

University of Arkansas, Fayetteville

ScholarWorks@UARK

Theses and Dissertations

5-2020

Improvement of Geometric Quality Inspection and Process Efficiency in Additive Manufacturing

Yu Jin

University of Arkansas, Fayetteville

Follow this and additional works at: <https://scholarworks.uark.edu/etd>



Part of the [Industrial Technology Commons](#), [Operational Research Commons](#), and the [Systems Engineering Commons](#)

Citation

Jin, Y. (2020). Improvement of Geometric Quality Inspection and Process Efficiency in Additive Manufacturing. *Theses and Dissertations* Retrieved from <https://scholarworks.uark.edu/etd/3586>

This Dissertation is brought to you for free and open access by ScholarWorks@UARK. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of ScholarWorks@UARK. For more information, please contact ccmiddle@uark.edu.

Improvement of Geometric Quality Inspection and Process Efficiency in Additive Manufacturing

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Engineering with a concentration in Industrial Engineering

by

Yu (Chelsea) Jin
Jinan University
Bachelor of Science in Network Engineering, 2014
University of Michigan
Master of Science in Manufacturing Engineering, 2015

May 2020
University of Arkansas

This dissertation is approved for recommendation to the Graduate Council.

Harry Pierson, Ph.D.
Dissertation Co-Director

Haitao Liao, Ph.D.
Dissertation Co-Director

Sarah Nurre Pinkley, Ph.D.
Committee Member

Zhenghui Sha, Ph.D.
Committee Member

Abstract

Additive manufacturing (AM) has been known for its ability of producing complex geometries in flexible production environments. In recent decades, it has attracted increasing attention and interest of different industrial sectors. However, there are still some technical challenges hindering the wide application of AM. One major barrier is the limited dimensional accuracy of AM produced parts, especially for industrial sectors such as aerospace and biomedical engineering, where high geometric accuracy is required. Nevertheless, traditional quality inspection techniques might not perform well due to the complexity and flexibility of AM fabricated parts. Another issue, which is brought up from the growing demand for large-scale 3D printing in these industry sectors, is the limited fabrication speed of AM processes. However, how to improve the fabrication efficiency without sacrificing the geometric quality is still a challenging problem that has not been well addressed. In this work, new geometric inspection methods are proposed for both offline and online inspection paradigms, and a layer-by-layer toolpath optimization model is proposed to further improve the fabrication efficiency of AM processes without degrading the resolution. First, a novel Location-Orientation-Shape (LOS) distribution derived from 3D scanning output is proposed to improve the offline inspection in detecting and distinguishing positional and dimensional non-conformities of features. Second, the online geometric inspection is improved by a multi-resolution alignment and inspection framework based on wavelet decomposition and design of experiments (DOE). The new framework is able to improve the alignment accuracy and to distinguish different sources of error based on the shape deviation of each layer. In addition, a quickest change point detection method is used to identify the layer where the earliest change of systematic deviation distribution occurs during the printing process. Third, to further improve the printing efficiency without sacrificing the quality of each layer, a toolpath allocation and scheduling optimization model is proposed based on a concurrent AM process that allows multiple extruders to work collaboratively on the same layer. For each perspective of improvements, numerical studies are provided to emphasize the theoretical and practical meanings of proposed methodologies.

Contents

1	Introduction	1
2	Scale and Pose Invariant Feature-based Quality Inspection	4
2.1	Introduction	4
2.2	Literature Review	8
2.3	Theory	11
2.4	Methods	13
2.4.1	Inspection based on LOS Distribution	13
2.4.2	Point Cloud Downsampling	18
2.4.3	Experimental Description	19
2.5	Results	20
2.5.1	Experiment 1	20
2.5.2	Experiment 2	23
2.5.3	Experiment 3	29
2.5.4	Experiment 4	32
2.6	Discussion	33
2.7	Conclusion	36
	Reference	37
3	Multi-resolution In-plane Alignment and Error Quantification	42
3.1	Introduction	42
3.2	Literature Review	46
3.2.1	Point Cloud Alignment	46
3.2.2	Deviation Modeling and Quantification in AM processes	47
3.3	Method	49
3.3.1	Shape Profile	50
3.3.2	Wavelet Decomposition	51

3.3.3	Multi-Resolution Alignment based on Coefficient Profile	53
3.3.4	Deviation quantification	54
3.3.5	Optimal setting for point cloud alignment	57
3.3.6	Quickest Change Point Detection	58
3.4	Experiment	63
3.4.1	The 1st layer — a simple layer	65
3.4.2	The 2nd layer — a defective layer	67
3.4.3	The 3rd layer — a layer with key internal features	70
3.4.4	A case study of QCD method	72
3.5	Conclusion	74
	Reference	74
4	Optimization of a Concurrent AM Process	78
4.1	Introduction	78
4.2	Literature Review	81
4.3	Method	84
4.3.1	Assumptions	84
4.3.2	Brief Description of Three Cases	86
4.3.3	Baseline Methods	86
4.3.4	Optimization model	88
4.3.5	Heuristic Algorithms	89
4.3.6	Adaptive Algorithm Modification	93
4.3.7	Collision Check Algorithm	96
4.4	Result and Discussion	99
4.4.1	Case 1	100
4.4.2	Case 2	104
4.4.3	Case 3	106
4.4.4	Discussion	107

4.5 Conclusion	109
Reference	110
5 Summary	114

List of Figures

1.1	An overview of three research problems	3
2.1	(a) Topology-optimized mounting bracket for an aerospace application (Image courtesy of EOS GmbH); (b) A freeform design lacking parametric surfaces for establishing datums and fixturing (Image courtesy of EOS North America)	6
2.2	Example of misalignment using the ICP method.	7
2.3	LOS inspection process overview.	14
2.4	Experimental Geometries	19
2.5	The point cloud associated with test object 1 was modified for scale, orientation, and point cloud density to test the sensitivity of the LOS distribution to these factors.	21
2.6	LOS distributions comparison for test object 1 are invariant to pose and scale, but are sensitive to point cloud density.	22
2.7	Objects printed for experiment 2(a-c); features of interest for experiment 2.	23
2.8	Shape inspection results for test objects 2a and 2b detect the shape defect introduced in test object 2b.	24
2.9	Datum feature and four target features.	25
2.10	LOS inspection result for test objects 2a and 2c detect the pose defect introduced in test object 2c.	26
2.11	The LOS distribution comparison for test object 2a indicates no features out of place with respect to each other.	27
2.12	The LOS distribution comparison for test object 2c indicates pose problems with feature 3 relative to features 1, 2, and 4.	27
2.13	An example when feature indexing of nominal model and sample parts is inconsistent.	28
2.14	Feature re-indexing was successful for test object 2a.	28
2.15	Feature re-indexing was successful for test object 2c despite the pose error associated with feature #3.	29

