results of semantic segmentation, presented using several real images. The neural network was trained on similar synthetic 3-D models. The color, printing surface texture, and slicing parameters, however, differ from those used in the training dataset.

Quantitative results are shown in Table 4.1. Test datasets include synthetic renders of STL models, both those included in the training dataset and those not included in it. The color, view angle, and environmental parameters of the 3-D models included in the test datasets were changed to avoid matching the training data.

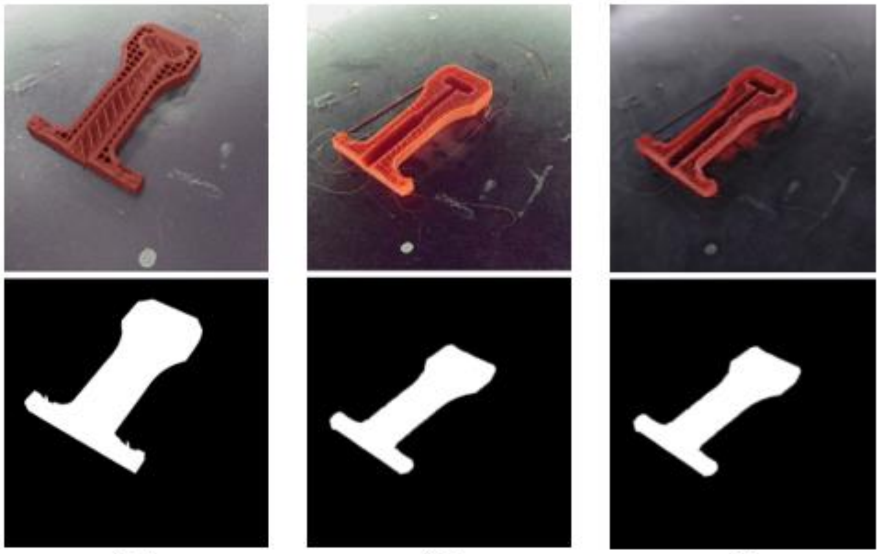
As can be seen from Table 4.1, the segmentation accuracy on real images (78.16%) is inferior to that of the synthetic data (94.90%), which indicates the need for additional research on domain adaptation. Detecting the top layer is a more complex task for the neural network compared to segmenting the entire part, which is clearly noticeable in the results within the same dataset (mIoU 73.33% for the top layer, versus 99.74% for the background). Shell segmentation has the lowest score (mIoU 55.31%). This, apparently, is due to the variety of geometric shapes and the lack of a characteristic texture that the infill and support areas have. Relatively low mIoU values for shell, support, and infill segmentation are due to the more complex geometry of these structural elements and the presence of a large number of thin lines in images. These scores can be improved by creating more images in the training dataset. The segmentation efficiency of individual part elements depends on their geometric complexity and the key factor for effective semantic segmentation is the number of image–mask pairs. With the developed open source methodology, this database can be significantly expanded by end users, which will lead to increased segmentation accuracy among all of available categories.

Table 4.1: Segmentation results for synthetic test datasets (mIoU scores, %).

No. of images	Test dataset	Background	Top layer	Shell	Support	Infill
89	Whole part segmentation (real images)	78.16	–	–	–	–
101	Whole part segmentation (synthetic renders images)	94.90	–	–	–	–
68	Top layer segmentation (synthetic renders)	99.74	73.33	–	–	–
57	Internal structure segmentation (synthetic renders)	94.52	–	55.31	69.45	78.93

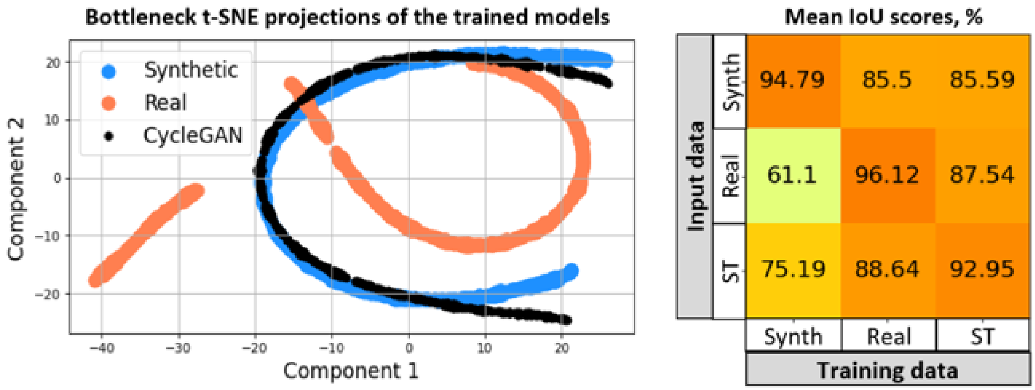
To analyze the influence of style transfer (ST) on semantic segmentation, separate CNN training of three datasets of one part was carried out (Figure 4.14). Synthetic and real datasets consist of 49 and 36 image–mask pairs, respectively.

To compare the domains, we used t-distributed stochastic neighbor embedding (t-SNE) [79,80] projections of the normalized bottleneck layers of trained U-Net models (Figure 4.14). The nonlinear dimensionality reduction technique was applied to 512-dimensional normalized vectors in the narrowest parts of the trained models to visualize the affinity of the domains in latent feature space. As can be seen from Figure 4.15a, the feature space of the real domain (orange) is getting closer to synthetic data (blue) after the image-to-image style translation (black).



(a) (b) (c)

Figure 4.14: Datasets for the style transfer influence analysis: (a) synthetic data, (b) real data, (c) real data after style transfer. The upper row shows sample images and the lower row illustrates the corresponding ground truth masks.



Bottleneck t-SNE projections of the trained models

Mean IoU scores, %

Input data \ Training data	Synth	Real	ST
Synth	94.79	85.5	85.59
Real	61.1	96.12	87.54
ST	75.19	88.64	92.95

(a) (b)

Figure 4.15: Domain comparison via t-SNE projections (a), and segmentation performance before and after style translation (b).

In addition to t-SNE projections, the segmentation performance of the real image data after ST was also analyzed (Figure 4.15b). The heatmap columns represent the data on which the neural network model was trained, and the rows stand for the input data to which segmentation was applied. The highest mIoU, as expected, was observed in those datasets on which the model was trained. When converting the real input data into ST using image-to-image translation, however, the segmentation score increased from 61.10% to 75.19%

for the model trained solely on synthetic data. This parameter is the most valuable, since in real conditions, training a convolutional network on real data may not be possible due to the lack of ground truth masks. This indicates that the ST method as a precursor to domain adaptation can significantly improve real 3-D printing image segmentation in situations where a model trained on synthetic data may be the only tool available. The sample results of image segmentation before and after style translation are shown in Figure 4.16.

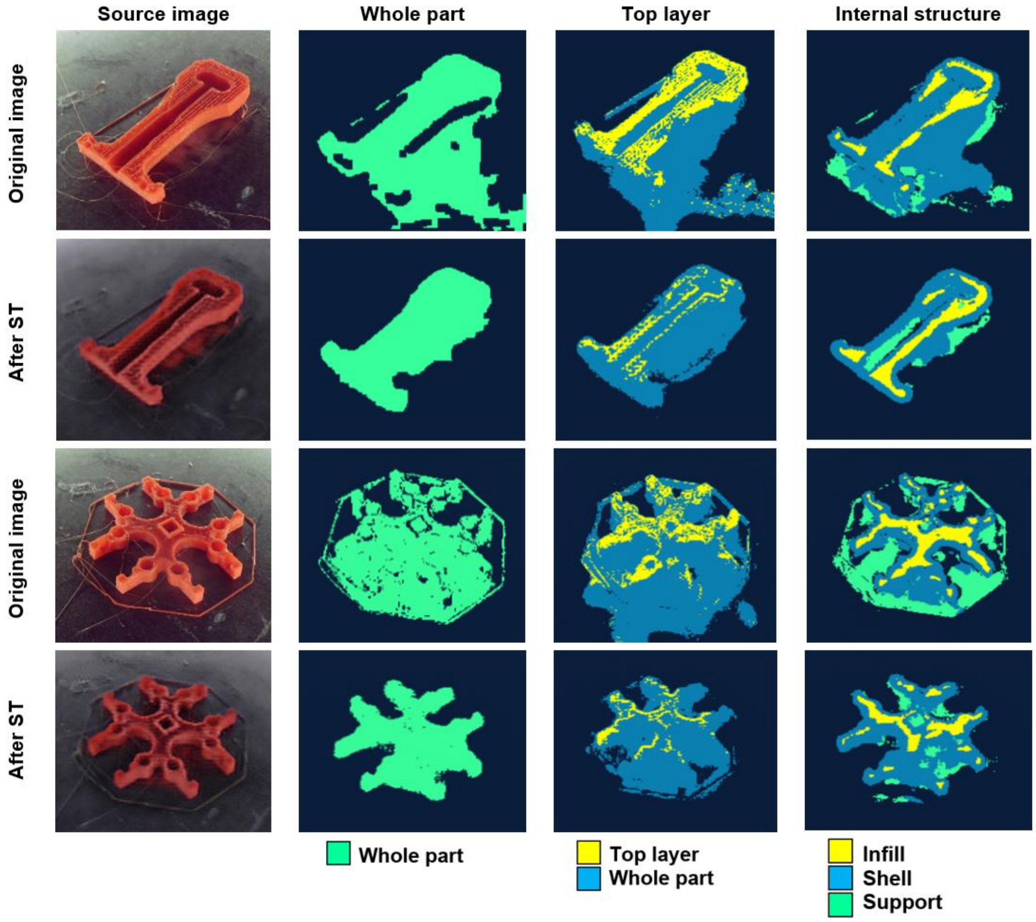


Figure 4.16: The results of image segmentation before and after style translation. Real-to-synthetic style transfer reduces the saturation of the incidental filament strings and reflections on the printing platform, which, in turn, affects the results of semantic segmentation.

As can be seen from Figure 4.16, real-to-synthetic style transferring reduces the saturation of the incidental filament strings and reflections on the printing platform, which, in turn, affects the results of semantic segmentation. Image-to-image translation, therefore, could be a powerful tool in further improving segmentation performance through domain adaptation techniques.

This work continues the previous authors' research on the use of physical rendering and demonstrates the significant potential of using synthetic data and machine learning in the field of additive manufacturing. Due to the relative simplicity of virtual printing and training data generation, segmentation of the contours of a manufactured part can be performed at every stage of its creation using a single camera in an arbitrary position. This reduces the requirements for camera calibration and eliminates the need to use visual markers to tightly bind the image frame to the coordinate system of the 3-D printing space. It also offers the flexibility to be used on any type of 3-D printing system with the addition of an after-market camera.

The limitations of the developed method are the need to create synthetic images and increase the training dataset for each new manufactured part, as well as the implementation of transfer learning to improve the segmentation accuracy. Additional research is also required in the field of domain adaptation applications based on existing state-of-the-art techniques [81,82,83].

Together with edge-based markerless tracking [84,85], the developed technique can become an integral part of a 3-D printing control and monitoring system such as OctoPrint [86]. In the future, this will make it possible to implement an inline comprehensive system for recognizing the type of part being produced and determining its location and orientation in the workspace, as well as for tracking its manufacturing deviations.

4.6 Discussion and Conclusions

The proposed method is a novel approach to segmenting key regions of a part during its fabrication utilizing the information in the G-code and synthetic image data.

In most cases, localization of detected anomalies is based on linking the camera position to reference visual markers. This allows image coordinates to be converted into 3-D coordinates on the workspace, which, however, requires precise calibration and a rigid, permanent camera mount that can be difficult for users. When using the developed semantic segmentation method, the need for precise camera positioning is eliminated, since the part and its structural elements can be segmented from almost any viewing angle. The semantic segmentation framework for additive manufacturing can enhance the visual analysis of manufacturing processes and allow the detection of individual manufacturing errors, while significantly reducing the requirements for positioning accuracy and camera calibration.

The results of this work will allow the localization of 3-D-printed parts in captured image frames, as well as the application of image processing techniques to the parts' structural elements to follow the tracking of manufacturing deviations. The use of image style transfer is of significant value for further research in the field of adapting the domain of synthetic renders to real images of 3-D-printed products.

The methodology demonstrated achieved the following mIoU scores for the synthetic test datasets: entire printed part, 94.90%; top layer, 73.33%; infill, 78.93%; shell, 55.31%; support, 69.45%. Increasing the number of image–mask pairs used for training neural networks will improve segmentation accuracy. The results illustrate the effectiveness of the developed method, but also indicate the need for additional experiments to eliminate the synthetic-to-real domain gap. Additional research is also required to expand the training database, as well as introduce production errors to analyze their impact on segmentation.

This research presents ways to expand the number of segmentation categories by using the information in the G-code about the characteristics of separate extruded sections. Further study, however, is required to analyze the impacts of material and texture shaders, as well as lighting and rendering parameters, on segmentation efficiency.

Revealing this system to end users will allow constant expansion of the synthetic image database for subsequent neural network training and improvement of segmentation results. Integrating it with web-based 3-D printing control systems can help to perform layer-wise analysis of manufactured parts, and also help to classify and track failures based on their bonding to a particular area of the model. In the case of small 3-D printing deviations, without significant deformation of the part, this method will be able to determine whether a manufacturing defect belongs to a particular structural element of the part. This, in turn, can facilitate the creation of procedures for correcting manufacturing defects.

4.7 Bibliography

[1] R. Geyer, J.R. Jambeck, and K.L. Law, Production, Use, and Fate of all Plastics Ever Made, Sci. Adv., 2017, 3(7), e1700782, doi: 10.1126/sciadv.1700782.

[2] J.R. Jambeck, R. Geyer, C. Wilcox, T.R. Siegler, M. Perryman, A. Andrady, R. Narayan, and K.L. Law, Plastic Waste Inputs From Land Into the Ocean, Science, 2015, 347(6223), pp. 768–771, doi: 10.1126/science.1260352.

[3] A.O. Laplume, B. Petersen, and J.M. Pearce, Global value chains from a 3D printing perspective, J Int Bus Stud, 2016, 47(5), pp. 595–609, doi: 10.1057/jibs.2015.47.

[4] E.E. Petersen and J.M. Pearce, Emergence of home manufacturing in the developed world: Return on investment for open-source 3-D printers, Technologies, 2017, 5(1), 7.

[5] J.M. Pearce and J.Y. Qian, Economic Impact of DIY Home Manufacturing of Consumer Products with Low-cost 3D Printing from Free and Open Source Designs, European Journal of Social Impact and Circular Economy, 2022, 3(2), 1-24, doi: 10.13135/2704-9906/6508.

[6] E. Hunt, C. Zhang, N. Anzalone, and J.M. Pearce, Polymer recycling codes for distributed manufacturing with 3-D printers, Resources, Conservation and Recycling, 2015, 97, 24-30.

[7] N. Shahrubudin, T.C. Lee, and R. Ramlan, An overview on 3D printing technology: Technological, materials, and ap-plications, Procedia Manufacturing, 2019, 35, 1286-1296.

[8] "Global 3D Printing Filament Market By Material, By Type, By End Use, By Region, Competition, Forecast & Opportunities, 2024," May, 2019. Available: https://www.reportbuyer.com/product/5778909/global-3d-printing-filament-market-by-material-by-typeby-end-use-by-regioncompetition-forecast-and-opportunities-2024.html (accessed January 10, 2024).

[9] B.T. Wittbrodt, A.G. Glover, J. Laureto, G.C. Anzalone, D. Oppliger, J.L. Irwin, and J.M. Pearce, Life-cycle economic analysis of distributed manufacturing with open-source 3-D printers, Mechatronics, 2013, 23(6), 713-726.

[10] S. Sharp, CEO 3DQue Systems. Personal communication. June 4, 2022.

[11] H.D. Kang, Analysis of furniture design cases using 3D printing technique, The Journal of the Korea Contents Association, 2015, 15(2), 177-186.

[12] J.K. Bow, N. Gallup, S.A. Sadat, and J.M. Pearce, Open source surgical fracture table for digitally distributed manufacturing, PloS one, 2022, 17(7), e0270328.

[13] J.I. Novak and J. O’Neill, A design for additive manufacturing case study: fingerprint stool on a BigRep ONE, Rapid Prototyping Journal, 2019, 25(6), pp. 1069–1079.

