Title

Improving precision of material extrusion 3D printing by in-situ monitoring and predicting 3D geometric deviation using Conditional Adversarial Networks

Author

Li, Ling

Publication Date

2020

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Improving precision of material extrusion 3D printing by in-situ monitoring and predicting 3D geometric deviation using Conditional Adversarial Networks

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Ling Li

2020

ABSTRACT OF THE DISSERTATION

Improving precision of material extrusion 3D printing by in-situ monitoring and predicting 3D geometric deviation using Conditional Adversarial Networks

by

Ling Li
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2020
Professor Robert Candler, Co-Chair
Professor Pirouz Kavehpour, Co-Chair

The field of additive manufacturing, especially 3D printing, has gained growing attention in the research and commercial sectors in recent years. Notwithstanding that the capabilities of 3D printing have moved on to enhanced resolution, higher deposition rate, and a wide variety of materials, the crucial challenge of verifying that the component manufactured is within the dimensional tolerance as designed continues to exist. Material extrusion 3D printing has long been established for rapid prototyping and functional testing in many research and industry fields. However, its inconsistency and intrinsic defects (surface roughness and geometric inaccuracies) hinder its application in several areas, most notably "certify-as-you-build" small-batch prototyping and large-batch production.

In this study, we present an approach to reduce both inconsistency and the 3D geometric inaccuracies of products fabricated by material extrusion.

1. This work developed and demonstrated an approach for layer-by-layer mapping of 3D printed parts, which can be used for validation of printed models and *in situ* adjustment of print parameters. This *in situ* metrology system scans each layer at the time of printing, providing a 3D model of the as-printed part. A high-speed optical scanning

system was integrated with a Material Extrusion type 3D printer to achieve *in situ* monitoring of dimensional inaccuracies during printing, which leaves the door open to implement a closed-loop feedback system to compensate geometric errors during printing in the future and fabricate "certify-as-you-build" products.

2. This work trained machine learning algorithms with data from this scanning system and predicted 3D geometric inaccuracies in new designs. Eight Conditional Adversarial Networks (CAN) machine learning models were trained on a limited number of scanned profile images of different layers, consisting of less than 50 actual images and 50 generated images, to predict the 3D geometric deviations of freeform shapes. The generated images were produced by randomly combining and cropping the actual images without any distortion. These CAN models produced predictions where at least 44.4%, 87.6%, 99.2% of data were within ±0.05 mm, ±0.10 mm, ±0.15 mm of the actual measured value, respectively.

3. This work developed an Iterative Forward approach to redesign the Computer-Aided-Design model by reverse engineering using the trained machine learning models, allowing for compensation of print imperfection at the design stage, in advance of the first printing. The compensation algorithms with eight different sets of different parameters were evaluated. It has been proven that the Iterative Forward approach improved the geometric deviation of the predicted profiles by making compensation to the CAD model.

The dissertation of Ling Li is approved.

Pei-Yu Chiou

Gregory Carman

Robert Candler, Committee Co-Chair

Pirouz Kavehpour, Committee Co-Chair

University of California, Los Angeles

2020

To my loving family . . .

who always supports me to chase my dream

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Professor Robert Candler, for his guidance, encouragement, and ingenuity. This dissertation would not have been possible without his steadfast support. I would like to thank Professors Pirouz Kavehpour, Pei-Yu Chiou, and Gregory Carman for serving on my dissertation committee, and for their insightful comments. I would like to thank Karl Gifford from Hyrel International Inc. for help with integrating the laser sensor to the 3D printer, Pedro Bernal for adding the USB camera to the system, Robert Issac for helping to expand the image dataset, and Max Ho for helpful discussions. I would like to thank all of my lab mates for all the support they have provided to me during my PhD.

VITA

2014	B.S. in Precision Instrument, Tsinghua University, Beijing, China.
2016	M.S. in Mechanical Engineering, University of California, Los Angeles (UCLA), Los Angeles, California.
2016–present	Ph.D. candidate in Mechanical Engineering, University of California, Los Angeles (UCLA), Los Angeles, California.

PUBLICATIONS

L. Li, R. McGuan, R. Issac, P. Kavehpour and R. Candler, (2020). Improving precision of material extrusion 3D printing by in-situ monitoring and predicting 3D geometric deviation using Conditional Adversarial Networks. *Additive Manufacturing*, submitted.

L. Li, A. A. Maccabi, Y.-Y. Juo, W. Zhang, Y.-J. Chang, G. N. Saddik, L. Jin, W. S. Grundfest, E. P. Dutson, J. D. Eldredge, P. Benharash, and R. N. Candler, (2019). Characterization of perfused and sectioned liver tissue in a full indentation cycle using a visco-hyperelastic model. *Journal of the Mechanical Behavior of Biomedical Materials*, 90, 591–603. https://doi.org/10.1016/j.jmbbm.2018.11.006

L. Li, R. McGuan, P. Kavehpour and R. Candler, (2018). Precison Enhancement Of 3D Printing Via In Situ Metrology. *Proceedings of the 2018 Solid Freeform Fabrication Symposium*, 251-260.

J. C. Wu, L. Li, J. C. Harrison and R. N. Candler, (2017). Micro-to millimeter scale magnetic

shielding. *19th International Conference on Solid- State Sensors, Actuators and Microsystems (TRANSDUCERS)*, Kaohsiung, 2017, pp. 838-841.

CHAPTER 1

Introduction

1.1 Motivation

Additive manufacturing (AM) has been broadly utilized to fabricate parts in free-form shapes with internal cavities, which are normally not practicable or are highly priced using conventional machining procedures. One of the challenges that hinders the extensive use of AM in manufacturing is that 3D printed parts suffer from poor consistency of geometric dimensioning and tolerating (GD&T). In traditional manufacturing industries, products are put through the requirements of the geometry specification (ISO 10303 [30]), tolerances (ISO 1101 [31] and ASME Y14.5 [32]), and surface finish to ensure the functionality. In AM, however, the differences in the manufacturing process compared to traditional methods (e.g., build direction, layer thickness, etc.) and current limitations of AM processes make it difficult to achieve and maintain the specification and tolerance of complex freeform surfaces and internal functional features.

1.2 Research Objectives

Motivated by the need to simultaneously reduce both job-to-job inconsistency and intrinsic 3D geometric inaccuracies, which are caused by inexact geometry conversion and the layer-by-layer printing mechanism of material extrusion, we propose an approach for predicting 3D geometric inaccuracies based on CAD models and *in situ* monitoring of the printing process. This method involves training conditional adversarial networks (CANs) on a limited set of primitive shapes to foresee the 3D geometric deviations of freeform complex shapes, which

could then be employed to reverse engineer and redesign the CAD model to compensate the geometric defects. This approach also integrated a laser sensor with a material extrusion 3D printer to accomplish *in situ* monitoring of dimensional inaccuracies during printing [33], which leaves the door open to implement a closed-loop feedback system to counteract geometric errors during printing in the future and fabricate "certify-as-you-build" products on the very first attempt.

1.3 Orgnaization of This Document

The remainder of this document is organized as follows:

Chapter 2 provides a detailed literature review of the state-of-the-art additive manufacturing techniques, the Material Extrusion type 3D printer that is used in this study, the *in-situ* monitoring technique, and closed-loop feedback system implemented in AM, and the statistical and machine learning models that can describe and predict geometric deviation from CAD models.

Chapter 3 introduces the metrology system that was built by integrating an optical scanning system with a Material Extrusion 3D printer and validated by using another high precision scanning system, including the hardware, optimized smart scan strategies, programs that can automatically generate customized scanning tool path and perform data acquisition on demands.

Chapter 4 presents the training and testing of the Conditional Adversarial Networks machine learning models on the cross-sections of the CAD models and the scanned profiles from our *in-situ* metrology system to make predictions of the scanned profiles on CAD models at the stage of design even before the first time of printing.

Chapter 5 develops two approaches, the Direct Reverse method and the Iterative Forward method, to redesign for Additive Manufacturing by implementing the predictions of the CAN machine learning models and trying to reduce the areas that are lack of material or should not have been printed.

Chapter 6 concludes this document by providing a roadmap for potential future research directions.

CHAPTER 2

Background

2.1 Seven Categories of Additive Manufacturing

Seven process classifications for additive manufacturing have been defined by ASTM F2792-12a: Binder Jetting, Directed Energy Deposition, Material Extrusion, Material Jetting, Powder Bed Fusion, Sheet Lamination, and Vat Photopolymerization. Table 2.1 summaries common names used for each category.

Binder Jetting is a 3D printing process that deposits a liquid bending agent to connect powder particles (Fig. 2.1(a)). After one layer is formed, the powder bed lowers and new powder is spread uniformly onto the top waiting for drops of a liquid binding agent (Fig. 2.1(b). This process repeats and Layers of material are bent to form a 3D object. Binder Jetting can print various materials including metals, sands, and ceramics. It is also capable of printing very large objects, like room-sized structures. Binder Jetting is distinctive that it doesn't involve heat source, which usually makes the process time sensitive and leaves residual stress, during the printing process. Without using a heat source, binder jetting is often more cost-efficient and faster compared to other additive manufacturing processes. However, some of the materials do need post-processing treatment, such as curing, sintering, or infiltration, to improve strength, depending on each application. For example, parts are often infiltrated with infiltrant that has a lower melting point to improve mechanical properties. Limited by the Binder Jetting process, parts with enclosed cavities should be avoided since the powder will be trapped inside of the hollow enclosure.

Directed Energy Deposition (DED) is a 3D printing approach that uses a laser or electron beam to melt a metal wire or powder onto a build stage or an existing part (Fig. 2.2). This

Classification Defined By ASTM F2792	Processes
Binding Jetting (BJ)	Powder bed and inkjet, drop-on- powder
Directed Energy Deposition (DED)	Laser engineered net shaping, directed light fabrication, direct metal deposition, 3D laser cladding
Material Extrusion (ME)	Fused Filament Fabrication (FFF), Fused Deposition Modeling (FDM)
Material Jetting (MJ)	Drop on demand
Powder Bed Fusion (PBD)	Laser Sintering, Selective Laser Melting, Electron Beam Melting
Sheet Lamination (SL)	ltrasonic additive manufacturing, laminated object manufacturing
Vat Photopolymerization (VP)	Selective Lithography, DLP

Table 2.1: Common names of seven additive manufacturing processes.

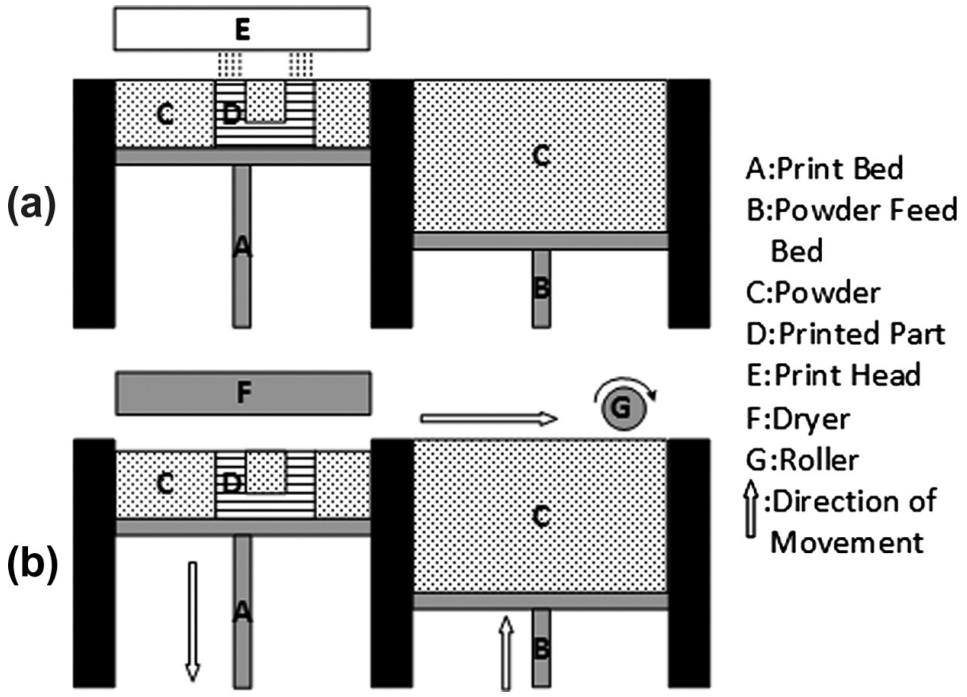


Figure 2.1: Working mechanism of Binder Jetting [1]: (a) rinting, and (b) drying and spreading.

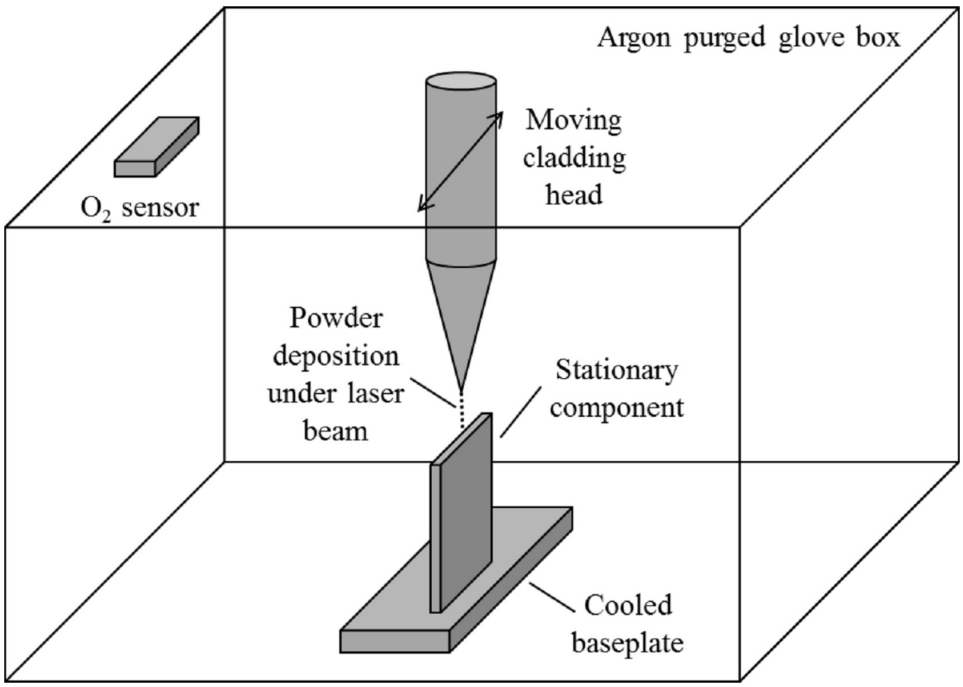
technique resembles metal welding. By using multi-axis robotic arms, the metal wire or powder can be fed and melted from nearly any angle, which makes it very convenient for repairing. Depending on the material size and power of the heat source, DED can print rapidly, which makes it suitable for printing very large parts with loose tolerances.

Material extrusion 3D printing, also known as Fused Filament Fabrication (FFF), is an additive manufacturing (AM) technique achieved by extruding melted thermoplastic or pliable material on a layer-by-layer basis (Fig. 2.3). Material is first fed into a heated nozzle and then extruded to a stage that is relatively moving regard the nozzle. Once a layer is being printed out, the print head and stage will move away from each other and start printing the new layer. Since the new layer is printed on the last layer, the overhang structure with a large negative slope needs proper support. Compared to processes that ultilize powder, material extrusion is often used to generate hollow structures with different levels of infills.

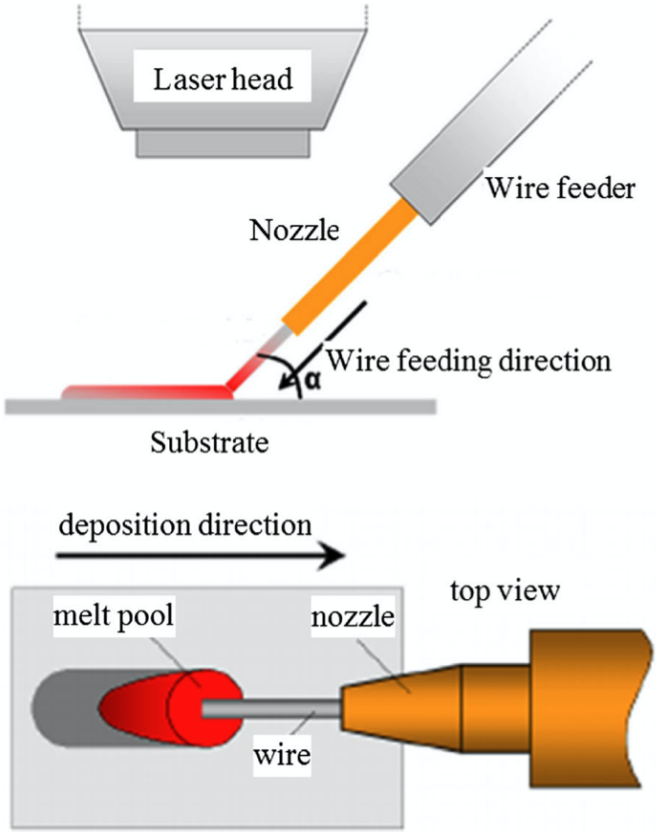
Material Jetting is a 3D printing technique that is very similar to 2D inkjet printing. Unlike 2D inkjet printer jets ink, material jetting 3D printer deposits drops of photosensitive viscous liquid resin. The liquid resin is first warmed up to achieve optical viscosity. Then, tiny droplets of material are deposited and solidified by UV light. Like the 2D color inkjet printer, the material jetting printer also has a line of print heads for different kinds of materials (like structural and support materials) (Fig. 2.4) or materials with different colors. Although the polymerization process solidifies the liquid material, no post-curing is necessary for the material to achieve its optimal properties since the layer height (droplet size) is very small.

Powder Bed Fusion (PBD) is a type of additive manufacturing process that uses a heat source (either laser or electron beam) to fuse the top layer of the preheated powder together (Fig. 2.5). Similar to Binder Jetting, the powder bed is recoated and a 3D part is printed on a layer-by-layer basis. PBD has the ability to print plastics, metal, ceramics, and glass.

Sheet Lamination is an additive manufacturing process that bonds sheets or ribbons of metal (or paper) together using ultrasonic welding (or adhesive) before or after the required shape is cut from the layer by CNC, laser, or knife (Fig. 2.6). This process is also a layer-by-

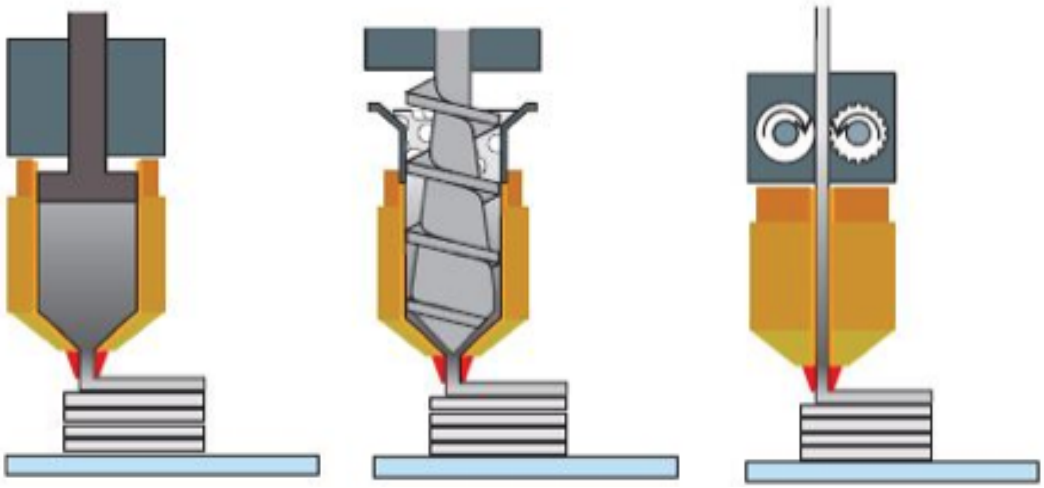


(a)



(b)

Figure 2.2: Working mechanism of Directed Energy Deposition: (a) powder feedstock [2], and (b) wire feedstock [3].



(a) (b) (c)

Figure 2.3: Working mechanism of Material Extrusion [4]: (a) syringe-type, (b)s crew-type, and (c) gear-type print head.

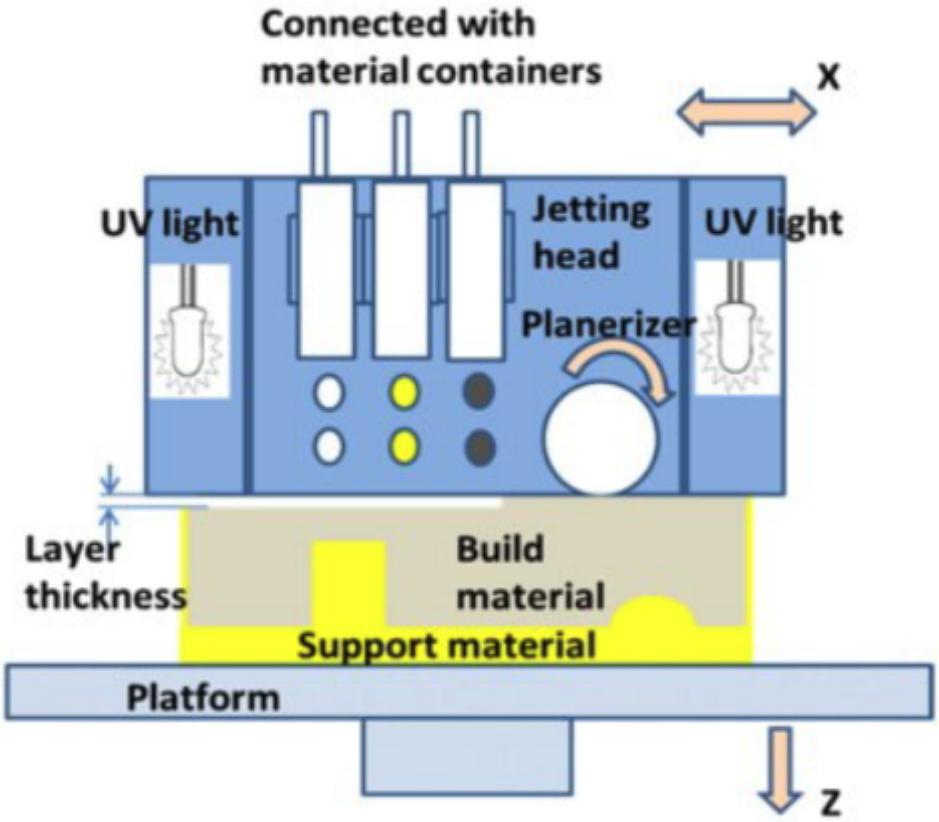


Figure 2.4: Working mechanism of a multi-material jetting 3D printer [5].

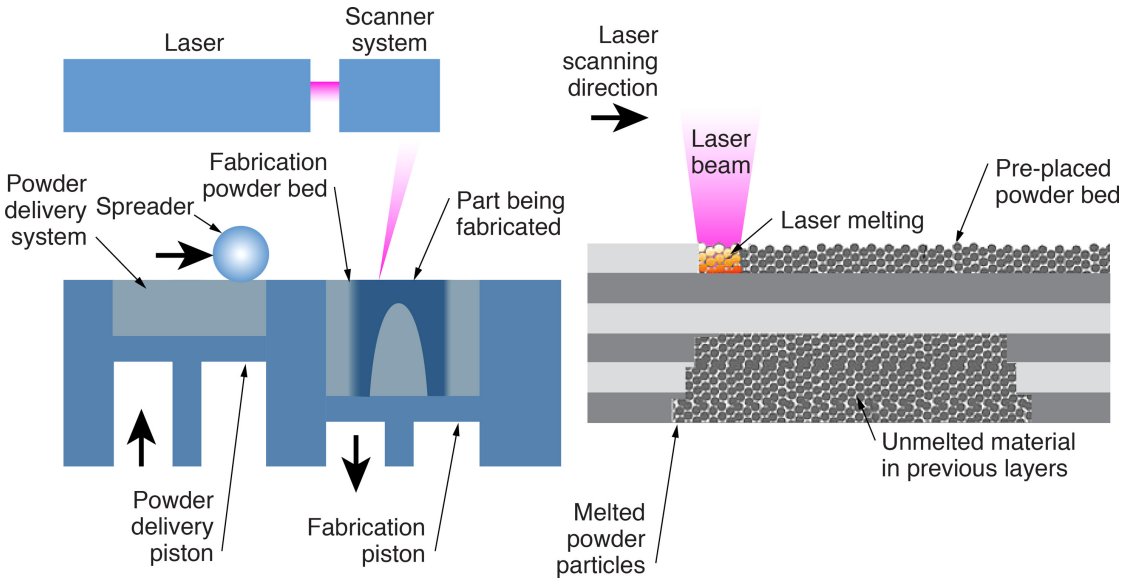


Figure 2.5: Working mechanism of Powder Bed Fusion [6].

layer process and each layer experience a cutting and bonding process. The Sheet Lamination process is low temperature and consumes little energy.

Vat Photopolymerization is another additive manufacturing process that uses liquid photopolymer. Unlike Material Jetting, the photopolymer is cured in a vat of liquid photo resin instead of a tiny photopolymer in air. There are many heat sources are used to cause the solidifying of a polymer due to cross-linking or degrading, including point light source (like focused UV light, laser) and plane light source (like digital projector, Liquid Crystal Display). There are two approaches in the Vat Photopolymerization (Fig. 2.7). The top-down approach puts the light below the resin tank, and the part is printed upside down on the bottom surface of the build platen. The bottom-up approach puts the light source above the vat and the part is printed on the top surface of the build plate. Vat Photopolymerization is capable of printing parts with high resolution and smooth surfaces. Thus, it is widely used in medical modeling to make anatomical replicas based on computer scans. However, parts that are fabricated by Vat Photopolymerization need post-printing cleaning processes to get rid of the extra liquid resin. Similar to PBF, Vat Photopolymerization cannot print void enclosure since liquid resin will be trapped inside.

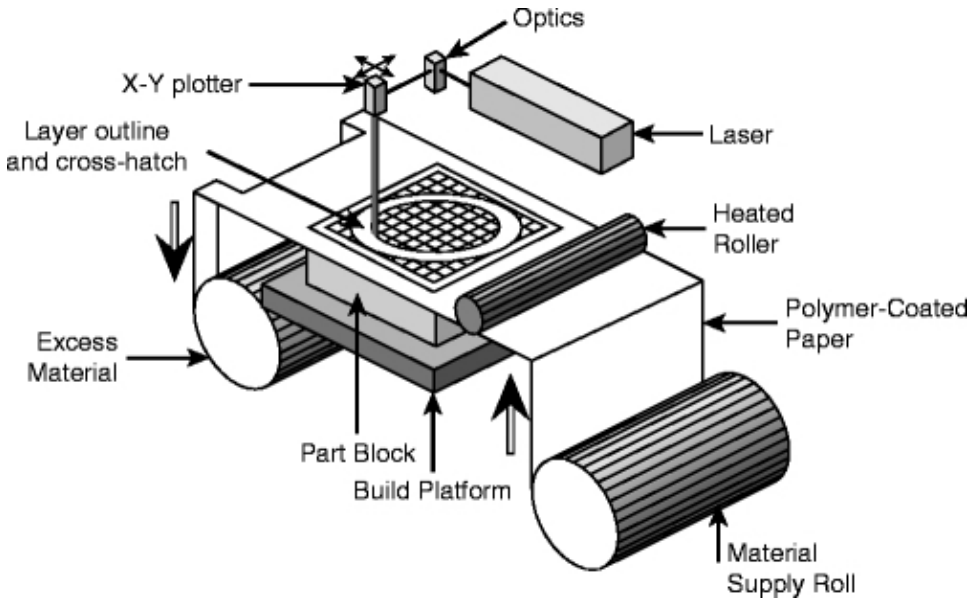


Figure 2.6: Working mechanism of Sheet Lamination [7].

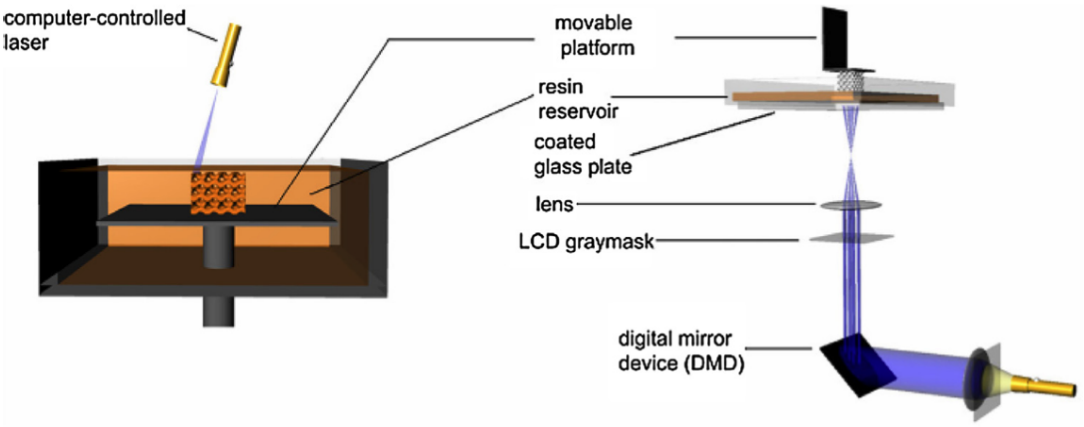


Figure 2.7: Working mechanism of Vat Photopolymerization: (Left) Bottom-up approach, (Right) Top-down apprach. [8, 9].

2.2 Material Extrusion 3D Printing

Material extrusion, especially the Fused Deposition Modeling (FDM) process, was first invented by Scott and Lisa Crump in 1988 to build their daughter a toy frog. Later, they patented the FDM technology and became the founders of Stratasys in 1989. Due to the constraint of trademark "Fused Deposition Modeling", "Fused Filament Fabrication" was used by the members of the "RepRap" project, which is in the form of a free desktop 3D printer that can replicate itself by printing its plastic parts. After the commercial patent on FDM [34] expired in 2009, many consumer-friendly low-cost FDM-based 3D printer communities and manufacturers emerged.

A diverse set of materials can be extruded by Material Extrusion 3D printing. Among them, the most widely used has being thermoplastics, such as Acrylonitrile Butadiene Styrene (ABS), PolyLactic Acid (PLA), High Impact Polystyrene (HIPS), Thermoplastic PolyUrethane (TPU), Polyethylene Terephthalate (PET), PolyEther Ether Ketone (PEEK), and Polypropylene (PP). As opposed to "thermoset", thermoplastic materials liquefy when heated, and this process is completely reversible. The more heat is directed, the less viscous they are. By contrast, thermoset plastics can only be heated once and will not be softened before being burned when heated again. Table 2.2 compares the common thermoplastic materials used in Material Extrusion 3D printing.