2.16	Test objects for Experiment 3; the shape distribution of each feature is compared in (c).	30
2.17	Feature re-indexing was successful for test object 3.	31
2.18	Three test objects were printed for Experiment 4: one unaltered and two with pose errors introduced.	32
2.19	LOS inspection results for Experiment 4 indicate good shape fidelity for all test objects, but clearly differentiate the pose errors that exist in test objects 4b and 4c.	33
3.1	An overview of the automatic multi-resolution alignment and inspection framework.	49
3.2	An example of shape representation in CCS, PCS, and shape profile.	51
3.3	Wavelet decomposition of shape profile.	52
3.4	An example of finding best matches for the 2nd level of the sample coefficient profile in the extended reference coefficient profile.	54
3.5	Find the optimal phase shift for the 2nd level of the coefficient profile.	55
3.6	Multi-resolution alignment of detail coefficient profiles and shape profiles.	55
3.7	Demonstration of change point in a time series.	59
3.8	Overview of simulation data.	61
3.9	The QCD result based on single post-change model.	61
3.10	The QCD result based on multiple post-change model.	62
3.11	The plot of ADD under different variance.	63
3.12	Performance of QCD under different post-change distribution $N(\mu, \sigma^2)$	63
3.13	Experiment setup.	64
3.14	Three sample layers.	65
3.15	Shape profile of the 1st layer.	65
3.16	Optimization of alignment parameters for the 1st layer.	66
3.17	4-level multi-resolution alignment results.	66
3.18	Alignment comparison with ICP.	67

3.19	Shape profile of the 2nd layer.	67
3.20	3-level multi-resolution alignment results.	68
3.21	Deviation profiles.	68
3.22	Result of error quantification based on 100 uniformly sampled points.	69
3.23	Result of error quantification based on 57 points around the defective area.	70
3.24	Feature segmentation by spectral clustering.	71
3.25	Shape profile of the outer contour shape.	71
3.26	3-level multi-resolution alignment results based on the datum feature.	72
3.27	Sample layers from the upper half of the human heart	73
3.28	Systematic error from layer 75 to layer 175	73
3.29	The ADD under different mean of post-change model	74
4.1	The overview of three different layers	86
4.2	Region division in the NS method (boundaries represented by dashed lines)	87
4.3	Region division in the HSS method (perimeter sub-paths omitted for clarity). (a) Sub-path #11 and #9 are two raster sub-paths of case 1. Both of them span two regions. Sub-path #11 is assigned to the left region, and sub-path #9 is assigned to the middle region. (b) Sub-path #8 is a raster sub-paths of case 2, which spans two regions and is assigned to the left region. (c) No sub-paths span multiple regions.	88
4.4	LPT algorithm flow chart	92
4.5	MOA algorithm flow chart	94
4.6	Computing overlapped area for two sub-paths	95
4.7	Example of finding the minimum delay	95
4.8	False positive example for collision detection. The shadowed area is the true area swept by the extruders.	96
4.9	Example of identifying inspection time period	97
4.10	Example of identifying inspection time period	99
4.11	Finding the bounding rectangle to form the extended sweeping area	99

4.12	Sub-paths of the layer (where axis values indicate the position of layer on the workspace)	101
4.13	Graphic illustration of the scheduling result from the LPT heuristic (a) The toolpath path is shown in time (vertical axis) indicating no intersection between extruder trajectories. (b) The relative size of the extruder's bounding rectangle to the layer is provided.	102
4.14	Result of LPT, MOA, NS and HSS for 3-extruder scheduling. Negative signs indicates that the path was reversed to avoid collision. Black indicates delay introduced to avoid collision.	103
4.15	The layer overview of Case 2.	105
4.16	Performance Comparison among LPT, MOA, NS, HSS for Case 2	105
4.17	The layer overview of Case 3	106
4.18	Performance Comparison among LPT, MOA, NS and HSS for Case 3	107

List of Tables

2.1	The performance of three rigid alignment methods.	7
2.2	LOS inspection without datum for a part with n features	16
2.3	Threshold of shape inspection for test object 2a and 2b.	24
2.4	Threshold of datum-based LOS inspection for test objects 2a and 2c.	25
2.5	Threshold of non-datum-based LOS inspection for test objects 2a and 2c.	26
2.6	Threshold of LOS inspection for test object 3.	31
2.7	LOS inspection result for the test object 3 indicating the pose problem with feature #4 relative to all other features.	31
2.8	Threshold of LOS inspection for test object 4a, 4b and 4c	32
3.1	Point-wise Deviation Based on the Multi-resolution Alignment	56
3.2	Result of ANOVA	69
3.3	Error Quantification of Each Feature	71
4.1	Symbols Used in the Optimization Model	90
4.2	Performance of LPT and MOA for Case 1 vs Number of Extruders)	101
4.3	Performance Comparison among LPT, MOA, NS and HSS for the Simple Layer . .	102

List of Published Papers

- **Jin, Y.**, Pierson, H., and Liao, H., “Toolpath Allocation and Process Scheduling for Concurrent Fused Filament Fabrication with Multiple Extruders”, *IISE Transactions*, 51(2): 192-208, 2019. [**Published**][**Chapter 4**]
- **Jin, Y.**, Pierson, H., and Liao, H., “Scale and Pose Invariant Feature Quality Inspection for Freeform Geometries in Additive Manufacturing”, *ASME Journal of Manufacturing Science and Engineering*, 141(12):121008, 2019. [**Published**][**Chapter 2**]
- **Jin, Y.**, Liao, H. and Pierson, H., “A Multi-Resolution Framework for Automated In-plane Alignment and Error Quantification in Additive Manufacturing”, *Rapid Prototyping Journal*, 2020. [**Under review**][**Chapter 3**]
- **Jin, Y.**, Liao, H., and Pierson, H., “Quickest Change Point Detection in Shape Inspection of Additively Manufactured Parts under a Multi-Resolution Framework”, *Proceedings of the ASME 2020 15th International Manufacturing Science and Engineering Conference*, Cincinnati, OH, 2020. [**Accepted**][**Chapter 3**]
- **Jin, Y.**, Pierson, H., and Liao, H., “Automatic Alignment Based on Datum Features”, *Procedia Manufacturing*, 2019. [**Published**][**Chapter 3**]

1 Introduction

Additive manufacturing (AM) is a form of creating a three-dimensional object layer by layer directly from a 3D CAD model. It now has been involved in many different industrial sectors to fabricate objects that have arbitrarily complex geometric features and can hardly be accomplished by traditional manufacturing processes. AM also shows its advantages in reducing material waste, improving part consolidation, and producing parts directly without the need for expensive part-specific tooling. However, many technical challenges have to be solved before AM reaches its full potential. One noticeable issue is the limited geometric accuracy of AM fabricated parts, which is still not sufficient to meet the requirements of some industries, such as biomedical or aerospace engineering. However, the traditional geometric inspection techniques might not perform well and even cannot be adopted in AM due to the geometric complexity of parts and additive layer-by-layer nature of AM processes. Moreover, the AM processes are no longer limited in producing small but complex geometries due to the growing demand for the low-volume fabrication of large products, especially in aerospace, architecture, and automotive industries. Nevertheless, upscaling the AM process is hindered by limited fabrication speed. One straightforward solution is to increase layer thickness, but this action sacrifices resolution and causes poor surface finish and dimensional accuracy. Aiming to address these limitations of AM processes, three research problems are illustrated in Figure 1.1 which are discussed and resolved in this work.