[14] A. Petsiuk, B. Lavu, R. Dick, and J.M. Pearce, Waste Plastic Direct Extrusion Hangprinter, Inventions, 2022, 7(3), p.70.

[15] A.L. Woern, D.J. Byard, R.B. Oakley, M.J. Fiedler, S.L. Snabes, and J.M. Pearce, Fused particle fabrication 3-D printing: Recycled materials’ optimization and mechanical properties, Materials, 2018, 11(8), p.1413.

[16] A. Oleff, B. Kuster, M. Stonis, L. Overmeyer, Process monitoring for material extrusion additive manufacturing: a state-of-the-art review, Prog Addit Manuf, 2021, 6, pp. 705–730, doi: 10.1007/s40964-021-00192-4.

[17] A. Ceruti, A. Liverani, T. Bombardi, Augmented vision and interactive monitoring in 3D printing process, Int J Inter Des Manuf, 2017, 11, pp. 385–395, doi: 10.1007/s12008-016-0347-y.

[18] S. Nuchitprasitchai, M.C. Roggemann, J.M. Pearce, Factors effecting real-time optical monitoring of fused filament 3D printing, Prog Addit Manuf J, 2017, 2(3), pp. 133–149, doi: 10.1007/s40964-017-0027-x.

[19] A. Johnson, H. Zarezadeh, X. Han, R. Bibb, R. Harris, Establishing in-process inspection requirements for material ex-trusion additive manufacturing, in Proceedings of the Fraunhofer Direct Digital Manufacturing Conference. Berlin: Fraunhofer-Gesellschaft, 2016.

[20] S. Hurd, C. Camp, J. White, Quality assurance in additive manufacturing through mobile computing, Int Conf Mob Comput Appl Serv, 2015, pp. 203–220.

[21] H. Jeong, M. Kim, B. Park, S. Lee, Vision-Based Real-Time Layer Error Quantification for Additive Manufacturing, in Proc ASME 2017 12th Int Manuf Sci Eng Conf, Los Angeles, California, USA, 2017.

[22] F. Wasserfall, D. Ahlers and N. Hendrich, Optical In-Situ Verification of 3D-Printed Electronic Circuits, IEEE 15th Int Conf Autom Sci and Eng (CASE), 2019, pp. 1302–1307, doi: 10.1109/COASE.2019.8842835.

[23] J. Straub, 3D printing cybersecurity: Detecting and preventing attacks that seek to weaken a printed object by changing fill level, Proc SPIE Dimens Opt Metrol Insp Pract Appl VI, 2017, doi: 10.1117/12.2264575.

[24] M.D. Kutzer, L.D. DeVries, C.D. Blas, Part monitoring and quality assessment of conformal additive manufacturing using image reconstruction, in Proc ASME 2018 Int Des Eng Tech Conf Comput Inf Eng Conf 5B, Quebec, Canada, 2018, doi: 10.1115/DETC2018-85370.

[25] Z. Chen and R. Horowitz, Vision-assisted Arm Motion Planning for Freeform 3D Printing, 2019 American Control Conference (ACC), pp. 4204–4209, doi: 10.23919/ACC.2019.8814699.

[26] H. Shen, W. Du, W. Sun, Y. Xu, J. Fu, Visual detection of surface defects based on self-feature comparison in robot 3-D printing, Appl Sci, 2020, 10(1), 235, doi: 10.3390/app10010235.

[27] A. Malik, H. Lhachemi, J. Ploennigs, A. Ba, R. Shorten, An application of 3D model reconstruction and augmented reality for real-time monitoring of additive manufacturing, Procedia CIRP, 2019, 81, pp. 346–351.

[28] A. Petsiuk, J.M. Pearce, Open source computer vision-based layer-wise 3D printing analysis, Addit Manuf, 2020, 36, 101473, doi: 10.1016/j.addma.2020.101473.

[29] A. Petsiuk, J.M. Pearce, Towards smart monitored AM: Open source in-situ layer-wise 3D printing image anomaly detection using histograms of oriented gradients and a physics-based rendering engine, Addit Manuf, 2022, 52, 102690, doi: 10.1016/j.addma.2022.102690.

[30] Spaghetti Detective. Available: https://www.obico.io/the-spaghettidetective.html (accessed January 10, 2024).

[31] The Spaghetti Detective Plugin. Available: https://github.com/TheSpaghettiDetective/OctoPrintTheSpaghettiDetective (accessed January 10, 2024).

[32] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz and D. Terzopoulos, Image Segmentation Using Deep Learning: A Survey, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(7), pp. 3523–3542, doi: 10.1109/TPAMI.2021.3059968.

[33] Blender: the free and open source 3D creation suite. Available: https://www.blender.org (accessed January 10, 2024).

[34] G. Csurka, R. Volpi, B. Chidlovskii, Unsupervised Domain Adaptation for Semantic Image Segmentation: a Comprehensive Survey, 2021, arXiv:2112.03241.

[35] A. Farahani, S. Voghoei, K. Rasheed, H.R. Arabnia, A Brief Review of Domain Adaptation, in Advances in Data Science and Information Engineering. Transactions on Computational Science and Computational Intelligence. Springer, Cham, 2021, doi: 10.1007/978-3-030-71704-9 65.

[36] B. Imbusch, M. Schwarz, S. Behnke, Synthetic-to-Real Domain Adaptation using Contrastive Unpaired Translation, 2022, arXiv:2203.09454.

[37] P. Li, X. Liang, D. Jia, E.P. Xing, Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption, 2018, arXiv:1801.01726.

[38] S. Lee, E. Park, H. Yi, S.H. Lee, StRDAN: Synthetic-to-Real Domain Adaptation Network for Vehicle Re-Identification, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[39] J. -Y. Zhu, T. Park, P. Isola and A. A. Efros, Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks, in 2017 IEEE

International Conference on Computer Vision (ICCV), pp. 2242–2251, doi: 10.1109/ICCV.2017.244.

[40] I. Ulku and E. Akagunduz, A Survey on Deep Learning-based Architectures for Semantic Segmentation on 2D Images, Applied Artificial Intelligence, 2022, 36(1), doi: 10.1080/08839514.2022.2032924.

[41] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The Cityscapes Dataset for Semantic Urban Scene Understanding, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[42] S.R. Richter, V. Vineet, S. Roth, V. Koltun, Playing for Data: Ground Truth from Computer Games, 2016, arXiv:1608.02192.

[43] G. Ros, L. Sellart, J. Materzynska, D. Vazquez and A. M. Lopez, The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3234–3243, doi: 10.1109/CVPR.2016.352.

[44] S.I. Nikolenko. Synthetic Data for Deep Learning. SOIA, vol. 174. Springer, Cham, 2021, doi: 10.1007/978-3-030-75178-4.

[45] C.M. de Melo, A. Torralba, L. Guibas, J. DiCarlo, R. Chellappa, J. Hodgins, Next-generation deep learning based on simulators and synthetic data, Trends Cogn Sci, 2022, 26(2), pp. 174–187, doi:10.1016/j.tics.2021.11.008.

[46] D. Ward, P. Moghadam, N. Hudson, Deep Leaf Segmentation Using Synthetic Data, 2018, arXiv:1807.10931.

[47] A. Boikov, V. Payor, R. Savelev, A. Kolesnikov, Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning, Symmetry, 2021, 13(7):1176, doi: 10.3390/sym13071176.

[48] M. Valizadeh, S.J. Wolff, Convolutional Neural Network applications in additive manufacturing: A review, Adv in Ind and Manuf Eng, 2022, 4, 100072, doi: 10.1016/j.aime.2022.100072.

[49] Y. Banadaki, N. Razaviarab, H. Fekrmandi, S. Sharifi, Toward Enabling a Reliable Quality Monitoring System for Additive Manufacturing Process using Deep Convolutional Neural Networks, 2020, arXiv:2003.08749.

[50] A. Saluja, J. Xie, K. Fayazbakhsh, A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks, J of Manuf Proc, 2020, 58, pp. 407–415, doi:10.1016/j.jmapro.2020.08.036.

[51] Z. Jin, Z. Zhang, G.X. Gu, Automated Real-Time Detection and Prediction of Interlayer Imperfections in Additive Manufacturing Processes Using Artificial Intelligence, Adv Intell Syst, 2019, 2, 1900130, doi:10.1002/aisy.201900130.

[52] D.A.J. Brion, S.W. Pattinson, Generalizable 3D printing error detection and correction via multi-head neural networks, Nat Commun 13, 4654 (2022), https://doi.org/10.1038/s41467-022-31985-y.V.W.H. Wong, M. Ferguson, K.H. Law, Y.T. Lee, P. Witherell, Automatic Volumetric Segmentation of Additive

Manufacturing Defects with 3D U-Net, in AAAI 2020 Spring Symposia, Stanford, CA, USA, March, 2020, arXiv:2101.08993.

[53] V.W.H. Wong, M. Ferguson, K.H. Law, Y.T. Lee, P. Witherell, Segmentation of Additive Manufacturing Defects Using U-Net, ASME J Comput Inf Sci Eng, 2022, 22(3), 031005, doi: 10.1115/1.4053078.

[54] D. Cannizzaro et al., In-Situ Defect Detection of Metal Additive Manufacturing: An Integrated Framework, in IEEE Transactions on Emerging Topics in Computing, 2022, 10(1), pp. 74–86, doi:10.1109/TETC.2021.3108844.

[55] O. Davtalab, A. Kazemian, X. Yuan, B. Khoshnevis, Automated inspection in robotic additive manufacturing using deep learning for layer deformation detection, J Intell Manuf, 2022, 33, pp. 771–784, doi:10.1007/s10845-020-01684-w.

[56] J.M. Pearce, A. Petsiuk, Synthetic-to-real composite semantic segmentation in additive manufacturing. OSF Source file repository. Available: https://osf.io/h8r45 (accessed January 10, 2024).

[57] Thingiverse: an open catalog of computer-aided designs for 3D printing. Available: https://www.thingiverse.com (accessed January 10, 2024).

[58] MatterControl: 3D Printing Software. Available: https://www.matterhackers.com/store/l/mattercontrol/sk/MKZGTDW6 (accessed January 10, 2024).

[59] H. Lopmeier, Blender-Gcode-Importer, Available: https://github.com/Heinz-Loepmeier/Blender-Gcode-Import (accessed January 10, 2024).

[60] Blender: Shader nodes library. Available: https://docs.blender.org (accessed January 10, 2024).

[61] Blender: Noise Texture Node. Available: https://docs.blender.org (accessed January 10, 2024).

[62] Blender: Voronoi Texture node. Available: https://docs.blender.org (accessed January 10, 2024).

[63] Blender: Principled BSDF. Available: https://docs.blender.org (accessed January 10, 2024).

[64] Blender: Glossy BSDF. Available: https://docs.blender.org (accessed January 10, 2024).

[65] Blender: Diffuse BSDF. Available: https://docs.blender.org (accessed January 10, 2024).

[66] Blender: Transparent BSDF. Available: https://docs.blender.org (accessed January 10, 2024).

[67] Blender API. Available: https://docs.blender.org/api/current/ (accessed January 10, 2024).

[68] Blender Compositing. Available: https://docs.blender.org (accessed January 10, 2024).

[69] Blender: Material Pass Index. Available: https://docs.blender.org (accessed January 10, 2024).

[70] Blender Cycles. Available: https://docs.blender.org (accessed January 10, 2024).

[71] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015, arXiv:1505.04597.

[72] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O.R. Zaiane, M. Jagersand, U2-Net: Going Deeper with Nested U-Structure for Salient Object Detection, 2020, arXiv:2005.09007.

[73] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, J. Wu, UNet 3+: A Full-Scale Connected UNet for Medical Image Segmentation, 2020, arXiv:2004.08790.

[74] M. Toldo, U. Michieli, P. Zanuttigh, Unsupervised Domain Adaptation in Semantic Segmentation via Orthogonal and Clustered Embeddings, 2020, arXiv:2011.12616.

[75] J. Yang, C. Li, W. An, H. Ma, Y. Guo, Y. Rong, P. Zhao, J. Huang, Exploring Robustness of Unsupervised Domain Adaptation in Semantic Segmentation, 2021, arXiv:2105.10843.

[76] X. Guo, C. Yang, B. Li, Y. Yuan, MetaCorrection: Domain-aware Meta Loss Correction for Unsupervised Domain Adaptation in Semantic Segmentation, 2021, arXiv:2103.05254.

[77] M. Buda, U-Net for brain segmentation, 2019. Available: https://pytorch.org/hub/mateuszbuda brain-segmentation-pytorch_unet (accessed January 10, 2024).

[78] F. Battocchio, U-Net architecture for Multiclass semantic segmentation, 2020. Available: https://github.com/France1/unet-multiclasspytorch (accessed January 10, 2024).

[79] G.E. Hinton and S.T. Roweis, Stochastic Neighbor Embedding, in Advances in Neural Information Processing Systems, S. Becker, S. Thrun, K. Obermayer (eds). MIT Press, vol. 15, 2002.

[80] L.J.P. van der Maaten, G.E. Hinton, Visualizing High-Dimensional Data Using t-SNE, Journal of Machine Learning Research, 2008, 9, pp. 2579–2605.

[81] T. Xu, W. Chen, P. Wang, F. Wang, H. Li, R. Jin, CDTrans: Cross-domain Transformer for Unsupervised Domain Adaptation, 2022, arXiv:2109.06165.

[82] B. Xie, S. Li, M. Li, C.H. Liu, G. Huang, G. Wang, SePiCo: Semantic Guided Pixel Contrast for Domain Adaptive Semantic Segmentation, 2022, arXiv:2204.08808.

[83] L. Hoyer, D. Dai, L. Van Gool, HRDA: Context-Aware High-Resolution Domain-Adaptive Semantic Segmentation, in European Conference on Computer Vision (ECCV), 2022, arXiv:2204.13132.

[84] P. Han, G. Zhao, A review of edge-based 3D tracking of rigid objects, Virtual Reality & Intelligent Hardware, 2019, 1(6), pp. 580–596, doi:10.1016/j.vrih.2019.10.001.

[85] B. Wang, F. Zhong, and X. Qin, Robust edge-based 3D object tracking with direction-based pose validation, Multimed Tools Appl, 2019, 78, pp. 12307–12331, doi: 10.1007/s11042-018-6727-5.

[86] OctoPrint: an open source 3D printer controller application. Available: https://octoprint.org (accessed January 10, 2024).

Chapter 5

5 Tool change Reduction in Multi-color 3-D Printing

This chapter[4] is adapted from the “Tool change Reduction for Multi-color Fused Filament Fabrication Through Interlayer Tool Clustering Implemented in PrusaSlicer” with minor modifications to the version submitted in Rapid Prototyping Journal (Emerald Publishing), 2024.

5.1 Abstract

The most popular type of 3-D printing globally fuses plastic filament into a 3-D printed object. Historically, this has been done with only a single polymer. Advanced 3-D printer manufacturers now allow multiple materials and/or colors to be part of a single print. Presently in multi-material fused filament-based 3-D printing, significant amounts of waste material is produced. Each time a change from one material to another occurs, waste is produced through nozzle priming and/or purging. 3-D printing software (slicers) that prepare the G-code for multi-material 3-D printing typically change the material on each layer meaning that every layer, waste is generated often resulting in wipe towers with greater mass than the 3-D printed target object. An alternative fabrication approach based on interlayer tool clustering (ITC) is presented here for the first time, which is compatible with any commercial 3-D printer without the need for hardware modifications. The theoretical time, mass and energy savings are calculated and validated with a series of experiments to evaluate the proposed algorithm qualitatively and quantitatively. The results show the novel ITC method can significantly increase the efficiency of multi-material printing, with an average 1.7-fold reduction in material used, and an average 1.4-fold reduction in both time and 3-D printing energy use. In addition, this approach reduces the likelihood of technical failures in the manufacturing of the entire part by reducing the

---

[4] A version of this chapter has accepted in Rapid Prototyping Journal. A. Petsiuk, B. Bloch, D. Vogt, M. Debora, J.M. Pearce (in press), Tool change Reduction for Multi-color Fused Filament Fabrication Through Interlayer Tool Clustering Implemented in PrusaSlicer. Rapid Prototyping Journal, 2024. A preprint is available at SSRN, http://dx.doi.org/10.2139/ssrn.4655383.

number of tool changes, or material transitions, on average by 2.4 times. These savings all support distributed recycling and additive manufacturing, which has both environmental and economic benefits, and increasing the number of colors in a 3-D print increases the savings and benefits.