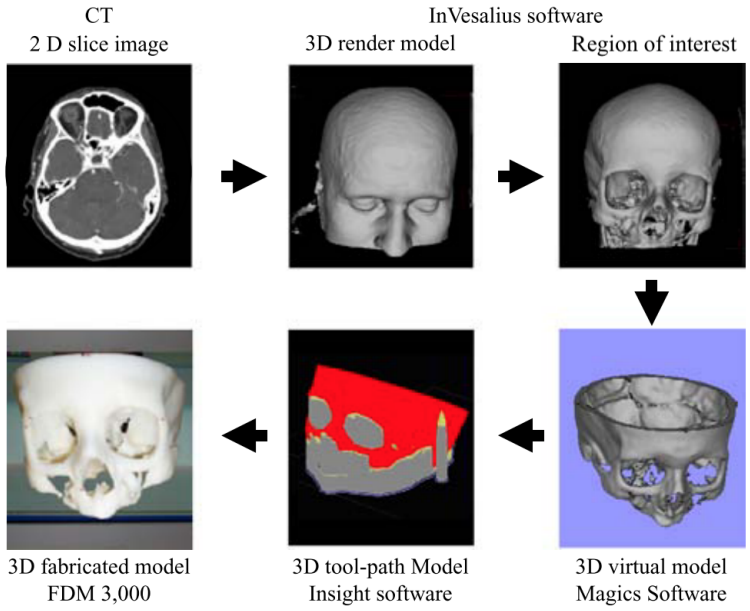
Material extrusion 3D printing is well-established for rapid prototyping and functional testing in the last twenty-five years. It has been widely used in automotive, aerospace, biomedical engineering, dentistry, consumer products, food industry, and the form tooling and casting areas.

Material extrusion 3D printing has been widely used to make complex anatomical replicas for medical education and surgical planning [10, 35, 11, 36, 37]. The 3D printed complex anatomical replicas, which were derived from computed tomography (Figure 2.8(a)), made a viable role in reconstructive surgery. El et al. [10] fabricated skulls for different human sizes (infant, female, or male) and evaluate the accuracy of those models. Smith et al. [11] used

Thermoplastic Materials	Pros	Cons	Applications
ABS	impact resistance, resistance to corrosive chemicals, very easy to machine, great electrical insulation, low cost, lost cost (∼$1.5/pound)	weatherability, solvent resistance, hazardous when burned	LEGO bricks, keyboard caps, tool housing
PLA	biodegradable, does not release toxic fumes when burned	slow rate of biodegradability when not controlled, unable to mix with other plastics in recycling	food packaging, medical applications
HIPS	low cost, impact and water resistant, lightweigth, dissolvable, dimensianlly stable	high printing temperature, ventilation required	support material used with ABS, coutertop point of purchase displays, indoor signs
TPU	rubber-like elasticity, thermal stability, resistant to chemicals and solvents, low warpage and shrinkage	hygroscopic, prone to clogging extruders, difficult to post-process	vibration-damping and shock absorption components
PET	recyclable, low cost, durable	low heat tolerance	FDA-approved packaging for food, beverages, medicines
PEEK	exceptional chemical resistance to organics, acids and bases, high mechanical strenth at high temperture (>250 °C), very good resistance to wear, abrasion, and fatigue, excellent dimensional stability	expensive (∼$60/pouond in raw form)	aircraft components, wear and load bearing applications
PP	low density, high flextural strength, loow coefficient of friction, reistant to moisture, chemicals, and fatigue, good impact strenth, good electrical insulator, easy to repair	high coefficient of thermal expansion, poor resistance to solvents, UV and scratches, difficult to paint, high flammability	food packging, consumer goods, automotive, medical applications

Table 2.2: Comparison of common thermoplastics used in Material Extrusion 3D printing.

FilaFlex flexible filament and PLA filament to print the vocal cords with movable arytenoids on the cricoid cartilage (Figure 2.8(b)), which can illustrate the action of intrinsic laryngeal muscles and offer rich possibilities for anatomy. This approach unchained the anatomist from restrictions of inflexible hard models.



(a)

(b)

Figure 2.8: (3D printing of anatomical models for education. a) fabrication process of the skull replica using CT and material extrusion 3D printing [10]; b) muscles, bones and cartilage of upper airway and neck [11].

Material extrusion 3D printing has also been used for devices/scaffolds for clinical reconstruction and implants [38, 39, 40]. Kirby et al. [40] used micro-extrusion technology to

print a biodegradable scaffold and load it with various kinds of growth factors. The scaffold was implanted in a sheep to reconstruct a critical size bone defect as shown in Figure 2.9.

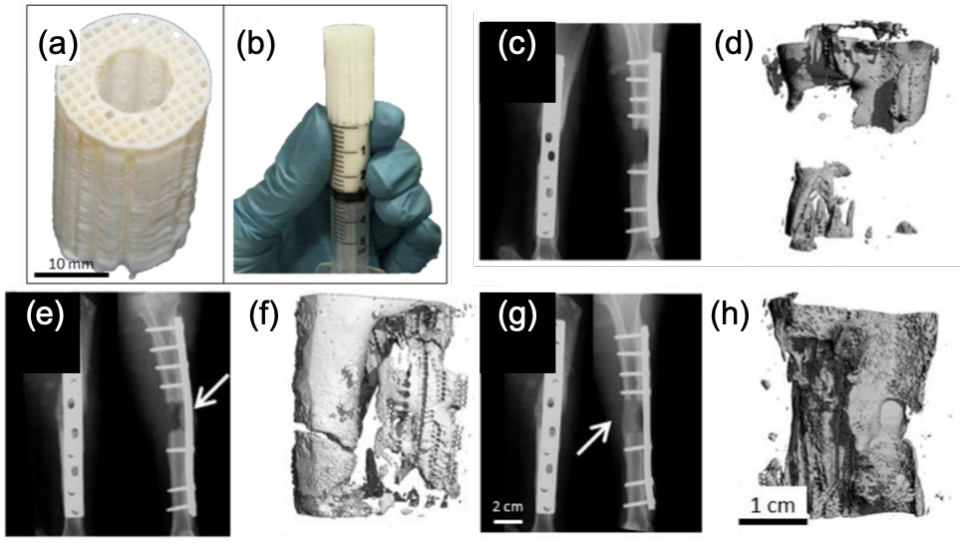


Figure 2.9: Use 3D printed scaffold in the regeneration of critically-sized bone defects. a) A 3D printed polycaprolactone scaffold with honeycomb pores larger than 500 microns. b) specific mixture was injected into the scaffold. Selected images of X-rays and CT show the reconstructions of the implant at 6 months: c) and d) the defects failed to bridge without growth factor inclusion; e)-h) different levels of bridging were observed with groups of specific growth factor inclusion.

Besides, material extrusion 3D printing is well-established for pharmaceuticals for precision medicine applications [41] and concrete for construction [42].

Notwithstanding the popularity of material extrusion 3D printing, the inconsistency of printing quality and intrinsic defects (surface roughness and geometric inaccuracies) hinders its potential applications to not only the "certify-as-you-build" small-batch prototyping but also large-batch production.

Compared to the popular plastic-based material extrusion process, metal extrusion is a fairly new process. Filament for metal extrusion is made of thermoplastic material and metallic particles. The printing process is the same but post-printing sintering is required to burn out the plastics and melt the metal particles together. Although metal extrusion has some limitations: the high percentage of the polymer as binder makes sintering difficult; part shrinks a lot due to binder content burnt out in the furnace; metal parts has a lower density

compared to solid material caused by small void from deposition and sintering processes. Metal extrusion still carries forward the advantages of plastic-based material extrusion, such as very low cost and fast build times compared to PBF or DED, which make it extremely suitable for rapid prototyping.

2.3 In-situ Monitoring

In situ monitoring of the printing process is an effective way to eliminate inconsistency of print quality. The prevalent methods used for *in situ* monitoring are machine vision [12, 15, 14, 43, 13, 16, 44, 45, 46, 47, 48, 49, 24], heterogeneous sensors [17, 50, 19, 51, 14, 52, 53, 20, 21, 18, 54, 55, 56], and acoustic emission [22, 57, 58, 59, 60, 61, 23].

Those methods are employed to observe nozzle clog [59, 17, 46, 55], stage and nozzle motion (position, velocity, acceleration, and vibration) [20, 14, 19, 18, 51, 53, 56], as well as stage and nozzle temperature [53, 51, 19, 21, 56].

Statistical methods [16, 22, 61, 43, 12, 59, 62] and Machine Learning (ML) [48, 52, 14, 60, 23] classification methods have been applied to decide whether a printer is working normally or not.

2.3.1 Machine Vision

Machine vision is the technology providing image-based automatic inspection and analysis [63]. This goal can be achieved by using a single camera. Baumann et al. [12] used a single consumer-grade-video-camera to recognize and monitor the relative motion between the extruder and the printed part. Error alert will be triggered if 1) the extruder is too far away from the printed part when it should not be, 2) the extruder is moving too fast, 3) filament is not recognized when it is missing (Figure 2.10(a)), or 4) the printed part is not recognized due to deviation from the model. Nuchitprasitchai et al. Lyngby et al. [13] also implemented a single camera to take images of the printed workpiece in real-time. Since the camera was calibrated, its location and orientation were used with the CAD model of the

object to generate a rendered image of the workpiece. Then, the real images were compared with the rendered images (Figure 2.10(b)) in real-time to determine if the printing process is abnormal. Roberson [14] applied an online image-based approach to analyze and quantify the surface roughness of specimens (Figure 2.10(c)) produced by cylindrical turning with an average prediction error of less than 8%.

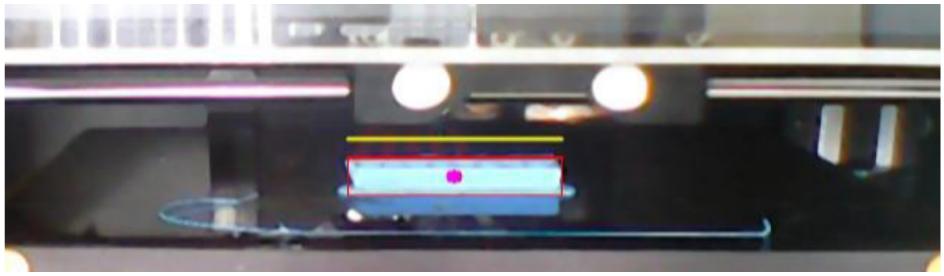
Nuchitprasitchai et al. [15] used two cameras to build an open-source low-cost reliable platform. This approach reconstruct a 3D model of the printed part using two 2D images (Figure 2.11(a)). This system is capable of monitoring the printing process and determine if the printing is normal or failed in real-time. Liu et al. [16] mounted two high-speed digital microscopes close to the extruder to achieve effectively monitoring of the as-printed surface quality during the entire printing process (Figure. 2.11(b)).

However, there are several limitations of machine vision methods. First, it requires a lot of computational power to analyze the tens of high-resolution images per second. Second, object detection is very sensitive to the change in lighting. The reflections due to movement of the stage or extruders can destabilize the object detection algorithms. Third, machine vision can only be used on materials that have a different color and texture than the background.

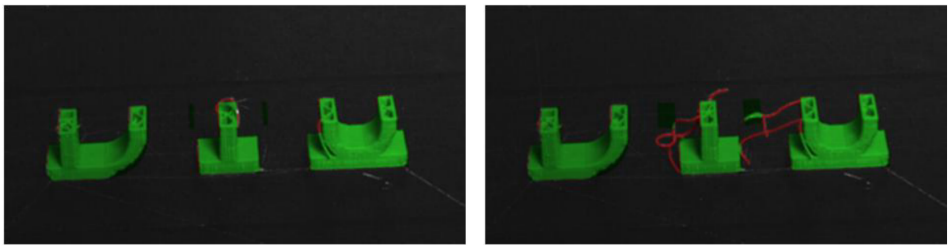
2.3.2 Heterogeneous Sensors

Vibration sensors (or accelerometer) have been widely in 3D printing monitoring. Tlegenov et al. [17] used a single vibration sensor to detect the feed force on a bar mount of the extruder as shown in Figure. 3.17. By extracting the natural frequency and acceleration from the data, he successfully detected the nozzle clogging. Li et al. [18] instrumented two vibration sensor (7251 A-500, single channel with 500 mV/g output on the build platform and 65-10, three channels with 10 mV/g output on the extruder) to *in-situ* monitor the printing process (Figure. 2.13). By using the least squares support vector machine algorithm, the nozzle clogging state was identified in real-time with an accuracy greater than 90

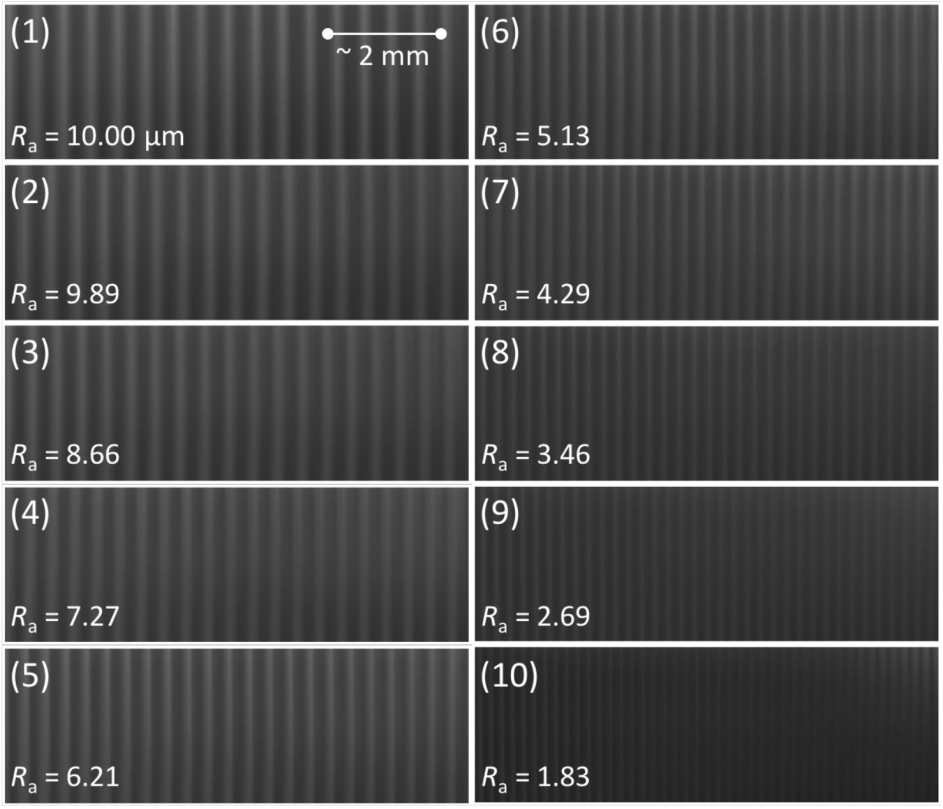
In addition to the use of a specific sensor, a heterogeneous sensor array was also implemented in a material extrusion 3D printer to collect data from a diverse set to sensors. Rao



(a)

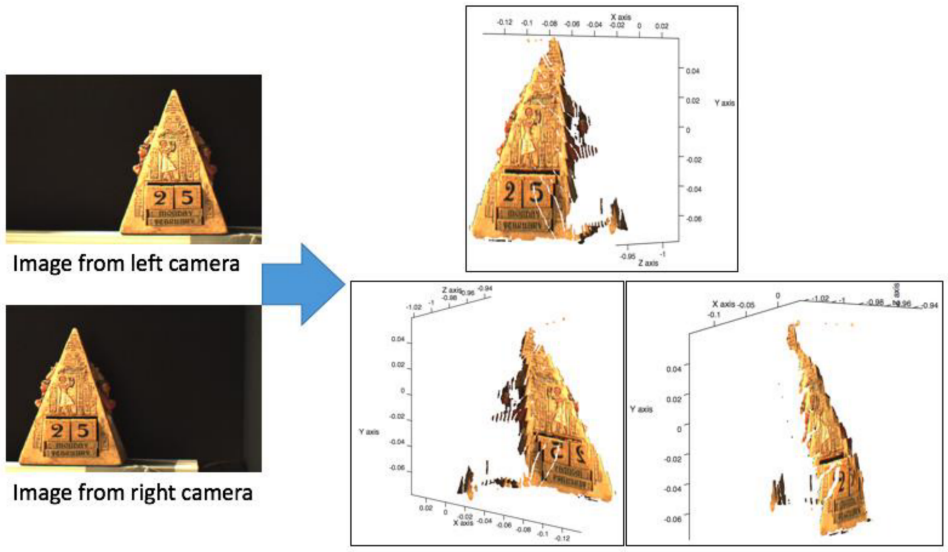


(b)

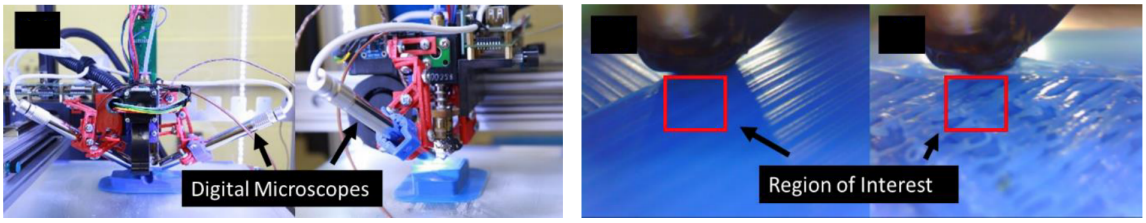


(c)

Figure 2.10: Use a single camera for *in-situ* monitoring. a) The filament is missing between the extruder reference line (yellow) and printed object bounding box (red) [12]. b) Images show the similarities (green) and differences(red) between the real images and the rendered images on layer 71 and layer 73 [13]. c) Correlation of images and surface roughness [14] .



(a)



(b)

Figure 2.11: Some applications of machine vision for *in-situ* monitoring using two cameras. a) Reconstruction of a 3D model from two 2D images [15]. b) Real-time online monitoring of surface quality [16].

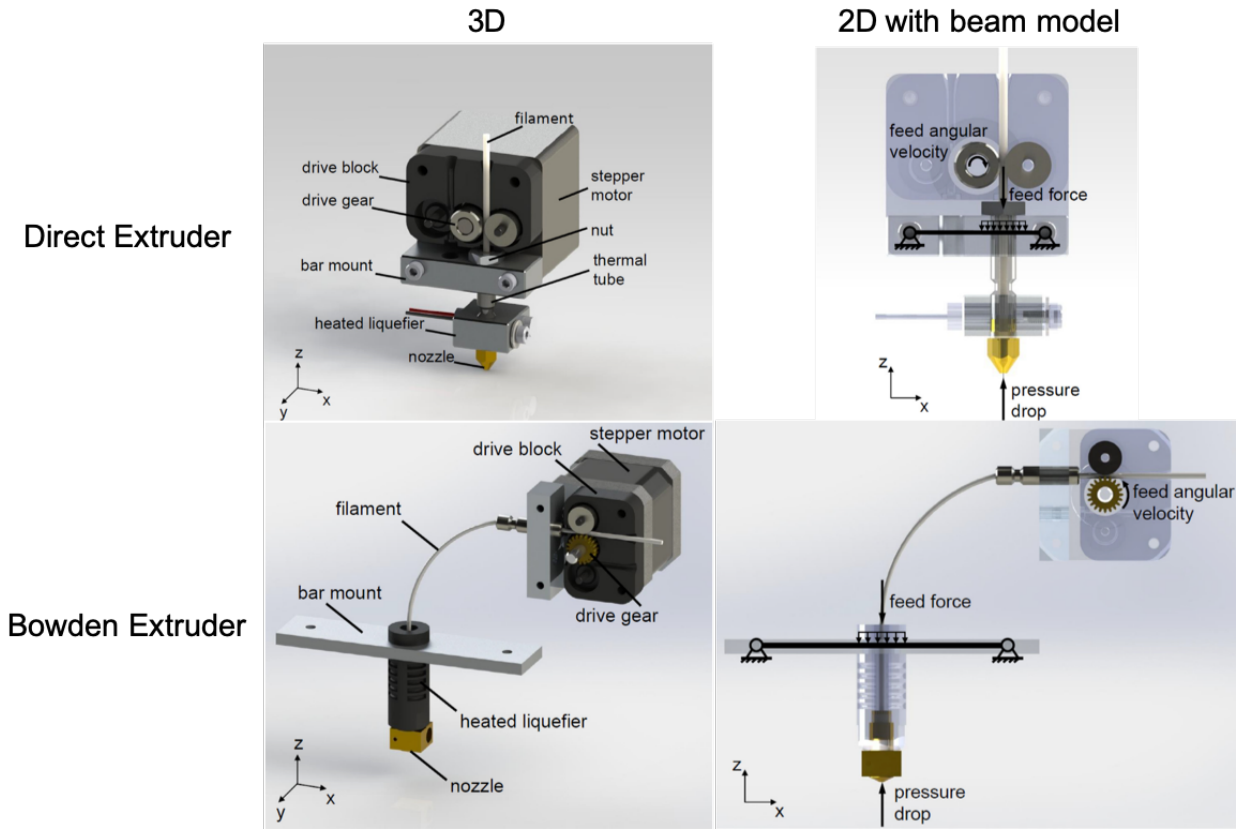


Figure 2.12: Schematic of the two types of the extruder and the analogical beam model for feed force detection using a vibration sensor [17].

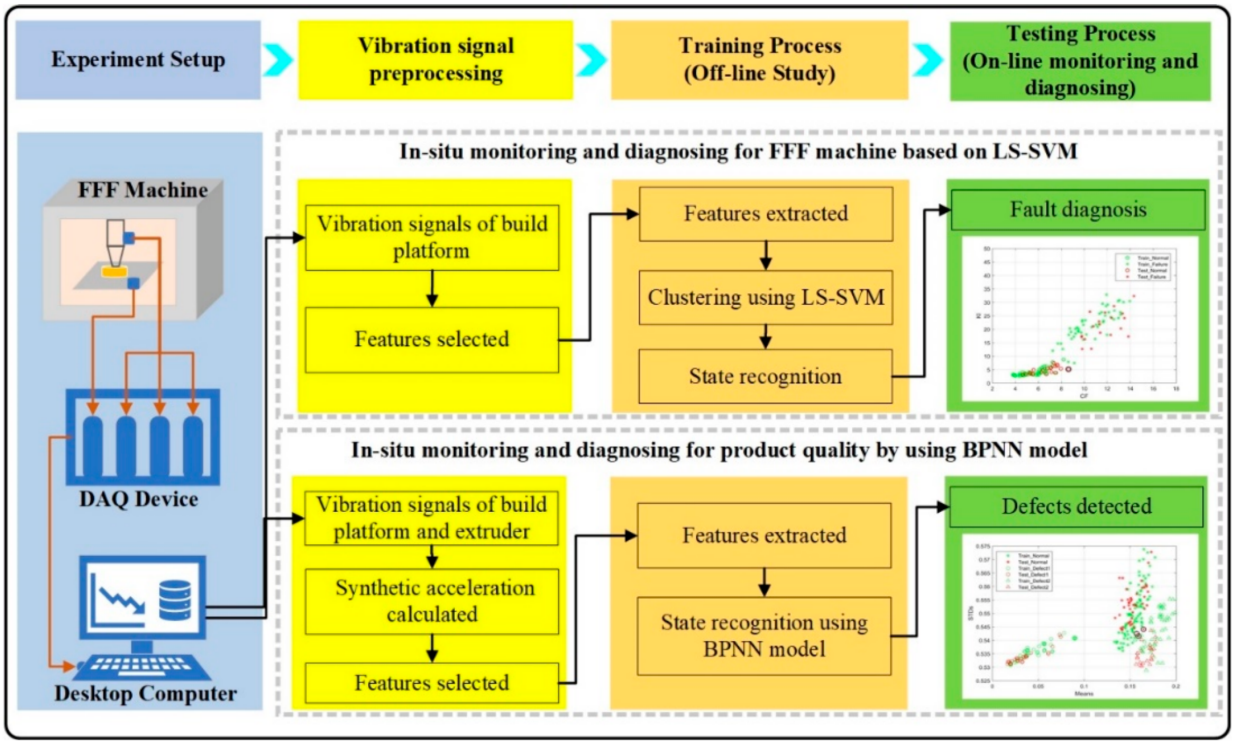
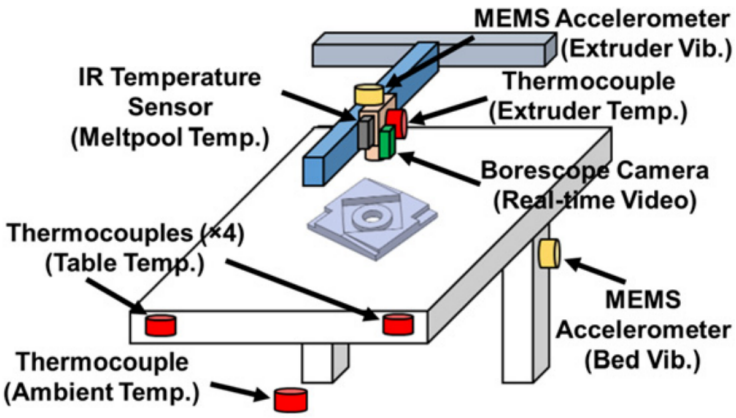
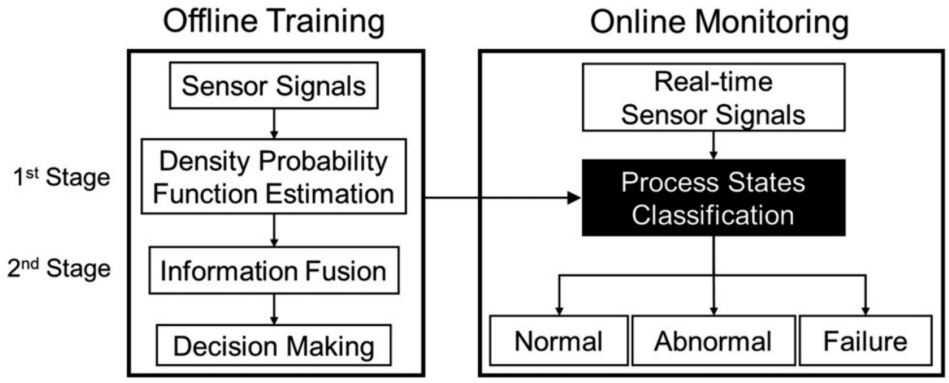


Figure 2.13: Flow chart of *in-situ* monitoring of printing process using two vibration sensors [18].

et al. [19] instrumented commercially available thermocouples, accelerometers, an infrared temperature sensor, and a video borescope to identify printing failures online in real-time. A nonparametric Bayesian Dirichlet process mixture model and evidence theory (Figure. 2.14) were used to analyze the data and detect the printing anomalies and failures with an average accuracy of 85%. Long et al. [20] instrumented a low-cost attitude sensor to a Delta 3D printer. The attitude sensor is capable of providing three-axial angular velocity, vibratory acceleration, and magnetic field intensity signals of a moving object. They employed a SEAEN approach, which combines an echo state network (ESN) with a sparse autoencoder (SAE), to extract features from the high dimensional data and diagnose fault state of the 3D printer (Figure. 2.15).

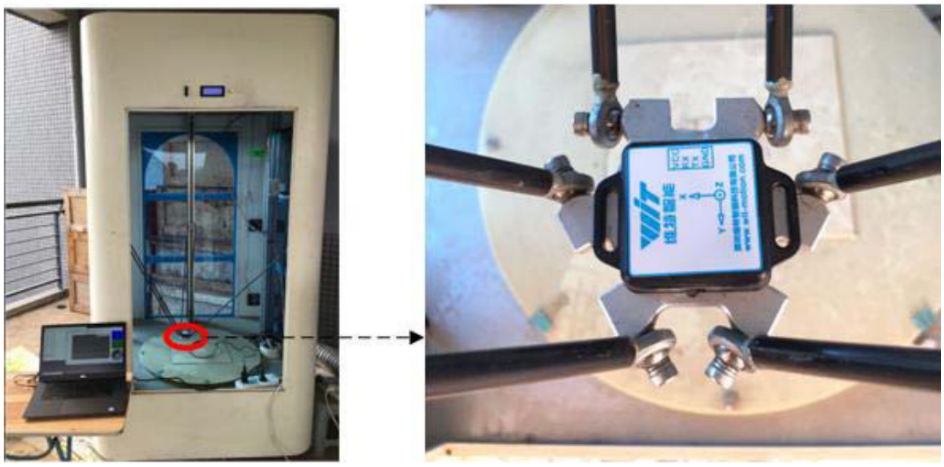


(a)

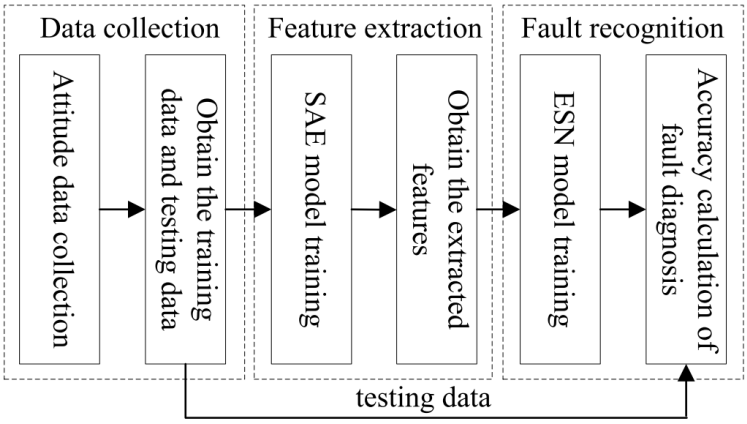


(b)

Figure 2.14: An application of the heterogeneous sensor array. a) Schematic of the printer setup with multiple sensors; b) Flow chart of the algorithm used in [19].