First, for offline inspection, the coordinate measuring machine (CMM) using the geometric dimensioning and tolerancing (GD&T) paradigm, which is applied in many traditional production environments, does have drawbacks in the context of AM. CMM inspection relies on datums established in the part design; however, this can be far from straightforward for AM designs, which in most cases lack parametric datum features. In addition, current AM techniques have shown the capability of producing complex or large-scaled AM parts with multiple fabrication cycles, or producing working assemblies in a single build, or printing on a pre-made surface. Each subpart, or so-called feature, is considered as an individual object of interest. In practice, the functionality of an AM produced part is dependent on the quality of features as well as their relative positioning.

Therefore, even if convenient datum surfaces exist in AM parts, designers are often interested in the relative position and orientation between two freeform features, rather than their individual poses with respect to a datum frame. Another drawback for CMM and GD&T paradigm is lacking accessibility for users outside of factory or laboratory environments. Fortunately, 3D scanning, as an alternative of CMM, can provide fast, high-density measurements and has now been widely applied in AM processes for geometric inspection. Based on the point cloud data obtained from 3D scanning, this research proposes a new feature-based scale- and pose- invariant quality inspection method using a novel Location-Orientation-Shape (LOS) distribution. The key technique of the new method is to describe the shape and pose of features via kernel density estimation and detect non-conformities based on statistical divergence. In section 2, technical details of this novel offline fast inspection method are provided and validated with numerical examples using both simulated and physical AM builds.

Second, due to the additive nature of AM, the online inspection should be blended with the offline geometric inspection to prevent errors of precedent layers getting accumulated on subsequent layers. The geometric accuracy of layers can be described by in-plane shape deviation. The shape of each inspected layer can be described by a 2D point cloud obtained by slicing a thin layer of 3D point cloud acquired from 3D scanning. In practice, a scanned shape must be aligned with the corresponding base-truth CAD model before evaluating its geometric accuracy. Then the shape deviation is described by point-wise deviation based on the aligned point clouds. Therefore, the calculated shape deviation is attributed to systematic, measurement, and alignment errors. However, how to distinguish errors from different sources has not been addressed in the previous studies. To fill this research gap, a multi-resolution point cloud alignment and inspection scheme based on wavelet decomposition and randomized completed block design (RCBD) is proposed in section 3. Based on the quantified alignment error, the performance of alignment is improved by a two-step optimization model. Moreover, since the systematic error is the one that triggers an alarm of system anomalies, a quickest change detection (QCD) algorithm is applied under the multi-resolution alignment and inspection framework to identify the layer where the earliest sys-

tematic deviation distribution changes during the printing process. Numerical experiments and a case study on a human heart is conducted to illustrate the performance of the proposed method in detecting the layer-wise geometric error.

Third, with demand for the low-volume fabrication of large products in industries such as aerospace, architecture, and automotive, there is a strong motivation to scale up the process. However, upscaling AM processes is restricted by the limited fabrication speed, especially for AM processes like Fuse Filament Fabrication (FFF). In practice, increasing layer thickness can speed up the process, but this sacrifices resolution and increases stair-stepping errors. This limitation forces manufacturers and customers to choose between long cycle times and extremely poor surface finishes and dimensional accuracy. Therefore, the fabrication efficiency of AM is desired to be improved without sacrificing the geometric quality. Fortunately, the idea of decomposing a 3D geometry into features provides a possibility of making multiple machines work collaboratively in parallel to accomplish a complex part. In this work, a general toolpath allocation and scheduling methodology is proposed in section 4 to accomplish the optimization of a concurrent AM process by distributing the fabrication tasks of each layer among multiple extruders. Two heuristic algorithms are proposed to solve the optimization model. The performance of the algorithms with respect to fabrication time and computational cost are demonstrated and compared using three case studies.

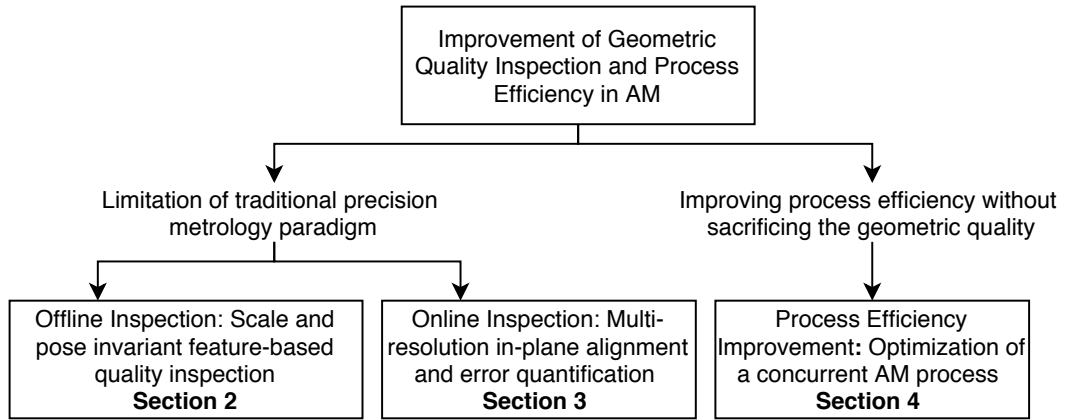


Figure 1.1: An overview of three research problems

2 Scale and Pose Invariant Feature-based Quality Inspection

Additive manufacturing (AM) has the unprecedented ability to create customized, complex, and nonparametric geometry, and it has made this ability accessible to individuals outside of traditional production environments. Geometric inspection technology, however, has yet to adapt to take full advantage of AM’s abilities. Coordinate measuring machines are accurate, but they are also slow, expensive to operate, and inaccessible to many AM users. On the other hand, 3D-scanners provide fast, high-density measurements, but there is a lack of feature-based analysis techniques for point cloud data. There exists a need for developing fast, feature-based geometric inspection techniques that can be implemented by users without specialized training in inspection according to geometric dimensioning and tolerancing conventions. This research proposes a new scale- and pose- invariant quality inspection method based on a novel Location-Orientation-Shape (LOS) distribution derived from point cloud data. The key technique of the new method is to describe the shape and pose of key features via kernel density estimation and detect non-conformities based on statistical divergence. Numerical examples are provided, and tests on physical AM builds are conducted to validate the method. The results show that the proposed inspection scheme is able to identify form, position, and orientation defects. The results also demonstrate how datum features can be incorporated into point cloud inspection; that datum features can be complex, nonparametric surfaces; and how the specification of datums can be more intuitive and meaningful, particularly for users without special training.