5.2 Introduction

Additive manufacturing (AM), or 3-D printing, stands as one of the foundational elements within the framework of Industry 4.0 [1,2]. Since the original inventions in 1970–80s [3,4], the primary emphasis of the technology has revolved around single-material rapid prototyping, a generation of a draft part representation to find the optimal shape before its final release. Subsequent advances in AM technology and materials development, as well as market demand for enhanced functionality and high geometric complexity, however, have shifted the focus towards the direct production of final products and the simultaneous use of multiple materials. Modern additive manufacturing is firmly established in such diverse areas as education [5], medicine [6], fitness [7], fashion [8], food industry [9], low-cost customized scientific equipment [10], architecture and construction industry [11], optics [12,13], electronics and robotics [14], automotive [15], aerospace [16], and completely new areas [17].

Multi-material 3-D printing greatly expands the horizons of manufacturing, providing innovative design opportunities and enhancing the quality of products through the optimal combination of properties, which is often impossible to achieve with traditional methods [18]. The main fabrication materials are polymers, metals, ceramics, and biological substances. The integration of electrically functional elements and shape memory polymers, in turn, marked the beginning of 3-D printing of smart devices and 4-D printing, respectively [19].

According to the ISO/ASTM 529000-21 standard [20], AM can be categorized based on fabrication principles, feedstock types, energy sources, and build volumes. Each of the various processing techniques has its own advantages and challenges [21,22]. Fused filament fabrication (FFF), however, is one of the most popular technologies and offers great opportunities for general users to become producing consumers (prosumers) [23–25] due to its simplicity and low cost [26,27]. Due to the increasing variety of filament materials with a wide range of physical properties, interest in multi-material and multi-color AM is constantly growing [21].

Multi-color FFF 3-D printing is a great way to create a wide range of consumer products including artistic and ornamental objects, such as figurines, avatars, reproductions of historical objects or buildings or famous characters. It also allows creating at a low cost and with a much-simplified process (manual part painting is removed from the steps) proof of concepts and industrial prototypes. Many 3-D printers on the market positioned as multi-color (FlashForge Creator [28], Sovol SV04 [29], Geeetech A30T [30], and others) have several extruders for individual materials, which limits the number of colors that can be used and has drawbacks like oozing, nozzle clogging and trajectory alignment, and others [21]. Prusa MMU [31] and Mosaic Palette [32] modules, in turn, allow to convert most of the commercial 3-D printers into 5-to-8-color fabrication stations utilizing a single-nozzle architecture. This opens a possibility for a general audience to create multi-color models for education, fashion, architectural concepts and so forth.

Most of the modern slicing software (slicers), however, utilize the traditional flat horizontal layer-based approach remaining from the single-material concept, where in the process of preparing a model, a part is sliced into multiple flat horizontal polygons. Traditional FFF 3-D printing is a fabrication technique that involves dividing a 3-D model into successive layers and sequentially adding (extruding) each layer to construct the final object. The printer nozzle moves along a flat plane while depositing a polymer to form each layer. Only once a layer is finished does the printer shift in the Z direction to commence the subsequent layer, and this process is repeated iteratively. Each time a change happens from one material to another, waste is produced through nozzle priming and/or purging. This disadvantage creates a large amount of waste and requires a lot of printing time and wasted energy [33]. Software that prepares the G-code for multi-material fabrication typically changes the material on each layer meaning that every layer, material waste is generated unlike in single material printing. For 3-D printing to maintain its environmentally-friendly reputation [34,35] and contribute to a low-waste circular economy [36,37], it must solve this multi-material waste problem.

Such popular free and open source slicers as PrusaSlicer [38], SuperSlicer [39], Bambu Studio [40], OrcaSlicer [41] and others, are based on Slic3r toolbox [42], which has this fundamental disadvantage described above. Increasing the number of colors will only increase the problem.

In this study, an enhanced technique known as interlayer tool clustering (ITC) method is introduced and applied to the open source PrusaSlicer [38]. The paper details a novel ITC upgrade [43] with a focus on optimizing multi-layer slicing to achieve substantial reductions in time and material waste. In addition, this approach reduces the chance of technical errors by minimizing the number of tool changes (material transitions). The presented approach is validated using a diverse set of models from areas such as biomedical education, architectural and landscape design, and product development. Material consumption, energy usage and print times are quantified and compared to the conventional layer-by-layer technique. The implications of widespread adoption of this technique are discussed in the context of distributed manufacturing and recycling.

Similar changes may also be applied to other slicing software based on the core “libslic3r” library: newer versions of PrusaSlicer, Bambu Studio, OrcaSlicer, SuperSlicer, etc., with appropriate individual modifications. Given the wide support for native and third-party 3-D printer profiles, these slicers can be used to prepare models for single-nozzle printers from vendors such as Prusa, BambuLab, Creality, Lulzbot, Ultimaker, Voron, and others. This significantly expands the capabilities of the presented ITC method.

5.3 Background: Traditional Slicing Approach and Related Works

A multi-material model, as a rule, consists of several stereolithography (STL) files, each corresponding to a specific color/material preset in the slicing program. The user can also manually recolor individual areas of the entire model using internal painting tools. After slicing, each flat vertical slice of the model represents a set of regions, each corresponding to a specific material.

In multiple-input, single-extruder prints with technologies like Mosaic Palette [32] and Prusa MMU [31], current slicing demands that 1) the transition between inputs must be purged externally from the desired printed part, to ensure a clean separation of different colors or materials, 2) toolpaths within the layer are grouped by input to minimize the number of transitions needed per layer, by avoiding switching back and forth more than necessary, and 3) because of the monotonically-increasing order of the heights in the toolpath, the input being printed must change back and forth on essentially a per-layer basis, and all of these transitions must be purged. This creates a substantial amount of waste material for the purge (i.e., wipe tower [33]) as well as the time and energy for manufacturing. This traditional fabrication approach is presented in Figure 5.1. The nozzle sequentially fills the inner and outer regions of the layer with material and moves to a new layer only after the completion of the current one.

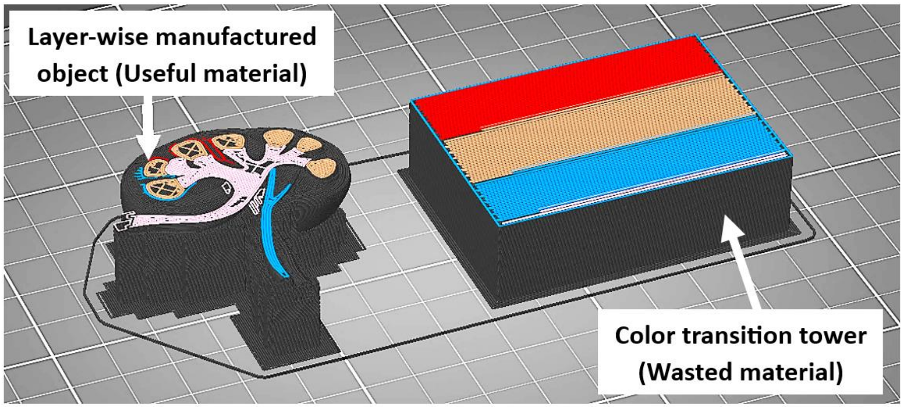
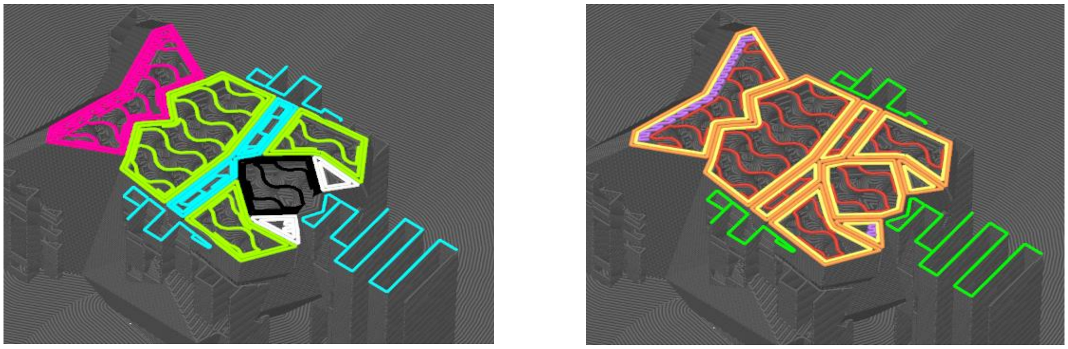

Layer-wise manufactured object (Useful material)
Color transition tower (Wasted material)

Figure 5.1: Traditional slicing approach. The nozzle sequentially fills the inner and outer regions of the layer (kidney model [44]) with material and moves to a new layer only after the completion of the current one. If the layer consists of several colored regions – the nozzle performs additional material extrusion (purging) for each transition between colors. This removes the remaining melted plastic in the nozzle and ensures sharp color transitions and stable filament flow.

It should also be noted that there are “purge into infill” and “purge into object” options, where the transition material section is used to fabricate an infill or another object, respectively. This approach, however, can compromise the material purity and aesthetic properties of the object, as well as increase the idle traveling time of the nozzle without extrusion due to changes in manufacturing order.

During slicing, each extruded segment (G-code line) for each fabrication layer has its own characteristic (Figure 5.2), such as type (external and internal perimeters, internal, solid, and bridge infills, support material and interface, skirt/brim, wipe tower, and others) and tool (material, color). Each line type can be assigned a specific print speed and temperature mode. This allows varying fabrication modes depending on the material used.



a) **b)**

Figure 5.2: Color coding of different types of extruded segments. During the slicing process, each extruded segment is assigned a series of specific properties that determine line length and type, material color, print temperature, fan speed, etc.: *a)* – sliced multi-color model with support, where each color represents a different material, *b)* – color coded extruded segments (green – support, red – internal infill, yellow – internal perimeter, orange – external perimeter, and purple – solid infill).

The extruded segments, however, are grouped according to their affiliation with a specific material within the layer, and it is not currently supported to assign individual materials to different types of segments (perimeter, infill, bridge, etc.). Only supports and support interfaces can belong to a separate material when manufacturing parts using soluble supports.

There are several approaches in the literature that attempt to solve the toolpath optimization problem to increase efficiency and quality of 3-D printing, as well as to reduce manufacturing time and minimize the number of nozzle transitions. Kaplan et al. [45]

presented a work on modifying the nozzle and creating a toolpath optimization algorithm that allows a vertical shift between layers during printing and maximizes the number of successively printed vertical layers for a single material. This improves print quality while reducing production time and overall nozzle travel. Wang et al. [46] presented an adaptive slicing algorithm that minimizes fabrication time and preserves salient features of a model. Ahlers et al. [47] presented a Slic3r-based curved layer fused deposition modeling implementation[48], which reduces the “staircase effect” of curved surfaces and shells. Similar works have been proposed by Etienne et al. [49] and Huang & Singamneni [50]. Molloy and Miller [51], Mueller et al. [52], and Gleadall [53] implemented alternative fabrication approaches utilizing direct freeform three-dimensional movement. Alexa et al. [54] introduced an adaptive discrete slicing algorithm that generates optimal slice heights and minimizes volumetric errors. Hergel et al. [55] developed a method that eliminates travel moves by generating continuous deposition paths across all of model layers. Zhong et al. [56] introduced a toolpath optimization framework for continuous extrusion of surface models. Liu et al. [57] proposed a minimization of the number of nozzle transitions within each layer using a traveling-salesman-problem approach. Similar approaches were presented by Fok et al. [58] and Aguilar-Duque et al. [59], utilizing the ant colony optimization technique and a genetic algorithm, respectively.

The developed method preserves the original layer-based slicing approach, but minimizes the material transitions by maximizing the batch of sequentially fabricated vertical layers. The developed ITC, or layer batching, is a variation of sequential printing where multiple layers of the same input can be printed successively, the input is switched and a transition is performed, multiple layers of the new input are printed successively, and so on. In multiple-input, single extruder prints, ITC presents a major opportunity for reducing the amount of material needed for all of the input transitions in the print, as well as the amount of time spent performing such transitions. The method described in this article allows printing the sliced layers out of order, where a part of a lower layer can be printed after a higher layer has already been fabricated.

The proposed algorithm can be applied to most of the commercial printers for the general audience. It allows using any stock nozzle without hardware modifications and slicing 3-

D models in a regular way using an open source upgrade for the widely used slicing library. This can improve the speed, material consumption, and energy usage of the resulting printed parts.

5.4 Theory and Application: Proposed Tool Clustering

The developed fabrication method uses the original slicing approach implemented by the “libslic3r” library [60] integrated in PrusaSlicer. The novelty in the approach is the interlayer clustering algorithm for reordering sliced planar regions of the same material/color to decrease the number of tool changes (material transitions) during manufacturing. Accounting for parts of the model that have already been printed and avoiding collisions are major challenges, as each completed material/color area makes the areas immediately below it no longer accessible.

Consider the example in Figure 5.3. The part to be produced contains three colors (color regions R0–R2) and consists of four layers (L0–L3). The traditional default printing order starts from a specific region and consists of sequentially bypassing all of the colored areas from layer to layer (Figure 5.3, c).

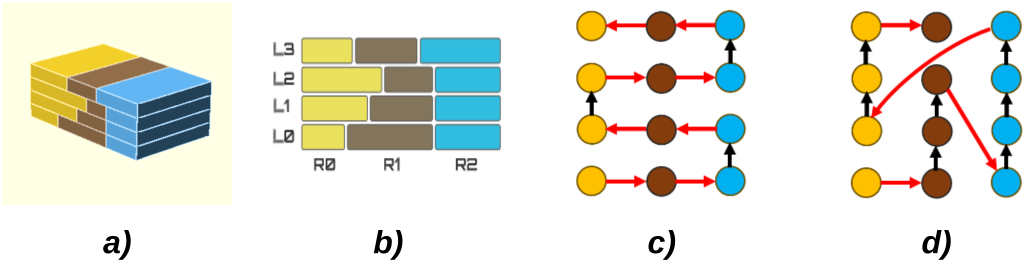


Figure 5.3: Fabrication order of a multi-color part: *a)* – source model, consisting of three colors (color regions R0–R2) and four layers (L0–L3), *b)* – schematic side view of the source model, *c)* – traditional (default) fabrication order with 8 tool changes (marked as red arrows), *d)* – fabrication order produced by the proposed tool clustering method (4 tool changes, marked as red arrows).

The proposed algorithm analyzes the possibility of printing the next region of the same color, moving through the layers. The main criteria for the range of nozzle movement are the maximum permissible height and the presence of intersections with other colored regions, making it impossible to further jump down (Figure 5.3, d). In this way, the overall structure of the manufactured part is maintained, where each material region is fabricated only once without printability violations, but the number of tool changes (marked as red arrows) is reduced. A step-by-step procedure for the same source model (Figure 5.3) is illustrated in Figure 5.4.