(a)

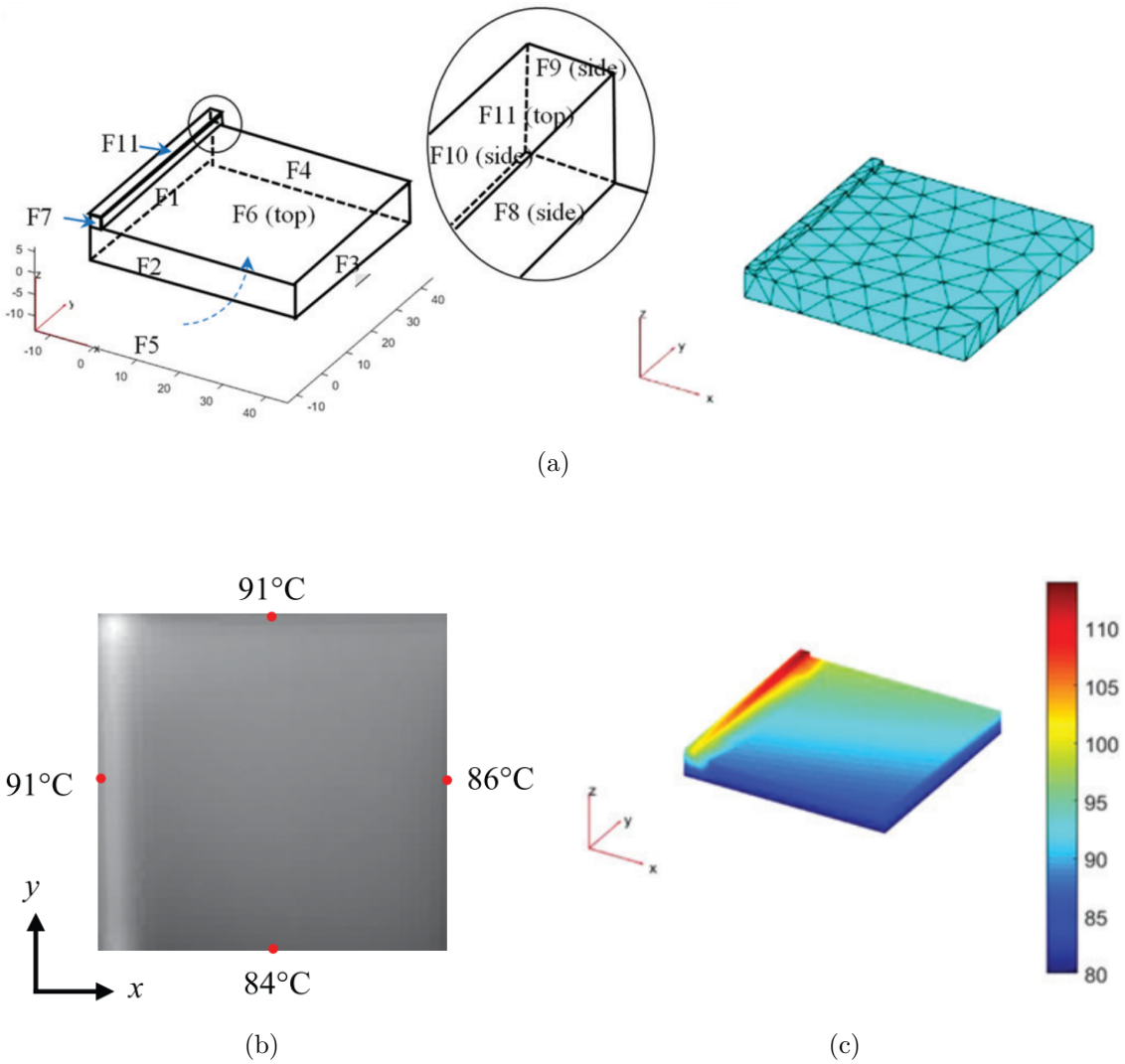


(b)

Figure 2.15: a) Experiment setup of the attitude sensor and Delta 3D printer. b) The framework of the SEAEN approach in [20].

Various kinds of sensors have been employed in *in-situ* fault diagnostics of the 3D printer. However, as a result, the cost of the overall monitoring system is rising, let alone the difficulty and feasibility to implement the sensor array to your own 3D printer. In addition to the cost, the real-time monitoring requires strong computational power, which becomes another obstacle for using a heterogeneous sensor array. Instead of processing high dimensional data, Lu et al. [21] reported a Physics-Based Compressive Sensing (PBCS) approach to reduce the amount of data collection by reconstructing the full data from sparse data points in both spatial and temporal domains. They reconstructed the thermal field of a printed part with

a data compression ratio of up to 163.96 and an average error of up to 2.54% (Figure. 2.16).



(a)

(b)

(c)

Figure 2.16: Use of the Physics-Based Compressive Sensing approach to reconstruct the trasient temperature field from sparse samplings [21]. a) The schematic and mesh model of the workpiece. b) The temperature at four points was measured. c) An example of the reconstructed temperature field.

2.3.3 Acoustic Emission

Acoustic emission is the emission of acoustic/elastic waves in solid material when the irreversible changes (like cracking or plastic deformation) occur due to temperature changes or external mechanical forces. Li et al. [22] used the acoustic emission approach to identify the

warping defects by sensing the elastic waves in real-time. The prediction results show strong correlation with the printing states (Figure. 2.17).

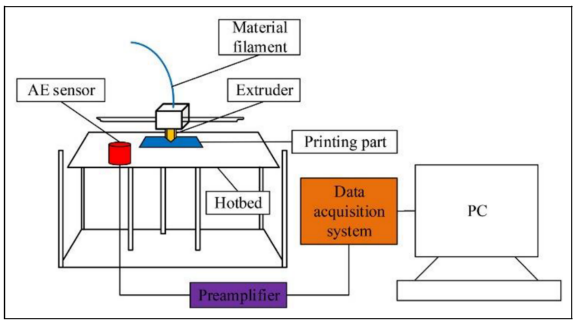
Although acoustic emission is widely used in the metal processing industry, it still requires complicated signal processing to remove noise and extract features (Figure. 2.18). And results are usually indirect and qualitative (not quantitative), which hinders its application in additive manufacturing.

2.4 Closed-loop Feedback System

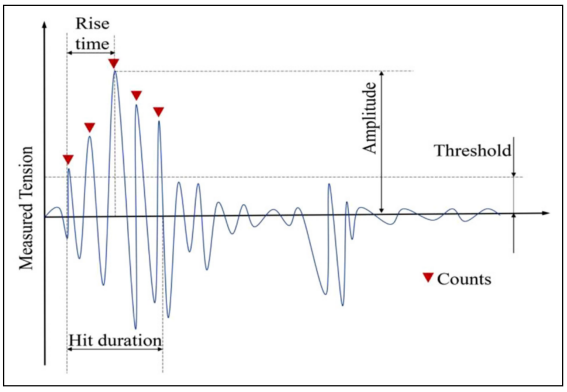
Most methods caught printing defects in real-time to abort the fruitless printing and save both time and material. A limited number of material extrusion 3D printing systems with closed-loop feedback control [16, 54, 55] have been reported to sustain the quality of the printing process.

Liu et al. [16] developed a machine-vision based closed-loop feedback system to analyze real-time images from two microscopes and improve the printed layer surface quality. By using a PID controller, this approach fixed an underfill defect by adjusting the feed rate to the optimal value after 6 additional layers of printing (Figure. 2.19). However, this approach only improved the top surface quality of the as printed part, but in reality, the largest factor that contributes to the surface quality of the overall part is the stair-step effect on the side surface.

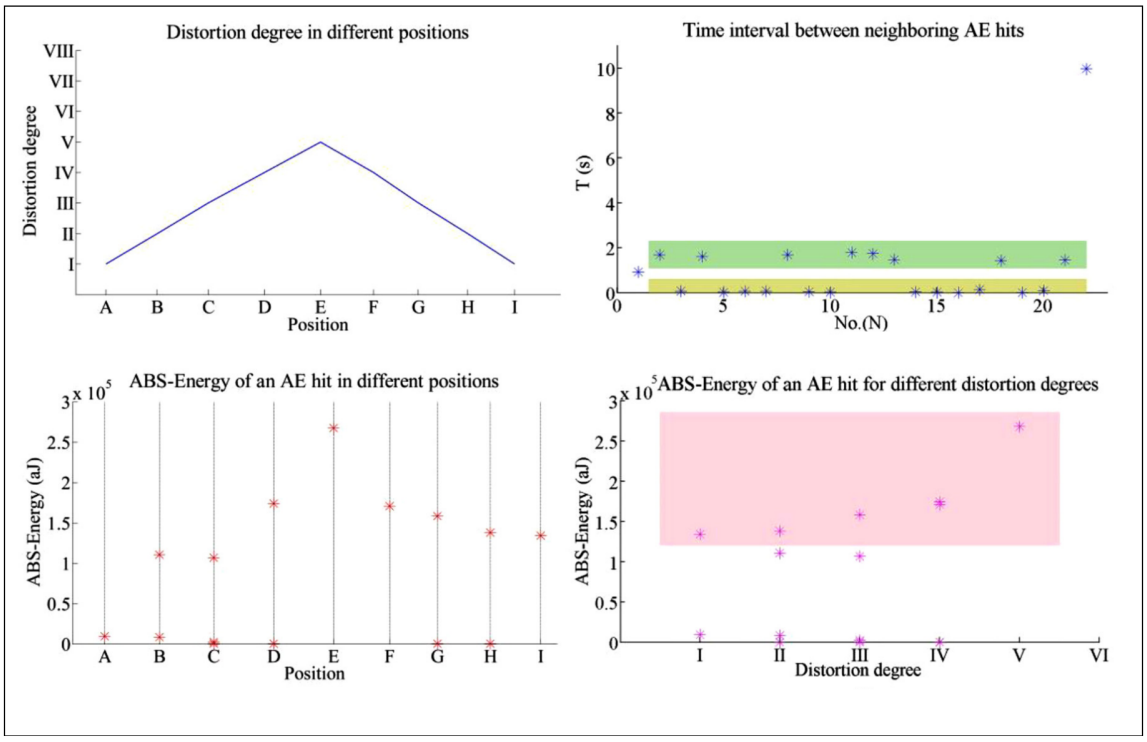
Sitthi-Amorn et al. [24] built a 3D printing platform with an integrated machine vision system, which allows for implementing a closed-loop feedback loop to realize print correction at the time of printing. The high-resolution imaging module was based on a full-filed optical coherence tomography, including a Michelson interferometer, with a scanning diameter of 15 mm and a resolution of 12.7 um/pixel (Figure. 2.20). By scanning the top surface of the printed part once every 15 layers, an additional correction layer was computed by comparing the measured depth map and the ideal surface. This correction layer was then sent to the printer to be printed out on top of the scanned layer.



(a)



(b)



(c)

Figure 2.17: Use of the acoustic emission approach to monitor the distortion area of the printed part in real-time [22]. a) The schematic of the system setup. b) An example of a detected elastic wave (AE hit) and the related features. c) The relationship between detected elastic waves and different distortion degrees of the printed part.

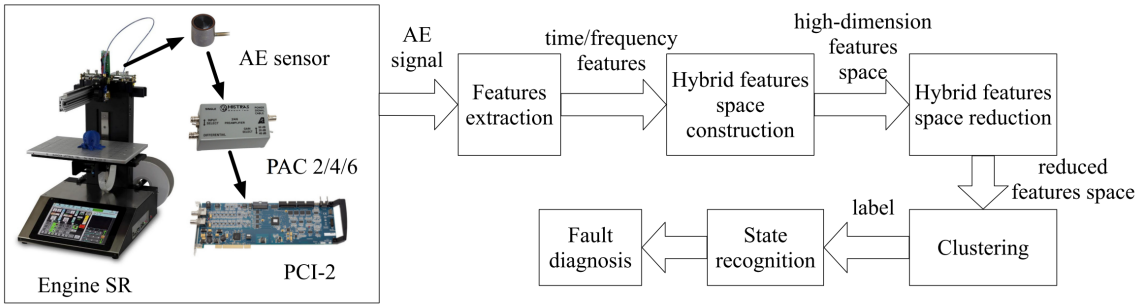


Figure 2.18: A flow chart shows the typical complicated data processing of signals obtained by acoustic emission [23].

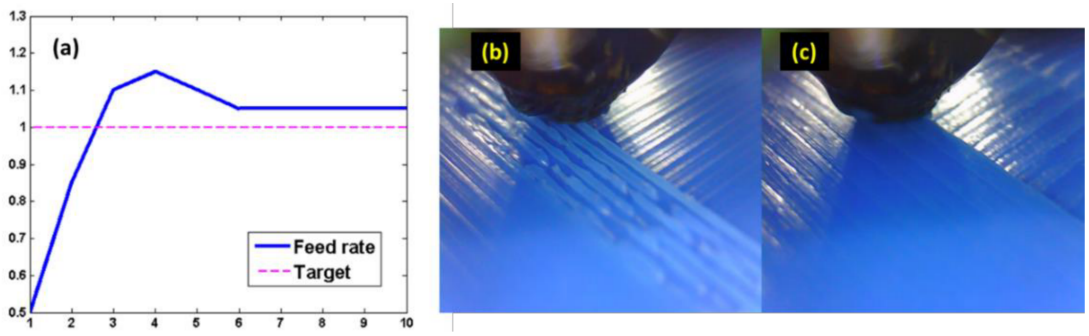


Figure 2.19: An online closed-loop feedback system to improve the surface quality of the printed layer. a) Controller performance; b) surface before adjustment; c) surface after the adjustment. [16]

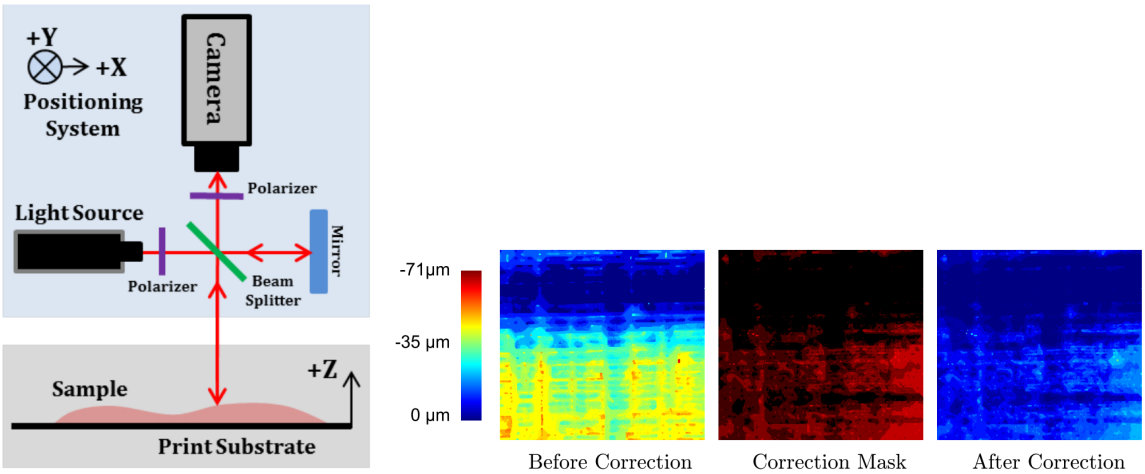
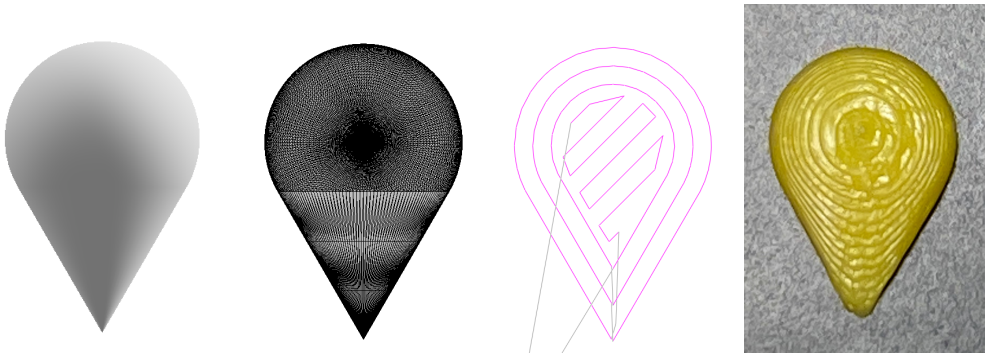


Figure 2.20: An example of a machine vision based closed-loop feedback system [24]. a) Schematic of the imaging module based on the optical coherence tomography. b) Depth maps of the printed surface before and after correction.

2.5 Predict Geometric Deviation From CAD

The methods in Section 2.3 can be used to *in-situ* monitor the printing process to eliminate inconsistency of print quality. However, even if printing conditions are assumed to be consistent between different printing sessions, material extrusion 3D printing still has trouble with inaccurate geometry and surface roughness caused by inexact geometry conversion and the layer-by-layer printing mechanism (Fig. 2.21).



(a) (b) (c) (d)

Figure 2.21: a) A "teardrop" shape [25] Computer-Aided Design (CAD) model was first translated to b) an approximate triangular meshed stereolithography (.stl) file, and then further converted to c) an estimated machine tool path (G-code). d) shows the round corner and stair steps caused by layer-by-layer material extrusion.

Both statistical models [64, 65, 66, 27, 28, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 26, 29, 82, 83, 84] and machine learning models [85, 86, 87, 88] were established to characterize and foresee the shape deviation.

Huang's group [26, 28, 27] demonstrated a methodology to predict and compensate the in-plane (x-y plane) geometric deviation of arbitrary freeform shapes from CAD model based on data of a limited number of cylinders and polyhedrons. The key points of this prescriptive modeling strategy are (Figure. 2.22): 1) transforming in-plane geometric errors from the Cartesian coordinate system into the polar coordinate system (PCS) [27]; 2) connecting the model of cylindrical shapes to those of polygons by butting them from their circumcircle [28]; 3) extending the strategy from cylinder and polyhedron to freeform shapes by making an approximation of the freeform shapes using either polygons with local compensation or a

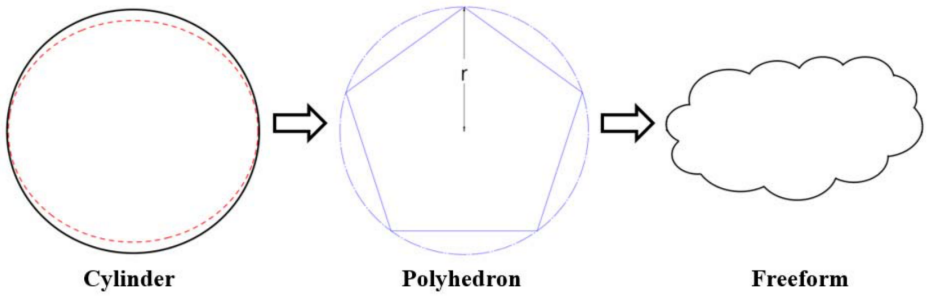
series of sectors with different radii [26]. The average shape deviations of the compensated freeform shapes were improved by 50%. Furthermore, Sabbaghi et al. [79] collaborated with Huang's group and presented a three-step adaptive Bayesian methodology to predict in-plane shape deformation of polygons and straight edges in freeform shapes by analyzing data from a small sample of previously fabricated cylinders and a single regular pentagon. The observed deviation fell in the 95% central posterior predictive intervals. Zhu et al. [87] worked with Huang's group and proposed a method that incorporated an affine transformation perspective and Gaussian process multi-task machine learning to represent the in-plane shape deviation of cylinders, regular pentagons and regular hexagons of three sizes.

Huang's group [29] also developed a prescriptive model for out-of-plane deformations caused by complex inter-layer interactions. Similar to the prescriptive modeling of the in-plane deformation, the out-of-plane was depicted in the spherical coordinate system. This model managed to describe the out-of-plane deformation of half-cylinder part and half-hexagon parts with limited accuracy (Figure. 2.23).

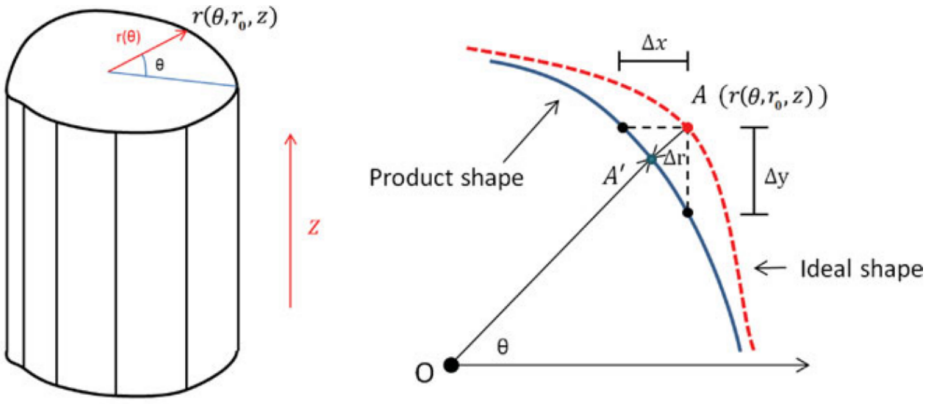
Ferreira et al. [88] further collaborated with Huang's group and presented Bayesian neural networks (NNs) that anticipated in-plane (freeform) and out-of-plane (vertical semi-cylinders) deviation separately with higher accuracy.

Nevertheless, these models are incapable of forecasting real 3D (both in-plane and out-of-plane) geometric deviation of freeform shapes. All of the 3D printed objects were scanned by a desktop scanner after the printing is done.

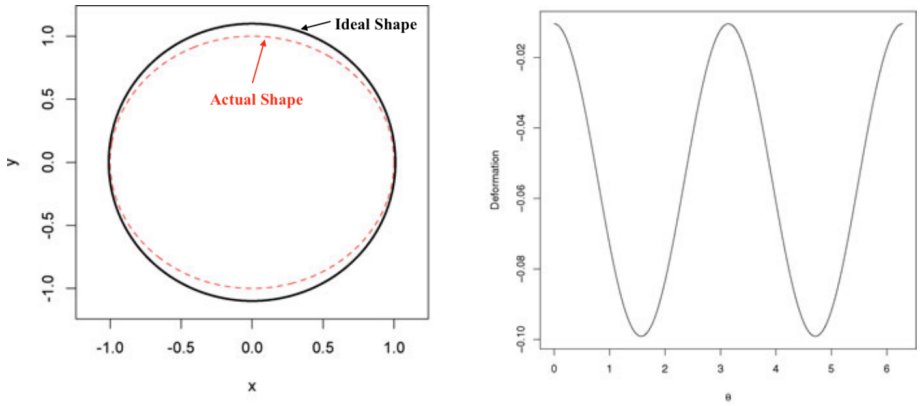
Huang's group [25] recently proposed a random forest machine learning model using a set of predictors generated by the triangular mesh .stl file to envision real 3D dimensional inaccuracies of the outer surface of freeform shapes. Eight predictors were extracted from the triangular mesh of the design. The first three predictors of a vertex are the x, y, and z coordinates of each vertex in the triangular mesh to represent the physical position of the printed part in a print bed. The next four predictors of a vertex are the median and the span of the elevation angle and azimuth angle from the normal vectors, which are expressed in spherical notation with radius 1, of the triangular faces adjacent to each vertex. These four


r
Cylinder
Polyhedron
Freeform

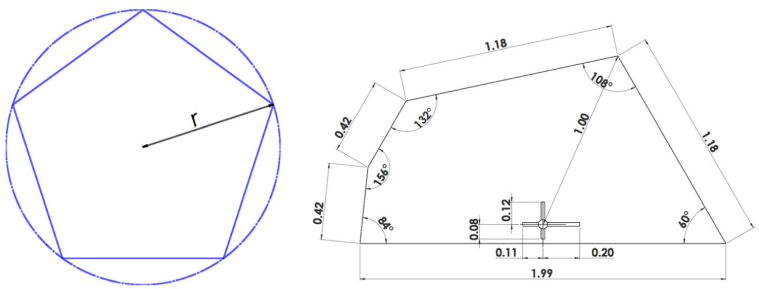
(a)


$r(\theta, r_0, z)$
r(θ)
θ
Z
Δx
$A\ (r(\theta, r_0, z))$
Product shape
A'
Δr
Δy
Ideal shape
θ
O

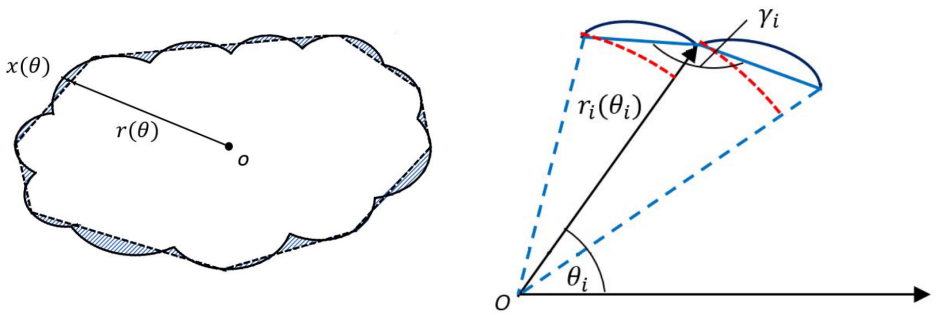
(b)


Ideal Shape
Actual Shape
y
x
1.0
0.5
0.0
-0.5
-1.0
Deformation
θ
-0.02
-0.04
-0.06
-0.08
-0.10
0
1
2
3
4
5
6

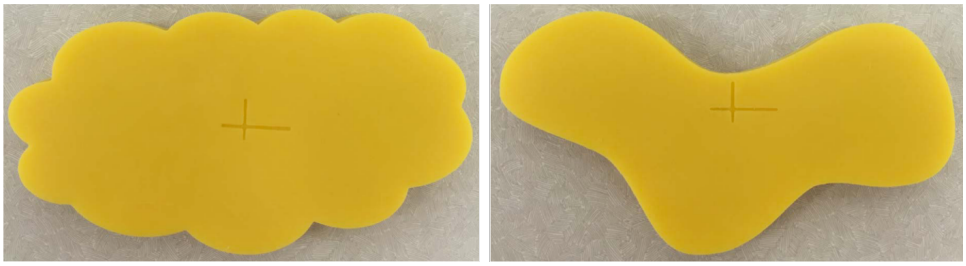
(c)


r
1.18
108°
132°
0.42
1.00
1.18
158°
0.42
88°
0.08
0.12
60°
0.11
0.20
1.99

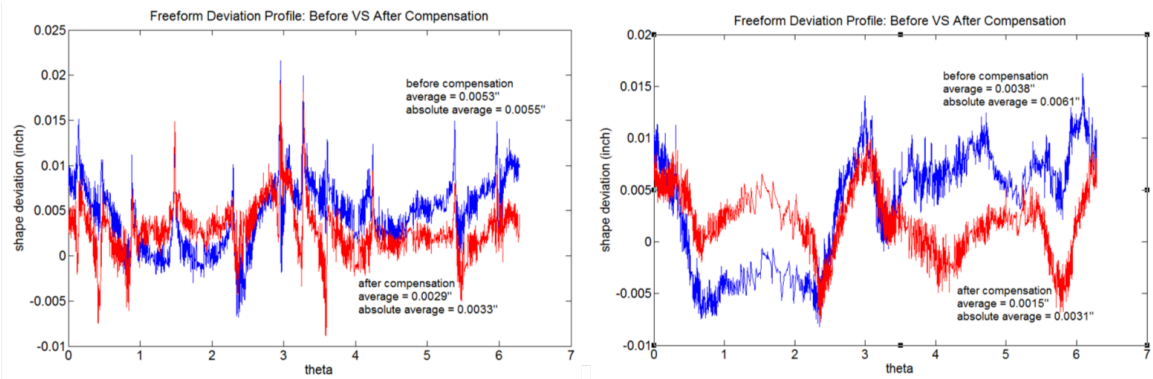
(d)



(e) (f)



(g)



(h)

Figure 2.22: A prediscriptive modeling methodology of in-plane (x-y plane) geometric errors of 3D printed freeform products. a) Schematic shows the idea to extend the modeling from cylinders to polyhedrons and further to freefrom shapes [26]. b) Represent the geometric error (shrinkage) of a cylinder under the polar coordinate system. c) Visualization of the geometric deformation in Cartesian coordinate system and the polar coordinate system [27]. d) Examples of polygons and their circumcircles [28]. e) Make approximation of a freeform shape using a polygon with local compensation. f) Make approximation of a freeform shape using a series of sectors with different radii. g) The printed convex and concave freeform shapes with circumcircle a radius of 2 inches. The cross marks the center of the related circumcircle. h) The shape deviation of the convex (Left) and concave (Right) freeform shapes before and after compensation [26].

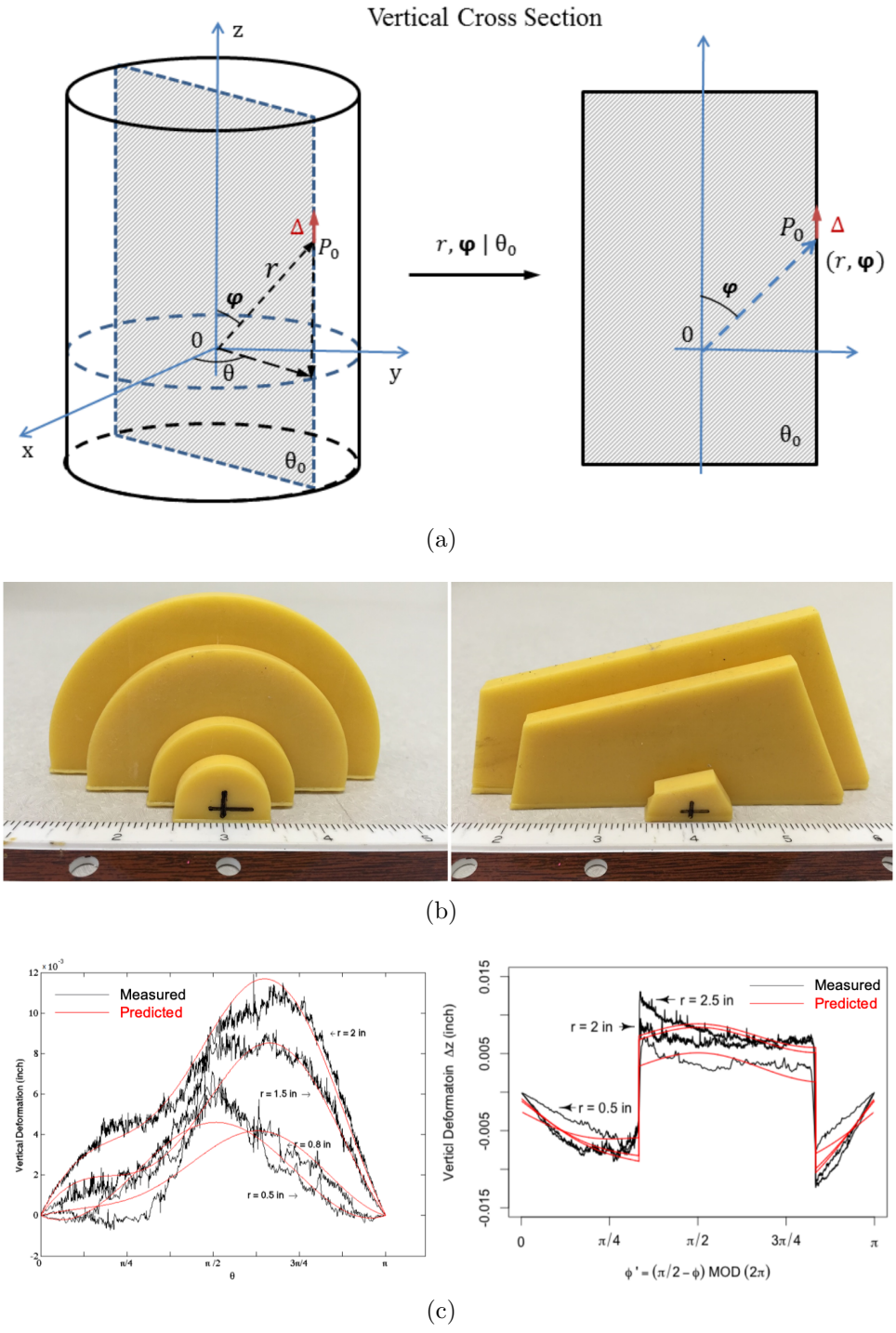
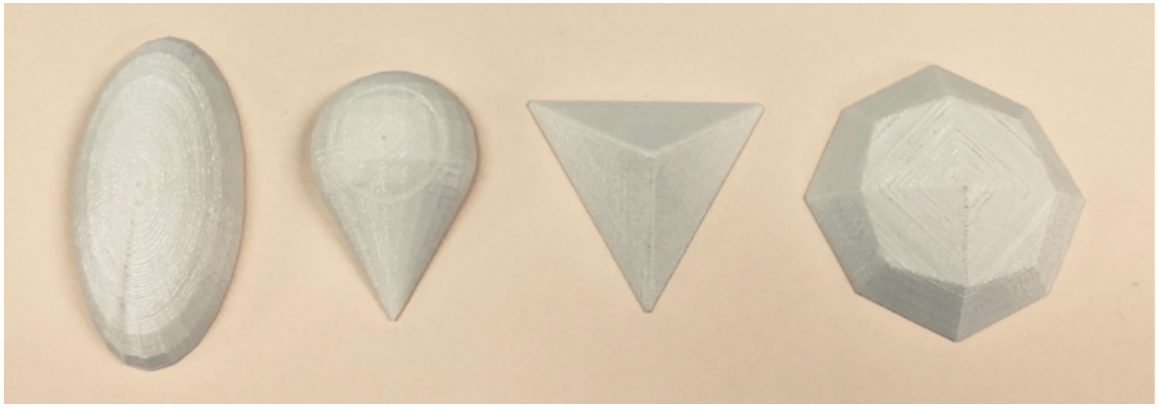
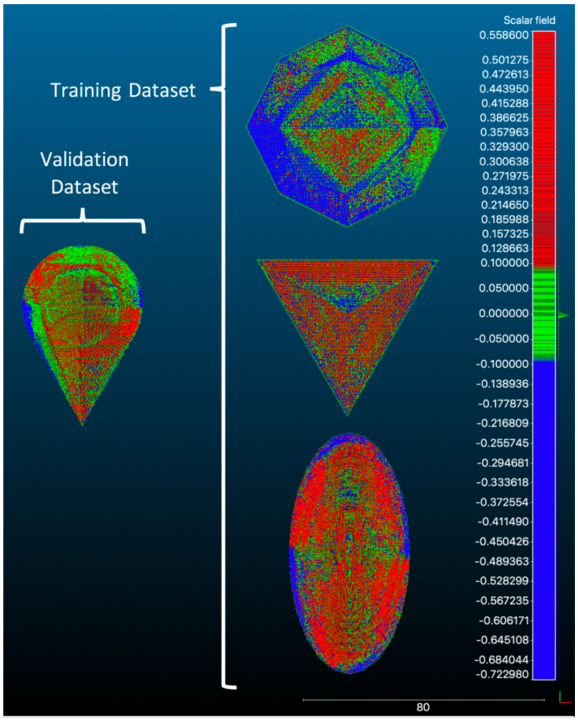


Figure 2.23: A prescriptive modeling methodology of out-if-plane geometric errors of 3D printed freeform products. a) Represent the geometric error (shrinkage) of a cylinder under the spherical coordinate system. b) (Left) Half cylinder parts with diameters of 0.5 in, 0.8 in, 1.5 in, and 2 in. (Right) Half hexagon parts with circumcircle radius of 0.5 in, 2 in, and 2.5 in. c) The measured and predicted out-of-plane deformation of the parts shown in b) [29].