2.1 Introduction

Additive manufacturing (AM) has shown notable potential for agile production of complex freeform geometric features. Application examples include topology-optimized structural components (Brackett et al., 2011), prosthetics (Jin et al., 2015), and biological tissues and organs (Murphy and Atala, 2014). AM is highly agile, offering good potential for economical low-volume production and the ability to easily scale builds on demand. AM has also democratized design and production by allowing people without specialized engineering skills or access to capital equipment to design

and manufacture objects. Despite these advantages, geometric quality inspection techniques that support these strengths are required before AM can reach its full potential.

The gold standard for precision metrology in many production environments is a coordinate measuring machine (CMM) using the geometric dimensioning and tolerancing (GD&T) (Meadows, 2009) paradigm, but this does have drawbacks in the context of AM. CMM inspection tends to be highly manual, especially in high-mix, low-volume production where AM can be the most impactful. Aside from making inspection slow and expensive, this tends to result in relatively low sampling density focused on a few key features. But in a topology-optimized part such as that shown in Fig.2.1a, stresses are highly sensitive to geometric deviations throughout the part, necessitating a more complete inspection. CMM inspection also relies on datums established in the part design; however, this can be far from straightforward for designs which lack parametric features such as the one depicted in Fig.2.1b. Even if convenient datum surfaces exist, designers are often interested in the relative position and orientation between two freeform features, rather than their individual poses with respect to a datum frame, (*e.g.*, the relative positions of teeth in a dental prosthesis (Juneja et al., 2018)). Moreover, CMM inspection for complex geometries often requires specialized tooling to fixture parts, undercutting the advantages of AM’s toolless nature, and the process is not scale invariant, making it more difficult to take advantage of AM’s ability to easily scale builds on demand. Finally, CMMs are relatively inaccessible outside of factory or laboratory environments, and establishing and inspecting against datums requires specialized knowledge, which are limiting factors for makers and entrepreneurs.

In contrast, 3D scanning is capable of high-density sampling of an entire part in a relatively short time without fixturing. It can also be cheaper and therefore more accessible outside of formal production environments, with many consumer-grade models under \$1000 (Anonymity, 2017a). In practice, however, the scanned point cloud must be aligned with the corresponding base-truth CAD model prior to evaluating the geometric accuracy. Common alignment methods coming with a point cloud processing software package rely on least-square fitting algorithms, such as iterative closest point (ICP) algorithm (Besl and McKay, 1992), to find the optimal rigid transformation

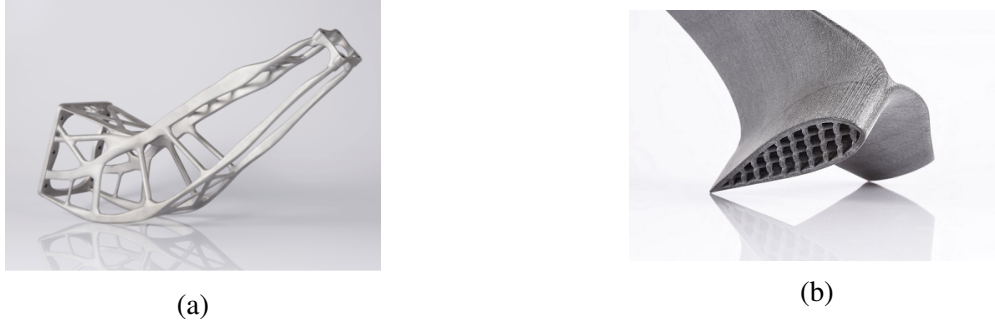


Figure 2.1: (a) Topology-optimized mounting bracket for an aerospace application (Image courtesy of EOS GmbH); (b) A freeform design lacking parametric surfaces for establishing datums and fixturing (Image courtesy of EOS North America)

that minimizes the sum of squared distances between the measured alignment points and their target reference points. However, it is possible that defects of the sample part might deteriorate the alignment accuracy, or cause the misalignment even around the non-defective area. This is illustrated in Fig.2.2, where a deviation in the ears of the Stanford bunny model (Turk and Levoy, 2005) results in misalignment that indicates geometric deviations in the head and foot features. To better justify this point, two more rigid registration methods based on Normal Distributions Transform (NDT) (Magnusson, 2009) and Coherent Point Drift (CPD) (Myronenko and Song, 2010) are implemented. The Root Mean Square Error (RMSE) of three different rigid alignment methods are provided in Table 2.1. It is observed that the influence of defects on the alignment result is significant. Indeed, there are many more advanced alignment methods in the literature, which are proposed for object recognition or 3D perception. For instance, local features-based alignment methods (Eckart et al., 2018) can resolve some shortcomings of the global ICP. However, the geometric accuracy of the selected features on a defective object are not able to be guaranteed. If the points associated with the selected feature have large deviations, then the alignment based on the feature apparently will still cause significant RMSE. Compared to rigid registration methods, non-rigid registration (Mateus et al., 2008) can perform a more complicated transformation that might change the shape of the point cloud to achieve better alignment. From the perspective of geometric inspection, the point cloud should not be changed or deformed in the registration

process. Therefore, in most literature or software associated with the shape inspection based on the point cloud data, only rigid registration methods are applied. (Hong-Seok and Mani, 2014; Li and Guskov, 2005; Shi et al., 2006; Minetola, 2012).

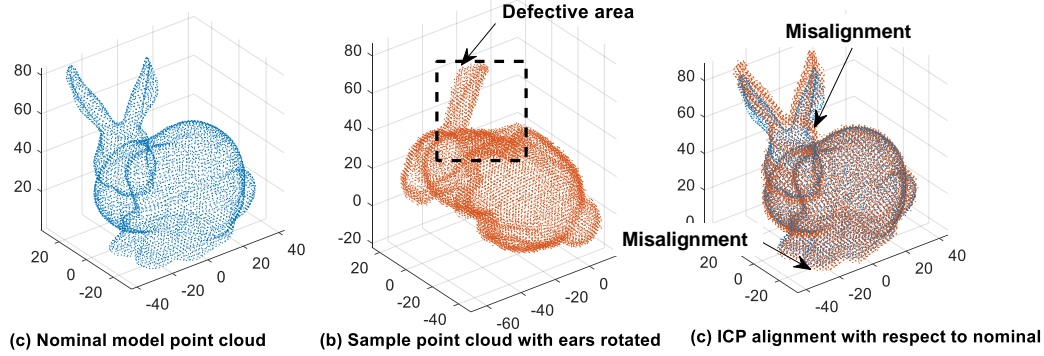


Figure 2.2: Example of misalignment using the ICP method.