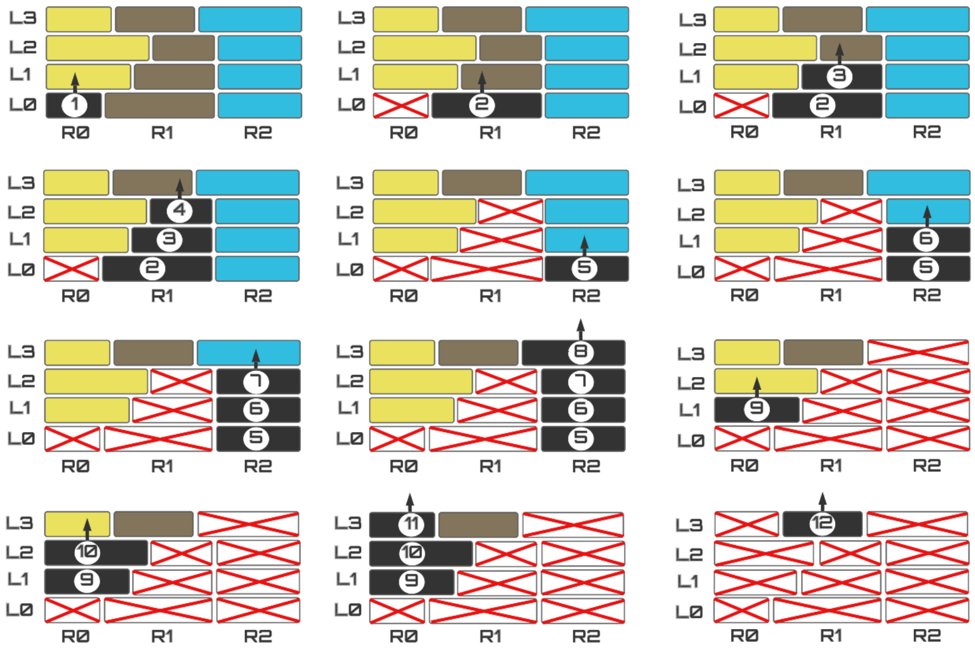


Figure 5.4: Tool clustering algorithm. The developed algorithm in action on the example of a simple part in the shape of a rectangular parallelepiped. The part is shown in the side view and each of the four layers (L0...L3) consists of three different colored regions (R0...R2). Starting from the first region, the program analyzes the possibility of printing a candidate region (the top layer of the same color as the previously selected region) according to two criteria: the allowable safe height and the presence of intersections in the unfinished underlying regions of other colors. If it is possible to print a candidate region – the region is added to a linked list (print map) and marked as processed (red crosshair). If printing is not possible, the candidate is discarded, and the pointer moves to the next unprocessed region.

As a result of the analysis, the initial printing map in the form of a linked list, based on the default slicing parameters, is transformed into a tool clustered batched map (Figure

5.5). As seen from the figure above (e.g., steps 4 to 5 and 8 to 9), the hot-end traversal may lead to inevitable damage to previously fabricated areas, described in detail further in the article. In many cases, however, this may have little effect on the cosmetic characteristics of the part being manufactured and is compensated for by a significant reduction in the resources used.

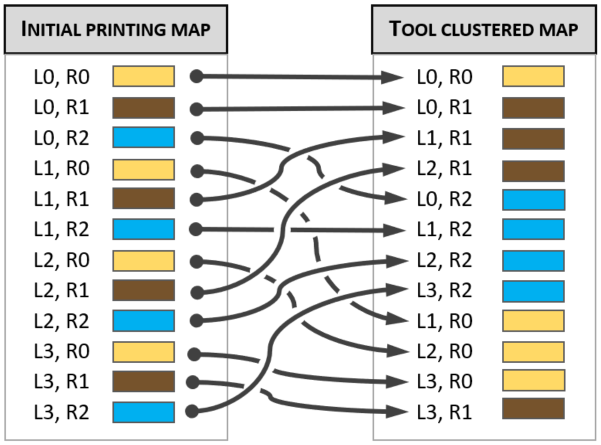


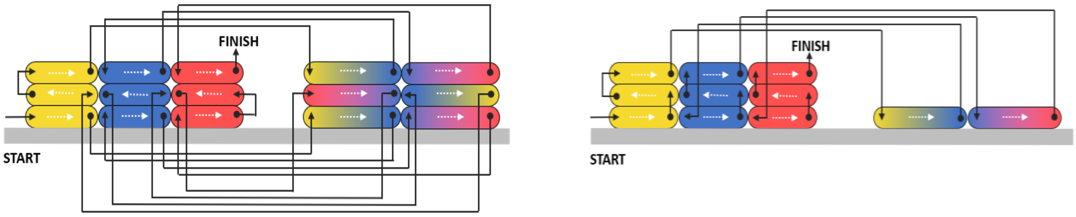
Figure 5.5: Converting the initial printing map into a clustered sequence. As a result of the analysis, the initial printing map in the form of a linked list, based on the default slicing parameters, is transformed into a tool clustered batched map.

The given algorithm is easily scalable into a large number of colors/materials and inherently supports the variable layer height fabrication according to the source PrusaSlicer approach. The open source modification is licensed under AGPL-3.0 [61] and is available on GitHub [43].

5.4.1 Material, Time, and Energy Savings

Given the huge variety of possible 3-D models and color/material distributions throughout their heights, it is hard to derive precise equations that describe all of the possible savings based on the applied tool clustering properties. Therefore, below is an analysis of the generic case, where each layer contains all of the materials used in the model, and a generalized equation is derived that allows estimating the potential savings in material, time, and energy.

Figure 5.6 depicts an example of a printing process of a three-color and three-layers part (side view) with object layers, wiping layers, and travel moves.



a) b)

Figure 5.6: Conventional *(a)* vs clustered *(b)* fabrication approaches. A cross section of a rectangular piece consisting of three layers, each containing three different materials (yellow, red, and blue colors) is shown. Gradient regions represent wiping layers with material transitions. The black arrows indicate travel nozzle moves without extruding material, while the white arrows within the layers illustrate nozzle movements with material extrusion.

The total time *T*, needed for the whole part fabrication, can be represented in the following way:

$$T = T_L + T_T + T_W + T_C + T_M = NR\frac{P_L}{S_L} + \frac{N(R+1)}{B} \cdot \frac{P_T}{S_T} + \frac{N(R-1)}{B} \cdot \frac{P_W}{S_W} + \frac{N(R-1)}{B} \cdot t_C + \frac{N(R-1)}{B} \cdot t_M \tag{5-1}$$

where T_L is the total time for all of the object layers fabrication, T_T is the total traveling time for the whole object fabrication, T_W is the total time required for wiping tower fabrication, T_C is the total time for all of the tool changes, T_M is the total waiting time for material properties (temperature, fan speed, etc.) adjustments for all of the material changes (equals to zero in case of using different colors of the same material), N is the number of layers, R is the number of material/color regions in the whole part, B is the batch size in number of layers, P_L is the average extruded path length within each layer (useful material), P_T is the average travel path length between object and wipe tower, P_W is the wiping path length (wasted material) for a single tool change, S_L is the average object layer printing speed, S_T is the travel speed, S_W is the wiping speed, t_C is the waiting time for a single tool change, t_M is the waiting time for material properties (temperature, fan speed, etc.)

adjustments for a single material change ($t_M = 0$ for a multi-color print with the same material).

All of the time parameters are measured in minutes, lengths – in meters, speeds – in m/min, $N(R-1)/B$ represents the number of tool changes during the fabrication of all of a part layers, $N(R+1)/B$ represents the number of travels to and from wipe tower during the part fabrication.

As it can be seen from the equation above, the useful time consists of T_L value only, all of the rest of the T-parameters are considered as wasted time and must be reduced. Therefore, the time loss equation can be expressed as follows:

$$Loss_T^{\%} = \frac{T_T+T_W+T_C+T_M}{T_L} = \frac{S_L}{BRP_L}\left[\frac{(R+1)P_T}{S_T} + \frac{(R-1)P_W}{S_W} + (R-1)t_C + (R-1)t_M\right] \cdot 100\%, R \geq 2 \qquad (5\text{-}2)$$

A similar approach is also applied to energy (5-3) and material (5-4) consumption.

$$\begin{cases} E = E_L + E_T + E_w + E_C + E_M = N \cdot e_L + \frac{N(R+1)}{B} \cdot e_T + \frac{N(R-1)}{B} \cdot e_W + \frac{N(R-1)}{B} \cdot e_C + \frac{N(R-1)}{B} \cdot e_M \\ Loss_E^{\%} = \frac{E_T+E_W+E_C+E_M}{E_L} = \frac{(R+1)e_T+(R-1)e_W+(R-1)e_C+(R-1)e_M}{B \cdot e_L} \cdot 100\%, R \geq 2 \end{cases} \qquad (5\text{-}3)$$

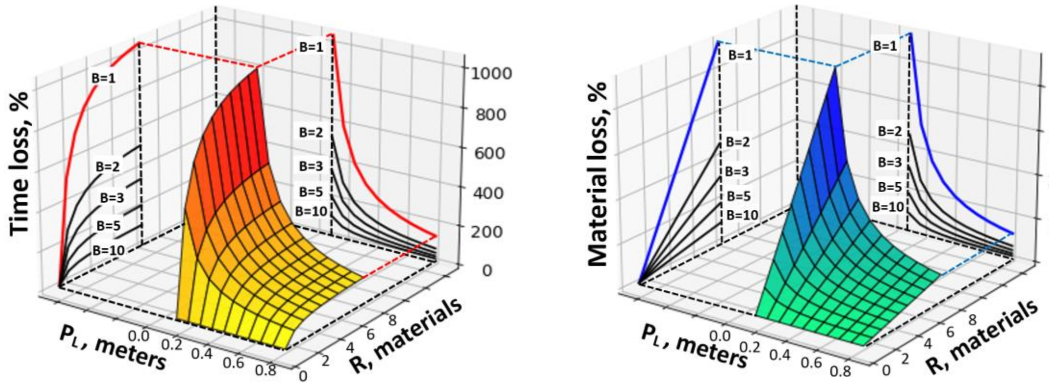
where E is the total amount of energy, required for part fabrication, e_L is the average amount of energy required for a single object layer fabrication, e_T is the energy for a single travel move, e_W is the energy for a single region fabrication of a transition tower required for a tool change, e_C is a single tool change energy required for a tool change device (Mosaic Palette, Prusa MMU, or similar), e_M is the nozzle and bed heating energy, respectively, wasted during the transition period for switching different materials ($e_M = 0$ for a multi-color print with the same material). All of the the energy parameters have kWh units.

$$\begin{cases} M = NP_L + \frac{N(R-1)}{B} P_W \\ Loss_M^{\%} = \frac{N(R-1)P_W}{NBP_L} = \frac{(R-1)P_W}{BP_L} \cdot 100\%, \mathrm{R} \geq 1 \end{cases} \qquad (5\text{-}4)$$

where M is the total amount of filament material in meters required for a whole part fabrication. The remaining parameters are described above.

Since the exact calculation of energy parameters is quite complex and can be rather inaccurate, energy losses can be further considered proportional to the time losses ($Loss_E^{\%} \propto Loss_T^{\%}$), since the printer consumes electrical power during operation, as well as when moving its parts and heating the bed platform, chamber and nozzle.

Figure 5.7 illustrates the graphical representation of the time (5-2) and material (5-4) loss functions. The colored surfaces are the default losses without layer batching depending on number of materials used and the fabrication part size, while the lines projected onto the orthogonal planes represent the losses after applying layer batching ($B = 2 \ldots 10$). The parameters for the plots were used as follows: $t_C = 0.3\ (min)$ is the time to change the tool, $S_L = 3.6\ (m/min)$ is the average printing speed, $S_T = 4.2\ (m/min)$ is the average travel speed, $S_W = 3.6\ (m/min)$ is the average wiping speed, $P_W = 0.066\ (m)$ is the average wiping length, $P_T = 0.05\ (m)$ is the average travel length.



a) **b)**

Figure 5.7: Time *(a)* and material *(b)* losses. The colored surfaces are the default losses without layer batching depending on number of materials used and the fabrication part size, while the lines projected onto the orthogonal planes represent the losses after applying layer batching ($\boldsymbol{B = 2 \ldots 10}$).

Also worth noting is the reduction in the overall manufacturing failure probability due to frequent tool changes (5-5).

$$Q = 1 - p^{\frac{N(R-1)}{B}} \tag{5-5}$$

where Q is the overall manufacturing failure probability, p is the probability of a single successful tool change, and $N(R-1)/B$ is the number of tool changes.

5.4.2 Implementation: PrusaSlicer Upgrade

PrusaSlicer is now one of the most popular open source slicers for the general audience around the world and is the successor of Slic3r developed by Alessandro Ranellucci in 2011 [62], which, in turn, replaced SkeinForge [63], a Python-based toolchain developed by Enrique Perez that converts 3-D models into G-code instructions for self-replicating rapid prototypers (RepRap) [64–66] class 3-D printers.

PrusaSlicer consists of several libraries, the main one being “libslic3r” [60], which processes geometric data and performs slicing directly. A high-level data flow is depicted in Figure 5.8. The process begins by loading a 3-D model and generating a printing object consisting of a set of resulting object and support layers. The data are further used to create a material transition tower and is converted into a set of corresponding layer-by-layer instructions for moving the extrusion tool. The fabrication speeds and cooling parameters for each material region, as well as region boundaries and the last position of the tool on the previous layer are considered.

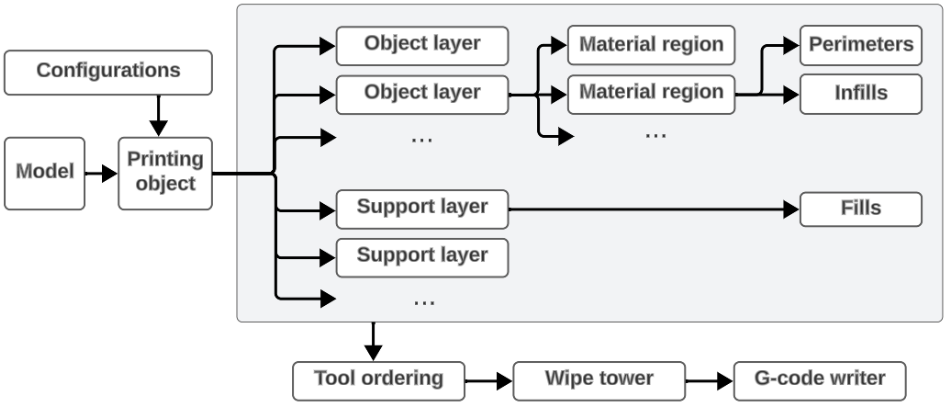
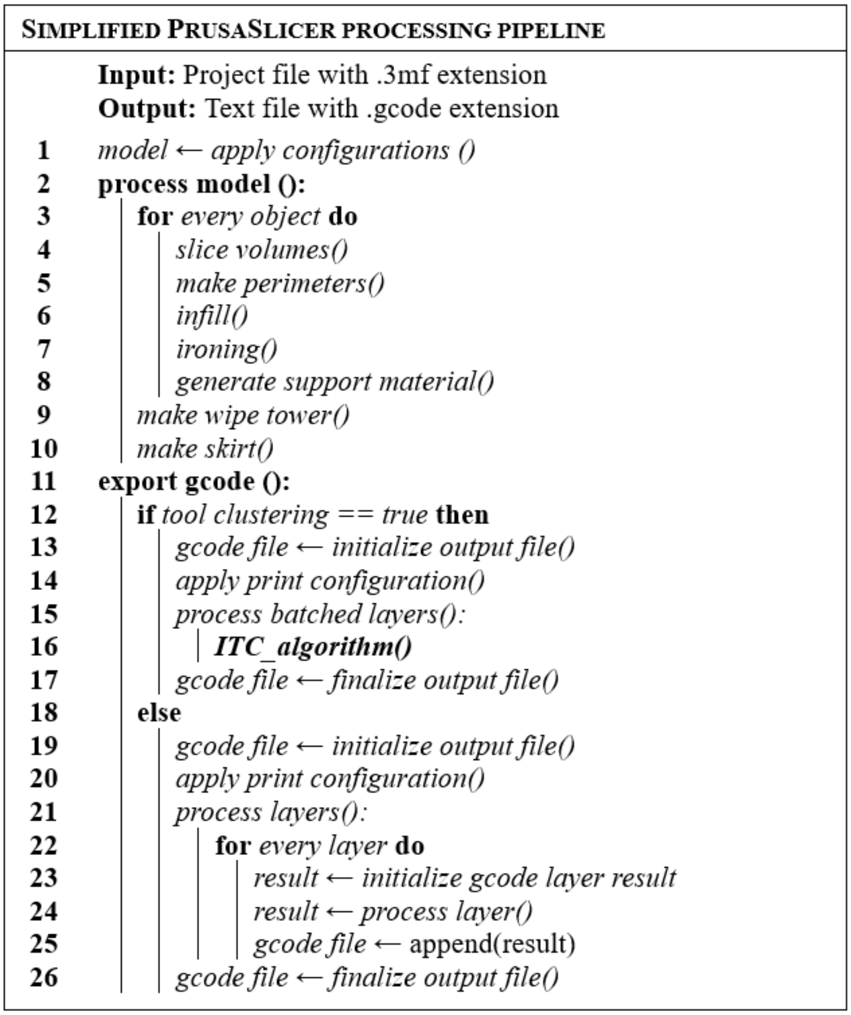


Figure 5.8: Internal data flow of the slicing process in the PrusaSlicer “libslic3r” library. The process begins by loading a 3-D model and generating a printing object consisting of a set of resulting object and support layers. The data are further used to create a material transition tower and is converted into a set of corresponding layer-by-layer instructions for moving the extrusion tool.