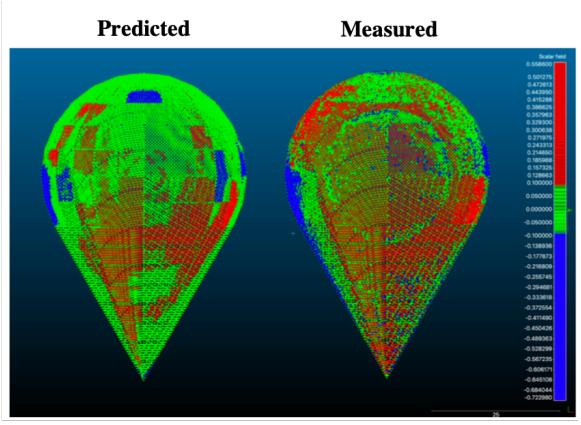
predictors show the significance of the orientation and curvature of the surface surrounding a given vertex. The last one predictor of a vertex is the distance to the z-axis which is placed at the center of each part. And this predictor represents the linear thermal expansion effects of the part when it is undergoing the phase change of the material during printing. Random forest method was used to make predictions of the 3D geometric deviation based on the predictors. The model produced fairly accurate predictions (38.3% of deviation predictions were within ± 0.05 mm of actual measured value) for a new shape from a training set consisting of just three shapes (Figure. 2.24).



(a)



(b)



Predicted – Measured (mm)	Percentage
±0.05	38.3%
±0.10	66.7%
±0.15	83.7%

(c)

Figure 2.24: Make predictions of real 3D geometric deviation using predictors from triangular mesh of the computer-aided designs and random forest machine learning algorithm [25]. a) A picture shows the four 3D printed objects that are scanned by a desktop scanner. b) The deviation values of the right three shapes were used to train the random forest machine learning model. And the "teardrop" shape on the left was used to validate the model. c) A comparison between the predicted values (Left) and the measured values (Right). The table shows the percentage of the predicted geometric deviations within given intervals of the measured values.

CHAPTER 3

Metrology System

3.1 System Setup

A material extrusion type 3D printer (Fig. 3.1) (System 30M, Hyrel International Inc., GA, USA) with an X, Y, and Z position accuracy of 50 um, 50 um, and 10 um, respectively, was operated with an MK1-250 hot flow print head with a nozzle diameter of 0.5 mm to print Acrylonitrile Butadiene Styrene (ABS) filament feedstock with a diameter of 1.75 mm. The printed trace was preset to be 0.55 mm wide and 0.2 mm thick.

A surface scanning laser confocal displacement meter (LT-9030M, Keyence Corporation, IL, USA), which is capable of measuring the displacement within ± 1.0 mm of the reference height with a resolution of 0.1 um, was integrated with the printer to achieve *in situ* monitoring during printing. The 3D printer was programmed to move the laser sensor as a non-extruding print head using the printer's built-in actuators. The Z and X/Y coordinates of the scanned surface profile were collected from the laser sensor and the 3D printer, respectively, using a data acquisition board (USB 6008, National Instruments, TX, USA) under the control of a LabVIEW program. Two microfabricated alignment marks (Fig. 3.2(a)), which have a circle ($\varnothing$3.138 mm, depth 150 um) in the center, were placed in the top left and bottom left corners of the stage. By scanning and locating the center of the circle (Fig. 3.2(b)) on the alignment marks, the scanned profiles of different layers were aligned to reconstruct the 3D profile of the printed part even if the stage was shifted unexpectedly.

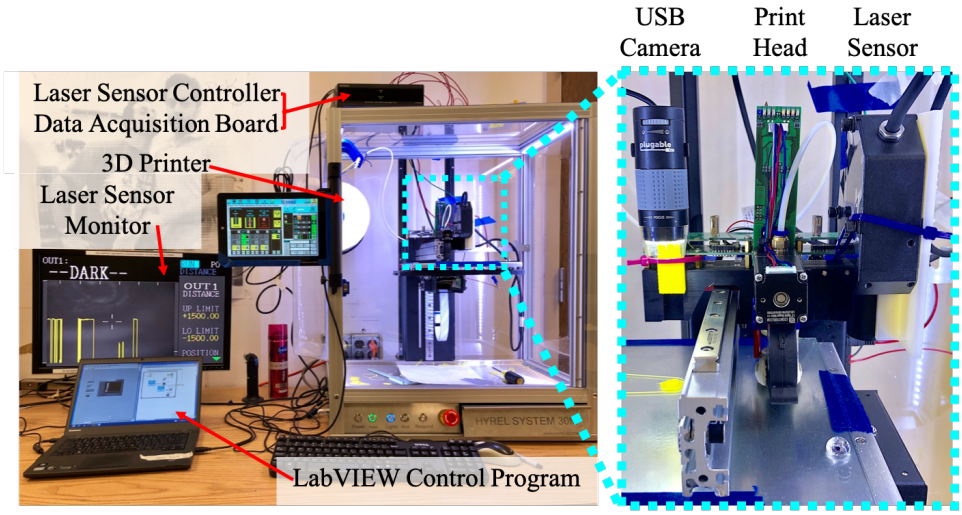
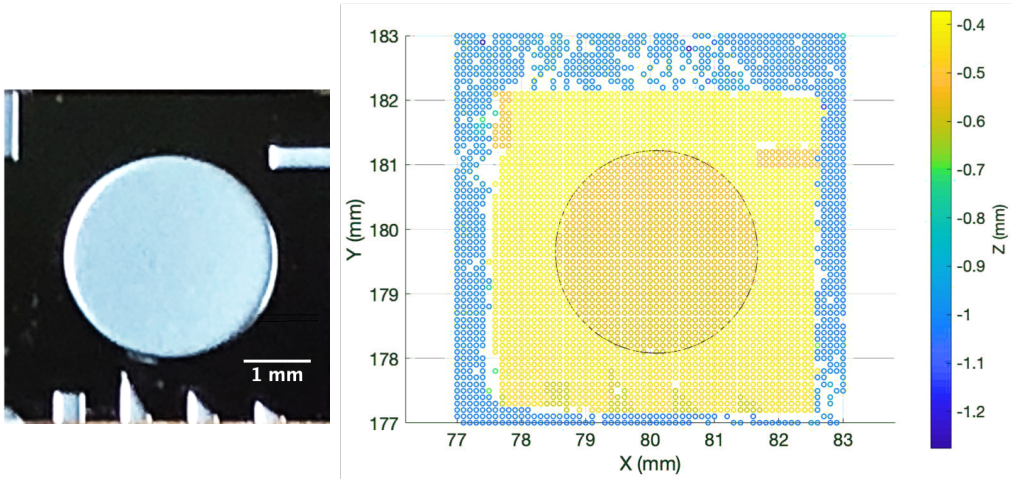


Figure 3.1: (Left) System setup of the *in situ* monitoring system (Right) A zoomed-in picture shows the laser sensor moves in parallel with the print head during printing/scanning.



(a) (b)

Figure 3.2: (a) Microfabricated alignment mark and its (b) scanned profile. The black circle in (b) shows the calculated minimum circumference of the circle on the alignment mark in (a).

3.2 Working Mechanism

Figure. 3.3 shows the working mechanism of our system. The blue blocks show the normal working flow of a typical material extrusion 3D printer.

The Computer-Aided-Design (CAD) model is first exported as a stereolithography (STL)

file, which is a format supported by most 3D printers. A STL file tiles the surfaces of any shapes using triangles which have no overlaps or gaps between each other and this process is also called "tessellation" (Figure. 2.21(b)). For each tiny triangle, the coordinates of the vertices and the unit normal vector which points outwards of the 3D model are stored in the STL file. Since the curved surfaces are approximated with numerous tiny triangles in the STL files, geometric errors caused by the approximation is already introduced to the final printed object. These geometric deviations can be reduced by using a finer size of triangles when exporting or using another more advanced file formats like AMF and 3MF.

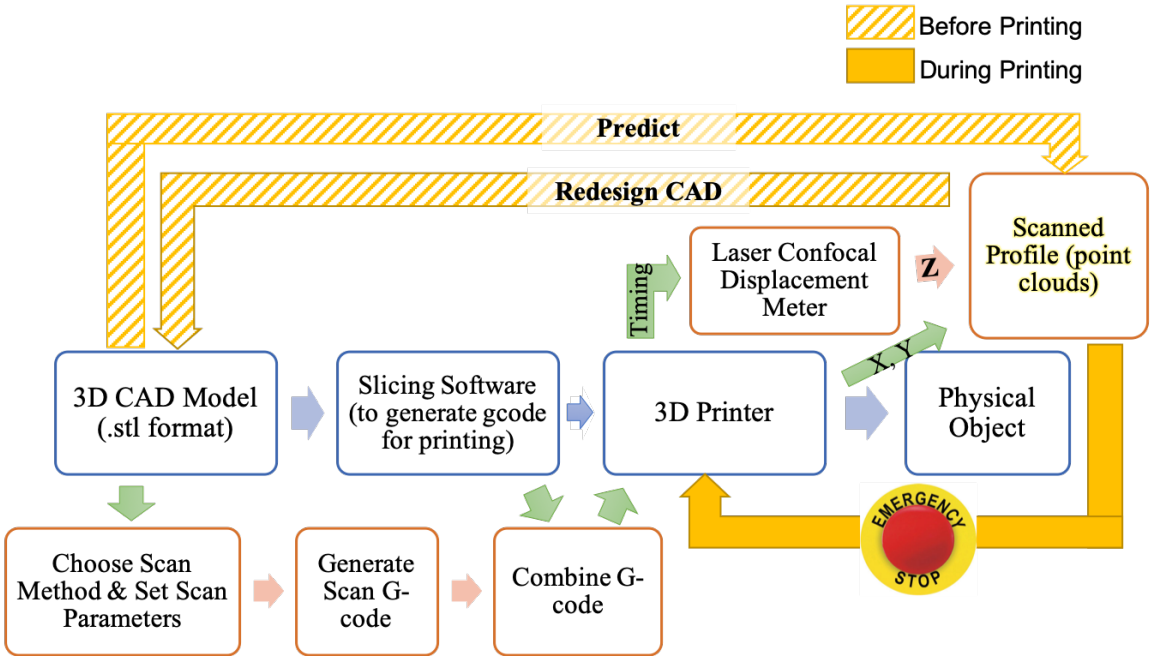


Figure 3.3: Flow chart shows the working mechanism of our system.

The 3D printer will then use a Slicing Software to convert the 3D design to 2D slices and further to the tool paths of the extruder. In this study, a free slicing software "Slic3r" was used to convert the STL files to the G-code. G-code is the most widely used programming language in computer-aided manufacturing to describe the tool paths. Then, the G-code is sent to the hardware of the 3D printer to print out the physical object.

The orange blocks indicate how the laser sensor was integrated with the 3D printer. First, the scan method is chosen based on the CAD model including scanning area, res-

olution, speed, and which layers you want to scan. Then, a piece of customized G-code for scanning is automatically generated by a Matlab Program. These codes were further combined with the original G-code and fed to the 3D printer together. During the printing process, the 3D printer will send periodic signals which will be further paired with the laser sensor signal to extract the coordinates of the points on the printed parts. And the scanning profiles are stored in the format of the point clouds, which are just groups of points with known coordinates. Furthermore, the scanned profiles were used to train a machine learning model to make predictions of the final printed part based on its CAD model. Finally, a compensation algorithm is developed by iteratively comparing the predicted geometry generated by the trained machine learning model with the ideal CAD model and then redesign the CAD model to minimize the difference.

3.3 Scan Strategy

For this study, the laser sensor was affixed in a location normally occupied by one of the print heads. The code was generated to cause the 3D printer to "print" a desired path with the laser sensor. Using this method, it is possible to design a custom scan path for the sensor simply by inputting a print path into the printer via G-code. The scanner location was then read from the printer, allowing for reconstruction of the scanned image. Thus, all built-in functions for print head control can be leveraged for commanding the laser sensor by modifying its path in the G-code. The printing and scanning speeds were set at 1800 and 300 mm/min, respectively. The scanning of printed layers runs after the printing of every layer, or periodically after a specified number of layers.

A MATLAB program was developed to generate the G-code for scanning and combine it with the original G-code automatically. The sampling frequency of the data acquisition system is checked in the MATLAB program. As shown in Figure. 3.4, the laser sensor settings are required to calculate the maximum sampling frequency of the laser sensor by carefully taking all sampling cycle time into consideration. The sampling frequency of the data acquisition system should not exceed the maximum sampling frequency of the laser

sensor to avoid duplicated data points in the output. Given the scan rate and the scan path separation, which is the distance between the adjacent scanning points, a minimum sampling frequency was calculated to ensure that at least one data point was collected at each scanning point. The sampling strategy of scanning can be burdensome for surfaces that are not smooth and possess bumpy or noisy texture. Depending on the sample size, the inspection time could be unnecessarily prolonged, or the measurement accuracy could be degraded. The gridpoint distribution sampling strategy was implemented in this work due to its low scan time relative to other methods [89]. The spacing of the grid pattern was set at 0.1 mm.

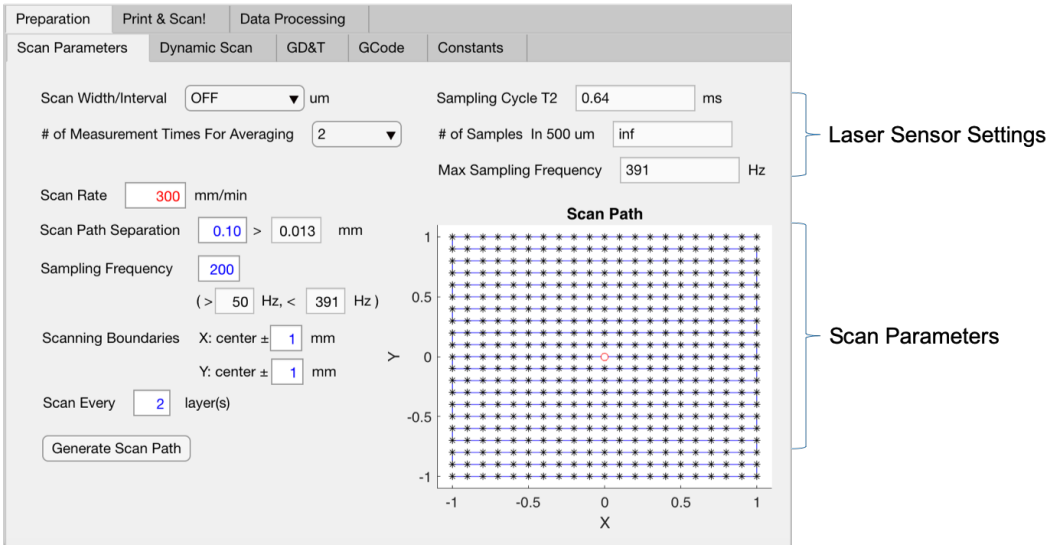


Figure 3.4: A screenshot of the MATLAB program shows the laser sensor settings and some basic parameters for scanning.

Two types of design strategies were developed in this study, the Static Scan and the Dynamic Design. The Static Scan, as shown in Figure. 3.4, is to scan a simple rectangular area with given boundaries. The static scan strategy is acceptable for a small part but will be very inefficient for a large scanning area. Thus, the Dynamic Scan, which can customize the scan path based on the geometry of each CAD model was developed. As shown in Figure. 3.5, the dynamic scan method can be chosen from "Basic Square", "Smart Solid", "Smart Sparse", and "Smart Boundary". The "Basic Square" scan method is similar to the Static Scan strategy but with self-adjusted boundaries based on the size of the part. The

"Smart Solid" scan method automatically covers the whole crosssection area of a part at a given height with a distance from edges set by users. If the center area of part's cross-section is less important than the areas near the edges, the "Smart Sparse" method can be used to save you some data storage space by sampling fewer points in the center area. If you only care about the areas near the edges, the "Smart Boundary" scan method can be implemented to save both the data storage space and the scanning time by only scanning the boundary areas. Fig. 3.6 is an example showing that 56% percent of the scanning time can be saved by using the Smart Boundary method instead of the Basic Square method. As shown in Figure. 3.5, you can choose to scan every several layers to further reduce the scanning time. Figure. 3.7 shows the options for generating G-code for printing and scanning with or without using the alignment marks. The "Estimated Total Print + Scan Time" was calculated automatically when generating the G-code.

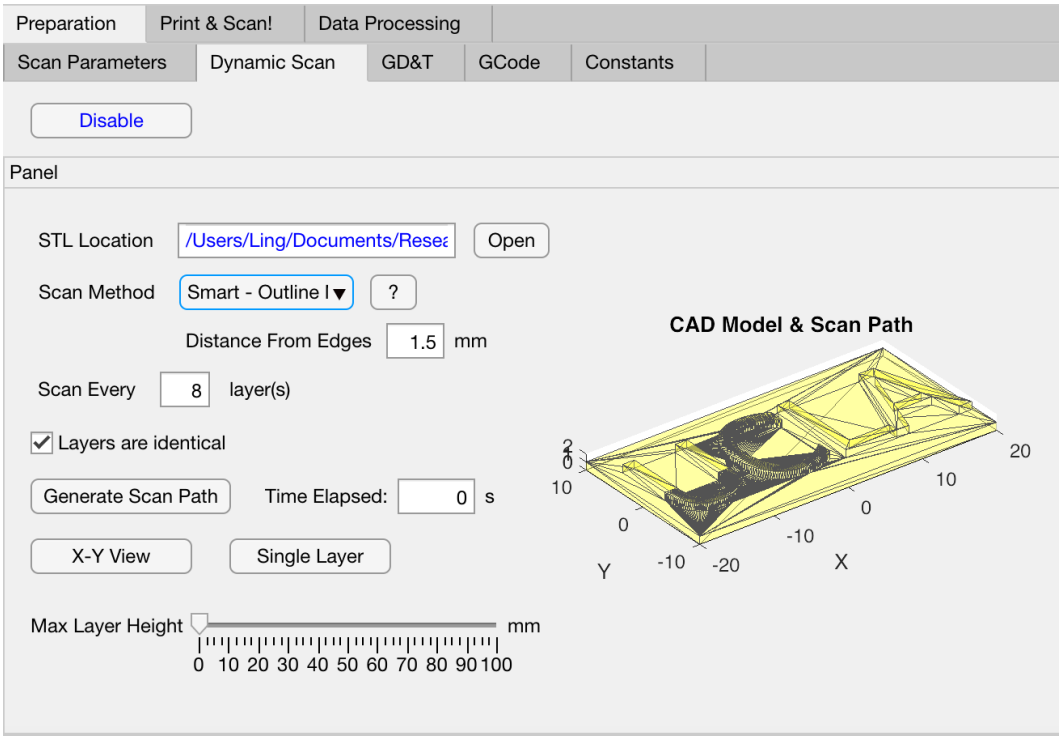


Figure 3.5: A screenshot of the MATLAB program shows the Dynamic Scan strategy.

One thing to note is that the laser sensor (Keyence LT-9030M) used in this study is a 1D sensor, which means it needs to work with the motion stage (built-in motors of the 3D

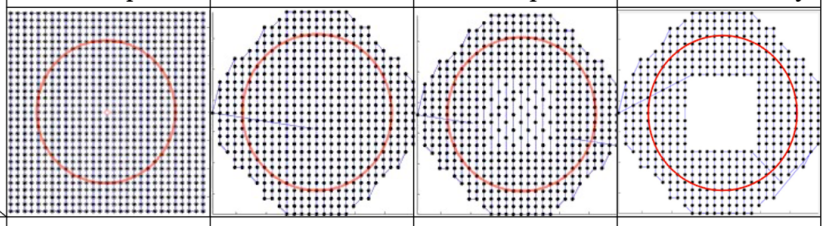
Scan Type / Estimated Total Printing + Scanning Time of Objects		Basic Square	Smart Solid	Smart Sparse	Smart Boundary
					
	UCLA Logo 40 x 20 mm^2, 2 layers	36 min 12 sec	24 min 52 sec	17 min 35 sec	16 min 2 sec

Figure 3.6: Examples of four customized smart scan methods using the grid point distribution strategy, each with a different number of sampling points. Black points, dark blue lines, and red circles represent sampling points, scan path, and the cross-sectional boundary of a printed sample, respectively.

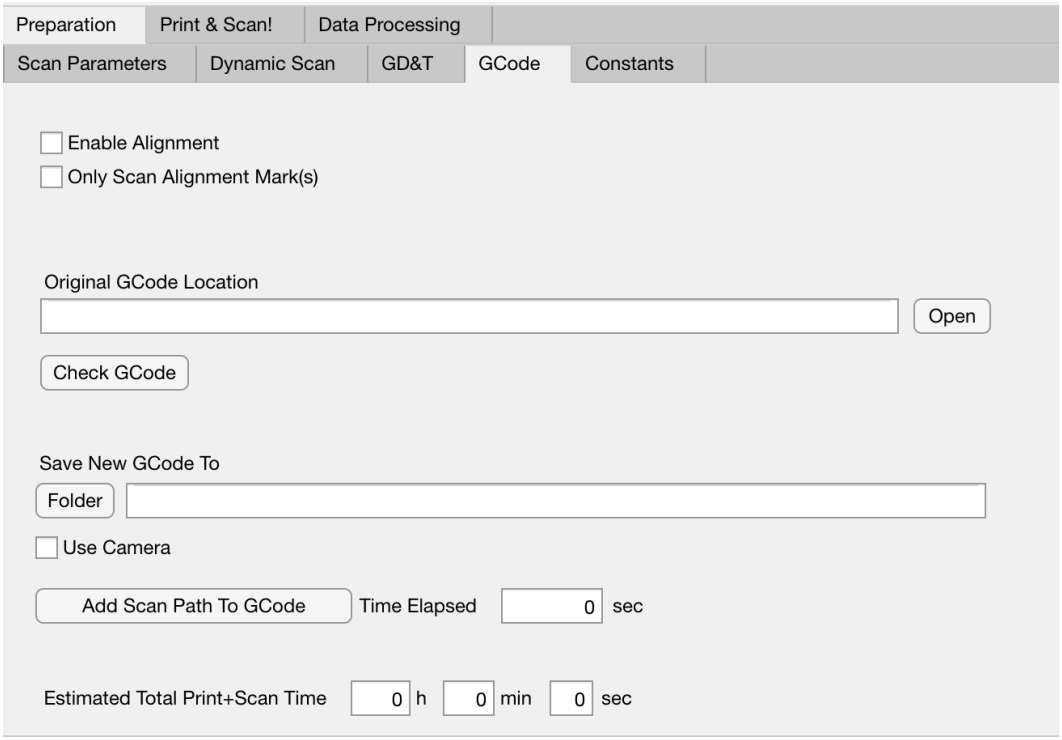


Figure 3.7: A screenshot of the MATLAB program shows the settings for generating G-code for printing and scanning.

printer) to perform 3D scanning. During scanning, the laser sensor is mechanically moving. Thus, the scanning speed can not be too high to avoid mechanical vibration. In this study, the scanning time is 6 times of the printing time because the scanning speed was set at 300 mm/min compared to 1800 mm/min for the printing. However, if you use a 3D laser sensor

that has a buit-in optical motion stage inside, the scanning time can be ignorable compared to the printing time. For example, the 3D laser sensor Keyence WI-010 can scan a 10 mm by 10 mm area in 266 ms with 80,000 sampling points. However, using the 1D laser sensor LT-9030 with printer's built-in motor at 300 mm/min, it will take 202 s to scan the same area. Thus, by upgrading the 1D laser sensor to a 3D laser sensor, the scanning time can be reduced to 1/750.

In some instances, molted plastic was observed dripping out of the nozzle when the hot extruder was moving between printing processes. Several actions were taken to minimize the excess material being printed to the next layer. First, in the slicing software, the extruder can be programmed to extract the filament back by 2 mm when the extruder is about to move without printing. This is the normal method for the common material extrusion 3D printer to avoid material dripping in between printing. However, this method only works great if the moving time is very short (e.g., several seconds). And this method results in a lack of material at the beginning of the next printing. So there are two other methods that will work with the first one. The second method is to turn off the heater of the extruder if the moving without printing (e.g., the scanning process) will take a longer time (more than several minutes). This method helps resolve the material burning issue in the extruder which will cause the material to appear black at the beginning of printing next time. However, even with the help of a USB fan integrated very close to the extruder, it still takes time for the extruder to cool down, and it will take another couple minutes to heat the extruder which can start simultaneously before the scanning process ends. So anyways, some excess material will be present. To address this issue, a sacrificial pattern (like a skirt) was printed around the formal part to consume and block the redundant material that was observed to drip while the print head waited during the scanning process. The skirt sacrificial pattern can be automatically generated by the slicing software and can have a height as tall as your printed part. Depending on the specific printer being used, this sacrificial pattern may not always be required. You can even manually put a sacrificial structure next to the printed part and program the 3D printer to swipe the nozzle before the formal printing of the next

layer.

3.4 Data Acquisition

In this study, the data acquisition focuses on acquiring X, Y, and Z coordinates of the sampling points on the as printed part. The Z coordinates can be directly extracted from the laser sensor signals. However, since the software (Repetrel version 3.83) of the Hyrel System 30M 3D printer did not provide/output the location of the extruder/laser sensor in real-time. Thus, extra effort was made to retrieve the X and Y coordinates of each point from the G-code. The G-code of printing usually consists of 3 sections: header to prepare the printing, commands to execute moving/printing, and the epilogue to end the printing. Figure. 3.8(a) shows the typical commands for moving and printing. It can be seen that when the (i+1) the line is about to be executed, the laser sensor had already moved to the point indicated by the ith line. Thus, the idea is to send a signal by using the (i+1)th line to manifest that the laser sensor had arrived the points with coordinates as shown in the ith line. The Hyrel System 30M 3D printer has AUX ports (Figure. 3.8(b)) which can be turned on and off by using "M7" or "M9" correspondingly (as shown in Figure. 3.8(c)). The output of the AuX ports is 12.5 V voltage signals which can be collected using the data acquisition board. These voltage signals were further used to pair the X, Y coordinates from the G-code with the Z coordinates from the laser sensor signals (Figure. 3.8(d)).

In this study, both the AUX signals and the laser sensor signals were from floating signal sources which were not connected to the ground of the building but was connected with an isolated referenced ground point. To reduce the noise and provide greater accuracy, Differential (DIFF) signal connection was used for the laser sensor signals (Figure. 3.9(a)). Instead, the AUX voltage signals were working as digital signals which did not require high accuracy of the data. Thus, Referenced Single-Ended (RSE) signal connection (Figure. 3.9(b)) was used for the AUX signals.