Table 2.1: The performance of three rigid alignment methods.

	RMSE of a conforming sample	RMSE of a defective sample
ICP	0.69	2.38
NDT	1.02	2.26
CPD rigid	0.81	2.53

This research introduces a novel point cloud quality inspection method that blends many of the advantages of CMM/GD&T and 3D-scanning-based approaches while mitigating some of the disadvantages. The theoretical contribution is a generalization of the shape distribution concept used in artificial intelligence for object recognition. This generalization is dubbed the location-orientation-shape (LOS) distribution and can describe the relative form and pose between features represented by point clouds. Methods for form, location, and orientation inspection are developed based on the LOS distribution and tested via physical experiments on 3D printed parts. Since the LOS distribution is a piece-wise pdf and is no longer associated with a specific Cartesian Coordinate system (CCS), or we say a specific pose and scale of the part, alignment is not required before inspection to eliminate the effect of alignment error. It is a scale- and pose-invariant approach that can incorporate intuitively defined datums based on complex and/or nonparametric features. It is also able to yield meaningful, high-level inspection results even without datum features. Although

we propose an alternative to CMM/GD&T, we do not claim to replace or supersede it, especially for traditional parametric geometries. Rather, we propose a method that has different strengths and weaknesses which can supplement or be used in place of CMM/GD&T when warranted by the application. Practical applications of the proposed method include a first-order inspection for reducing the number of samples sent to CMM; an alternative inspection method for freeform geometry that is difficult to tolerance and/or establish datums; and an alternative for non-engineers who are not well versed in the intricacies of GD&T or have access to a CMM.

2.2 Literature Review

GD&T has limitations in tolerancing freeform geometries. Two major GD&T standards are ASME Y14.5 (Meadows, 2009) and ISO 1101 (Standard, 2004). In ASME Y14.5, manufacturing variations are categorized into different classes with specific tolerance types (*e.g.*, form, orientation, location, and profile). Among all defined tolerance types, the profile tolerance is the one which has potential for capturing the dimensional accuracy of freeform surfaces. The profile of a surface describes a 3D tolerance zone around a surface by two offset surfaces. The size of the tolerance zone could be either uniform or non-uniform along the surface. The non-uniform tolerance zone can potentially be useful by adopting variable tolerances along the surface (Ameta et al., 2015b). However, it is not clear in the standard how to specify such a tolerance zone and its parameters (Savio et al., 2007). Moreover, an AM-produced freeform part cannot simply be modeled as a continuous surface by a mathematical formulation (Yamaguchi, 2012). In practice, a freeform part can be decomposed into multiple surfaces for the ease of formulation, but the position information of these surface segments essential to the functionality of the part, is unfortunately neglected. In addition, there are some other AM-relevant issues associated with GD&T (Ameta et al., 2015a) including (1) the feature and tolerance information lost during tessellation, (2) the lack of analysis in terms of building assemblies as a single component, (3) the error accumulations among layers, (4) the influence of datum reference in each layer, and (5) lack of methods for processing measurements obtained by non-CMM techniques. The method proposed in this work does not require

datum references and is able to process the point cloud measurement to quickly evaluate the quality of the part.

Motivated by the limitation of GD&T in tolerancing freeform geometries, alternatives have been proposed based on the result of non-contact measuring techniques (*e.g.*, 3D scanner). Prior work in the area of point-cloud-based inspection may be classified into two categories: (1) modeling the as-printed surface via non-parametric approaches and comparing statistics to corresponding in-control values, and (2) comparing the as-printed geometry to a base-truth CAD model and calculating point-wise, layer-wise, or surface-wise deviation directly. The Gaussian Process (GP) and its extended methods are the main statistical models adopted in the first category of literature (Del Castillo et al., 2015; Colosimo et al., 2014, 2015; Jahani et al., 2018). The main idea is to build a parametric model for a surface using a Gaussian process and then establish a control chart on the GP parameters. This method has proven effective for inspecting simple geometries such as cylinders' and tunnels' inner surface (Zhang et al., 2017). However, a freeform AM part has more complicated geometric features and surfaces. Additionally, such methods are not computationally efficient for large datasets like point clouds. The methods in the second category focus more on deviation modeling. In Choi et al. (2002), Silva et al. (2008), and El-Katatny et al. (2010) geometric error is quantified by measuring the position of specific landmarks on the part. For example, in (El-Katatny et al., 2010), the linear distances of 11 landmarks on a printed skull are measured and then compared to the values obtained from the nominal model. However, error estimation based on relatively few landmarks is unlikely to be as accurate as those quantified by calculating point-wise distances between two point clouds (Ciocca et al., 2009; Germani et al., 2010; Meakin et al., 2004), calculating the deviation from the printed surface to a nominal one using a polar coordinate system (Zhu et al., 2018), or using some commercial software to generate a deviation profile or color-map (Rao et al., 2016). One limitation of these works is that the fabricated geometry must be aligned to the nominal model, which implies that both must be of the same scale. To improve inspection efficiency and inspect scaled parts within a single inspection framework, scale and orientation invariant methods for error estimation are desirable. In addition, these works focus on quantifying

the dimensional accuracy of individual features and do not consider quality inspection in terms of the relative positioning of complex features. The inspection methods based on shape distribution proposed in this work are intended to address these issues.

The methods proposed in this research are based on the concept of a shape distribution, which is a shape descriptor originally proposed by Osada et al. (2002) for object recognition. A shape descriptor is a simplified representation of a 2D or 3D shape in the form of a mathematical formulation. Shape distribution is a probability distribution sampled from a shape function measuring geometric properties of the 3D model such as distance, angle, area, or volume between random surface points. It evaluates the similarity of the objects by comparing probability distributions. The method is implemented in three steps: (1) select a shape function to create a shape vector, (2) construct the shape distribution for the object based on the shape vector, and (3) compare the shape distribution to those of known objects. In the first step of this method, common shape functions include the D2, D3, A3 and D4, where D2 indicates the Euclidean distance between any two points in the point cloud, D3 measures the square root of the area of the triangle between three random points on the surface, A3 measures the angle between three random points on the surface of a 3D model, and D4 measures the cube root of the volume of the tetrahedron between four random points on the surface. Then the shape distribution is constructed by a piecewise linear function based on the histogram of the shape vectors. The distance between two distributions can be compared via many standard approaches. Examples include the Minkowski L_N norms, Kolmogorov-Smirnov distance, Kullback-Leibler divergence, match distances, earth mover's distance, and Bhattacharyya distance. Some comprehensive surveys (Kazmi et al., 2013; Tangelder and Velthkamp, 2004) have been conducted for different types of 3D shape descriptors and their computational complexity. Compared to other shape descriptors, shape distribution is a more robust, efficient, and transformation-invariant method to quantify the dissimilarity between 3D shapes. However, the original shape distribution describes only the property of a single geometry/feature, and it does not contain location or orientation information. The proposed methods inherit the advantages of shape distribution while adding the capability of encoding the relative

pose between features.