By manipulating the order of the material regions, it is possible to disrupt the conventional layered mode of object fabrication and change the order of the material regions by stepping over the layers. The proposed ITC algorithm is based on this feature and is injected into the main slicing pipeline as shown in Figure 5.9.

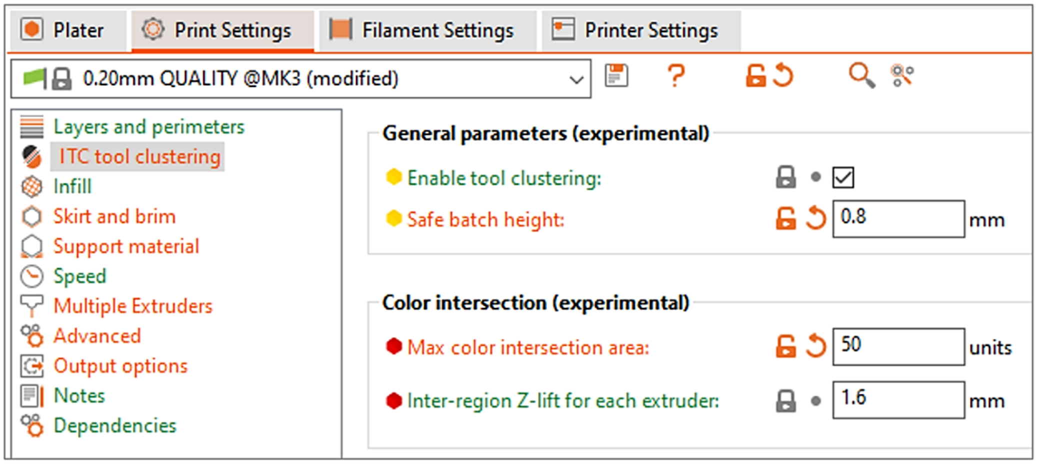


SIMPLIFIED PRUSASLICER PROCESSING PIPELINE

```
Input: Project file with .3mf extension
Output: Text file with .gcode extension
model ← apply configurations ()
process model ():
    for every object do
        slice volumes()
        make perimeters()
        infill()
        ironing()
        generate support material()
    make wipe tower()
    make skirt()
export gcode ():
    if tool clustering == true then
        gcode file ← initialize output file()
        apply print configuration()
        process batched layers():
            ITC_algorithm()
        gcode file ← finalize output file()
    else
        gcode file ← initialize output file()
        apply print configuration()
        process layers():
            for every layer do
                result ← initialize gcode layer result
                result ← process layer()
                gcode file ← append(result)
        gcode file ← finalize output file()
```

Figure 5.9: High-level slicing pipeline of the developed PrusaSlicer upgrade. The main functionality of the developed ITC algorithm is integrated in the G-code generation process.

The developed ITC algorithm is implemented in PrusaSlicer version 2.6.0, mainly affecting the “libslic3r” library. From a user's perspective, the above modifications represent an additional feature tab in the graphical user interface (GUI), where a maker can specify the desired layer batching parameters (Figure 5.10). The main control variables are the allowable height of a stack of layers processed within one material transition ("Safe batch height") and the maximum acceptable intersection with underlying regions of other colors ("Max color intersection area"). The "Clipper" library [67] integrated into "libslic3r" is used to calculate the intersections of sliced regions. The "Interregional Z-lift" sets the extruder to rise during retraction, which occurs throughout every material change. This avoids intersections with already fabricated areas.



General parameters (experimental)

Enable tool clustering: ☑

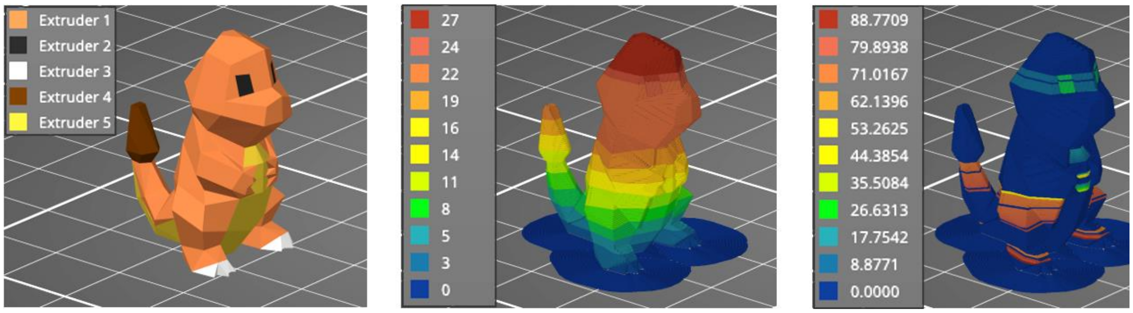
Safe batch height: 0.8 mm

Color intersection (experimental)

Max color intersection area: 50 units

Inter-region Z-lift for each extruder: 1.6 mm

Figure 5.10: Tool clustering control panel in the PrusaSlicer GUI. The main control variables are the allowable height of a stack of layers processed within one material transition and the maximum acceptable intersection with underlying regions of other colors.



a) **b)** **c)**

Figure 5.11: Colormap visualization of the clustered sliced model in the PrusaSlicer GUI: *a)* – five-color source model, *b)* – sliced material regions grouped into layer clusters, *c)* – critical region intersections.

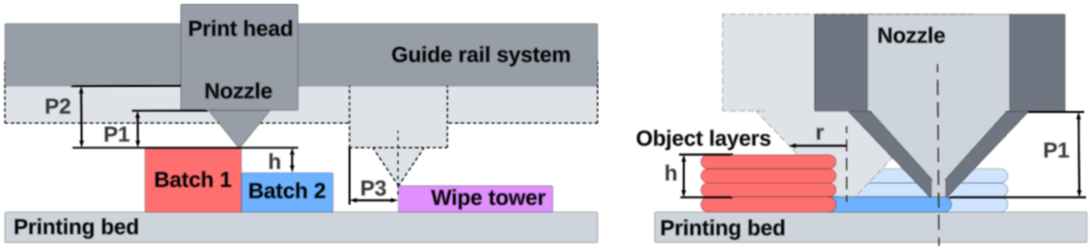
Figure 5.11 shows a colormap visualization of the clustered layer batches and critical region intersections implemented in the PrusaSlicer GUI for a sample multi-material source model [68]. This feature allows more efficient process debugging and the tool clustering potential analysis of source models with varying geometric complexity.

Tool clustering imposes certain limitations on the manufacturing process, since most available 3-D printers are designed for conventional single-layer slicing. The typical FFF 3-D printing nozzle system and its clearance are shown in Figure 5.12.

Regardless of the nozzle design, the current approach results in some deformation of the previous layers. The maximum damage volume, V_D (in mm^3), equals the volume of the intersection with the nozzle that can be represented as a cone with volume (equation 5-6). These deformations, however, are not critical in most cases.

$$V_D = \pi r^2 \frac{h}{3} \tag{5-6}$$

Where h is the batch height in mm, and r is the nozzle radius (in mm) at a given height h.



a) **b)**

Figure 5.12: Nozzle system and its clearance. The extrusion system of each printer has a certain gap and filled space in the immediate vicinity of the nozzle, which prevents free movement in non-planar printing *(a)*. When printing batched layers, subsequent surrounding regions are damaged in most cases. The maximum damage volume *(b)* is equal to the volume of the intersection with the nozzle, which can be represented in equation (5-6). Here, the parameters P1, P2 and P3 define the physical constraints imposed by the dimensions of the print head system, $\boldsymbol{h}$ is the batch height, and $\boldsymbol{r}$ is the nozzle radius at a given height $\boldsymbol{h}$.

An important part of a good tool clustering implementation is to ensure that the print head will not collide with paths that have already been printed. The ideal implementation is one where, given reasonably accurate information about the printer's print head geometry, the user does not need to perform any manual work to ensure neither the 3-D prints nor the printer is damaged. Automatic avoidance checks when planning the order of these paths should consider avoidance of paths in the XY plane that will have already been printed (i.e., not running into them from the side) as well as the fact that the nozzle itself needs to be able to reach the points of a path from above (i.e., the nozzle geometry fits sufficiently into any space where a path is being printed, and has direct access from above). While certain optimizations can be performed automatically before slicing (such as adjusting the layout of objects on the build plate to increase spacing as needed), most parameters affecting the ability to batch layers should be considered fixed at the time of slicing (e.g., build plate layout, layer height) and should not be automatically adjusted for layer batching. The paths that are generated in the XY plane should be the same as without any layer batching; this algorithm is simply concerned with the order in which these paths are printed. In the proposed algorithm, the goal is to batch as many layers as possible, up to the point where it is physically not possible to include more layers because the inclusion of another layer would mean that a different input would no longer reach some path that has not yet been printed.

Even with automatic collision avoidance, the ideal solution should allow for a configurable maximum Z differential (that is, the maximum height difference between the current layer being printed and the lowest layer that still needs to be printed). There are a few reasons such a configuration is beneficial or even necessary:

1) If too much time passes between a batch of layers with a given input and the next batch with that input, the hot material extruded along the paths of this input may have cooled sufficiently that inter-layer bonding is adversely affected [69], reducing the print quality and/or strength of the finished part, and thus preventing the use of the feature in practical settings like production of end-user products (where surface finish is a major factor) or engineering-grade parts (where dimension accuracy and strength are major factors).

2) Many common printer gantries will have an inherent Z differential limit because at a certain point, moving back down will cause the already-printed paths to collide with the printer's gantry itself even if print head avoidance is calculated correctly.

Even sequential printing as implemented in most slicers today does not have such a configuration, meaning the burden for ensuring printability is once again put on the user. The ideal implementation would instead take this configuration and account for it automatically and reliably.

5.5 Experimental Methods

For all of the experiments on the interlayer tool clustering proposed here a Prusa MK3S FFF RepRap-class 3-D printer was used with an MMU2S [70] filament changing system. The time consumption and the amount of material (filament length) were quantified based on PrusaSlicer internal calculations, energy was monitored with a digital power meter EZ P4460 (±0.005kWh).

In the experiments carried out, a model of a multi-color rectangular parallelepiped with dimensions of 48×24 mm was used. The height of the model was gradually increased from 10 to 20 mm to analyze material, time and energy consumption depending on the number of fabricated layers (Figure 5.13). The same slicing parameters were chosen for each part, namely 2 perimeters, 2 top and bottom layers, 0.2 mm layer height with 0.4 mm nozzle diameter and 30% rectilinear infill. The temperature parameters are 205°C for the nozzle and 60°C for the bed. Polylactic acid (PLA) was used for all of the experiments.

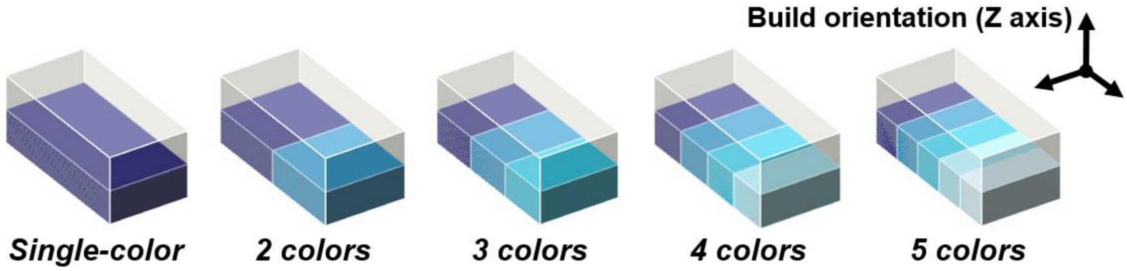


Figure 5.13: Multi-color experimental model. The height of the model was gradually increased from 10 mm (solid color parts) to 20 mm (semitransparent gray volumes) to

analyze material, time and energy consumption depending on the number of fabricated layers.

In addition to the experimental parallelepipeds, a comparative slicing of a random set of models [71] was also carried out. All of the models were sliced with the same parameters: material – PLA filament of 1.75 mm diameter, 2 perimeters, 2 top and bottom layers, 0.2 mm layer height with 0.4 mm nozzle diameter and 15% gyroid infill, default wiping volume (15 mm^3) for each color transition, interregional lift Z=0.4 mm for the default slicing and Z=1.2 mm for batching (1.00 mm batch height). The temperature parameters are 205°C for the nozzle and 60°C for the bed. When analyzing material consumption, the contribution of the object layer extrusion was calculated in comparison to the extrusion amount of the wiping tower, without considering the individual contribution of materials of different colors.

5.6 Results and Discussion

Figure 5.14 shows the results of the time and material consumption analysis for the multi-color rectangular parallelepiped fabrication.

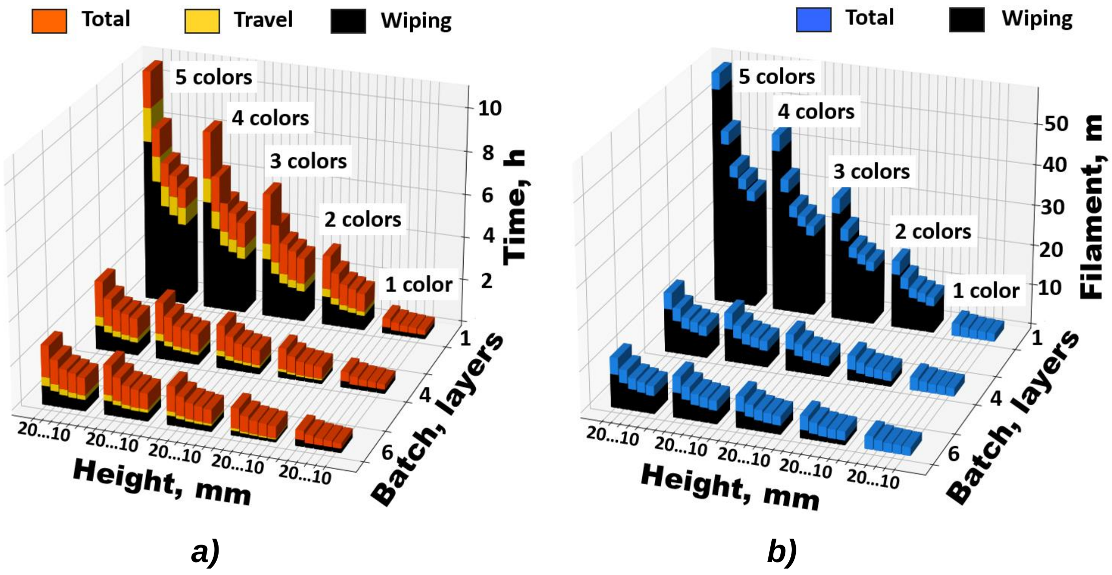


Figure 5.14: Analysis of time *(a)* and material *(b)* consumption during the manufacturing of the experimental models depending on the layer batch size.

During the experiments, the average energy consumption was 0.11 kWh per hour. Figure 5.15 shows the dependence of energy consumption, fabrication time and the number of tool changes on the height of the parts manufactured using variable layer batch size. The shaded areas demonstrate the degree of reduction in resource consumption when using the ITC fabrication approach.

As can be seen from the graphs, printing models in batches of 4 layers leads to a reduction in material consumption by three or more times due to a significant reduction in the wipe tower caused by the number of tool changes, or transitions from one material to another. The total fabrication time can be reduced by 3 or more times by reducing the travel movements and time required to form the wipe tower.