A LabVIEW program was written to collect the data. Besides the AUX signals to extract

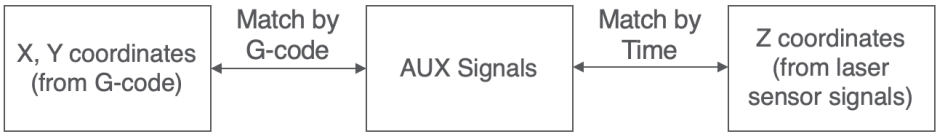
```
;---
G1 Z0.200 F1800.000 ; move to next layer (0)
G1 X52.749 Y111.777 F1800.000 ; move to first skirt point
G1 X55.500 Y111.111 E0.11935 F1800.000 ; skirt
G1 X98.500 Y111.111 E1.93240 ; skirt
G1 X100.648 Y111.508 E2.02450 ; skirt
G1 X102.513 Y112.645 E2.11660 F1800.000 ; skirt
```

(a)

(b)

```
;---
G1 Z0.200 F1800.000 ; move to next layer (0)
G1 X52.749 Y111.777 F1800.000 ; move to first skirt point
M7; start Aux 1
G1 X55.500 Y111.111 E0.11935 F1800.000 ; skirt
M9; stop all Aux
```

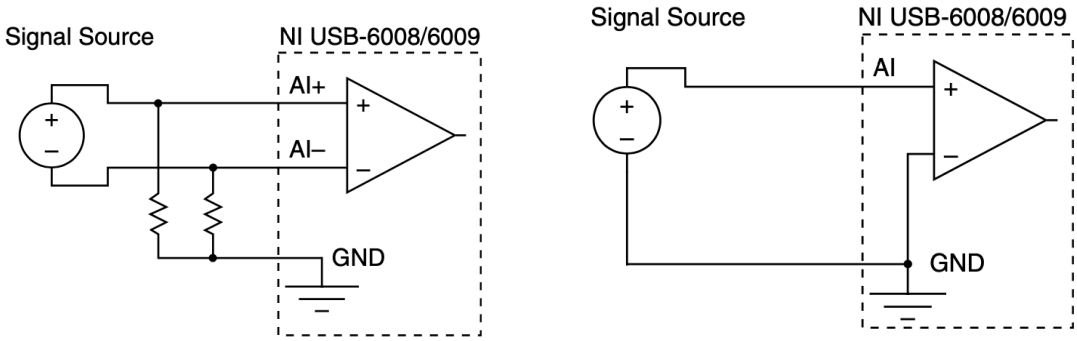
(c)



(d)

Figure 3.8: Get the coordinates of the scanned profile from G-code and laser sensor signals. (a) G-code for printing ONLY. G1 - working move. X/Y/Z - coordinates of the next points to move to. F - moving speed. E - absolute length of the filament to feed into the nozzle. (b) A picture shows the AUX port on the 3D printer. (c) Modified G-code for both printing and scanning. M7/M9 - turn on/off Aux ports. (d) Schematic shows how to pair X, Y coordinates with the Z coordinates using the AUX signals.

the X and Y coordinates from the G-code, another AUX signals which was controlled by "M8/M9" in the G-code was also implemented to trigger a USB camera to take some images of the as-printed part before and after the scanning of each layer. The first generation of the data acquisition program (Figure. 3.10(a)) used a single "For Loop" for both data collection of three signals (including laser sensor, AUX, and camera) and data-consuming. Here, the LabVIEW program accomplished two types of data consuming tasks: 1) saving the laser sensor signals and AUX signals to a TXT file which will be further analyzed with

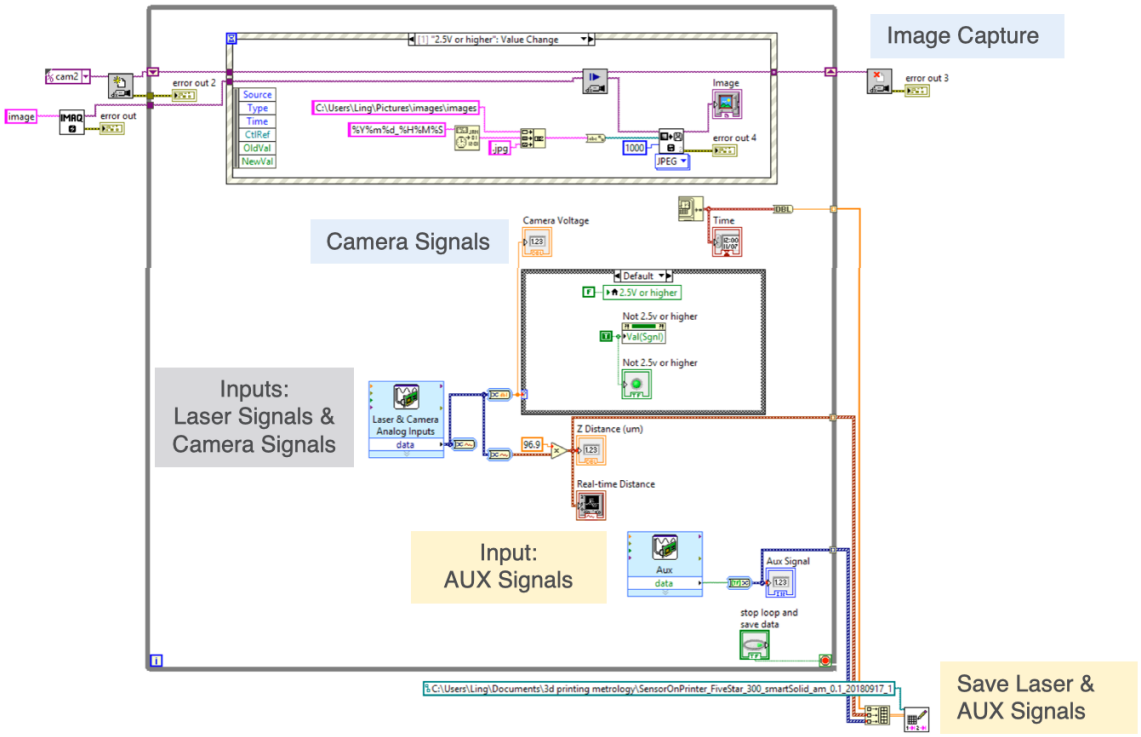


(a) (b)

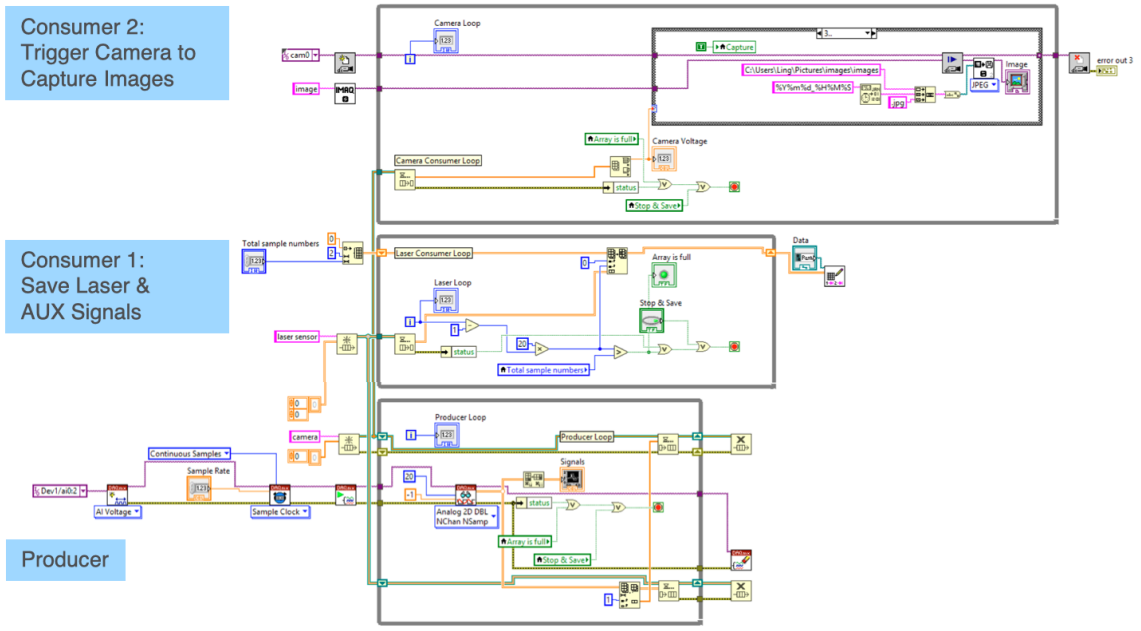
Figure 3.9: Analog input modes for floating signal sources. (a) Differential signal connection for the laser sensor signals. (b) Referenced Single-Ended signal connection for the AUX signals. Both images are from the User Guide for NI USB-6008/6009.

G-code using MATLAB; 2) processing the camera signals and trigger the LabVIEW Image Module to take images on demands. However, the Image Module takes so much longer to achieve image capture than simply saving data, which delayed the single "For Loop" and further caused data missing in saved AUX signals. Since the AUX signals are paired with the G-code in order, one data point missing will cause all following data points to be shifted and mismatched with the G-code, which is not acceptable. Thus, a new design pattern - the Producer/Consumer design pattern - was implemented in the second version of the LabVIEW data acquisition program (Figure. 3.10(b). The Producer/Consumer design pattern aims at enhancing data communication between multiple loops running at different speeds, which is exactly what we were looking for. By using this design pattern, all 3 signals were collected using a Producer and temporarily saved to a queue, which is a data structure following a First In First Out rule. Then two consumers were used to save the laser and AUX signals and trigger a camera to capture images. Even though the data were retrieved from the queue at different rates, there was no data missing that occured in the process. Figure. 3.11 shows a series of images taken *in-situ* after the layer was done printing and before the scanning started.

A partial Kaplan turbine with a logo that rotated through the structure thickness was



(a)



(b)

Figure 3.10: LabVIEW block diagrams of the data acquisition program: (a) 1st generation - single "For Loop" and (b) 2nd generation - Producer/Consumer.

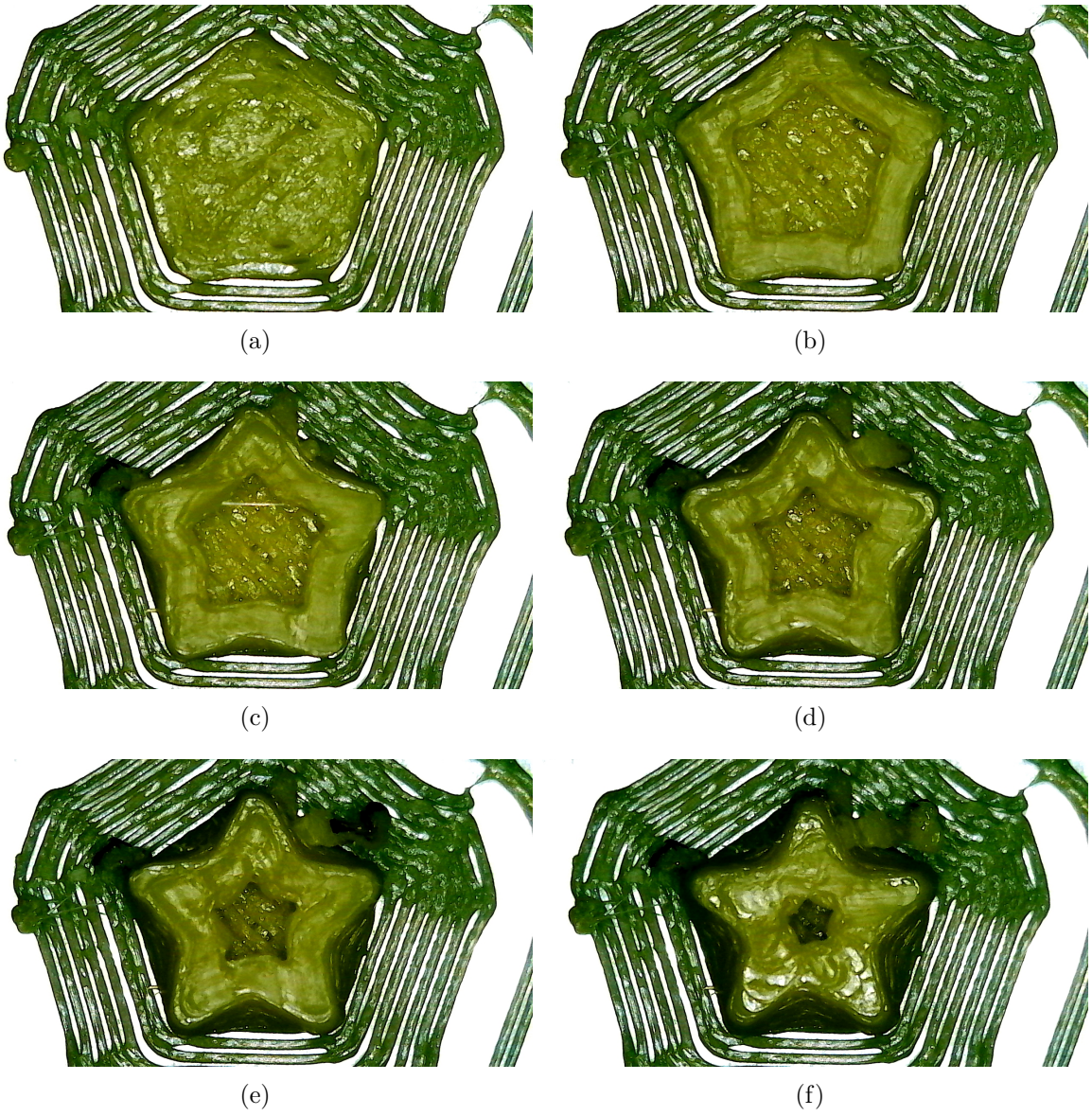
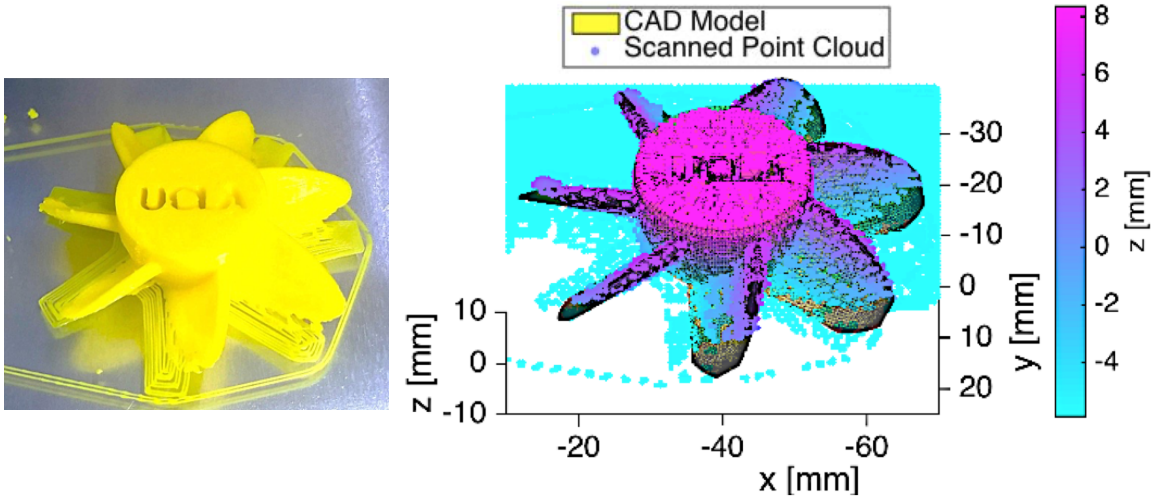


Figure 3.11: Images were automatically taken on demands by using AUX signals to trigger the Image Module to capture images in LabVIEW. (a)-(e) were images of the as printed layer presented in chronological order.

printed and scanned every 5 layers in order to investigate a structure with complex features. The scan path separations in both the X and Y axis were 0.5 mm. The scanned point clouds in comparison with CAD models are shown in Figure. 3.12. To get a better view of the inside structures, the scanned point cloud was sliced into four horizontal planes at different heights (from top to bottom) as shown in Figure. 3.13.



(a) (b)

Figure 3.12: Pictures and scanned point clouds in comparison with the corresponding CAD models of the complicated hollow part.

3.5 3D Geometric Deviation

The scanned profile of the printed sample, which is also called a point cloud (PC), consists of the coordinates of all sampling points. The scanned PC is usually noisy [90, 91]. Thus, denoising (Fig. 3.14) was applied to the PC to remove the unphysical outliers whose average distance to its k-nearest neighbors exceeds a threshold, where k is a constant number.

To evaluate the geometric inaccuracies of a printed object, its scanned PC must first be aligned with the corresponding ground truth CAD model. The actual alignment operation consists of two steps. First, convert the CAD model to a grid PC that is 4 times denser than the sampling grid of the scanned PC. Second, use an iterative closest point (ICP) algorithm [92] to align the scanned PC ("Moving") (Fig. 3.15) to the CAD PC ("Reference"). ICP is a method to find the optimal rigid transformation that registers two shapes. ICP iteratively identifies the nearest neighbor of each point in the scanned PC from the CAD PC to minimize the sum of squared distances between the paired points. The 3D geometric deviation was defined as the distance between each point of the scanned PC and its corresponding nearest neighbor from the CAD PC (Fig. 3.16).

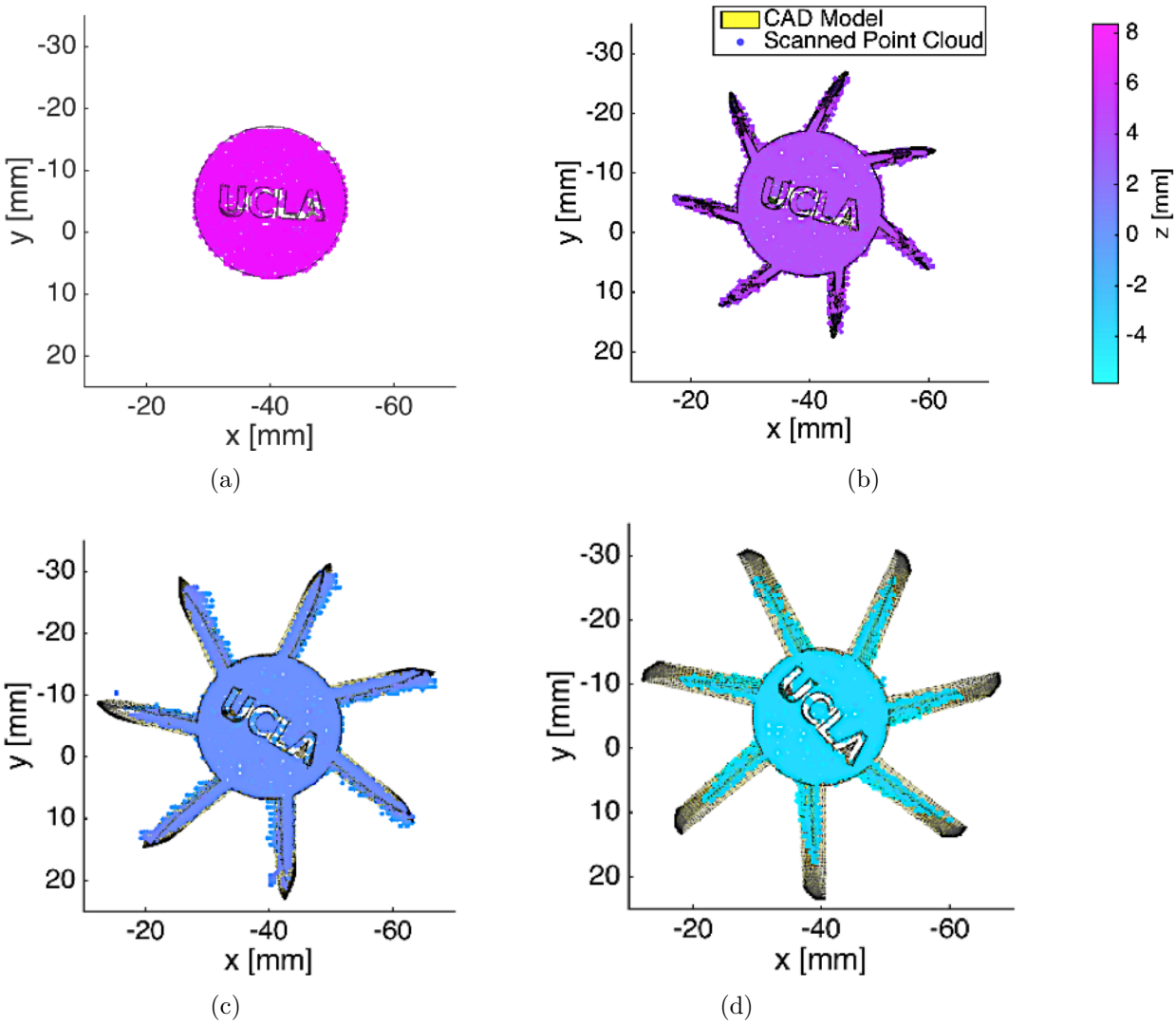


Figure 3.13: Slices in the X-Y plane of the scanned point clouds of the complicated hollow part at (a) $6 \leq z \leq 8$, (b) $3 \leq z \leq 5$, (c) $-1 \leq z \leq 1$, and (d) $-5 \leq z \leq -3$. Here the Z coordinates of the scanned point cloud are the same as those in the CAD design.

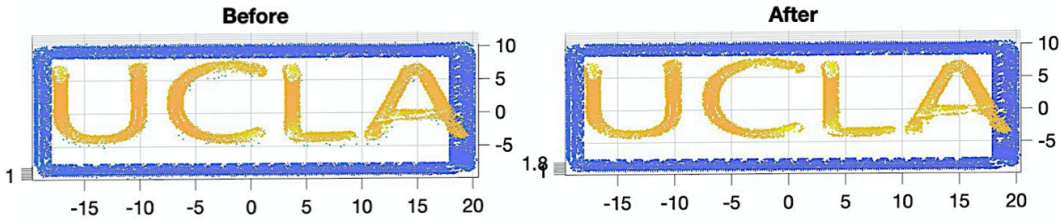


Figure 3.14: Outlier removal of the scanned PC of a printed UCLA logo.

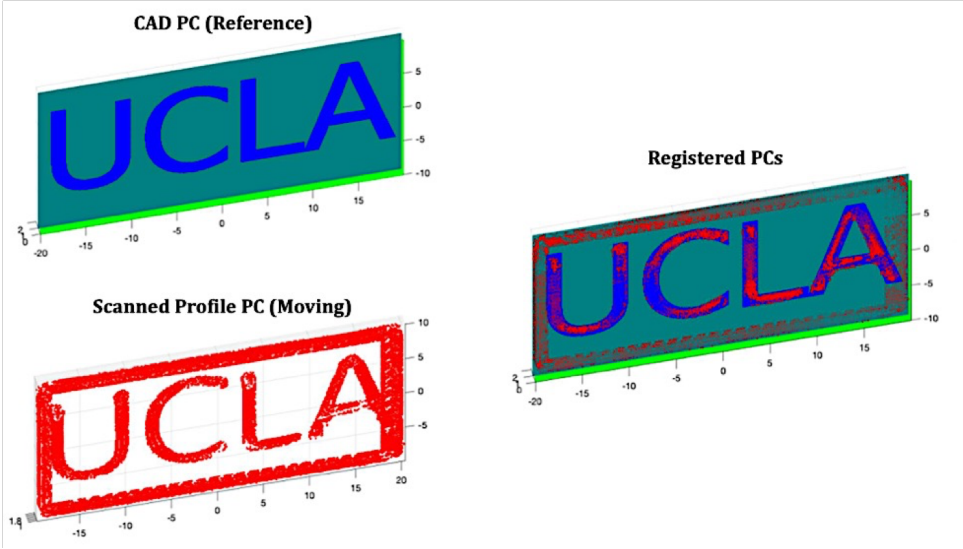


Figure 3.15: ICP was used to register the scanned PC to the corresponding CAD PC.

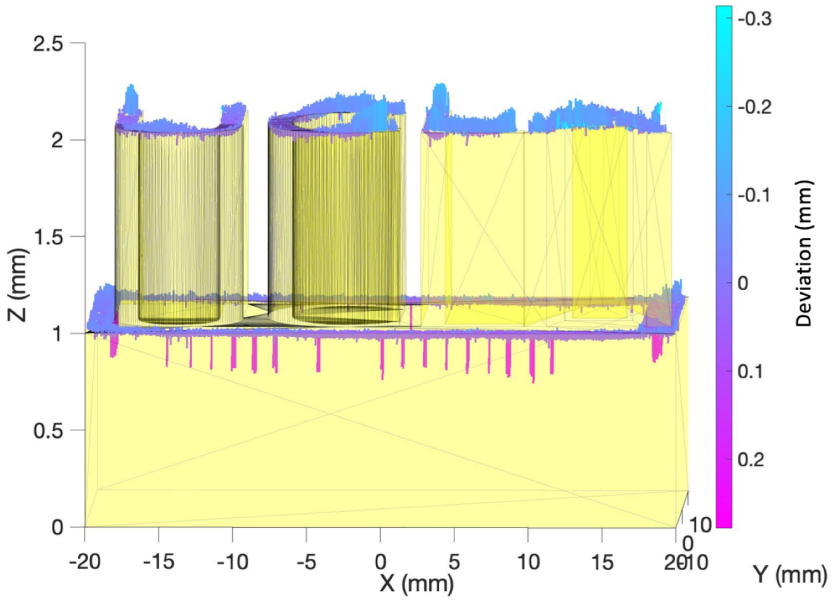
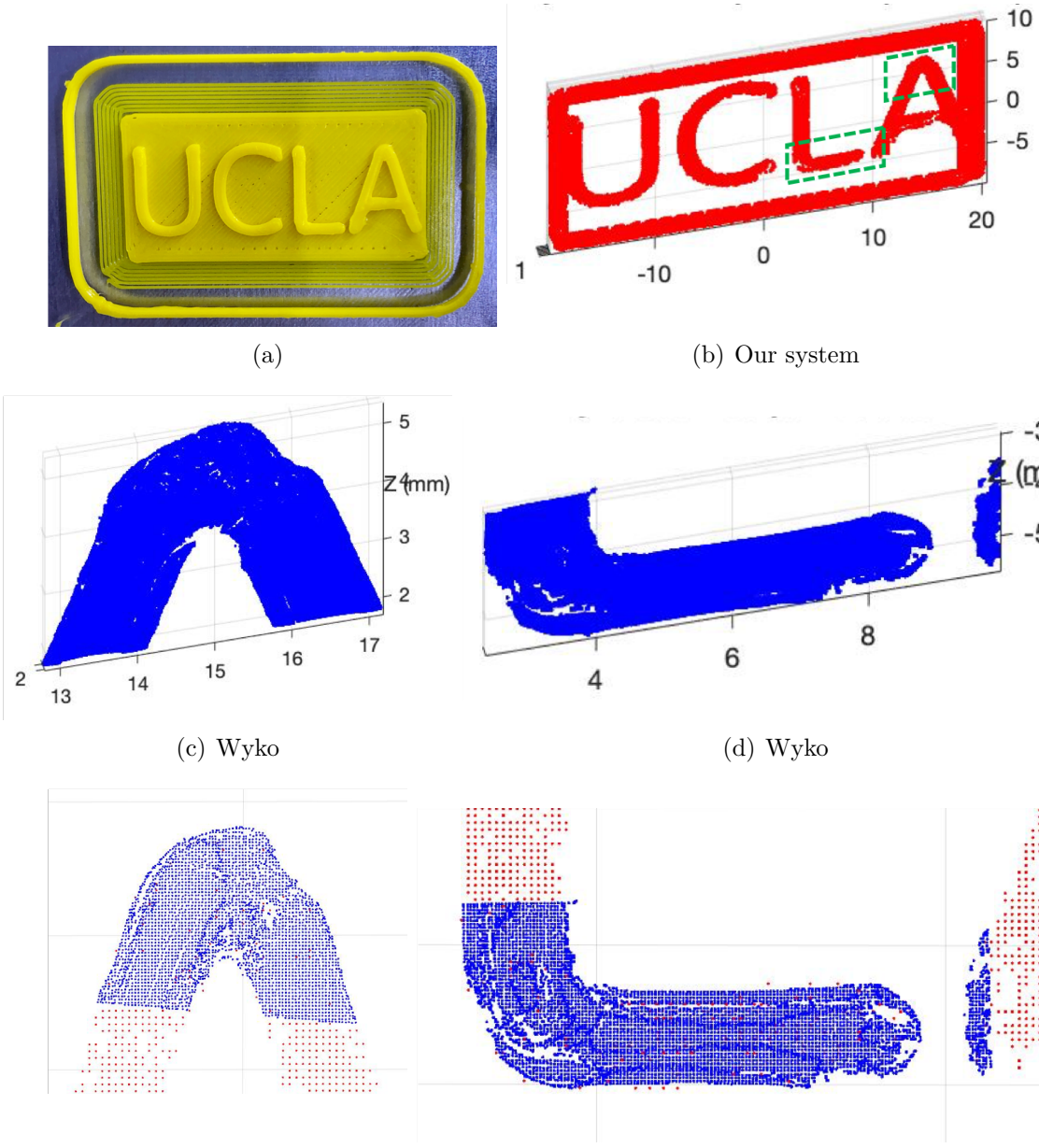


Figure 3.16: Illustration of 3D geometric deviation between CAD design and printed structure.

3.6 Metrology System Validation

To validate our *in situ* monitoring system, four UCLA logo samples (as shown in Fig. 3.6 and Fig. 3.17(a)) were scanned by both our system (Fig. 3.17(b)), as well as a high precision optical 3D profiling system Wyko® NT3300 (Veeco Instrument Inc., NY, USA). The Wyko

surface profiler has two modes: phase-shifting interferometry (PSI) mode for measurement of smooth surfaces and vertical scanning interferometry (VSI) for rough surfaces. In this study, VSI mode with a measurement range of 2 mm and a vertical resolution of 3 nm was used to scan the surface of the printed UCLA logo sample (Fig. 3.17(c)(d)). Outlier removal was then applied to the scanned profiles. The scanned point clouds obtained by Wyko and our system were aligned to each other (Fig. 3.17(e)(f). The mean and standard deviation of the 3D geometric deviation between two profiles were calculated as 0.00 mm and 0.02 mm, respectively.



(a) (b) Our system

(c) Wyko (d) Wyko

(e) Aligned (c) with (b) (f) Aligned (d) with (b)

Figure 3.17: (a) A UCLA logo was scanned by a Wyko surface profiler and our system. (b) A scanned profile of the UCLA logo obtained by our system with Smart Boundary scan method. The green boxes indicate the areas shown in (c)-(f). (c) A scanned profile of the top of letter A by Wyko. (d) A scanned profile of the bottom of the letter L by Wyko. (e) Aligned profiles of the top of letter A. (f) Aligned profiles of the top of letter L.

CHAPTER 4

Predict 3D Geometric Deviation From CAD Using CAN Models

4.1 Conditional Adversarial Networks (CAN)

Many machine learning methods, such as support vector machine (SVM) [60, 93, 58, 53, 94, 95, 96, 16, 97], random forests [93, 98, 25, 99], Naïve Bayesian [97], neural network (NN) [97, 99, 100], and k-nearest neighbor [97, 101], have been applied in AM for tool condition monitoring as well as defect detection and classification. Nonetheless, limited methods [25] have been reported to predict the 3D geometric deviation of freeform shapes.

In this work, conditional adversarial networks (CAN) [102], a general-purpose solution to image-to-image translation problems, were implemented to translate an input image of a slice of a CAD model into an output image of the corresponding slice from the 3D scanned profile, with color indicating the surface height of each point. CAN consists of an image generator and an image discriminator. The generator is trained to map an input image and a random noise vector to the ground truth image. The discriminator is trained adversarially to detect the fake images (produced by the generator) from the ground truth images.

4.2 Image Dataset Preparation

It is widely known that machine learning relies on tens of thousands of training data samples, which limits the application of machine learning in AM because only a limited number of physical samples can be printed out for data acquisition. Fortunately, CAN has been shown

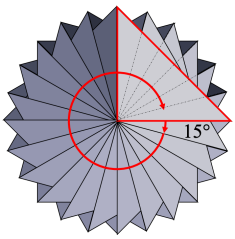
to have an adequate performance on smaller datasets. For example, a CAN was trained on a small dataset of only 400 images to translate day images to night images successfully [102]. To train the CAN models to make the best predictions of freeform shapes on a limited input dataset, samples in the input image dataset were carefully selected. By using the Minimum Near-Convex Shape Decomposition [103], arbitrary shapes can be decomposed into a minimum number of "near-convex" shapes, which can be categorized as transformations of circles, triangles, and quadrilaterals. In addition to shape, the orientation of part also has a big impact on the final product and was carefully considered when the input image dataset was established.

In this study, we printed and scanned 53 transformational samples in total from 3 primitive shapes (circle, isosceles right triangle, and square) with different orientations and various levels of transformation (Fig.4.1). The 3 standard primitive shapes were rotated from with an increment of 15 degrees until they completely overlapped with themselves (Fig.4.1(a)(b)). Both the isosceles right triangle and the square were transformed to quasi-circular shapes with round or sharp corners (Fig.4.1(c)) to represent arbitrary "near-convex" shapes. This selection of shapes is chosen to account for the wide variety of geometry commonly found in freeform additive manufacturing.