Defining features and segregating the point cloud is prerequisite to any feature-based inspection method. First, features should be defined based on the function of the part. For instance, each of the separable working assemblies, which are produced in a single build (Ameta et al., 2015a) or by using multi-stage AM processes (Monika Mahto, 2017), can be identified as a feature. In addition, 6-axis printing (Song et al., 2015) started to gain interest for its potential ability in improving quality and mechanical properties of the printed part. The possibility and performance of printing on a pre-made surface have been investigated (Pierson and Chivukula, 2018). In this case, the geometry printed in each fabrication cycle is recognized as a feature. Once features are defined, the point cloud must be segmented accordingly, either manually or by machine learning techniques such as Clustering (Zhang et al., 2008; Lu et al., 2016) and Deep Learning (Charles et al., 2017; Ben-Shabat et al., 2017). Different machine learning techniques for point cloud segmentation are reviewed by Grilli et al. (2017). In this work, to focus on the key idea of quality inspection, features are pre-determined by users. Then k-means clustering is applied to segment the point cloud into desired pieces. The clustering result is verified by the user to remove any inconsistencies in the cluster labeling. In the future, more intelligent methods can be embedded into the system without influencing the remainder of the underlying inspection method.

2.3 Theory

The novel location-orientation-shape (LOS) distribution extends the concept of a shape distribution to include the relative pose between features. Whereas a shape distribution is obtained based on points *within a single point cloud*, the LOS distribution is based on analogous metrics between two point clouds. Additionally, the LOS distribution considers all points within the subject point clouds, rather than random points as in Osada et al. (2002). Given point cloud \mathcal{P}_i , its LOS vector associated with a reference point cloud \mathcal{P}_j is denoted as $LOS\text{-}fun\{\mathcal{P}_i, \mathcal{P}_j\}$, where fun is a geometric function selected by the user. Generalizing the D2, D3, A3 and D4 shape vectors, the

$LOS\text{-}fun\{\mathcal{P}_i, \mathcal{P}_j\}$ can be calculated by

$$LOS\text{-}D2\{\mathcal{P}_i, \mathcal{P}_j\} = \{d(\mathbf{p}, \mathbf{q}) \mid d(\mathbf{p}, \mathbf{q}) = (\mathbf{p} - \mathbf{q})(\mathbf{p} - \mathbf{q})^T, \forall \mathbf{p} \in \mathcal{P}_i, \mathbf{q} \in \mathcal{P}_j\} \quad (2.1)$$

$$LOS\text{-}D3\{\mathcal{P}_i, \mathcal{P}_j\} = \{d(\mathbf{p}, \mathbf{q}, \mathbf{t}) \mid d(\mathbf{p}, \mathbf{q}, \mathbf{t}) = \sqrt{\frac{1}{2} |(\mathbf{p} - \mathbf{q}) \times (\mathbf{p} - \mathbf{t})|}, \forall \mathbf{p} \in \mathcal{P}_i, \mathbf{q}, \mathbf{t} \in \mathcal{P}_j\} \quad (2.2)$$

$$LOS\text{-}A3\{\mathcal{P}_i, \mathcal{P}_j\} = \{a(\mathbf{p}, \mathbf{q}, \mathbf{t}) \mid a(\mathbf{p}, \mathbf{q}, \mathbf{t}) = \arccos \frac{(\mathbf{q} - \mathbf{p}) \cdot (\mathbf{t} - \mathbf{p})}{\|\mathbf{q} - \mathbf{p}\| \|\mathbf{t} - \mathbf{p}\|}, \forall \mathbf{p} \in \mathcal{P}_i, \mathbf{q}, \mathbf{t} \in \mathcal{P}_j\} \quad (2.3)$$

$$LOS\text{-}D4\{\mathcal{P}_i, \mathcal{P}_j\} = \{d(\mathbf{p}, \mathbf{r}, \mathbf{q}, \mathbf{t}) \mid d(\mathbf{p}, \mathbf{q}, \mathbf{t}) = \sqrt[3]{\frac{|(\mathbf{r} - \mathbf{p}) \cdot ((\mathbf{q} - \mathbf{p}) \times (\mathbf{t} - \mathbf{p}))|}{6}}, \forall \mathbf{p}, \mathbf{r} \in \mathcal{P}_i, \mathbf{q}, \mathbf{t} \in \mathcal{P}_j\} \quad (2.4)$$

where $LOS\text{-}D2$ is a distance vector measuring distances between any two points, $LOS\text{-}D3$ measures the square root area of triangles formed by three points, $LOS\text{-}A3$ measures angles between two vectors formed by three points, and $LOS\text{-}D4$ measures the cube root of the volumes of a tetrahedron formed by four points. Given these definitions, the shape vector created in Osada et al. (2002) is a special case of the LOS vector when the target point cloud \mathcal{P}_i itself is considered as the reference point cloud. For instance, the D2 shape vector for point cloud i is

$$LOS\text{-}D2\{\mathcal{P}_i\} = \{d(\mathbf{p}, \mathbf{q}) \mid d(\mathbf{p}, \mathbf{q}) = (\mathbf{p} - \mathbf{q})(\mathbf{p} - \mathbf{q})^T, \forall \mathbf{p}, \mathbf{q} \in \mathcal{P}_i\} \quad (2.5)$$

Equations for the other shape functions follow similarly.

The LOS distribution is constructed from an LOS vector in three steps: 1) The LOS vector is standardized such that its components are in the interval $[0,1]$ to eliminate the effect of scale. 2) A histogram is constructed for the values in the LOS vector. 3) A piece-wise kernel estimate is obtained from the histogram, which can be determined directly using a kernel fitting algorithm for the selected kernel function. In this work, a standard Gaussian kernel function is chosen and applied in Matlab to perform the kernel density estimation. Given the LOS vector $LOS\text{-}fun\{\mathcal{P}_i, \mathcal{P}_j\}$, the

corresponding LOS distribution is denoted as $g_{ij} = g\{\mathcal{P}_i, \mathcal{P}_j\}$.