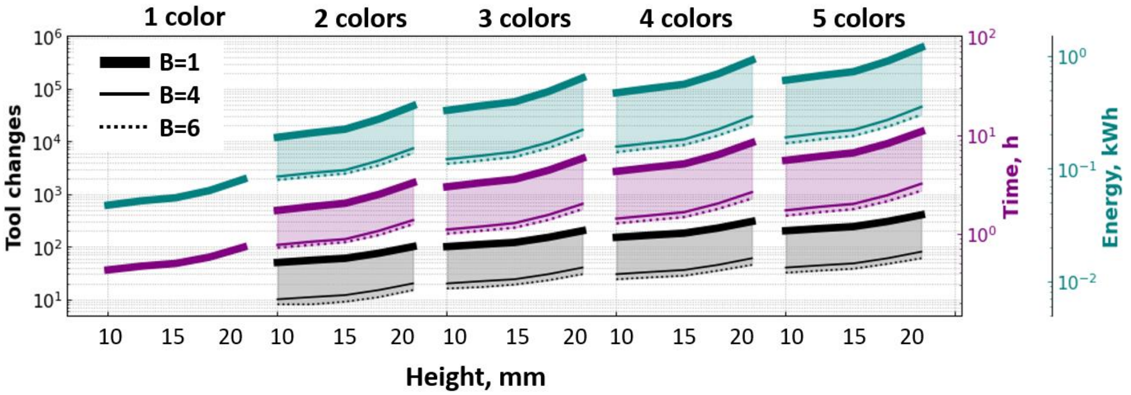
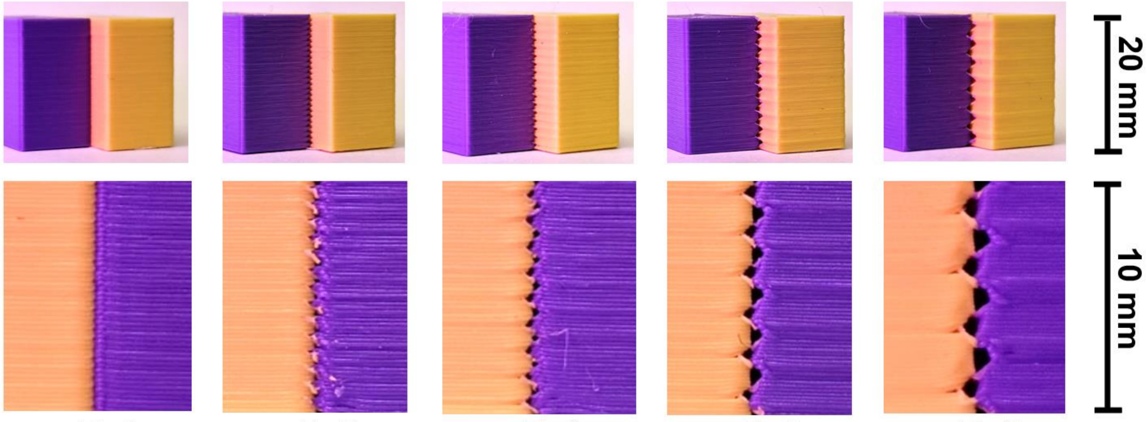


Figure 5.15: Dependence of energy consumption, fabrication time, and number of tool changes on the height of the parts manufactured using variable layer batch size. Batch size B=1 corresponds to conventional layer-wise manufacturing, and batch sizes B=4 and B=6 correspond to the developed ITC method with grouping of fabricated layers of the same material into batches of 4 and 6 layers, respectively.

Reducing production costs, however, is reflected in the quality of the resulting products. An increase in the number of layers processed in one batch leads to the appearance of a zigzag gap at the material boundaries, as well as a wave pattern that reflects the contours of the nozzle in the immediate vicinity of these boundaries (Figure 5.16). This feature may not be suitable for some models, but the effect can be significantly reduced by using a thinner and longer nozzle. Future work is needed to determine if a change in slicing parameters could be used to eliminate this effect.



B=1 **B=2** **B=3** **B=5** **B=6**

Figure 5.16: The boundaries of the material of adjacent batches. The layer thickness is 0.2 mm, and the B-parameter indicates the number of layers in a single batch.

The use of the ITC was tested on a set of models, characterized by a wide variety of shapes and summarized in Table 5.1. The number of colors and the dimensions of each part are presented in the table. Comparative consumptions of time and materials are illustrated in Figures 5.17 and 5.18, respectively. In this case, the energy parameters were not measured, but were calculated theoretically based on the coefficient obtained earlier (0.11 kWh per hour).

Table 5.1: Comparison of time and material allocation for different fabrication techniques of a selected set of 3-D models [71].

Model	Slicing	Printing time, h			Tool changes	Filament length, m		Energy, kWh
		Total	*Wipe tower*	*Travel*		*Total*	*Wipe tower*	
Citrus coaster	***ITC***	***2.37***	***0.15***	***0.2***	***8***	***9.05***	***1.13***	***0.26***
3 colors, 100x100x3 mm	Default	2.85	0.43	0.23	30	11.98	4.06	0.31
Plant cell	***ITC***	***3.98***	***0.53***	***0.50***	***32***	***15.60***	***4.34***	***0.44***
8 colors, 87x95x26 mm	Default	6.62	2.38	0.80	112	31.00	19.75	0.73
Kidney	***ITC***	***5.73***	***0.85***	***0.93***	***63***	***20.37***	***8.30***	***0.63***
4 colors, 85x80x34 mm	Default	7.82	2.37	1.20	135	33.00	21.15	0.86
Computer mouse,	***ITC***	***7.90***	***1.78***	***1.10***	***134***	***34.04***	***17.93***	***0.87***
3 colors, 100x58x37 mm	Default	9.97	3.12	1.28	221	46.46	30.35	1.10
Parrot,	***ITC***	***7.82***	***2.35***	***1.53***	***172***	***31.88***	***22.19***	***0.86***
4 colors, 49x73x60 mm	Default	13.81	6.10	2.23	429	67.06	57.38	1.52
Mandala,	***ITC***	***13.57***	***1.23***	***2.43***	***91***	***45.39***	***12.2***	***1.49***
3 colors, 120x120x50 mm	Default	17.03	3.52	2.52	245	67.04	33.85	1.87
Earth,	***ITC***	***18.03***	***3.57***	***3.57***	***277***	***65.77***	***36.15***	***1.98***
4 colors, 75x75x80 mm	Default	20.47	5.52	3.22	389	82.68	53.05	2.25

Toronto CN tower,	***ITC***	***21.33***	***5.07***	***5.38***	***336***	***65.39***	***38.06***	***2.35***
8 colors, 104x104x150 mm	Default	33.28	13.40	6.50	820	142.92	115.60	3.66
Hourglass,	***ITC***	***15.85***	***4.42***	***3.43***	***328***	***63.02***	***43.81***	***1.74***
4 colors, 80x80x125 mm	Default	33.92	14.95	5.18	1118	170.04	150.82	3.73
Integumentary system	***ITC***	***30.48***	***5.63***	***7.37***	***429***	***108.40***	***56.25***	***3.35***
7 colors, 67x95x94 mm	Default	39.98	12.43	7.37	880	172.5	120.55	4.40
Leaf tissue,	***ITC***	***40.52***	***5.22***	***9.50***	***377***	***127.42***	***49.73***	***4.46***
5 colors, 83x102x67 mm	Default	44.73	8.98	8.15	634	164.08	86.53	4.92
Coffee shop,	***ITC***	***37.35***	***6.73***	***6.45***	***494***	***144.72***	***65.53***	***4.11***
7 colors, 100x160x49 mm	Default	53.43	17.32	8.83	1204	241.92	162.73	5.88
Bike helmet,	***ITC***	***41.27***	***3.73***	***7.08***	***274***	***133.39***	***36.22***	***4.54***
4 colors, 129x166x93 mm	Default	54.55	12.20	8.00	898	217.46	121.20	6.00
Tooth model,	***ITC***	***46.72***	***10.05***	***8.53***	***768***	***184.62***	***102.19***	***5.14***
5 colors, 100x52x160 mm	Default	78.30	28.95	11.20	2206	378.04	295.60	8.61
Total	***ITC***	***292.92***	***51.31***	***58.00***	***3783***	***1049.06***	***494.03***	***32.22***
	Default	416.76	131.67	66.71	9321	1826.18	1272.62	45.84
Savings, %		***29.71***	***61.03***	***13.06***	***59.41***	***42.55***	***61.18***	***29.71***

As can be seen from Table 5.1, the total production time of models has been reduced by more than 1.4 times, material consumption has been reduced by 1.7 times, and the number of tool changes – by 2.4 times. In addition, the energy use was decreased by about 30%, which given a constant energy source would have substantive impacts on the greenhouse gas emissions and other environmental impacts of AM from energy consumption [72]. The minimum gain occurs during travel movement, as when using the tool clustering method, the interregional lift Z parameter is increased, resulting in longer overall travels per each region.

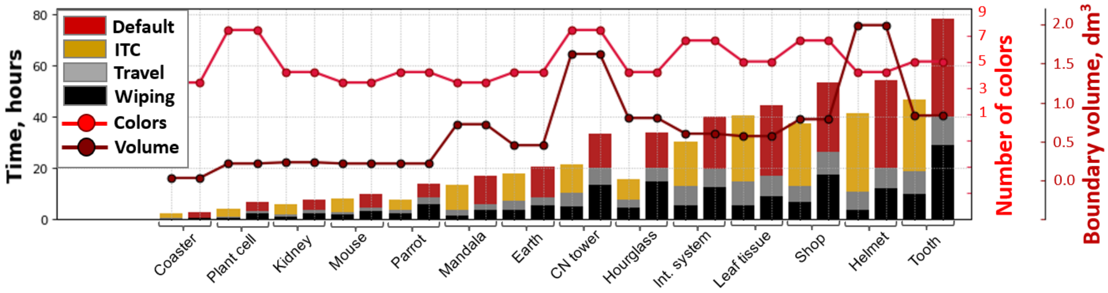


Figure 5.17: Comparative time consumption when fabricating a random set of models.

Figure 5.19 shows an example of the printed mandala model with different slicing parameters (default slicing vs ITC method with a batch height of 1.0 mm, or 5 layers of 0.2 mm). A total of 48.50 meters of plastic were used to print the mandala, of which 27.77

meters were used to produce the wipe tower. Using the developed ITC method, the filament consumption was reduced to 30.86 meters, 10.13 of which are needed for the wipe tower. The total production time is 13.5 hours when printing with default parameters and 10.0 hours using the tool clustering method. Savings in energy costs amount to 0.39 kWh (1.10 vs 1.49 kWh with the default slicing). The experimental data obtained thus supports the theoretical findings, however, caution is needed in overgeneralizing the results found here as the results for individual models may differ due to geometric features.

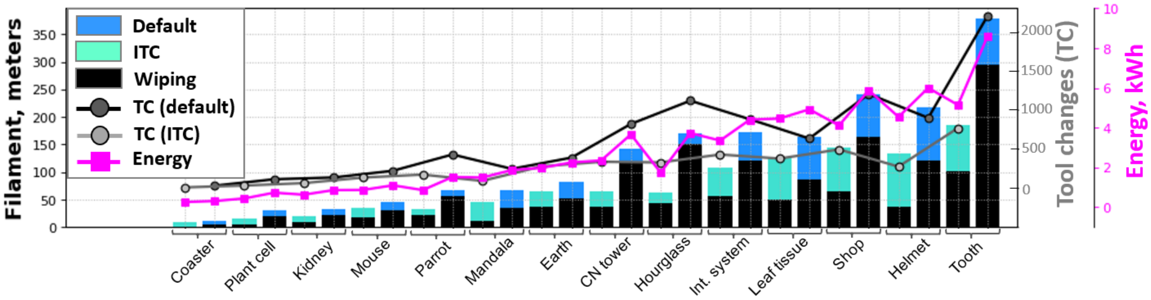


Figure 5.18: Comparative material consumption when fabricating a random set of models.

The advantages of ITC for multi-material 3-D printer users are clear from an efficiency standpoint in both time savings and economic savings from reduced material use and energy consumption. From a wider societal perspective as the cost advantages of distributed digital manufacturing [25] cause greater adoption rates, ITC savings will expand to have a substantiative environmental impact. This can be done with both virgin materials as well facilitating the use of recycled materials. For example, to reduce the embodied energy of transportation needed for centralized recycling [73], while at the same time potentially improving the financial situation of waste pickers a distributed recycling paradigm has been proposed [73–76]. If a two-material print can use high-quality filament (virgin or recycled) on the exterior and lower-quality recycled filament for the infill a large amount of waste plastic can be used in more solid 3-D prints, which may not be economic otherwise. The ITC reduces the time and energy to leverage the ‘fill with waste’ approach, which further enhances the economics of distributed recycling and additive manufacturing (DRAM) [77–79]. One method of DRAM is to upcycle plastic waste into 3-D printing filament with a recyclebot, which is an open source waste plastic extruder [80,81]. Previous research on the life cycle analysis (LCA) of the recyclebot process using post-consumer plastics instead of raw materials, showed a 90% decrease in the embodied energy of the filament from the mining, processing of natural resources and synthesizing compared to traditional manufacturing [82,83]. The ITC further improves the potential for recycled filament to be used by prosumers to recycle plastic to save money by offsetting purchased filament [77,82,83].

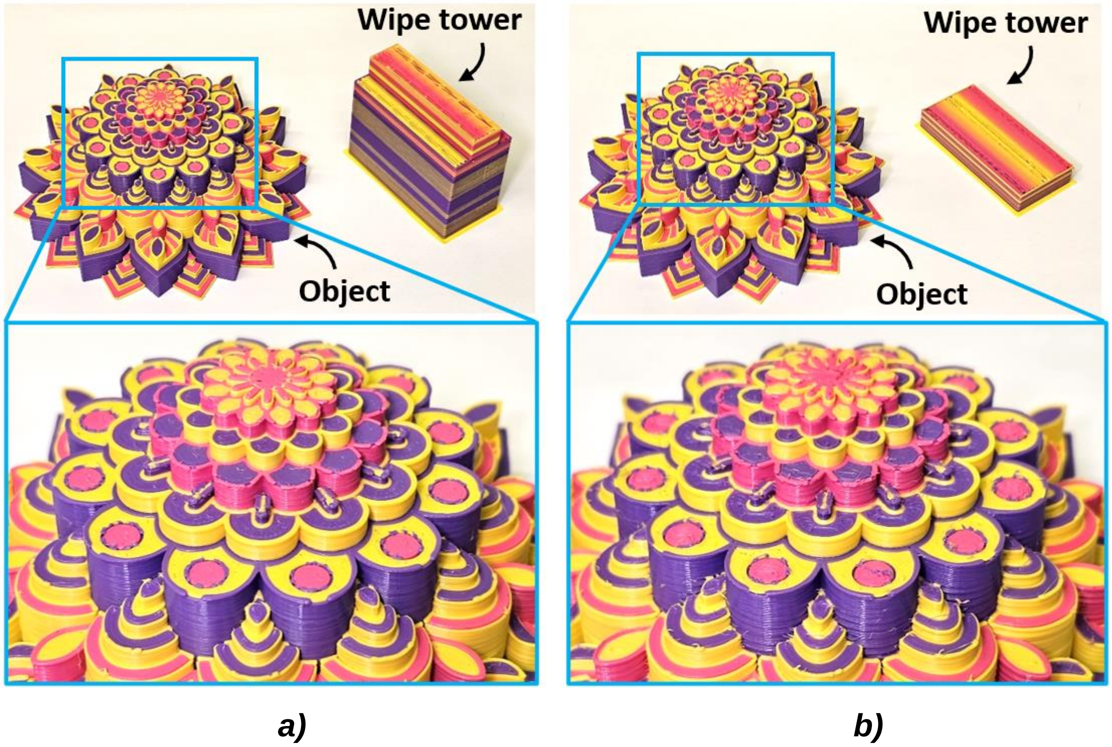


Figure 5.19: Default *(a)* vs tool clustering *(b)* fabrication of the multi-color mandala model. With the default method, tool changes occur on each layer, while with the ITC method, tool changes occur after each batch (1.0 mm, or 5 layers of 0.2 mm) is printed.

Future work is needed to adjust the slicing settings using the interlayer tool clustering to allow printing of several different types of materials by changing the 3-D printing parameters like temperature and print speed for individual extrusion sections. Additional work is also needed to eliminate the boundary issues with layer batching shown in Figure 5.16 by perhaps using changes in slicer settings at the transition. More research is also needed on the use of wiping into infill option, which would eliminate the waste of a wipe tower all together for some prints. Finally, future work is needed to quantify the aesthetic quality as well as the potential impacts on strength or other mechanical properties of finished prints using the ITC vs conventional multi-material slicing.