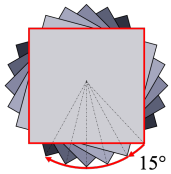
Each printed shape has 10 layers and every layer was scanned *in situ* during printing (Fig.4.2(a)). The observed scanned profile (ground truth) image was paired with the image of the corresponding cross-section of the CAD model. The input and output images of the CAN models are 256 pixels by 256 pixels, which represent physical areas of 14 mm by 14 mm. For parts with cross-section larger than 14 mm by 14 mm, the cross-section shape will be decomposed into multiple 14 mm by 14 mm areas to use the CAN models.

4.3 Image Dataset Expansion

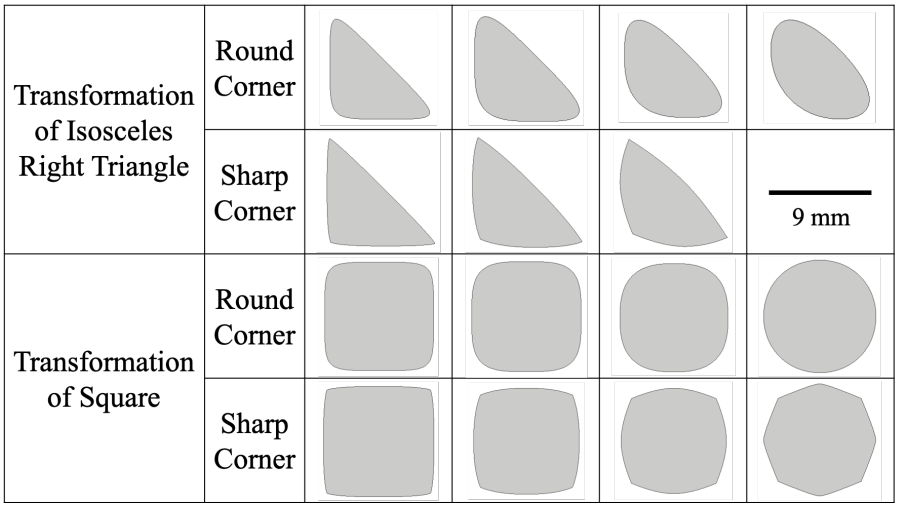
After performing the image dataset preparation as mentioned in Section 4.2, there were around 50 images in the image dataset for each layer. Although the CAN algorithm is already



(a)

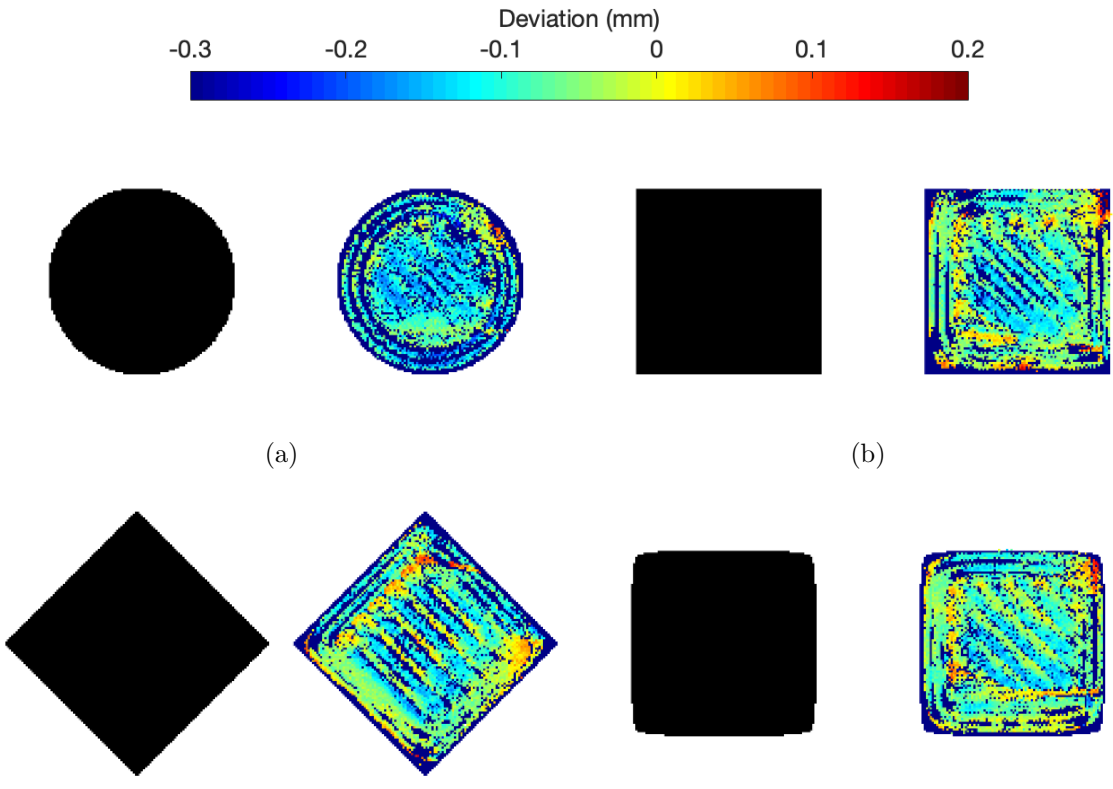


(b)



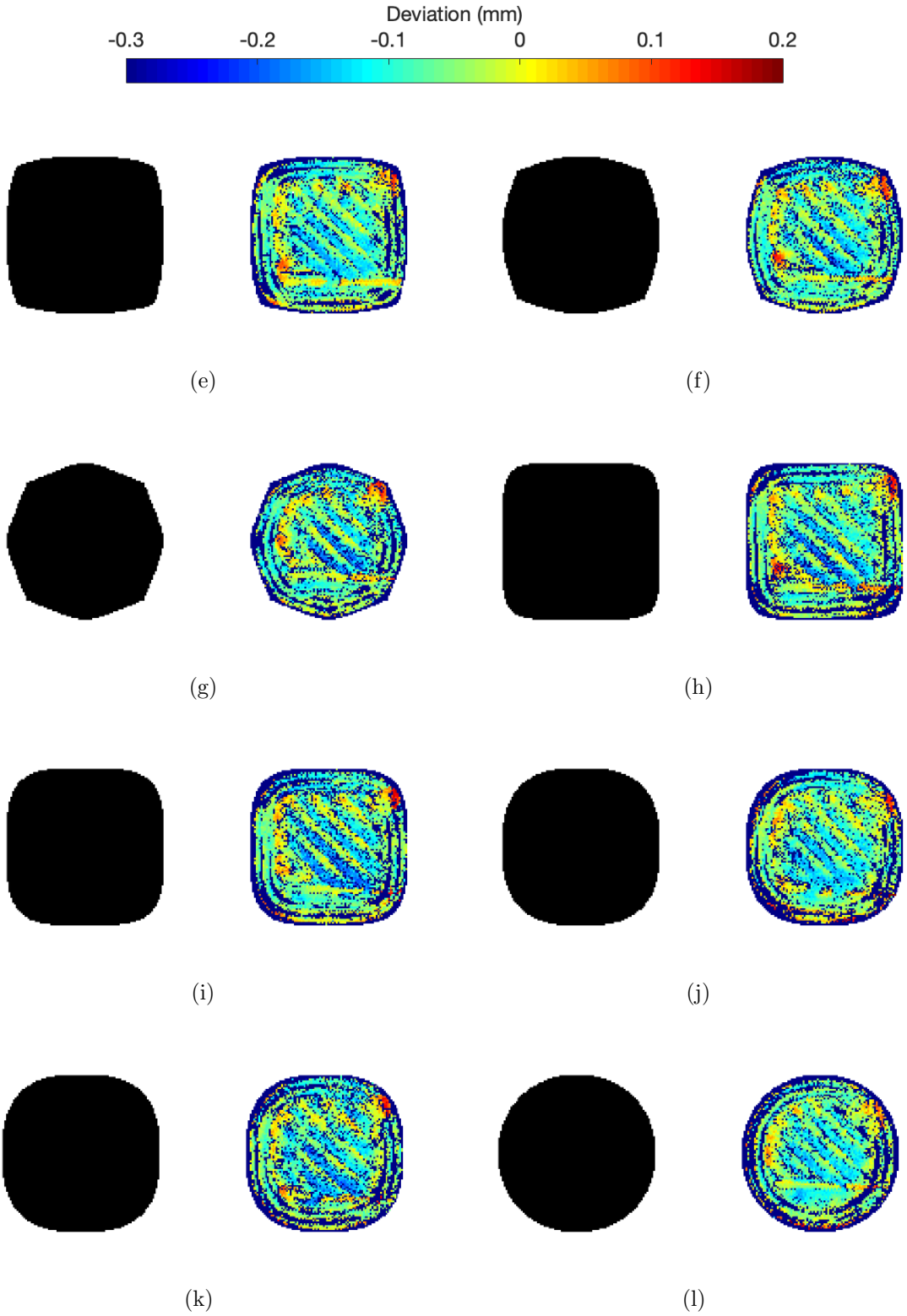
(c)

Figure 4.1: Selected transformational shapes from 3 primitive shapes (circle, triangle, and square) (a)(b) with different orientations and (c) various levels of transformation. Those shapes were picked to maximize the diversity of features in the training dataset of the CAN models since only a limited number of physical samples can be printed out for data acquisition. The characteristic length for these shapes is 9 mm.



(a) (b)

(c) (d)



(e) (f)

(g) (h)

(i) (j)

(k) (l)

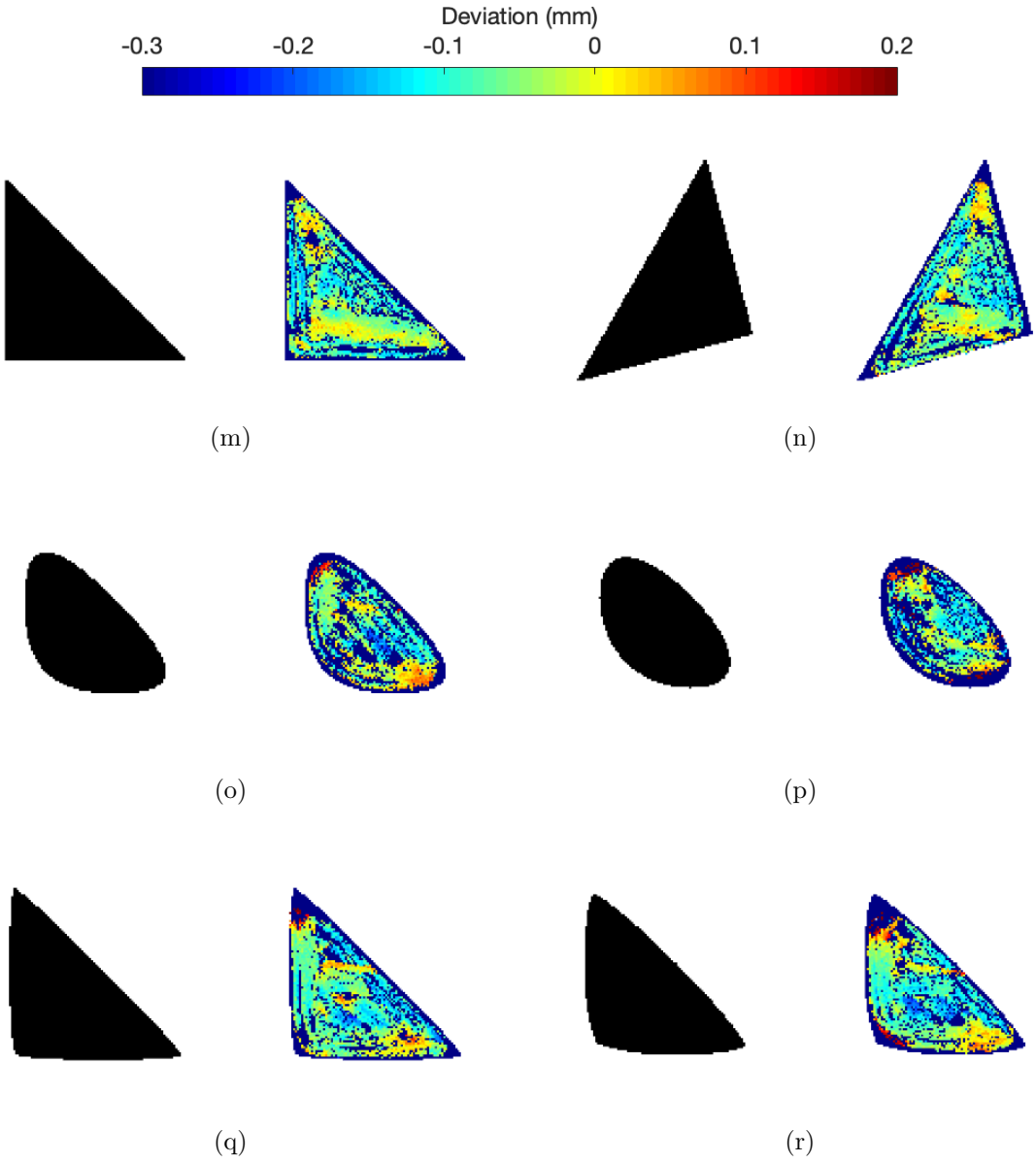


Figure 4.2: 18 Pairs of images from the image dataset, each consisting of the input image (left) and its corresponding ground truth (right), were used to train the CAN machine learning model. (a)-(r)The input image (left) is a cross-section of the CAD model. The ground truth image (right) is the scanned profile of the corresponding layer, in which the color indicates 3D geometric deviation. The images are 256 pixels by 256 pixels, which corresponds to a real area of 14 mm by 14 mm.

proved to work well for a small image dataset with 400 images, the total number of images in our image dataset is still not enough. Several approaches were explored to expand the current image dataset. Common methods to expand image datasets are taking more scans,

flipping/rotating images, cropping images,combining images by overlapping empty space, and image augmentations. Image Augmentation techniques includes resizing + cropping images, shearing images, adding noise to images, and dilation + erosion of images. However, one thing to note that the quality of printing is influenced by both the geometry (dimensions and aspect ratio) and the orientation of the object being printed. Thus, any approaches that will cause the deformation of the image are not acceptable. Finally, the only two methods left are cropping and combining the images. The image dataset for each layer was expanded by first randomly combining and then cropping the input images to 50 different images (256 pixels by 256 pixels). Figure. 4.3 shows examples of the generated images.

4.4 Tune Parameters

In this study, CAN was used to generate images of predicted profiles of a printed part based on images of its CAD model. For best prediction results, parameters of the CAN model (λ and patch size) were tuned to find a good balance between spatial sharpness and colorfulness. λ controls how much L1 loss, the sum of absolute differences between the predictions and the truth, is added to the CAN model. Small λ might introduce visual artifacts. On the other hand, large λ leads to reasonable but blurry images. Patch size determines the scale of image patches at which incorrect structures will be penalized by the discriminator. A smaller patch size of discriminator receptive fields encourages greater color diversity. A larger patch size tends to generate images with sharp structural features and colorfulness. After being trained for 30 epochs on a training dataset containing 95 images, the CAN models with different parameter settings were inspected on a test dataset consisting of 40 images. The predicted profiles of one shape are shown in Fig. 4.4. All of the 2D predicted images were converted to the 3D predicted profiles by inferring the geometric deviation of each pixel based on its color. Then the predicted scanned profiles were compared with the real scanned profiles (ground truth) and the error was calculated and statistically analyzed as shown in Table 4.1. The error was defined as the difference between the predicted deviation of a point and its corresponding real deviation. The CAN model with $\lambda = 125$ and patch size = 142 showed

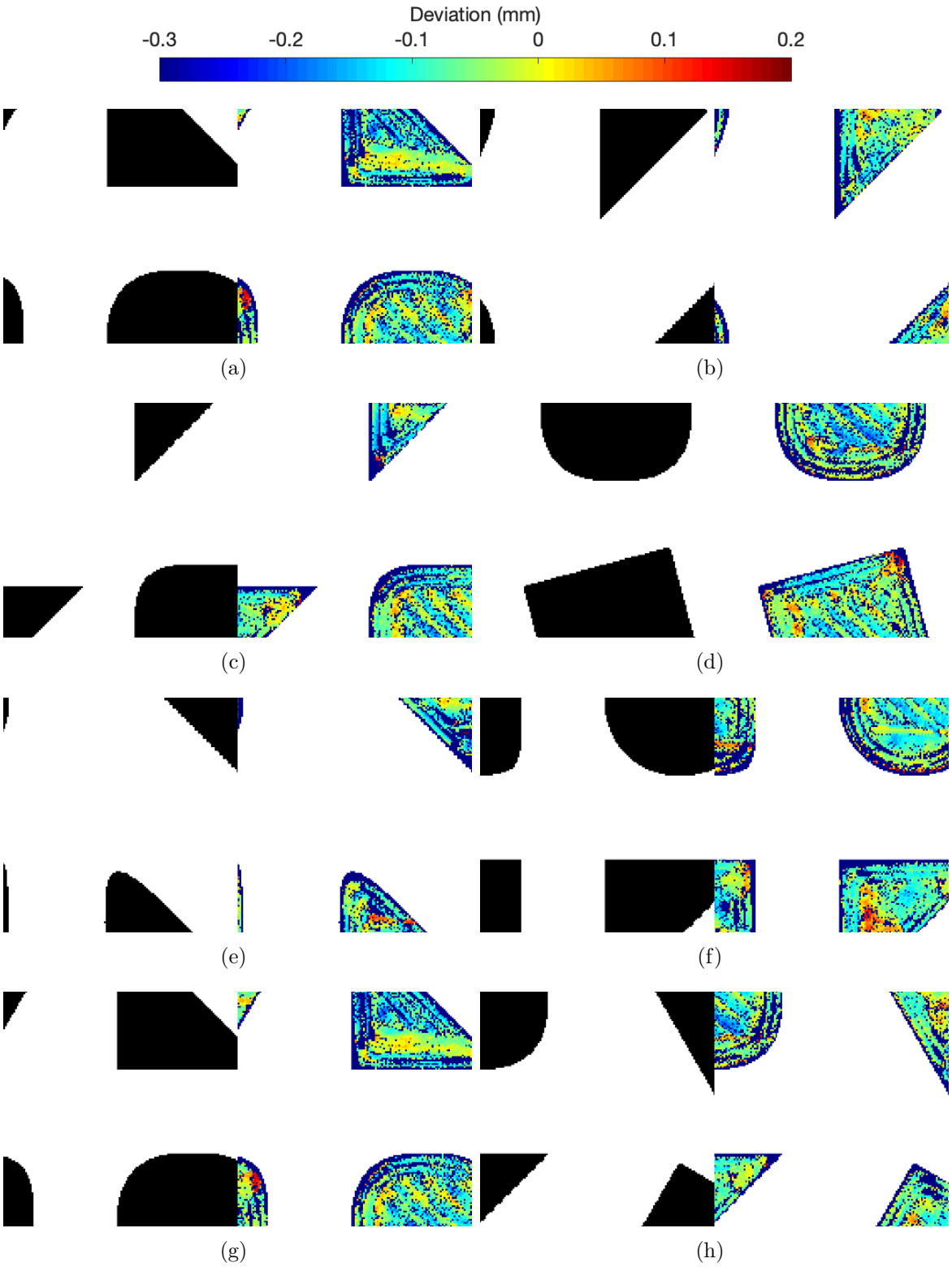
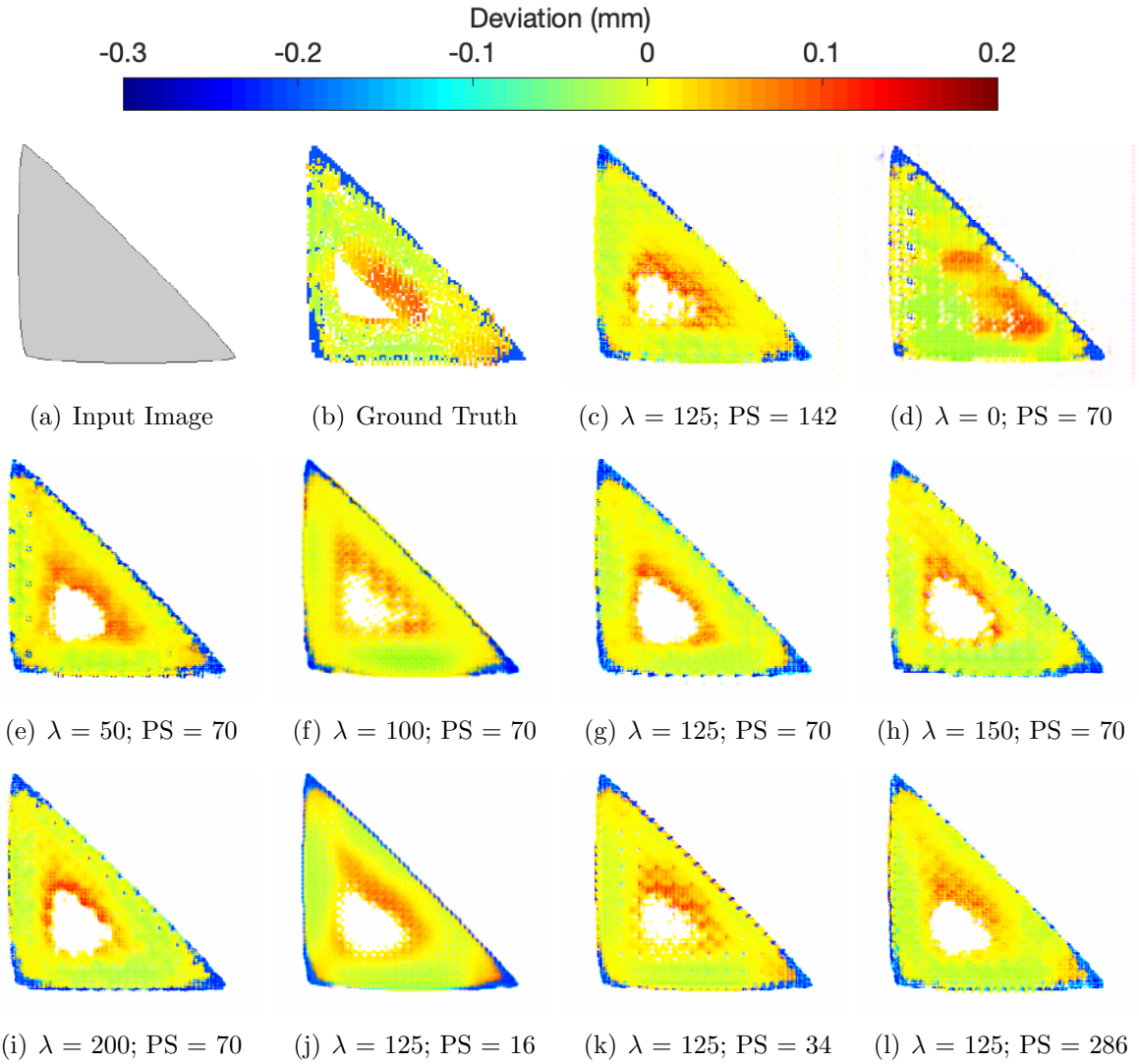


Figure 4.3: 8 Pairs of images were generated by randomly combining and cropping images to effectively expand the image dataset.

the best performance, and those parameters were used in the following sections.



(a) Input Image (b) Ground Truth (c) $\lambda = 125$; PS = 142 (d) $\lambda = 0$; PS = 70

(e) $\lambda = 50$; PS = 70 (f) $\lambda = 100$; PS = 70 (g) $\lambda = 125$; PS = 70 (h) $\lambda = 150$; PS = 70

(i) $\lambda = 200$; PS = 70 (j) $\lambda = 125$; PS = 16 (k) $\lambda = 125$; PS = 34 (l) $\lambda = 125$; PS = 286

Figure 4.4: (a) Input image - a cross-section of CAD model, (b) ground truth - scanned profile of the corresponding surface, and (c)-(i) predicted images using CAN model with different λ and patch size (PS) after 30 epochs of training. The CAN model with $\lambda = 125$ and PS = 142 generated the most similar image to the ground truth.

4.5 Training and Testing

53 shapes with a characteristic length of 9 mm and a thickness of 2 mm (10 layers) were scanned *in situ* during printing. The quality of the first printed layer relies on 1) the preset

CAN Parameters		Error = CAN Predicted Deviation - Experimentally Measured Value				
λ	Patch Size (PS)	Percentage within Given Range of Error			Mean (mm)	Standard Deviation (mm)
		±0.05 (mm)	±0.10 (mm)	±0.15 (mm)		
0	70	88.15%	100.00%	100.00%	-0.0006	0.0179
50	70	88.77%	99.99%	100.00%	-0.0009	0.0176
100	70	90.11%	100.00%	100.00%	0.0018	0.0162
125	70	88.08%	99.99%	100.00%	-0.0023	0.0182
150	70	87.06%	100.00%	100.00%	-0.0012	0.0189
200	70	88.02%	99.92%	100.00%	-0.0014	0.0184
125	16	88.64%	100.00%	100.00%	-0.0014	0.0178
125	34	89.49%	100.00%	100.00%	0.0009	0.0170
125	**142**	**90.24%**	**99.99%**	**100.00%**	**0.0007**	**0.0162**
125	286	88.06%	100.00%	100.00%	-0.0010	0.0180

Table 4.1: The deviation between predicted and printed structure for different CAN parameters. The CAN model with $\lambda = 125$ and PS = 142 provided the highest percentage of predictions with error less than ±0.05 mm error, as well as the lowest mean and standard deviation of error.

print head's relative height to the stage, which is manually adjusted before each printing in this study, and 2) the adhesion between the printing material and the stage, which is enhanced by applying adhesion agent. Larger distance and smaller adhesion contribute to a narrower deposited trace on the first layer. Thus, the quality of the first printed layer can vary from one printing job to another and the scanned profiles of the first printed layer were not used in this study. The scanned profiles of the sixth printed layer were very noisy because the stage height was right at the edge of the working range of the laser sensor. Thus, the scanned profiles of the sixth printed layer were not used in this study. Except for the 1st and the 6th layer, the scanned profiles of the other 8 layers were used to train 8 CAN models independently. For each CAN model, the image dataset consists of 18-42 images of the scanned profiles and 50 images generated by randomly combining and cropping of the scanned profiles of the same layer. Each CAN model was trained on 70% of the images from the dataset for 100 epochs and tested on the remaining 30% of images that the model had never seen before.

Some of the predictions made by CAN models of different layers are shown in Fig. 4.5. Table 4.2 shows the percentage of pixels whose deviation predictions were within specific ranges from the actual deviation values. As we can see from the Tab. 4.2, at least 48.6%, 86.4%, 98.6% of data were within ±0.05 mm, ±0.10 mm,w ±0.15 mm of the actual measured

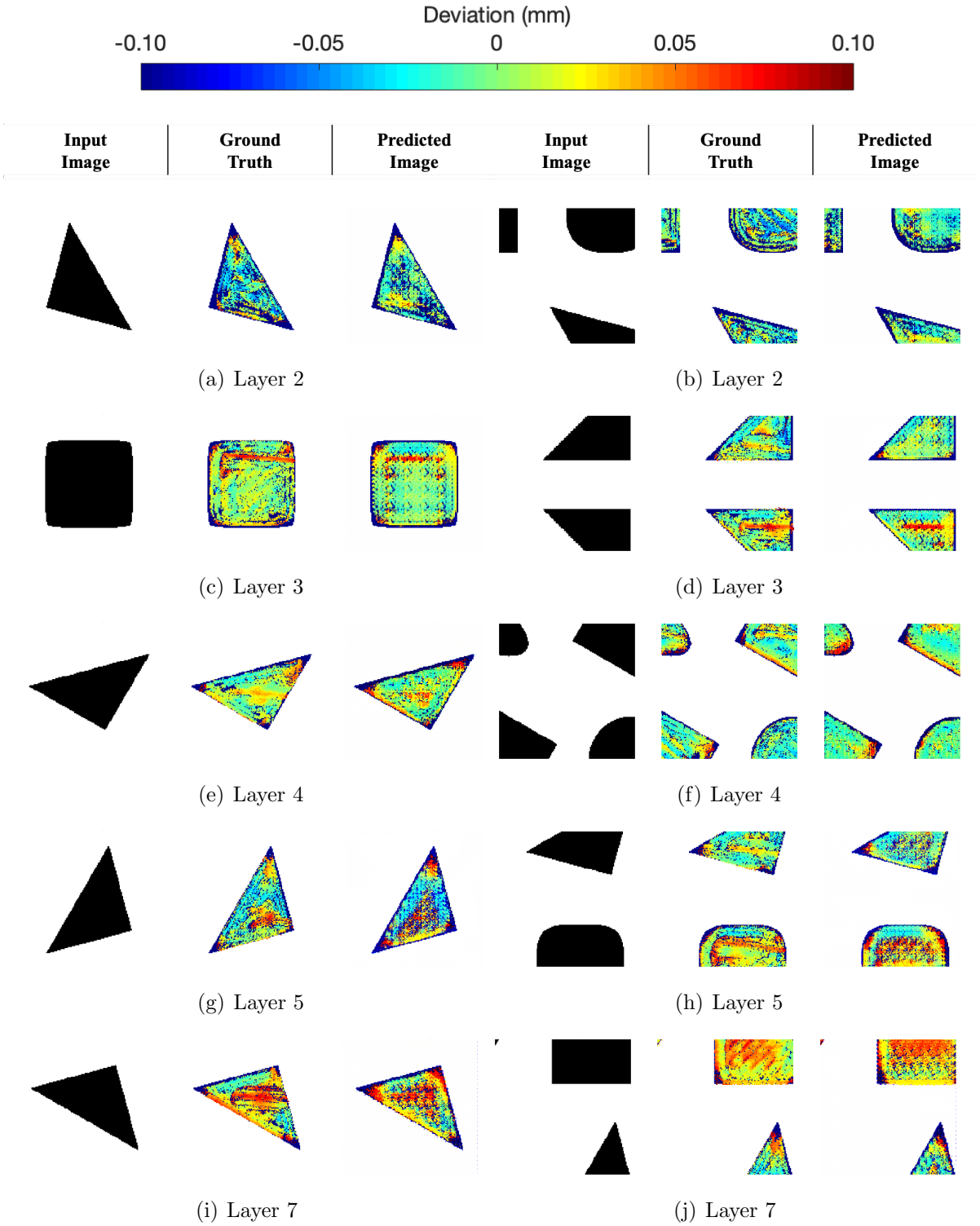
value, respectively.

Range (mm)	Percentage							
	Layer 2	**Layer 3**	**Layer 4**	**Layer 5**	**Layer 7**	**Layer 8**	**Layer 9**	**Layer 10**
±0.05	48.6%	50.9%	52.7%	53.2%	53.2%	56.0%	58.1%	59.6%
±0.10	91.3%	89.7%	89.6%	88.1%	86.4%	88.6%	87.5%	88.5%
±0.15	99.9%	99.7%	99.4%	99.0%	98.7%	99.3%	99.0%	98.6%

Table 4.2: Deviation of prediction from CAN versus measured values. Values reported as the percentage of measurements within each given range of deviation.