The distance between an LOS distribution generated from 3D scan information and the one generated from the base-truth model can be used as a quality metric. A smaller distance value indicates more similarity between the two distributions. The distance between any two LOS distributions can be measured by a number of methods: the χ^2 statistic, Bhattacharyya distance, Minkowski L_N norms, Kolmogorov-Smirnov (K-S) distance, and so forth (Puzicha et al., 1999). In particular, the χ^2 statistic and Bhattacharyya distance are commonly used for comparing binned data (histograms). On the other hand, the Minkowski L_N norms measures the average difference between two *pdfs* while K-S distance is preferred in measuring the maximum difference between two cumulative distribution functions (*cdfs*). In practice, users have the flexibility to choose any similarity measure. K-S distance is selected for this work without loss of generality. Given two LOS distributions $g_{ij}^{(1)}$ and $g_{ij}^{(2)}$ constructed from the point clouds of two sample parts, the K-S distance between the two distributions is calculated by

$$d_{ij} = \max_{x_k} |G_{ij}^{(1)}(x_k) - G_{ij}^{(2)}(x_k)|, \forall x_k \in [0, 1] \quad (2.6)$$

where $G_{ij}^{(1)}$ and $G_{ij}^{(2)}$ are *cdfs* corresponding to $g_{ij}^{(1)}$ and $g_{ij}^{(2)}$.

2.4 Methods

2.4.1 Inspection based on LOS Distribution

A summary of the LOS quality inspection method is as follows (Fig.2.3): 1) A point cloud for the fabricated part is obtained via 3D scanning and segmented into features. Features may be parametric or non-parametric and should include datum features when defined. Each feature point cloud is downsampled using a Voxel Grid Filter (Anonymity, 2017b) to reduce the size of the data set and computation time. The result is a sampled feature point cloud $\mathcal{P}_i^{(s)}$, for each feature i . 2) The base-truth CAD model is sampled to obtain a nominal point cloud, which is similarly segmented and downsampled. The result is a set of nominal feature point clouds $\mathcal{P}_i^{(n)}$. 3) LOS

distributions $g^{(s)}$ and $g^{(n)}$ are computed based on these point clouds. The specific point cloud combinations used to form $g^{(s)}$ and $g^{(n)}$ depend on the inspection objective, as detailed below. 4) The geometric quality metric for a given feature is quantified by the distance, d , between $g^{(s)}$ and $g^{(n)}$.

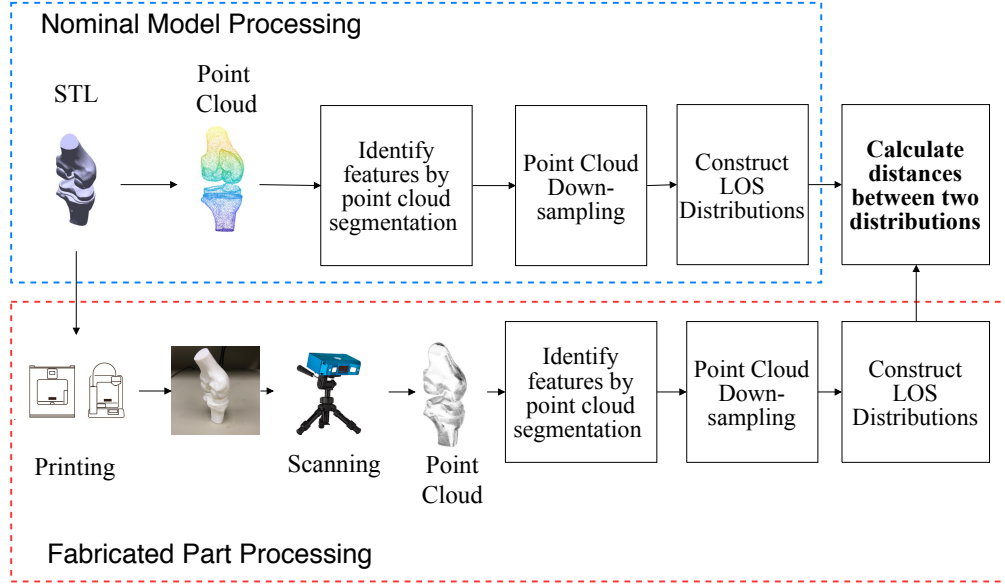


Figure 2.3: LOS inspection process overview.

LOS Shape Inspection. When an LOS vector is constructed for a point cloud with respect to itself, as with Eq.5, the resulting distribution contains information about that point cloud's shape. Let n denote the number of features to be inspected and i be a feature index where $i \in \{1, 2, \dots, n\}$. The distance, d_{ii} , between LOS distributions $g_{ii}^{(s)} = g\{\mathcal{P}_i^{(s)}, \mathcal{P}_i^{(s)}\}$ and $g_{ii}^{(n)} = g\{\mathcal{P}_i^{(n)}, \mathcal{P}_i^{(n)}\}$ indicates shape deviation. This is analogous to form inspection in GD&T, but is not restricted to simple parametric geometries such as planes and cylinders.

LOS Inspection with Datum Features. When an LOS vector is constructed between a given point cloud and a reference point cloud, as in Eq.1-4, the resulting distribution contains information about the given point cloud's location, orientation, and shape with respect to the reference. Let n denote the number of inspection features, m the number of datum features, i the inspection feature index, and j the datum feature index, where $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, m\}$.

The distance, d_{ij} , between LOS distributions $g_{ij}^{(s)} = g\{\mathcal{P}_i^{(s)}, \mathcal{P}_j^{(s)}\}$ and $g_{ij}^{(n)} = g\{\mathcal{P}_i^{(n)}, \mathcal{P}_j^{(n)}\}$ is a combined measure of the shape and pose of inspection feature i with respect to datum feature j . Although datum features may be simple surfaces as is conventional in GD&T, inspection via LOS distributions allows the concept of a datum feature to be expanded to include complex surfaces and/or non-parametric surfaces.

The coupling between shape and pose information in LOS inspection can be advantageous in a go/no-go inspection context because it requires only one set of computations and yields a single numeric result. In other contexts, however, it may be desirable to have separate information about shape and pose. In this situation, shape inspection may be combined with LOS inspection. If a feature's shape deviation, d_{ii} , is low, but its LOS deviation, d_{ij} , is high, one may conclude that the primary problem with the feature is its location and/or orientation.

LOS Inspection without Datum Features. Outside of industrial production environments, models intended for AM often lack defined datum features, yet there is still a need for meaningful, quantitative geometric quality assessment. In this case, LOS inspection may be applied between all features of interest. Let n denote the number of features of interest and i, j are the indices for a pair of features where $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, n\}$. The distances, d_{ij} , between LOS distributions $g_{ij}^{(s)} = g\{\mathcal{P}_i^{(s)}, \mathcal{P}_j^{(s)}\}$ and $g_{ij}^{(n)} = g\{\mathcal{P}_i^{(n)}, \mathcal{P}_j^{(n)}\}$ form an LOS inspection matrix, as shown in Table 2.2.