5.7 Conclusions

In this study, a novel technique known as interlayer tool clustering is introduced and applied to the open source PrusaSlicer. The theoretical as well as experimental toolpath optimization in multi-material slicing has been shown to achieve substantial reductions in time, energy, and material waste. The presented approach allows fabricating material regions out of sequential layer order and is validated using a diverse set of models from areas such as biomedical education, architectural and landscape design, and product development. These improvements will directly assist in the UN Sustainable Development Goals 9 (Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation) and 12 (Ensure sustainable consumption and production patterns).

The results show the novel method can significantly increase the efficiency of multi-material printing, which is expressed in an average 1.7-fold reduction in material used, and an average 1.4-fold reduction in both time and 3-D printing energy use. In addition, this approach reduces the likelihood of technical failures in the manufacturing of the entire part by reducing the number of tool changes, or material transitions, on average by 2.4 times. These savings all support distributed recycling and additive manufacturing, which has both environmental and economic benefits. Increasing the number of colors in fabrication leads to an increase in savings and thus the benefits.

The proposed open source additions are adapted to PrusaSlicer version 2.6.0 and presented under the AGPL-3.0 license. Similar changes can also be applied to other slicing software based on the core “libslic3r” library: newer versions of PrusaSlicer, Bambu Studio, OrcaSlicer, SuperSlicer, etc., with appropriate individual modifications. With the wide support for native and third-party 3-D printer profiles, these slicers can be used to prepare models for single-nozzle printers from vendors such as Prusa, BambuLab, Creality, Lulzbot, Ultimaker, Voron, and others. This significantly expands the capabilities of the developed method. The same ITC can be used for multi-material 3-D printing and additional work is required to enable variants of slicing parameters for each of the materials.

As future work, it is important to note the necessity for extensive economic analysis and systematic study on multi-color polymer 3-D printing applications in academia and industry from a joint perspective of hardware, materials, and slicing algorithms. Additional experimental research is required to adjust the batched support structure generation and variable layer height implementation. More studies are also needed to analyze the impact of auxiliary inter-batch waiting time on layer adhesion and mechanical properties of manufactured parts.

5.8 Bibliography

[1] B. Esmaeilian, S. Behdad, B. Wang, The evolution and future of manufacturing: A review, J. Manuf. Syst. 39 (2016) 79–100. https://doi.org/10.1016/j.jmsy.2016.03.001.

[2] M. Ghobakhloo, The future of manufacturing industry: a strategic roadmap toward Industry 4.0, J. Manuf. Technol. Manag. 29 (2018) 910–936. https://doi.org/10.1108/JMTM-02-2018-0057.

[3] C.W. Hull, Apparatus for production of three-dimensional objects by stereolithography, 4,575,330, 1986. https://patents.google.com/patent/US4575330A/en (accessed August 31, 2023).

[4] Pierre Alfred Leon Ciraud, Method and device for manufacturing any articles from any meltable material, DE2263777A1, n.d. https://patents.google.com/patent/DE2263777A1/en (accessed September 8, 2023).

[5] S. Ford, T. Minshall, Invited review article: Where and how 3D printing is used in teaching and education, Addit. Manuf. 25 (2019) 131–150. https://doi.org/10.1016/j.addma.2018.10.028.

[6] C.-Y. Liaw, M. Guvendiren, Current and emerging applications of 3D printing in medicine, Biofabrication. 9 (2017) 024102. https://doi.org/10.1088/1758-5090/aa7279.

[7] Mark Zastrow, 3D printing gets bigger, faster and stronger, Nature. 578 (2020) 20–23.

[8] J. Jeong, H. Park, Y. Lee, J. Kang, J. Chun, Developing parametric design fashion products using 3D printing technology, Fash. Text. 8 (2021) 22. https://doi.org/10.1186/s40691-021-00247-8.

[9] I. Dankar, A. Haddarah, F.E.L. Omar, F. Sepulcre, M. Pujolà, 3D printing technology: The new era for food customization and elaboration, Trends Food Sci. Technol. 75 (2018) 231–242. https://doi.org/10.1016/j.tifs.2018.03.018.

[10] J.M. Pearce, Cut costs with open-source hardware, Nature. 505 (2014) 618–618. https://doi.org/10.1038/505618d.

[11] P. Wu, J. Wang, X. Wang, A critical review of the use of 3-D printing in the construction industry, Autom. Constr. 68 (2016) 21–31. https://doi.org/10.1016/j.autcon.2016.04.005.

[12] R. Orange Kedem, N. Opatovski, D. Xiao, B. Ferdman, O. Alalouf, S. Kumar Pal, Z. Wang, H. Von Der Emde, M. Weber, S.J. Sahl, A. Ponjavic, A. Arie, S.W. Hell, Y. Shechtman, Near index matching enables solid diffractive optical element fabrication via additive manufacturing, Light Sci. Appl. 12 (2023) 222. https://doi.org/10.1038/s41377-023-01277-1.

[13] T. Gissibl, S. Thiele, A. Herkommer, H. Giessen, Sub-micrometre accurate free-form optics by three-dimensional printing on single-mode fibres, Nat. Commun. 7 (2016) 11763. https://doi.org/10.1038/ncomms11763.

[14] T.J. Wallin, J. Pikul, R.F. Shepherd, 3D printing of soft robotic systems, Nat. Rev. Mater. 3 (2018) 84–100. https://doi.org/10.1038/s41578-018-0002-2.

[15] J.C. Vasco, Additive manufacturing for the automotive industry, in: Addit. Manuf., Elsevier, 2021: pp. 505–530. https://doi.org/10.1016/B978-0-12-818411-0.00010-0.

[16] J. Pierre, F. Iervolino, R.D. Farahani, N. Piccirelli, M. Lévesque, D. Therriault, Material extrusion additive manufacturing of multifunctional sandwich panels with load-bearing and acoustic capabilities for aerospace applications, Addit. Manuf. 61 (2023) 103344. https://doi.org/10.1016/j.addma.2022.103344.

[17] D. Wangpraseurt, S. You, F. Azam, G. Jacucci, O. Gaidarenko, M. Hildebrand, M. Kühl, A.G. Smith, M.P. Davey, A. Smith, D.D. Deheyn, S. Chen, S. Vignolini, Bionic 3D printed corals, Nat. Commun. 11 (2020) 1748. https://doi.org/10.1038/s41467-020-15486-4.

[18] M. Rafiee, R.D. Farahani, D. Therriault, Multi-Material 3D and 4D Printing: A Survey, Adv. Sci. 7 (2020) 1902307. https://doi.org/10.1002/advs.201902307.

[19] J. Lee, H.-C. Kim, J.-W. Choi, I.H. Lee, A review on 3D printed smart devices for 4D printing, Int. J. Precis. Eng. Manuf.-Green Technol. 4 (2017) 373–383. https://doi.org/10.1007/s40684-017-0042-x.

[20] F42 Committee, Terminology for Additive Manufacturing - General Principles - Terminology, ASTM International, n.d. https://doi.org/10.1520/F3177-21.

[21] D. Han, H. Lee, Recent advances in multi-material additive manufacturing: methods and applications, Curr. Opin. Chem. Eng. 28 (2020) 158–166. https://doi.org/10.1016/j.coche.2020.03.004.

[22] A. Bandyopadhyay, B. Heer, Additive manufacturing of multi-material structures, Mater. Sci. Eng. R Rep. 129 (2018) 1–16. https://doi.org/10.1016/j.mser.2018.04.001.

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

[24] E.E. Petersen, J. Pearce, Emergence of Home Manufacturing in the Developed World: Return on Investment for Open-Source 3-D Printers, Technologies. 5 (2017) 7. https://doi.org/10.3390/technologies5010007.

[25] J. Pearce, J.-Y. Qian, Economic Impact of DIY Home Manufacturing of Consumer Products with Low-cost 3D Printing from Free and Open Source Designs, Eur. J. Soc. Impact Circ. Econ. 3 (2022) 1–24. https://doi.org/10.13135/2704-9906/6508.

[26] Martin Placek, Most used 3D printing technologies worldwide 2021, (2023). https://www.statista.com/statistics/560304/worldwide-survey-3d-printing-top-technologies/ (accessed September 1, 2023).

[27] Sophian Beyerlein, M. Aboushama, Evaluation of Continuous Fiber Reinforcement Desktop 3D Printers Desktop 3D Printers Overview, (2020). https://doi.org/10.13140/RG.2.2.16640.87040.

[28] FlashForge Creator Pro2, (n.d.). https://www.flashforge.com/product-detail/flashforge-creator-pro-2-3d-printer (accessed September 13, 2023).

[29] Sovol SV04 IDEX, (n.d.). https://sovol3d.com/products/sv04 (accessed September 13, 2023).

[30] Geeetech A30T Three-Colors Printing, (n.d.). https://www.geeetech.com/geeetech-a30t-threecolors-printing-large-printing-area-320320420mm-quick-assembly-p-1135.html (accessed September 13, 2023).

[31] Original Prusa MMU3, (n.d.). https://www.prusa3d.com/category/original-prusa-mmu3/ (accessed August 21, 2023).

[32] Mosaic Palette 3 Pro, (n.d.). https://www.mosaicmfg.com/products/palette-3-pro (accessed August 21, 2023).

[33] Wipe tower - Prusa Knowledge Base, (n.d.). https://help.prusa3d.com/article/wipe-tower_125010 (accessed August 21, 2023).

[34] M. Kreiger, J.M. Pearce, Environmental Life Cycle Analysis of Distributed Three-Dimensional Printing and Conventional Manufacturing of Polymer Products, ACS Sustain. Chem. Eng. 1 (2013) 1511–1519. https://doi.org/10.1021/sc400093k.

[35] M. Kreiger, J.M. Pearce, Environmental Impacts of Distributed Manufacturing from 3-D Printing of Polymer Components and Products, MRS Online Proc. Libr. OPL. 1492 (2013) 85–90. https://doi.org/10.1557/opl.2013.319.

[36] M.I. Mohammed, M. Mohan, A. Das, M. D. Johnson, P. Singh Badwal, D. McLean, I. Gibson, A low carbon footprint approach to the reconstitution of plastics into 3D-printer filament for enhanced waste reduction, KnE Eng. 2 (2017) 234. https://doi.org/10.18502/keg.v2i2.621.

[37] A. Das, E. Gomez-Kervin, D. Wilson, I. Gibson, M.I. Mohammed, EcoPrinting: Investigating the use of 100% recycled Acrylonitrile Butadiene Styrene (ABS) for Additive Manufacturing, in: Solid Free. Fabr. 2017, 2017. https://core.ac.uk/download/pdf/327067487.pdf.

[38] PrusaSlicer, (n.d.). https://www.prusa3d.com/page/prusaslicer_424/ (accessed August 21, 2023).

[39] SuperSlicer, (n.d.). https://github.com/supermerill/SuperSlicer (accessed September 8, 2023).

[40] Bambu Studio, (n.d.). https://bambulab.com/en/download (accessed September 8, 2023).

[41] Orca Slicer, (n.d.). https://github.com/SoftFever/OrcaSlicer (accessed September 8, 2023).

[42] A. Ranellucci, Slic3r: Open source 3-D printing toolbox, (n.d.). https://slic3r.org/ (accessed August 21, 2023).

[43] A. Petsiuk, PrusaSlicer: Interlayer Tool Clustering Fork, (n.d.). https://github.com/apetsiuk/PrusaSlicer (accessed August 21, 2023).

[44] Mosaic Manufacturing, Multi-color kidney model, (n.d.). https://cults3d.com/en/3d-model/tool/multi-color-kidney-model (accessed August 21, 2023).

[45] D. Kaplan, S. Rorberg, M. Ben Chen, Y. Sterman, NozMod: Nozzle Modification for Efficient FDM 3D Printing, in: Symp. Comput. Fabr., ACM, Seattle WA USA, 2022: pp. 1–9. https://doi.org/10.1145/3559400.3561999.

[46] W. Wang, H. Chao, J. Tong, Z. Yang, X. Tong, H. Li, X. Liu, L. Liu, Saliency-Preserving Slicing Optimization for Effective 3D Printing: Saliency-Based Slicing Optimization, Comput. Graph. Forum. 34 (2015) 148–160. https://doi.org/10.1111/cgf.12527.

[47] D. Ahlers, F. Wasserfall, N. Hendrich, J. Zhang, 3D Printing of Nonplanar Layers for Smooth Surface Generation, in: 2019 IEEE 15th Int. Conf. Autom. Sci. Eng. CASE, IEEE, Vancouver, BC, Canada, 2019: pp. 1737–1743. https://doi.org/10.1109/COASE.2019.8843116.

[48] G.A. Nisja, A. Cao, C. Gao, Short review of nonplanar fused deposition modeling printing, Mater. Des. Process. Commun. 3 (2021). https://doi.org/10.1002/mdp2.221.

[49] J. Etienne, N. Ray, D. Panozzo, S. Hornus, C.C.L. Wang, J. Martínez, S. McMains, M. Alexa, B. Wyvill, S. Lefebvre, CurviSlicer: slightly curved slicing for 3-axis printers, ACM Trans. Graph. 38 (2019) 1–11. https://doi.org/10.1145/3306346.3323022.

[50] B. Huang, S.B. Singamneni, Curved Layer Adaptive Slicing (CLAS) for fused deposition modelling, Rapid Prototyp. J. 21 (2015) 354–367. https://doi.org/10.1108/RPJ-06-2013-0059.

[51] I. Molloy, T. Miller, Digital Dexterity. Freeform 3D printing through direct toolpath manipulation for crafted artifacts, in: Mexico City, Mexico, 2018: pp. 266–275. https://doi.org/10.52842/conf.acadia.2018.266.

[52] S. Mueller, S. Im, S. Gurevich, A. Teibrich, L. Pfisterer, F. Guimbretière, P. Baudisch, WirePrint: 3D printed previews for fast prototyping, in: Proc. 27th Annu. ACM Symp. User Interface Softw. Technol., ACM, Honolulu Hawaii USA, 2014: pp. 273–280. https://doi.org/10.1145/2642918.2647359.

[53] A. Gleadall, FullControl GCode Designer: Open-source software for unconstrained design in additive manufacturing, Addit. Manuf. 46 (2021) 102109. https://doi.org/10.1016/j.addma.2021.102109.

[54] M. Alexa, K. Hildebrand, S. Lefebvre, Optimal Discrete Slicing, ACM Trans. Graph. 36 (2017) 1–16. https://doi.org/10.1145/2999536.

[55] J. Hergel, K. Hinz, S. Lefebvre, B. Thomaszewski, Extrusion-based ceramics printing with strictly-continuous deposition, ACM Trans. Graph. 38 (2019) 1–11. https://doi.org/10.1145/3355089.3356509.

[56] F. Zhong, Y. Xu, H. Zhao, L. Lu, As-Continuous-As-Possible Extrusion-Based Fabrication of Surface Models, ACM Trans. Graph. 42 (2023) 1–16. https://doi.org/10.1145/3575859.

[57] H. Liu, R. Liu, Z. Liu, S. Xu, Minimizing the Number of Transitions of 3D Printing Nozzles Using a Traveling-Salesman-Problem Optimization Model, Int. J. Precis. Eng. Manuf. 22 (2021) 1617–1637. https://doi.org/10.1007/s12541-021-00512-2.

[58] K.-Y. Fok, C.-T. Cheng, N. Ganganath, H.H.-C. Iu, C.K. Tse, An ACO-Based Tool-Path Optimizer for 3-D Printing Applications, IEEE Trans. Ind. Inform. 15 (2019) 2277–2287. https://doi.org/10.1109/TII.2018.2889740.

[59] J.I. Aguilar-Duque, C.O. Balderrama-Armendáriz, C.A. Puente-Montejano, A.S. Ontiveros-Zepeda, J.L. García-Alcaraz, Genetic algorithm for the reduction printing time and dimensional precision improvement on 3D components printed by Fused Filament Fabrication, Int. J. Adv. Manuf. Technol. 115 (2021) 3965–3981. https://doi.org/10.1007/s00170-021-07314-w.