4.6 Case Study On Larger Prints

To further investigate the performance of CAN models, three freeform shapes (Fig. 4.6) were printed and scanned. These three freeform shapes are miniatures of the "teardrop" shape in [25] and the "convex freeform" and "concave freeform" shapes in [26]. The CAD models and scanned deviation data of these shapes were used to test the CAN models. One thing to note is that the input and predicted images of the CAN models are 256 pixels by 256 pixels, which correspond to physical areas of 14 mm by 14 mm. The "teardrop" shape in this study has a characteristic length of 9 mm, which can fit entirely into one image. The "convex freeform" and "concave freeform" shapes, however, have a characteristic length of 18 mm, which needs two input/output images to cover the whole shape. Fortunately, the CAN models were trained using the extended image dataset including the images generated by randomly combining and cropping of the original images. Thus, the CAN models are capable of predicting deviation of shapes with a characteristic length larger than 9 mm. Examples of predicted geometric deviations of these three shapes are shown in Fig. 4.7. The percentage of predictions within specific ranges from the actual deviation values are shown in Table 4.3. CAN models produced more accurate predictions than that of the model in [25] and at least 44.4%, 87.6%, 99.2% of data were within ±0.05 mm, ±0.10 mm, ±0.15 mm of the actual measured value, respectively.



(a) Layer 2

(b) Layer 2

(c) Layer 3

(d) Layer 3

(e) Layer 4

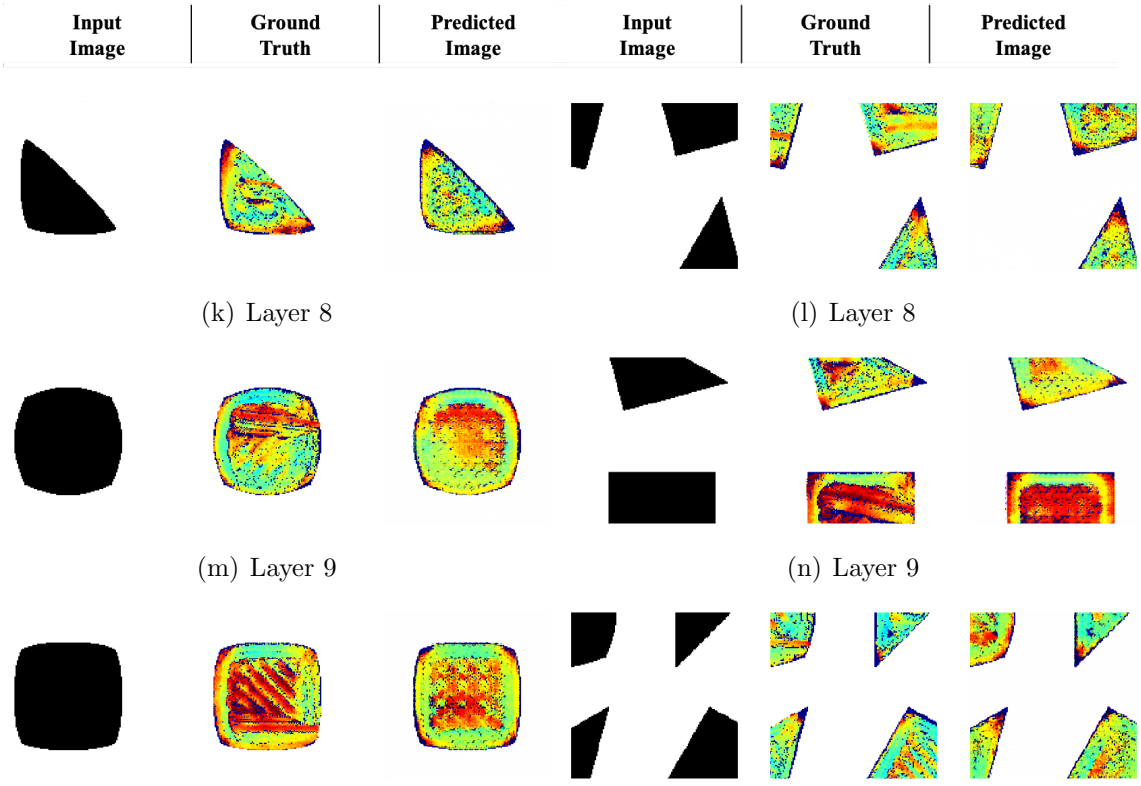
(f) Layer 4

(g) Layer 5

(h) Layer 5

(i) Layer 7

(j) Layer 7



(k) Layer 8 (l) Layer 8

(m) Layer 9 (n) Layer 9

(o) Layer 10 (p) Layer 10

Figure 4.5: Examples of predicted images of different layers on the test dataset.

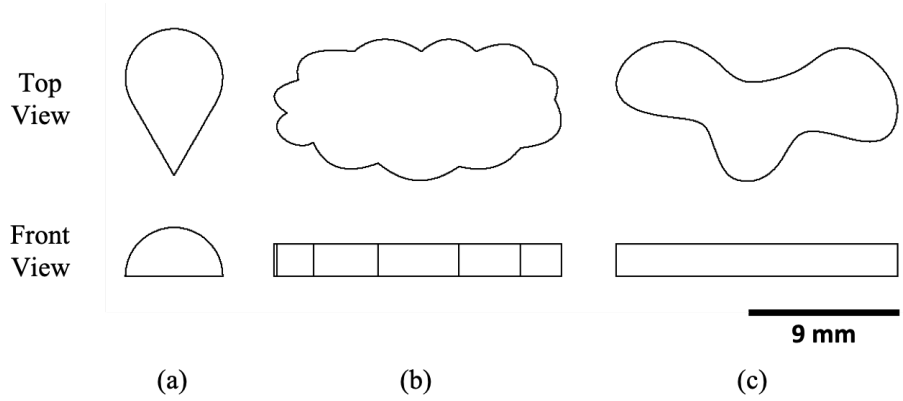


Figure 4.6: Top and front views of CAD models of the (a) teardrop [25], (b) convex freeform, and (c) concave freeform [26] shapes.

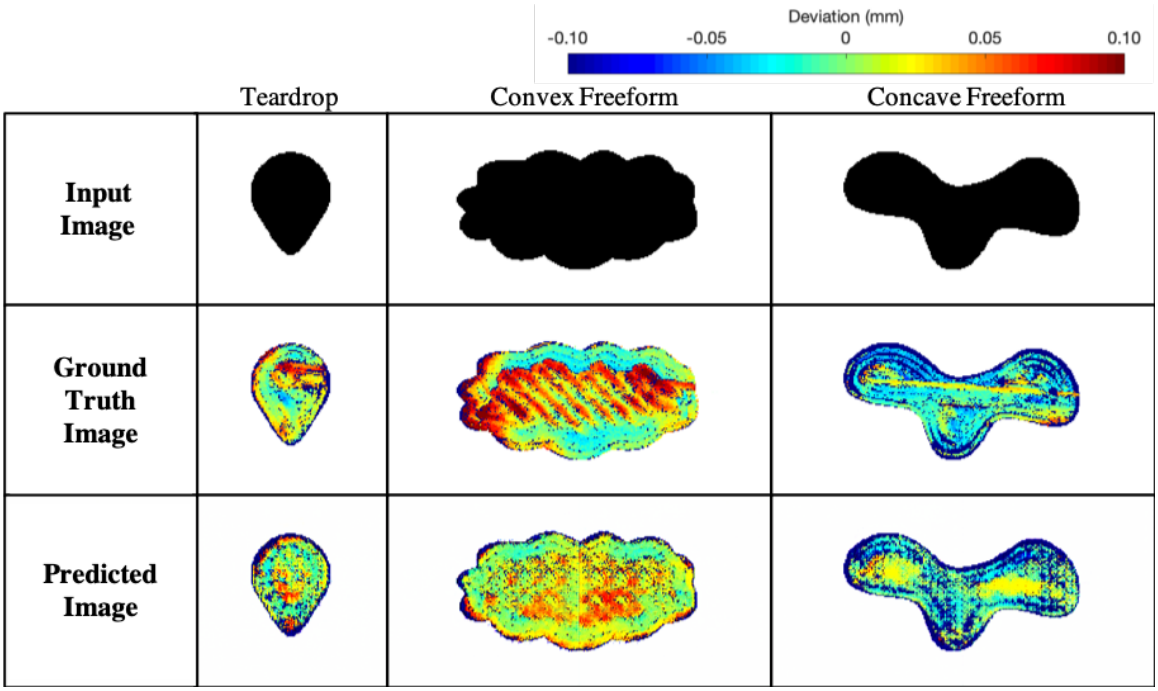


Figure 4.7: Examples of predicted images of freeform shapes.

Range (mm)	**Percentage (%)**								
	Proposed Model on Different Layers of 3 Shapes								***Decker et al.*** [25]
	L2	L3	L4	L5	L7	L8	L9	L10	
±0.05	45.9	46.3	44.4	44.9	47.1	48.5	55.2	50.5	***38.3***
±0.10	90.6	90.6	90.0	89.3	87.6	90.9	91.0	91.6	***66.7***
±0.15	99.9	99.9	99.8	99.6	99.2	99.8	99.7	99.6	***83.7***

Table 4.3: Comparison between the predictions made by our proposed CAN model and the Random Forest model in [25]. Our proposed model made predictions on different layers of the "Teardrop", "Convex Freeform", and "Concave Freeform" shapes. The model reported in [25] made predictions on the "Teardrop" shape. A higher percentage of predicted deviation within given intervals of the experimentally measured value implies that the model is more accurate.

CHAPTER 5

Redesign For AM By Compensating CAD Based On Predicted Geometric Deviation

We now have the capability of predicting 3D geometric inaccuracies based on CAD models. Looking forward, CAN models can conceivably be used to reverse engineer and redesign the CAD model, allowing for compensation of print imperfection at the design stage, in advance of the first printing.

There are two ways to use the CAN machine learning model to redesign the CAD model: Direct Reverse method and Iterative Forward method.

5.1 Redesign Using Direct Reverse Method

The Direct Reverse (DR) method works by reverse training another CAN model to predict the CAD model from the scanned profiles. This model is first trained on scanned profiles as input images and cross-sections of CAD models as ground truth images. Then, an imaginary desired scanned profile, which is created by filling the cross-section of the CAD model with the color indicating the ideal height, is fed into the DR model to anticipate the CAD model (Figure. 5.1).

This CAN machine learning model was trained the same way as those models in Chapter 4. To see if this CAN model can redesign the geometry of the object to achieve a better geometric accuracy, an ideal scanned profile of an object with an equilateral triangle cross-section was created. This desired scanned profile image is in light green color which indicates all of the points on the scanned surface have 0 mm geometric deviation. Figure. 5.2 shows

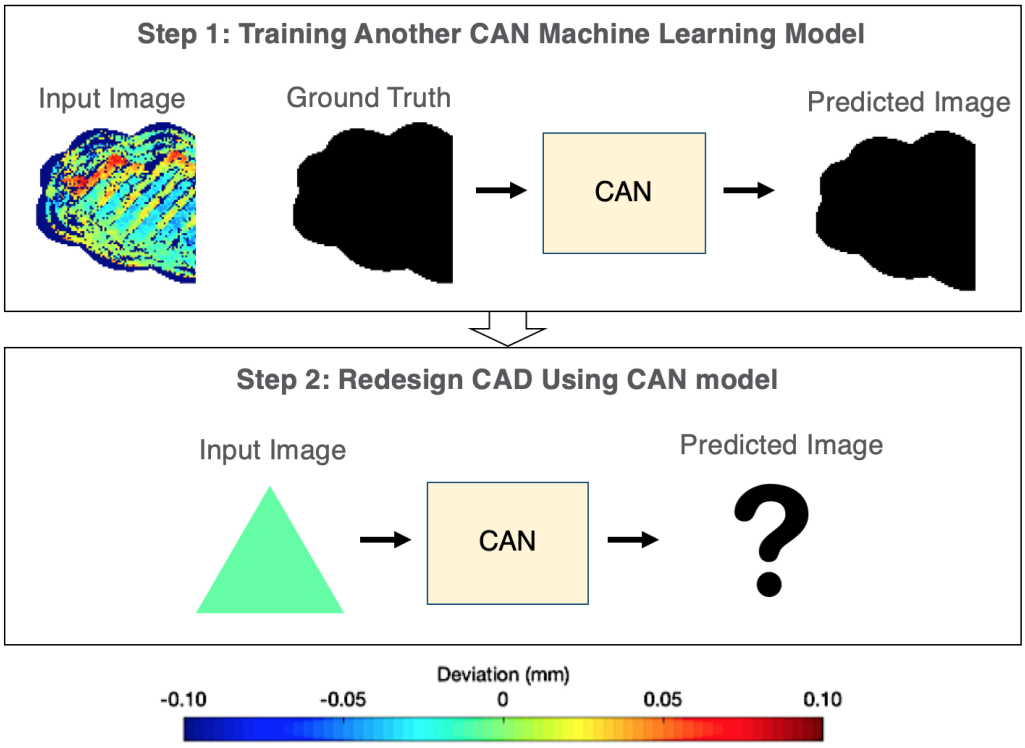


Figure 5.1: A flow chart shows the working mechanism of the Direct Reverse method.

the prediction made by the CAN model based on the input image. However, the geometry of the predicted CAD is identical to the shape of the input image. And we already know that this predicted CAD will have rounded corners when it is being printed out. So this trained CAN will not correctly anticipate the desired CAD images based on the imaginary ideal scanned profile images.

Two reasons contribute to the incorrect prediction. First, the image dataset for training the CAN model is not suitable for the reverse training purpose. If you take a closer look at the Input images and Ground Truth images in Figure. 4.2 and Figure. 4.5, almost every scanned profile image has a dark blue edge which means the printed part is smaller than the designed shape. Thus, the CAN model can easily learn that the CAD model cross-section image can be generated successfully by simply turning every colorful pixel in the scanned profile image into black. And it only took the CAN model two training epochs to find out that trick. Another reason is that the ideal scanned profile image was manually created and it is nothing similar to the true scanned profile images although they share the same

Figure 5.2: The CAN model made a prediction of design based on its ideal scanned profile image. The input image (left) is the scanned profile of the corresponding layer, in which the color indicates 3D geometric deviation. The ground truth image (right) is a cross-section of the CAD model.

colormap. And the ideal scanned profile image is not realistic and not what the CAN model can anticipate by reverse training on the same image dataset used in Chapter 4. Thus, the Iterative Forward method is developed to redesign the CAD to achieve better printing quality at the first time of printing.

5.2 Redesign Using Iterative Forward Method

The Iterative Forward (IF) method does not need to train another CAN model. It repeatedly used the trained CAD models in Chapter 4 to make predictions of the scanned profiles based on the cross-sections of the CAD model. Then the predicted scanned profiles are compared with the ideal geometry of the object (the uncompensated cross-section of the CAD) to find out which areas in the CAD need to be expanded ("grow") or reduced ("shrink"). Then the CAD is compensated and fed into the CAN machine learning model again to repeat the whole process. The goal is to minimize the number of pixels in the "grow" and "shrink" areas by iteratively making compensations to CAD based on the comparison between the predicted scanned profile and the ideal CAD (Figure. 5.3).

In Figure. 5.3, Step 1 is the same as the process introduced in Chapter 4. Step 2 is to compare the predicted image generated in Step 1 to the ideal geometry of the design, which

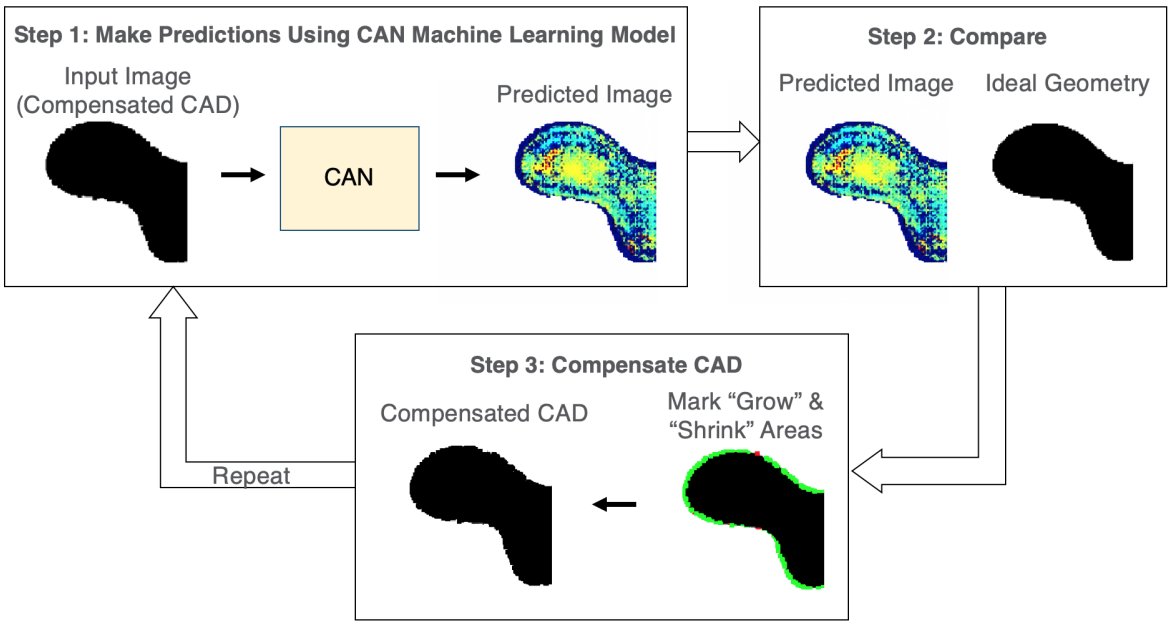


Figure 5.3: A flow chart shows the working mechanism of the Iterative Forward method. The green and red areas indicate the "grow" and "shrink" areas, respectively.

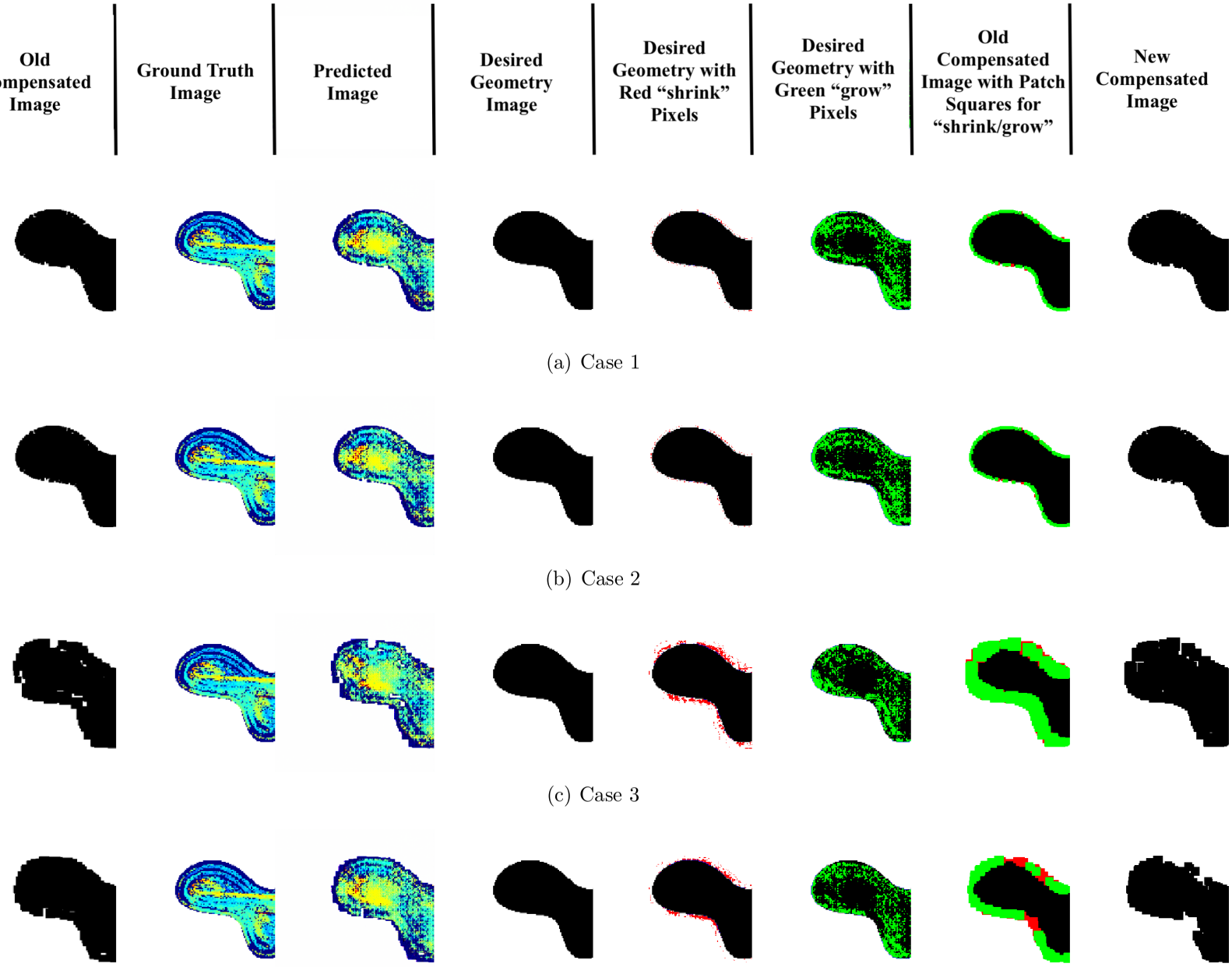
is also known as the original CAD image. A pixel is marked as "shrink" (red) if this pixel in the ideal geometry image is white, which means no material should be printed on this point, but the pixel with the same coordinates in the predicted image is not white or the geometric deviation indicating by its color is larger than -0.05 mm, which means, in reality, there was some material been accidentally printed there. Similarly, a pixel is marked as "grow" (green) if this pixel in the ideal geometry image is black, which means there should be some material there, but the pixel with the same coordinate in the predicted image is white or the related geometric deviation is less than -0.05 mm, which means the material was missing on that point. Those "shrink" and "grow" pixels are shown in the 5th and 6th columns in Figure. 5.4, respectively.

In Step 3, the closest boundary pixel was found for each of the "grow" and "red" pixels in Step 2. The closed boundary pixel is defined by two criteria: 1) it must be a pixel that is marked "grow" or "shrink" in Step 2; 2) it must be on the boundary in the input image in Step 1 since it is meaningless to expand or shrink areas in the middle of a cross-section. In addition to the closest boundary pixel, the maximum distance among all distances between

Case #	Parameters				Results			
	Range N	Range (mm)	Length of Patch Square		# of Shrink Pixels		# of Grow Pixels	
			Shrink	Grow	Initial	Final	Initial	Final
1	24	1	2log(Max Distance+1)	2log(Max Distance+1)	19	16	382	358
2	12	0.5	2log(Max Distance+1)	2log(Max Distance+1)	22	16	385	365
3	12	0.5	Max Distance/$\sqrt{2}$	Max Distance/$\sqrt{2}$	7	126	386	247
4	9	0.375	Max Distance/$\sqrt{2}$	Max Distance/$\sqrt{2}$	6	129	389	206
5	6	0.25	Max Distance/$\sqrt{2}$	Max Distance/$\sqrt{2}$	6	58	386	286
6	3	0.125	Max Distance/$\sqrt{2}$	Max Distance/$\sqrt{2}$	4	6	383	377
7	6	0.25	Max Distance/2	Max Distance/$\sqrt{2}$	7	51	380	293
8	6	0.25	Max Distance/3	Max Distance/$\sqrt{2}$	5	53	381	293

Table 5.1: Eight sets of parameters were used in Step 3 of the Iterative Forward method. The best results were selected from the first 20 iterations. All unstated units are pixels.

the "grow"/"shrink" pixels (within a range N from the boundary) and the coorelated closest boundary point was found as well. The length of square "grow"/"shrink" areas as shown in the second image in Step 3 in Figure. 5.3 is defined as a function of the Maximum Distance. Table. 5.1 shows eight sets of parameters have been used to determine the length of the path square for "Shrink" or "Grow". As it was mentioned before in Section 5.1, the printer was set to print the part which seems always to be a little bit smaller than the design. Thus, the initial number of "shrink" pixels is much smaller than the initial number of "grow" pixels, which means the CAD needs to be expanded to cover more pixels. However, Table 5.1 also shows that there is a tradeoff between the final number of "shrink" and "grow" pixels. If you want to compensate the areas that are lack material by expanding the CAD model, there is a large chance that you will over-expand the CAD model and introduced more areas that need to be removed/shrank as shown in Figure. 5.4. All MATLAB codes have been included in Appendix A.

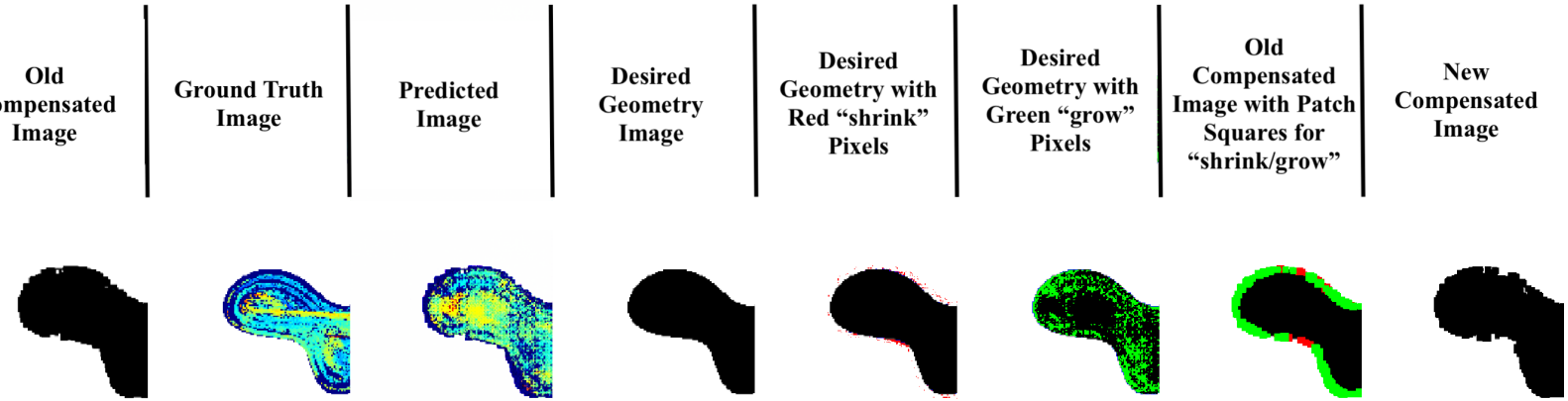


(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4



(e) Case 5

(f) Case 6

(g) Case 7

(h) Case 8

Figure 5.4: The images generated by the eight cases in 5.1.

CHAPTER 6

Summary and conclusions

6.1 Chapter 2: literature review

This chapter provided a detailed literature review of the state-of-the-art additive manufacturing techniques, the Material Extrusion type 3D printer that is used in this study, the *in-situ* monitoring technique, and closed-loop feedback system implemented in AM, and the statistical and machine learning models that can describe and predict geometric deviation from CAD models.

6.2 Chapter 3: system setup, working machnism, scan strategy, and validation of the custom built *in-situ* metrology system

This chapter developed and demonstrated an approach for layer-by-layer mapping of 3D printed parts, which can be used for validation of printed models and *in situ* adjustment of print parameters. This *in situ* metrology system scans each layer at the time of printing, providing a 3D model of the as-printed part. A high-speed optical scanning system was integrated with a Material Extrusion type 3D printer to achieve *in situ* monitoring of dimensional inaccuracies during printing, which leaves the door open to implement a closed-loop feedback system to compensate geometric errors during printing in the future and fabricate "certify-as-you-build" products.

6.3 Chapter 4: train, test, and tune parameters of the Conditional Adversarial Networks (CAN) machine learning models

This chapter trained the CAN machine learning algorithms with data from this scanning system and predicted 3D geometric inaccuracies in new designs. Eight conditional adversarial networks (CAN) machine learning models were trained on a limited number of scanned profile images of different layers, consisting of less than 50 actual images and 50 generated images, to predict the 3D geometric deviations of freeform shapes. The generated images were produced by randomly combining and cropping the actual images without any distortion. These CAN models produced predictions where at least 44.4%, 87.6%, 99.2% of data were within ±0.05 mm, ±0.10 mm, ±0.15 mm of the actual measured value, respectively.

6.4 Chapter 5: redesign for Additive Manufacturing using the predictions of the CAN machine learning models

This chapter developed an Iterative Forward approach to redesign the Computer-Aided-Design model by reverse engineering using the trained machine learning models, allowing for compensation of print imperfection at the design stage, in advance of the first printing. The compensation algorithms with eight different sets of different parameters were evaluated. It has been proven that the Iterative Forward approach improved the geometric deviation of the predicted profiles by making compensation to the CAD model.

APPENDIX A

MATLAB codes for Iterative Forward method

```
clc;
clear all;
close all;

maxNumOfIteration = 20;
% column 1 - green; column 2 - red; column 3 - total pixels
numOfGreenRedPoint = zeros(maxNumOfIteration,3);

for ARG1 = 1:maxNumOfIteration
    % %% Section 1.1: test HelloWorld.py
    % [result, status] = python('HelloWorld.py')

    % %% Section 1.2: run ML code
    %
    % [result, status] = python('Pix2Pix_improve_v2_layer2-Copy1.py')


    %% Section 1.3: run ML code with arguments
    % % pass ARG1 to CAN code
    % ARG1 = 1;
    [result, status] = python('Pix2Pix_improve_v2_layer2-Copy1.py',num2str(...
        ARG1))

    %% Section 2.1: read images from a specific folder
```

```
    myFolder = sprintf('/Users/Ling/Documents/Research_Related/3...
        D_Printing_Metrology/Feedback System/outputOfCAN/');
    color = jet(100);

    %load images into array "all_images" if we havent already
    if ¬exist('all_images','var')
        if ¬isfolder(myFolder)
            errorMessage = sprintf('Error: The following folder does not ...
                exist:\n%s', myFolder);
            uiwait(warndlg(errorMessage));
            return;
        end
        filePattern = fullfile(myFolder, '*.png');
        pngFiles = dir(filePattern);
        totImages = length(pngFiles);
        images = cell([totImages 3]);
        % first column input
        % second column ground truth
        % third column predicted image
        errorValue = [];
        for k = ARG1
%             baseFileName = pngFiles(k).name;
            baseFileName = sprintf(['predicted_layer2_epoch100_' num2str(k)...
                '.png']);
            fullFileName = fullfile(myFolder, baseFileName);
            imageArray = imread(fullFileName);
            %       imshow(imageArray);
            if k == 1
                inputImage0 = imageArray(588:928,189:530,:);
                groundTruthImage0 = imageArray(588:928,599:940,:);
            end
            images{k,1} = imageArray(588:928,189:530,:);
            images{k,2} = imageArray(588:928,599:940,:);
            images{k,3} = imageArray(588:928,1009:1350,:);
```

```
        %   drawnow; % Force display to update immediately.
            [compensatedImage,illustrationImage,numOfGreenRedPoint(k,1),...
               numOfGreenRedPoint(k,2),numOfGreenRedPoint(k,3),shrinkImage...
               ,growImage] = growAndShrink_v2(inputImage0,images{k,3},...
               color);
            newImageArray1 = [compensatedImage groundTruthImage0];
            newImageArray11 = imresize(newImageArray1, [256 512]);
            fullFileName_1 = sprintf('/Users/Ling/Documents/...
               Research_Related/3D_Printing_Metrology/Feedback System/...
               inputOfCAN/%d.png',ARG1+1);
            imwrite(newImageArray11,fullFileName_1);
            newImageArray2 = [images{k,1} groundTruthImage0 images{k,3} ...
               inputImage0 shrinkImage growImage illustrationImage ...
               compensatedImage];
            fullFileName_2 = sprintf('/Users/Ling/Documents/...
               Research_Related/3D_Printing_Metrology/Feedback System/...
               records/%d.png',ARG1);
            imwrite(newImageArray2,fullFileName_2);
        end
    end
end
load gong.mat;
sound(y);
```

```
function [compensatedImage,illustrationImage,tempGreen,tempRed,tempTotal,...
   shrinkImage,growImage] = growAndShrink_v2(input,predicted,color)
dim = size(input);
growArea = zeros(dim(1:2));
growBoundary = zeros(dim(1:2));
shrinkArea = zeros(dim(1:2));
shrinkBoundary = zeros(dim(1:2));
compensatedImage = input;
illustrationImage = input;
```

```
shrinkImage = input;
growImage = input;
tempGreen = 0;
tempRed = 0;
tempTotal = 0;
figure(1);
imshow(input);
figure(2);
imshow(predicted);
%%
for i = 1:dim(1)
    for j = 1:dim(2)
        % recognize grow and shrink area and boundary
        inputColor = double(input(i,j,:))/255;
        predictedColor = double(predicted(i,j,:))/255;
        % criteria is white in the predicted image
        criteria = predicted(i,j,1) > 220 && predicted(i,j,2) > 220 && ...
            predicted(i,j,3) > 220;
        % check shrink area
        if inputColor(:,:,1) > 0.5 && inputColor(:,:,2) > 0.5 && inputColor...
            (:,:,3) > 0.5   % input color is white
            if ¬criteria
                color_diff_temp = [color(:,1) - predictedColor(:,:,1),color...
                    (:,2) - predictedColor(:,:,2),color(:,3) - ...
                    predictedColor(:,:,3)];
                color_diff = sum((color_diff_temp).^2,2);
                [useless colorIdx] = min(color_diff);
                if colorIdx > 25   % predicted deviation is > -0.05 mm
                    shrinkArea(i,j) = 1;
                    shrinkImage(i,j,:) = [255 0 0];
                    if i>1 && i<dim(1) && j>1 && j<dim(2)
                        upColor = double(input(i-1,j,:))/255;
                        up = (1-upColor(:,:,1)) * (1-upColor(:,:,2)) * (1-...
                            upColor(:,:,3));
```

```
                downColor = double(input(i+1,j,:))/255;
                down = (1-downColor(:,:,1)) * (1-downColor(:,:,2)) ...
                    * (1-downColor(:,:,3));
                leftColor = double(input(i,j-1,:))/255;
                left = (1-leftColor(:,:,1)) * (1-leftColor(:,:,2)) ...
                    * (1-leftColor(:,:,3));
                rightColor = double(input(i,j+1,:))/255;
                right = (1-rightColor(:,:,1)) * (1-rightColor...
                    (:,:,2)) * (1-rightColor(:,:,3));
                if up == 1 || down == 1 || left == 1 || right == 1
                    shrinkBoundary (i,j) = 1;
                    shrinkImage(i,j,:) = [0 0 255];
                end
            end
        end
    end
end
% check grow area and grow boundary
if inputColor(:,:,1) <= 0.5 && inputColor(:,:,2) <= 0.5 && inputColor...
    (:,:,3) <= 0.5   % input color is black
    color_diff_temp = [color(:,1) - predictedColor(:,:,1),color...
        (:,2) - predictedColor(:,:,2),color(:,3) - predictedColor...
        (:,:,3)];
    color_diff = sum((color_diff_temp).^2,2);
    [useless colorIdx] = min(color_diff);
    if colorIdx <= 25 || criteria   % predicted deviation is <-0.05...
        mm
        growArea(i,j) = 1;
        growImage(i,j,:) = [0 255 0];
        if i>1 && i<dim(1) && j>1 && j<dim(2)
            upColor = double(input(i-1,j,:))/255;
            up = upColor(:,:,1) * upColor(:,:,2) * upColor(:,:,3);
            downColor = double(input(i+1,j,:))/255;
            down = downColor(:,:,1) * downColor(:,:,2) * downColor...
```

```
                    (:,:,3);
                leftColor = double(input(i,j-1,:))/255;
                left = leftColor(:,:,1) * leftColor(:,:,2) * leftColor...
                    (:,:,3);
                rightColor = double(input(i,j+1,:))/255;
                right = rightColor(:,:,1) * rightColor(:,:,2) * ...
                    rightColor(:,:,3);
                if up == 1 || down == 1 || left == 1 || right == 1
                    growBoundary(i,j) = 1;
                    growImage(i,j,:) = [0 0 255];
                end
            end
        end
    end
end

%% calculate shrinkMap and growMap
% the number on each pixel is the range of area to be grown or shrunk
shrinkMap = zeros(dim);
growMap = zeros(dim);
for i = 1:dim(1)
    for j = 1:dim(2)
        if shrinkArea(i,j) == 1
            % check 0.5 mm / 12 pixel area of this specific point
            range = 24;
            shrinkMap = calculateMap(range,i,j,dim,shrinkBoundary,shrinkMap...
                );
        end
        if growArea(i,j) == 1
            range = 24;
            growMap = calculateMap(range,i,j,dim,growBoundary,growMap);
        end
    end
```

```
end
growMap = floor(log(growMap+1)*2);
shrinkMap = floor(log(shrinkMap+1)*2);

%% color/compensate CAD input image based on shrinkMap and growMap
for i = 1:dim(1)
    for j = 1:dim(2)
        if shrinkMap(i,j) > 0
            % shrink boundary by shrinkMap(i,j) pixels to white
            tempRed = tempRed + 1;
            tempRange = floor(sqrt(shrinkMap(i,j)));
            upTo = max(i-tempRange,1);
            downTo = min(i+tempRange,dim(1));
            leftTo = max(j-tempRange,1);
            rightTo = min(j+tempRange,dim(2));
            for m = upTo:downTo
                for n = leftTo:rightTo
                    compensatedImage(m,n,:) = [255,255,255];
                    illustrationImage(m,n,:) = [255,0,0];
                end
            end
        end
    end
end
for i = 1:dim(1)
    for j = 1:dim(2)
        if growBoundary(i,j) == 1
            % grow boundary by growMap(i,j) pixels to black
            tempGreen = tempGreen + 1;
            tempRange = floor(sqrt(growMap(i,j)));
            upTo = max(i-tempRange,1);
            downTo = min(i+tempRange,dim(1));
            leftTo = max(j-tempRange,1);
            rightTo = min(j+tempRange,dim(2));
```

```
            for m = upTo:downTo
                for n = leftTo:rightTo
                    compensatedImage(m,n,:) = [0,0,0];
                    illustrationImage(m,n,:) = [0,255,0];
                end
            end
        end
    end
end
tempTotal = dim(1)*dim(2);
```

```
function shrinkMap = calculateMap(range,i,j,dim,shrinkBoundary,shrinkMap)
upTo = max(i-range,1);
downTo = min(i+range,dim(1));
leftTo = max(j-range,1);
rightTo = min(j+range,dim(2));
% find the closest boundary point
tempClosestBoundary = zeros(1,2);
lastDistance = 0;
for m = upTo:downTo
    for n = leftTo:rightTo
        if shrinkBoundary(m,n) == 1
            if tempClosestBoundary(1) == 0
                tempClosestBoundary = [m n];
                lastDistance = (m-i)^2 + (n-j)^2;
            else
                tempDistance = (m-i)^2 + (n-j)^2;
                if tempDistance < lastDistance
                    tempClosestBoundary = [m n];
                    lastDistance = tempDistance;
                end
            end
        end
```

```
    end
end
if tempClosestBoundary(1) > 0
    oldDistance = shrinkMap(tempClosestBoundary(1),tempClosestBoundary(2));
    if lastDistance > oldDistance
    shrinkMap(tempClosestBoundary(1),tempClosestBoundary(2)) = lastDistance...
        ;
    end
end
```

REFERENCES

[1] X. Xu, S. Meteyer, N. Perry, and Y. F. Zhao, "Energy consumption model of binder-jetting additive manufacturing processes," *International Journal of Production Research*, vol. 53, no. 23, pp. 7005–7015, 2015.

[2] Z. Wang, T. A. Palmer, and A. M. Beese, "Effect of processing parameters on microstructure and tensile properties of austenitic stainless steel 304l made by directed energy deposition additive manufacturing," *Acta Materialia*, vol. 110, pp. 226–235, 2016.

[3] D. Ding, Z. Pan, D. Cuiuri, and H. Li, "Wire-feed additive manufacturing of metal components: technologies, developments and future interests," *The International Journal of Advanced Manufacturing Technology*, vol. 81, no. 1-4, pp. 465–481, 2015.

[4] J. Gonzalez-Gutierrez, S. Cano, S. Schuschnigg, C. Kukla, J. Sapkota, and C. Holzer, "Additive manufacturing of metallic and ceramic components by the material extrusion of highly-filled polymers: A review and future perspectives," *Materials*, vol. 11, no. 5, p. 840, 2018.

[5] H. Yang, J. C. Lim, Y. Liu, X. Qi, Y. L. Yap, V. Dikshit, W. Y. Yeong, and J. Wei, "Performance evaluation of projet multi-material jetting 3d printer," *Virtual and physical prototyping*, vol. 12, no. 1, pp. 95–103, 2017.

[6] W. E. King, A. T. Anderson, R. M. Ferencz, N. E. Hodge, C. Kamath, S. A. Khairallah, and A. M. Rubenchik, "Laser powder bed fusion additive manufacturing of metals; physics, computational, and materials challenges," *Applied Physics Reviews*, vol. 2, no. 4, p. 041304, 2015.

[7] I. Gibson, D. W. Rosen, and B. Stucker, "Sheet lamination processes," in *Additive Manufacturing Technologies*, pp. 223–252, Springer, 2010.

[8] J. Frketic, T. Dickens, and S. Ramakrishnan, "Automated manufacturing and processing of fiber-reinforced polymer (frp) composites: An additive review of contemporary and modern techniques for advanced materials manufacturing," *Additive Manufacturing*, vol. 14, pp. 69–86, 2017.

[9] F. P. Melchels, J. Feijen, and D. W. Grijpma, "A review on stereolithography and its applications in biomedical engineering," *Biomaterials*, vol. 31, no. 24, pp. 6121–6130, 2010.

[10] I. El-Katatny, S. Masood, and Y. Morsi, "Error analysis of fdm fabricated medical replicas," *Rapid Prototyping Journal*, vol. 16, no. 1, pp. 36–43, 2010.

[11] M. L. Smith and J. F. Jones, "Dual-extrusion 3d printing of anatomical models for education," *Anatomical sciences education*, vol. 11, no. 1, pp. 65–72, 2018.

[12] F. Baumann and D. Roller, "Vision based error detection for 3d printing processes," in *MATEC web of conferences*, vol. 59, p. 06003, EDP Sciences, 2016.

[13] R. A. Lyngby, J. Wilm, E. R. Eiríksson, J. B. Nielsen, J. N. Jensen, H. Aanæs, and D. B. Pedersen, "In-line 3d print failure detection using computer vision," in *euspen and ASPE Special Interest Group Meeting: Additive Manufacturing*, 2017.

[14] D. M. Roberson III, *Sensor-based online process monitoring in advanced manufacturing*. PhD thesis, Virginia Tech, 2016.

[15] S. Nuchitprasitchai, M. Roggemann, and J. Pearce, "Three hundred and sixty degree real-time monitoring of 3-d printing using computer analysis of two camera views," *Journal of Manufacturing and Materials Processing*, vol. 1, no. 1, p. 2, 2017.

[16] C. Liu, D. Roberson, and Z. Kong, "Textural analysis-based online closed-loop quality control for additive manufacturing processes," in *Proceedings of the IIE Annual Conference*, pp. 20–23, 2017.

[17] Y. Tlegenov, G. S. Hong, and W. F. Lu, "Nozzle condition monitoring in 3d printing," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 45–55, 2018.

[18] Y. Li, W. Zhao, Q. Li, T. Wang, and G. Wang, "In-situ monitoring and diagnosing for fused filament fabrication process based on vibration sensors," *Sensors*, vol. 19, no. 11, p. 2589, 2019.

[19] P. K. Rao, J. P. Liu, D. Roberson, Z. J. Kong, and C. Williams, "Online real-time quality monitoring in additive manufacturing processes using heterogeneous sensors," *Journal of Manufacturing Science and Engineering*, vol. 137, no. 6, p. 061007, 2015.

[20] J. Long, Z. Sun, C. Li, Y. Hong, Y. Bai, and S. Zhang, "A novel sparse echo autoencoder network for data-driven fault diagnosis of delta 3-d printers," *IEEE Transactions on Instrumentation and Measurement*, 2019.

[21] Y. Lu and Y. Wang, "An efficient transient temperature monitoring of fused filament fabrication process with physics-based compressive sensing," *IISE Transactions*, vol. 51, no. 2, pp. 168–180, 2019.

[22] F. Li, Z. Yu, Z. Yang, and X. Shen, "Real-time distortion monitoring during fused deposition modeling via acoustic emission," *Structural Health Monitoring*, p. 1475921719849700, 2019.

[23] J. Liu, Y. Hu, B. Wu, and Y. Wang, "An improved fault diagnosis approach for fdm process with acoustic emission," *Journal of Manufacturing Processes*, vol. 35, pp. 570–579, 2018.

[24] P. Sitthi-Amorn, J. E. Ramos, Y. Wangy, J. Kwan, J. Lan, W. Wang, and W. Matusik, "Multifab: a machine vision assisted platform for multi-material 3d printing," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, pp. 1–11, 2015.

[25] N. Decker and Q. Huang, "Geometric accuracy prediction for additive manufacturing through machine learning," in *ASME 2019 International Manufacturing Science and Engineering Conference*, American Society of Mechanical Engineers, 2019.

[26] H. Luan and Q. Huang, "Prescriptive modeling and compensation of in-plane shape deformation for 3-d printed freeform products," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 73–82, 2016.

[27] Q. Huang, J. Zhang, A. Sabbaghi, and T. Dasgupta, "Optimal offline compensation of shape shrinkage for three-dimensional printing processes," *Iie transactions*, vol. 47, no. 5, pp. 431–441, 2015.

[28] Q. Huang, H. Nouri, K. Xu, Y. Chen, S. Sosina, and T. Dasgupta, "Statistical predictive modeling and compensation of geometric deviations of three-dimensional printed products," *Journal of Manufacturing Science and Engineering*, vol. 136, no. 6, p. 061008, 2014.

[29] Y. Jin, S. J. Qin, and Q. Huang, "Prescriptive analytics for understanding of out-of-plane deformation in additive manufacturing," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 786–791, IEEE, 2016.

[30] S. ISO, "Step application handbook," *SCRA, North Charleston, SC*, 2006.

[31] D. E. ISO, "Geometrical product specifications (gps)–geometrical tolerancing–tolerances of form, orientation, location and run-out," 2006.

[32] J. D. Meadows, *Geometric dimensioning and tolerancing: applications, analysis & measurement (per ASME Y14. 5-2009)*. American Society of Mechanical Engineers, 2009.

[33] L. Li, R. McGuan, P. Kavehpour, and R. N. Candler, "Precision enhancement of 3d printing via in situ metrology," in *Proceedings of the 2018 Solid Freeform Fabrication Symposium*, pp. 251–260, 2018.

[34] S. S. Crump, "Apparatus and method for creating three-dimensional objects," June 9 1992. US Patent 5,121,329.

[35] Y. Gur, "Additive manufacturing of anatomical models from computed tomography scan data," *Mol Cell Biomech*, vol. 11, no. 4, pp. 249–258, 2014.

[36] V. Bagaria and K. Chaudhary, "A paradigm shift in surgical planning and simulation using 3dgraphy: Experience of first 50 surgeries done using 3d-printed biomodels," *Injury*, vol. 48, no. 11, pp. 2501–2508, 2017.

[37] M. Macko, Z. Szczepański, D. Mikołajewski, J. Nowak, E. Mikołajewska, and J. Furtak, "Cae/fdm methods for design and manufacture artificial organs for exercises purposes," in *International Conference on Computer Aided Engineering*, pp. 462–469, Springer, 2018.

[38] Y. Wen, S. Xun, M. Haoye, S. Baichuan, C. Peng, L. Xuejian, Z. Kaihong, Y. Xuan, P. Jiang, and L. Shibi, "3d printed porous ceramic scaffolds for bone tissue engineering: a review," *Biomaterials science*, vol. 5, no. 9, pp. 1690–1698, 2017.

[39] A. Youssef, S. J. Hollister, and P. D. Dalton, "Additive manufacturing of polymer melts for implantable medical devices and scaffolds," *Biofabrication*, vol. 9, no. 1, p. 012002, 2017.

[40] G. T. Kirby, L. J. White, R. Steck, A. Berner, K. Bogoevski, O. Qutachi, B. Jones, S. Saifzadeh, D. W. Hutmacher, K. M. Shakesheff, *et al.*, "Microparticles for sustained growth factor delivery in the regeneration of critically-sized segmental tibial bone defects," *Materials*, vol. 9, no. 4, p. 259, 2016.

[41] M. Trivedi, J. Jee, S. Silva, C. Blomgren, V. M. Pontinha, D. L. Dixon, B. Van Tassel, M. J. Bortner, C. Williams, E. Gilmer, *et al.*, "Additive manufacturing of pharmaceuticals for precision medicine applications: A review of the promises and perils in implementation," *Additive Manufacturing*, vol. 23, pp. 319–328, 2018.

[42] B. Nematollahi, M. Xia, S. H. Bong, and J. Sanjayan, "Hardened properties of 3d printable ?one-part?geopolymer for construction applications," in *RILEM International Conference on Concrete and Digital Fabrication*, pp. 190–199, Springer, 2018.

[43] W. Yi, H. Ketai, Z. Xiaomin, and D. Wenying, "Machine vision based statistical process control in fused deposition modeling," in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 936–941, IEEE, 2017.

[44] J. law Fastowicz and K. Okarma, "Quality assessment of photographed 3d printed flat surfaces using hough transform and histogram equalization," *Journal of Universal Computer Science*, vol. 25, no. 6, pp. 701–717, 2019.

[45] S. Nuchitprasitchai, M. Roggemann, and J. M. Pearce, "Factors effecting real-time optical monitoring of fused filament 3d printing," *Progress in Additive Manufacturing*, vol. 2, no. 3, pp. 133–149, 2017.

[46] K. He, H. Wang, and H. Hu, "Approach to online defect monitoring in fused deposition modeling based on the variation of the temperature field," *Complexity*, vol. 2018, 2018.

[47] W. Yi, H. Ketai, H. Huaqing, and Z. Xue, "Process monitoring of fused deposition modeling through profile control," in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, pp. 346–350, IEEE, 2018.

[48] M. Wu, V. V. Phoha, Y. B. Moon, and A. K. Belman, "Detecting malicious defects in 3d printing process using machine learning and image classification," in *ASME 2016 International Mechanical Engineering Congress and Exposition*, pp. V014T07A004–V014T07A004, American Society of Mechanical Engineers, 2016.

[49] O. Holzmond and X. Li, "In situ real time defect detection of 3d printed parts," *Additive Manufacturing*, vol. 17, pp. 135–142, 2017.

[50] Y. Tlegenov, *Model-Based Monitoring of Nozzle Clogging in Fused Deposition Modelling Process*. PhD thesis, National University of Singapore (Singapore), 2018.

[51] F. Baumann, M. Schön, J. Eichhoff, and D. Roller, "Concept development of a sensor array for 3d printer," *Procedia CIRP*, vol. 51, pp. 24–31, 2016.

[52] K. Bastani, P. K. Rao, and Z. Kong, "An online sparse estimation-based classification approach for real-time monitoring in advanced manufacturing processes from heterogeneous sensor data," *IIE Transactions*, vol. 48, no. 7, pp. 579–598, 2016.

[53] P. K. Rao, J. P. Liu, D. Roberson, and Z. J. Kong, "Sensor-based online process fault detection in additive manufacturing," in *ASME 2015 International Manufacturing Science and Engineering Conference*, pp. V002T04A010–V002T04A010, American Society of Mechanical Engineers, 2015.

[54] I. T. Cummings, M. E. Bax, I. J. Fuller, A. J. Wachtor, and J. D. Bernardin, "A framework for additive manufacturing process monitoring & control," in *Topics in Modal Analysis & Testing, Volume 10*, pp. 137–146, Springer, 2017.

[55] G. P. Greeff *et al.*, *Applied Metrology in Additive Manufacturing*. Mensch und Buch Verlag, 2018.

[56] H. Sun, P. K. Rao, Z. J. Kong, X. Deng, and R. Jin, "Functional quantitative and qualitative models for quality modeling in a fused deposition modeling process," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 393–403, 2017.

[57] Z. Yang, L. Jin, Y. Yan, and Y. Mei, "Filament breakage monitoring in fused deposition modeling using acoustic emission technique," *Sensors*, vol. 18, no. 3, p. 749, 2018.

[58] F. Li, Z. Yu, X. Shen, and H. Zhang, "Status recognition for fused deposition modeling manufactured parts based on acoustic emission," in *E3S Web of Conferences*, vol. 95, p. 01005, EDP Sciences, 2019.

[59] H. Wu, Z. Yu, and Y. Wang, "Real-time fdm machine condition monitoring and diagnosis based on acoustic emission and hidden semi-markov model," *The International Journal of Advanced Manufacturing Technology*, vol. 90, no. 5-8, pp. 2027–2036, 2017.

[60] H. Wu, Y. Wang, and Z. Yu, "In situ monitoring of fdm machine condition via acoustic emission," *The International Journal of Advanced Manufacturing Technology*, vol. 84, no. 5-8, pp. 1483–1495, 2016.

[61] H. Wu, Z. Yu, and Y. Wang, "A new approach for online monitoring of additive manufacturing based on acoustic emission," in *ASME 2016 11th International Manufacturing Science and Engineering Conference*, pp. V003T08A013–V003T08A013, American Society of Mechanical Engineers, 2016.

[62] Z. Zhu, N. Anwer, and L. Mathieu, "Statistical modal analysis for out-of-plane deviation prediction in additive manufacturing based on finite element simulation," *Journal of Manufacturing Science and Engineering*, vol. 141, no. 11, 2019.

[63] Wikipedia contributors, "Machine vision — Wikipedia, the free encyclopedia," 2020. [Online; accessed 19-April-2020].

[64] Z. Zhu, N. Anwer, and L. Mathieu, "Geometric deviation modeling with statistical shape analysis in design for additive manufacturing," *Procedia CIRP*, vol. 84, pp. 496–501, 2019.

[65] B. S. Rupala, K. G. Mostafaa, Y. Wangb, and A. J. Qureshia, "A reverse cad approach for estimating geometric and mechanical behavior of fdm printed parts," *Procedia Manufacturing*, vol. 34, pp. 535–544, 2019.

[66] L. Xu, Q. Huang, A. Sabbaghi, and T. Dasgupta, "Shape deviation modeling for dimensional quality control in additive manufacturing," in *ASME 2013 International Mechanical Engineering Congress and Exposition*, pp. V02AT02A018–V02AT02A018, American Society of Mechanical Engineers, 2013.

[67] S. Song, A. Wang, Q. Huang, and F. Tsung, "Shape deviation modeling for fused deposition modeling processes," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 758–763, IEEE, 2014.

[68] Q. Huang, H. Nouri, K. Xu, Y. Chen, S. Sosina, and T. Dasgupta, "Predictive modeling of geometric deviations of 3d printed products-a unified modeling approach for cylindrical and polygon shapes," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 25–30, IEEE, 2014.

[69] Q. Huang, "An analytical foundation for optimal compensation of three-dimensional shape deformation in additive manufacturing," *Journal of Manufacturing Science and Engineering*, vol. 138, no. 6, p. 061010, 2016.

[70] Q. Huang, "3d printing shrinkage compensation using radial and angular layer perimeter point information," Feb. 6 2018. US Patent 9,886,526.

[71] Q. Huang, T. Dasgupta, and S. Sosina, "Statistical predictive modeling and compensation of geometric deviations of 3d printed products," Nov. 28 2017. US Patent 9,827,717.

[72] L. Cheng, A. Wang, and F. Tsung, "A prediction and compensation scheme for in-plane shape deviation of additive manufacturing with information on process parameters," *IISE Transactions*, vol. 50, no. 5, pp. 394–406, 2018.

[73] Z. Huang, J.-Y. Dantan, A. Etienne, M. Rivette, and N. Bonnet, "Geometrical deviation identification and prediction method for additive manufacturing," *Rapid Prototyping Journal*, vol. 24, no. 9, pp. 1524–1538, 2018.

[74] A. Wang, S. Song, Q. Huang, and F. Tsung, "In-plane shape-deviation modeling and compensation for fused deposition modeling processes," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 2, pp. 968–976, 2016.

[75] T. Lieneke, V. Denzer, G. A. Adam, and D. Zimmer, "Dimensional tolerances for additive manufacturing: Experimental investigation for fused deposition modeling," *Procedia CIRP*, vol. 43, pp. 286–291, 2016.

[76] A. M. Aboutaleb, M. A. Tschopp, P. K. Rao, and L. Bian, "Multi-objective accelerated process optimization of part geometric accuracy in additive manufacturing," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 10, p. 101001, 2017.

[77] H. Luan, *Statistical Modeling and Machine Learning for Shape Accuracy Control in Additive Manufacturing*. PhD thesis, University of Southern California, 2018.

[78] Y. Jin, S. J. Qin, and Q. Huang, "Modeling inter-layer interactions for out-of-plane shape deviation reduction in additive manufacturing," *IISE Transactions*, no. just-accepted, pp. 1–16, 2019.

[79] A. Sabbaghi, Q. Huang, and T. Dasgupta, "Bayesian model building from small samples of disparate data for capturing in-plane deviation in additive manufacturing," *Technometrics*, vol. 60, no. 4, pp. 532–544, 2018.

[80] H. Luan, B. K. Post, and Q. Huang, "Statistical process control of in-plane shape deformation for additive manufacturing," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pp. 1274–1279, IEEE, 2017.

[81] Y. Jin, S. J. Qin, and Q. Huang, "Offline predictive control of out-of-plane shape deformation for additive manufacturing," *Journal of Manufacturing Science and Engineering*, vol. 138, no. 12, p. 121005, 2016.

[82] Y. Jin, S. J. Qin, and Q. Huang, "Out-of-plane geometric error prediction for additive manufacturing," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 918–923, IEEE, 2015.

[83] A. Sabbaghi, Q. Huang, and T. Dasgupta, "Bayesian additive modeling for quality control of 3d printed products," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 906–911, IEEE, 2015.

[84] H. Luan and Q. Huang, "Predictive modeling of in-plane geometric deviation for 3d printed freeform products," in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 912–917, IEEE, 2015.

[85] M. Khanzadeh, P. Rao, R. Jafari-Marandi, B. K. Smith, M. A. Tschopp, and L. Bian, "Quantifying geometric accuracy with unsupervised machine learning: using self-organizing map on fused filament fabrication additive manufacturing parts," *Journal of Manufacturing Science and Engineering*, vol. 140, no. 3, p. 031011, 2018.

[86] P. Jaiswal, *Geometric Reasoning and Machine Learning: A Set of Design and Manufacturing Problems*. PhD thesis, State University of New York at Buffalo, 2019.

[87] Z. Zhu, N. Anwer, Q. Huang, and L. Mathieu, "Machine learning in tolerancing for additive manufacturing," *CIRP Annals*, vol. 67, no. 1, pp. 157–160, 2018.

[88] R. d. S. B. Ferreira, A. Sabbaghi, and Q. Huang, "Automated geometric shape deviation modeling for additive manufacturing systems via bayesian neural networks," *IEEE Transactions on Automation Science and Engineering*, 2019.

[89] S. H. Mian, A. Al-Ahmari, and H. Alkhalefah, "Analysis and realization of sampling strategy in coordinate metrology," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[90] X. Ning, F. Li, G. Tian, and Y. Wang, "An efficient outlier removal method for scattered point cloud data," *PloS one*, vol. 13, no. 8, p. e0201280, 2018.

[91] K. Wolff, C. Kim, H. Zimmer, C. Schroers, M. Botsch, O. Sorkine-Hornung, and A. Sorkine-Hornung, "Point cloud noise and outlier removal for image-based 3d reconstruction," in *2016 Fourth International Conference on 3D Vision (3DV)*, pp. 118–127, IEEE, 2016.

[92] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586–606, International Society for Optics and Photonics, 1992.

[93] D. Wu, Y. Wei, and J. Terpenny, "Predictive modelling of surface roughness in fused deposition modelling using data fusion," *International Journal of Production Research*, vol. 57, no. 12, pp. 3992–4006, 2019.

[94] K. He, Z. Yang, Y. Bai, J. Long, and C. Li, "Intelligent fault diagnosis of delta 3d printers using attitude sensors based on support vector machines," *Sensors*, vol. 18, no. 4, p. 1298, 2018.

[95] J. Mazumder, "Design for metallic additive manufacturing machine with capability for ?certify as you build?," *Procedia CIRP*, vol. 36, pp. 187–192, 2015.

[96] C. Gobert, E. W. Reutzel, J. Petrich, A. R. Nassar, and S. Phoha, "Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging.," *Additive Manufacturing*, vol. 21, pp. 517–528, 2018.

[97] M. S. Tootooni, A. Dsouza, R. Donovan, P. K. Rao, Z. J. Kong, and P. Borgesen, "Classifying the dimensional variation in additive manufactured parts from laser-scanned three-dimensional point cloud data using machine learning approaches," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 9, p. 091005, 2017.

[98] D. Wu, Y. Wei, and J. Terpenny, "Surface roughness prediction in additive manufacturing using machine learning," in *ASME 2018 13th International Manufacturing Science and Engineering Conference*, American Society of Mechanical Engineers Digital Collection, 2018.

[99] D. Ye, G. S. Hong, Y. Zhang, K. Zhu, and J. Y. H. Fuh, "Defect detection in selective laser melting technology by acoustic signals with deep belief networks," *The International Journal of Advanced Manufacturing Technology*, vol. 96, no. 5-8, pp. 2791–2801, 2018.

[100] G. Tourloukis, S. Stoyanov, T. Tilford, and C. Bailey, "Data driven approach to quality assessment of 3d printed electronic products," in *2015 38th International Spring Seminar on Electronics Technology (ISSE)*, pp. 300–305, IEEE, 2015.

[101] R. Jafari-Marandi, M. Khanzadeh, W. Tian, B. Smith, and L. Bian, "From in-situ monitoring toward high-throughput process control: cost-driven decision-making framework for laser-based additive manufacturing," *Journal of Manufacturing Systems*, vol. 51, pp. 29–41, 2019.

[102] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[103] Z. Ren, J. Yuan, and W. Liu, "Minimum near-convex shape decomposition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 10, pp. 2546–2552, 2013.