[60] libslic3r: Library for generating gcode from 3d models, (n.d.). https://manual.slic3r.org/libslic3r-doc/namespace_slic3r.html (accessed September 8, 2023).

[61] GNU Affero General Public License v3.0, (2017). https://github.com/prusa3d/PrusaSlicer/blob/master/LICENSE (accessed September 6, 2023).

[62] Alessandro Ranellucci, 1.5 years of Slic3r development, (2013). https://slic3r.org/blog/1.5-years-of-slic3r-development/ (accessed September 14, 2023).

[63] Enrique Perez, Skeinforge, RepRap. (n.d.). https://reprap.org/wiki/Skeinforge (accessed September 14, 2023).

[64] E. Sells, S. Bailard, Z. Smith, A. Bowyer, V. Olliver, RepRap: The Replicating Rapid Prototyper: Maximizing Customizability by Breeding the Means of Production, in: Handb. Res. Mass Cust. Pers., World Scientific Publishing Company, 2009: pp. 568–580. https://doi.org/10.1142/9789814280280_0028.

[65] R. Jones, P. Haufe, E. Sells, P. Iravani, V. Olliver, C. Palmer, A. Bowyer, RepRap – the replicating rapid prototyper, Robotica. 29 (2011) 177–191. https://doi.org/10.1017/S026357471000069X.

[66] A. Bowyer, 3D Printing and Humanity’s First Imperfect Replicator, 3D Print. Addit. Manuf. 1 (2014) 4–5. https://doi.org/10.1089/3dp.2013.0003.

[67] Angus Johnson, Clipper2: A Polygon Clipping and Offsetting library, (n.d.). https://github.com/AngusJohnson/Clipper2 (accessed September 14, 2023).

[68] Flowalistik, Low-poly Charmander, (n.d.). https://cults3d.com/en/3d-model/game/low-poly-charmander-multi-and-dual-extrusion-version (accessed September 14, 2023).

[69] B. Wijnen, P. Sanders, J.M. Pearce, Improved model and experimental validation of deformation in fused filament fabrication of polylactic acid, Prog. Addit. Manuf. 3 (2018) 193–203. https://doi.org/10.1007/s40964-018-0052-4.

[70] Original Prusa MMU2S, (n.d.). https://help.prusa3d.com/tag/mmu2 (accessed September 14, 2023).

[71] Mosaic Manufacturing: A digital collection of multimaterial 3D models, (n.d.). https://www.thingiverse.com/mosaicmanufacturing/designs (accessed September 14, 2023).

[72] T. Peng, K. Kellens, R. Tang, C. Chen, G. Chen, Sustainability of additive manufacturing: An overview on its energy demand and environmental impact, Addit. Manuf. 21 (2018) 694–704. https://doi.org/10.1016/j.addma.2018.04.022.

[73] U. Arena, M.L. Mastellone, F. Perugini, Life Cycle assessment of a plastic packaging recycling system, Int. J. Life Cycle Assess. 8 (2003) 92. https://doi.org/10.1007/BF02978432.

[74] S.R. Feeley, B. Wijnen, J.M. Pearce, Evaluation of Potential Fair Trade Standards for an Ethical 3-D Printing Filament, J. Sustain. Dev. 7 (2014) p1. https://doi.org/10.5539/jsd.v7n5p1.

[75] T. Birtchnell, W. Hoyle, The 3D4D Challenge, in: 3D Print. Dev. Glob. South, Palgrave Macmillan UK, London, 2014: pp. 13–35. https://doi.org/10.1057/9781137365668_2.

[76] S. Zhong, P. Rakhe, J. Pearce, Energy Payback Time of a Solar Photovoltaic Powered Waste Plastic Recyclebot System, Recycling. 2 (2017) 10. https://doi.org/10.3390/recycling2020010.

[77] S. Zhong, J.M. Pearce, Tightening the loop on the circular economy: Coupled distributed recycling and manufacturing with recyclebot and RepRap 3-D printing, Resour. Conserv. Recycl. 128 (2018) 48–58. https://doi.org/10.1016/j.resconrec.2017.09.023.

[78] S.C. Dertinger, N. Gallup, N.G. Tanikella, M. Grasso, S. Vahid, P.J.S. Foot, J.M. Pearce, Technical pathways for distributed recycling of polymer composites for distributed manufacturing: Windshield wiper blades, Resour. Conserv. Recycl. 157 (2020) 104810. https://doi.org/10.1016/j.resconrec.2020.104810.

[79] F.A. Cruz Sanchez, H. Boudaoud, M. Camargo, J.M. Pearce, Plastic recycling in additive manufacturing: A systematic literature review and opportunities for the circular economy, J. Clean. Prod. 264 (2020) 121602. https://doi.org/10.1016/j.jclepro.2020.121602.

[80] C. Baechler, M. DeVuono, J.M. Pearce, Distributed recycling of waste polymer into RepRap feedstock, Rapid Prototyp. J. 19 (2013) 118–125. https://doi.org/10.1108/13552541311302978.

[81] A.L. Woern, J.R. McCaslin, A.M. Pringle, J.M. Pearce, RepRapable Recyclebot: Open source 3-D printable extruder for converting plastic to 3-D printing filament, HardwareX. 4 (2018) e00026. https://doi.org/10.1016/j.ohx.2018.e00026.

[82] M. Kreiger, G.C. Anzalone, M.L. Mulder, A. Glover, J.M. Pearce, Distributed Recycling of Post-Consumer Plastic Waste in Rural Areas, MRS Proc. 1492 (2013) 91–96. https://doi.org/10.1557/opl.2013.258.

[83] M.A. Kreiger, M.L. Mulder, A.G. Glover, J.M. Pearce, Life cycle analysis of distributed recycling of post-consumer high density polyethylene for 3-D printing filament, J. Clean. Prod. 70 (2014) 90–96. https://doi.org/10.1016/j.jclepro.2014.02.009.

Chapter 6

6 Conclusion

6.1 Summary

The presented research demonstrates novel approaches to layer-wise monitoring and analysis of FFF AM processes based on computer vision, as well as an alternative fabrication method for multicolor 3-D printing.

Revealing these methods to end users as open source techniques will provide research motivation and allow constant expansion of the information base for subsequent continuous improvement. Integrating the discussed approaches with web-based 3-D printing control systems can help to perform layer-wise analysis of manufactured parts, and classify and track manufacturing deviations based on their bonding to a particular area of the model. These studies all support distributed and additive manufacturing, which has both environmental and economic benefits.

6.2 Contributions

The developments presented in this dissertation formed the basis for the concept of multifaceted visual analysis of 3-D printing processes. This will help improve FFF AM technology in quality control and reduce the amount of time, materials, and energy required to manufacture physical objects. The contributions of the work presented in this dissertation are as follows:

1. Presented a conceptually new approach to comprehensive monocular layer-wise visual analysis of extrusion-based FFF AM processes. This approach is built upon multi-stage image examination, allowing control of the height, the external shape of the printed object, and the internal structure of its layers.
2. Introduced a basis for adaptive 3-D printing control, allowing G-code updates during fabrication.
3. Developed a novel method for detecting 3-D printing anomalies by comparing images of printed layers from a stationary monocular camera with photorealistic

synthetic G-code-based reference images of an ideal process generated in a physics-based rendering engine. Recognition of visual deviations was accomplished by analyzing the similarity of histograms of oriented gradients (HOG) of local image areas. The developed method can be applied to material extrusion 3-D printers of any size with a resolution of detecting anomalous regions of 5–10% of the overall observation area.

4. Developed a novel approach to segment key regions of a part during manufacturing utilizing the information in the G-code and synthetic image data. The following mIoU scores were achieved for synthetic test data sets: entire printed part, 94.90%; top layer, 73.33%; infill, 78.93%; shell, 55.31%; support, 69.45%. The developed segmentation framework can enhance visual analysis of manufacturing processes and enable the detection of fabrication errors, while significantly reducing camera positioning accuracy and calibration requirements. Revealing this system to end users will allow constant expansion of the synthetic image database for subsequent neural network training and improvement of segmentation results. Integrating it with web-based 3-D printing control systems can help to perform layer-wise analysis of manufactured parts, and also help to classify and track failures based on their bonding to a particular area of the model.
5. Developed and applied to the open source PrusaSlicer software a novel interlayer tool clustering method for multi-color 3-D printing. The given experimental toolpath optimization in multi-material slicing has been shown to achieve substantial reductions in time, energy, and material waste. The results show the method can significantly increase the efficiency of multi-material printing, which is expressed in an average 1.7-fold reduction in material used, and an average 1.4-fold reduction in both time and 3-D printing energy use. In addition, this approach reduces the likelihood of technical failures in the manufacturing of the entire part by reducing the number of tool changes, or material transitions, on average by 2.4 times.

6.3 Limitations and Future Work

The development of adaptive control algorithms is a comprehensive and complex problem since it is challenging to (1) uniquely visually determine the type of failure, (2) establish a direct causal relationship between the type of failure and the fabrication setting involved, and (3) declare in advance what parameter value (scaling coefficients, feed rate, temperature, traveling speed, etc.) should be used to correct the failure.

The presented study does not cover all of possible manufacturing anomalies and requires additional research in the future, utilizing constantly evolving machine learning and computer vision technologies. It is necessary to supplement the introduced developments with feedback systems that will allow adaptive control and toolpath generation based on image analysis results.

In the application of semantic segmentation in AM, more research is needed on the use of image style transfer and domain adaptation to bridge the gap between synthetic and real domains.

In the field of multi-color 3-D printing, it is necessary to implement general toolpath optimization based on the geometry and materials of local areas of manufactured models.

Integration of the presented methods with existing web-based 3-D printing control systems will attract a considerable number of users to supplement a centralized database of manufacturing deviations. This will stimulate the development of advanced data-driven additive manufacturing and enable performing continuous analysis of manufactured parts, classifying and tracking anomalies based on their bonding to a specific region of the model.

In conclusion, recommended future projects can be listed as follows:

- Implementation of precise contour-based tracking of the produced part layers;
- Development of image style transfer and domain adaptation systems for the effective use of labeled synthetic data;
- Development of a real-time semantic segmentation system for all of structured areas of manufactured layers;

- Development of a global labeled image dataset of 3-D printing errors;
- Development of a system for classifying manufacturing errors based on a centralized global image dataset;
- Development of adaptive G-code generation systems to correct localized 3-D printing defects;
- Implementation of the above developments in web-based 3-D printing control systems;
- Graph-based advanced fabrication algorithms for multi-color 3-D printing.

Curriculum Vitae

Name:	Aliaksei Petsiuk
Post-secondary Education and Degrees:	Belarusian National Technical University Minsk, Belarus 2005-2010 B.S. in Robotics
	National Academy of Sciences of Belarus Minsk, Belarus 2010-2011 M.S. in Physics
	The University of Western Ontario London, Ontario, Canada 2022-2024 Ph.D. in Electrical and Computer Engineering
Related Work Experience	Machine Learning Engineer (part-time) Mosaic Manufacturing (London, Ontario, Canada) 2023-2024
	Research Assistant Michigan Technological University (Houghton, MI, USA) 2020-2021
	Teaching Assistant Michigan Technological University (Houghton, MI, USA) 2017-2019
	Engineer (full-time/part-time) Multiservisnaya Set (Minsk, Belarus) 2013-2016
	Mathematical Engineer K-Nowledge Lab (Competentum Group) (Minsk, Belarus) 2011 – 2014
	Research Assistant Institute of Physics, NASB (Minsk, Belarus) 2011 – 2013

Publications:

1. **A. Petsiuk,** B. Bloch, D. Vogt, M. Debora, J.M. Pearce (in press), Tool change Reduction for Multi-color Fused Filament Fabrication Through Interlayer Tool Clustering Implemented in PrusaSlicer. *Rapid Prototyping Journal*, 2024.
2. **A. Petsiuk,** H. Singh, H. Dadhwal, J.M. Pearce, Synthetic-to-real composite semantic segmentation in additive manufacturing. *J. Manuf. Mater. Process.*, vol. 8, no. 66, 2024, doi:10.3390/jmmp8020066.
3. C. Baldus-Jeursen, **A. Petsiuk,** S.-A. Rheault, S. Pelland, A. Côté, Y. Poissant, J.M. Pearce, Snow losses for photovoltaic systems: validating the Marion and Townsend models. *IEEE Journal of Photovoltaics,* vol. 13, no. 4, pp. 610–620, 2023, doi:10.1109/JPHOTOV.2023.3264644.
4. P.J. Mayville, **A. Petsiuk,** J.M. Pearce, Thermal post-processing of 3D printed polypropylene parts for vacuum systems. *J. Manuf. Mater. Process.,* vol. 6, no. 98, 2022, doi:10.3390/jmmp6050098.
5. **A. Petsiuk,** B. Lavu, R. Dick, J.M. Pearce, Waste plastic direct extrusion hangprinter. *Inventions,* vol. 7, no. 70, 2022, doi:10.3390/inventions7030070.
6. K.S. Hayibo, **A. Petsiuk,** P. Mayville, L. Brown, J.M. Pearce, Monofacial vs bifacial solar photovoltaic systems in snowy environments. *Renewable Energy*, vol. 193, pp. 657–668, 2022, doi:10.1016/j.renene.2022.05.050.
7. **A. Petsiuk,** J.M. Pearce, Towards smart monitored AM: open source *in situ* layer-wise 3D printing image anomaly detection using histograms of oriented gradients and a physics-based rendering engine. *Additive Manufacturing,* vol. 52, no. 102690, 2022, doi:10.1016/j.addma.2022.102690.
8. M. Mohammed, D. Wilson, E. Gomez-Kervin, **A. Petsiuk,** R. Dick, J.M. Pearce, Sustainability and feasibility assessment of distributed E-waste recycling using additive manufacturing in a bi-continental context. *Additive Manufacturing,* vol. 50, no. 102548, 2022, doi:10.1016/j.addma.2021.102548.
9. **A. Petsiuk,** J.M. Pearce, Open Source Filament diameter sensor for recycling, winding, and additive manufacturing machines. *ASME. J. Manuf. Sci. Eng.*, 143(10): 105001, 2021, doi:10.1115/1.4050762.
10. **A. Petsiuk,** J.M. Pearce, Open source computer vision-based layer-wise 3D printing analysis. *Additive Manufacturing,* vol. 36, no. 101473, 2020, doi:10.1016/j.addma.2020.101473.
11. A.M. Pringle, S. Oberloier, **A. Petsiuk,** P.G. Sanders, J.M. Pearce, Open source arc analyzer: multi-sensor monitoring of wire arc additive manufacturing. *HardwareX,* vol. 8, no. e00137, 2020, doi:10.1016/j.ohx.2020.e00137.
12. **A. Petsiuk,** N.G. Tanikella, S. Dertinger, A. Pringle, S. Oberloier, J.M. Pearce, Partially RepRapable automated open source bag valve mask-based ventilator. *HardwareX,* vol. 8, no. e00131, 2020, doi:10.1016/j.ohx.2020.e00131.
13. S. Sule, **A. Petsiuk,** J.M. Pearce, Open source completely 3-D printable centrifuge. *Instruments,* vol. 3, no. 2: 30, 2019, doi:10.3390/instruments3020030.
14. **A. Petsiuk,** J.M. Pearce, Low-cost open source ultrasound-sensing based navigational support for the visually impaired. *Sensors,* vol. 19, no. 17: 3783, 2019, doi:10.3390/s19173783.
15. S. Hryshyn, A. Batischev, S. Koldashov, **A. Petsiuk,** V. Seliantev, S. Marach, V. Doktarau, S. Grishin, Procedure and tools for developing thermal control system

for scientific spacecraft equipment. *Heat Pipe Science and Technology*, vol. 4(3), pp. 169–180, 2013, doi:10.1615/HeatPipeScieTech.2014005521.

16. S. Hryshyn, **A. Petsiuk,** S. Grishin, V. Doktarau, V. Maziuk, Use of heat pipes for electronics block thermostating of magnetic measurement system fixed on space vehicle. *Heat Pipe Science and Technology*, vol. 2(1-4), pp. 31–41, 2011, doi:10.1615/HeatPipeScieTech.v2.i1-4.50.