The diagonal contains the shape inspection results for all features. Each row corresponds to one feature, with the elements representing the LOS inspection results with respect to each of the other part features. Any large element in the matrix is cause for rejecting the part, but more importantly, the LOS inspection matrix gives the user *insight into why* the part is rejected. For instance, if d_{33} and d_{67} are two elements of the matrix that give cause to reject the part, the user may deduce that 1) feature #3 is not well shaped and 2) feature #6 is not positioned accurately with regard to feature #7. This information can prove valuable in diagnosing the cause of the failure and devising a means to correct the problem.

It is worth noting that the threshold on distance values to differentiate acceptable and unaccept-

Table 2.2: LOS inspection without datum for a part with n features

	\mathcal{P}_1	\mathcal{P}_2	...	\mathcal{P}_i	...	\mathcal{P}_n
\mathcal{P}_1	d_{11}	d_{12}	...	d_{1i}	...	d_{1n}
\mathcal{P}_2	d_{21}	d_{22}	...	d_{2i}	...	d_{2n}
...
\mathcal{P}_i	d_{i1}	d_{i2}	...	d_{ii}	...	d_{in}
...
\mathcal{P}_n	d_{n1}	d_{n2}	...	d_{ni}	...	d_{nn}

able parts can be determined by the distribution of divergence considering natural variation of AM process and user-defined tolerance. Technically, the threshold can be obtained based on simulated models which are created by adding only random deviations to the nominal model. The random deviation follows a normal distribution which has a zero mean and a standard deviation equal to the worst variability a user can tolerate and determine. After the distributions of K-S distances based on simulated models are established, the threshold of a certain K-S distance value, in this work, is defined by $\mu_d + 3\sigma_d$, where μ_d and σ_d are denoted as the mean and standard deviation of the K-S distance distribution obtained from simulated models. Specifically, if the observed deviation (i.e., K-S distance) is larger than the threshold, then the part is considered as an unacceptable sample.

Feature indexing. To this point it has been tacitly assumed that feature indexing is consistent between the sampled and nominal point clouds, $\mathcal{P}^{(s)}$ and $\mathcal{P}^{(n)}$. If feature definition is a manual process, with or without assistance from the automated feature segmentation algorithms mentioned in section 2, this is a reasonable assumption; however, inconsistent indexing could be problematic in fully automated feature definition and would result in false rejections. In such circumstances, comparison of the LOS distributions can assist in establishing the correct feature correspondence.

The procedure begins by matching and re-indexing similarly shaped features: Given feature j

in the nominal model, the corresponding feature in the scanned sample can be determined by

$$i = \arg \min_i (\max |G_{ii}^{(s)} - G_{jj}^{(n)}|) \quad (2.7)$$

where $\max |G_{ii}^{(s)} - G_{jj}^{(n)}|$ measures the K-S distance between the shape-based LOS distribution of the given nominal feature j and any sample feature i . For any feature j , if the corresponding $i \neq j$, then feature i in the scanned model is re-indexed as j . The matched results can further be interpreted in the context of LOS shape inspection as discussed above. If the shape inspection results are acceptable, and if all features are uniquely shaped, then the indexing result is unambiguous and LOS inspection can proceed; however, substantially similar features may still be mis-indexed.

LOS distributions may further be used to resolve the indexing ambiguity caused by features with similar form if one correctly indexed feature can be identified. This can be done automatically if at least one feature has a unique shape. The unique features can be differentiated based on shape distribution comparisons. The procedure proposed in Osada et al. (2002) can be implemented by considering one feature as the test shape and the rest as the reference shape database. If the shape distribution of the test shape cannot be grouped into the shape distribution of any reference shape, then the test shape is considered as a unique feature. If no unique features exist, user intervention (via an appropriately designed interface) is required to identify one correctly indexed feature.

Once a correctly indexed feature has been identified, let i be that feature's index. Given feature l of the nominal model, the corresponding feature in the scanned point cloud can be determined by

$$k = \arg \min_k (\max |G_{ik}^{(s)} - G_{il}^{(n)}|) \quad (2.8)$$

where $\max |G_{ik}^{(s)} - G_{il}^{(n)}|$ is the K-S distance between two pose-based LOS distributions, measuring the difference between the position of sample feature i relative to feature k of the scanned sample and position of nominal feature i with regard to feature l in the nominal model. If $k \neq l$, then feature k in the scanned sample is re-indexed as l . The rationale: If feature k in the scanned sample is actually feature l of the nominal model, their position relative to a correctly indexed feature i should be similar.

2.4.2 Point Cloud Downsampling

Due to the large size of the raw point cloud generated by a 3D scanner, downsampling is desirable to reduce the size of the dataset and improve computational efficiency (Steinberg, 2016). A Voxel Grid Filter (VGF) is used as a uniform sampling strategy that superimposes a voxel grid of user-defined resolution over the input point cloud and approximates all points within each voxel using their centroid.

In the original application of the shape distribution (Osada et al., 2002), it was shown that variation in point cloud density does not affect the overall curvature of the distribution, but it can generate small local differences which are undesirable for quality inspection. Even though a dense point cloud can provide more details on the distribution curve, the computation cost forces users to make a trade-off between the computational efficiency and inspection accuracy. In practice, for a point cloud with thousands of points, the LOS distribution can be constructed in seconds.

More importantly, since the proposed method is designed for scale-invariant quality inspection, the nominal model and the fabricated sample do not need to have the same scale. In this case, the voxel grid sizes chosen to downsample the nominal and fabricated point clouds should result in similar point cloud densities. Mathematically, a point cloud is represented as an $N \times 3$ matrix, where N is the number of points in the point cloud. Let $\mathcal{OP}_i^{(s)}$ and $\mathcal{OP}_i^{(n)}$ denote the original point cloud of feature i generated from the 3D scanner and the nominal model, respectively. Then the corresponding downsampled point clouds are denoted as $\mathcal{DP}_i^{(s)}$ and $\mathcal{DP}_i^{(n)}$. Assume the downsampled nominal point cloud $\mathcal{DP}_i^{(n)}$ has density $N_i^{(n)}$, then the downsampling grid size δ for the scanned point cloud $\mathcal{OP}_i^{(s)}$ should satisfy

$$\delta^* = \arg \min_{\delta} |N_i^{(n)} - N_i^{(s)}|, \quad N_i^{(s)} = \text{size}\{VGF(\mathcal{OP}_i^{(s)}, \delta)\} \quad (2.9)$$

where $N_i^{(s)}$ is the density of downsampled scanned point cloud of feature i using VGF under grid size δ . Since the two downsampled point clouds might not have equal density, the optimal grid size δ^* is thus selected for achieving the minimum difference between the densities of the two

downsampled point clouds.

2.4.3 Experimental Description

Four experiments were conducted to evaluate the proposed methods on both parametric and non-parametric geometries. Each considers one of the test objects in Fig.2.4.

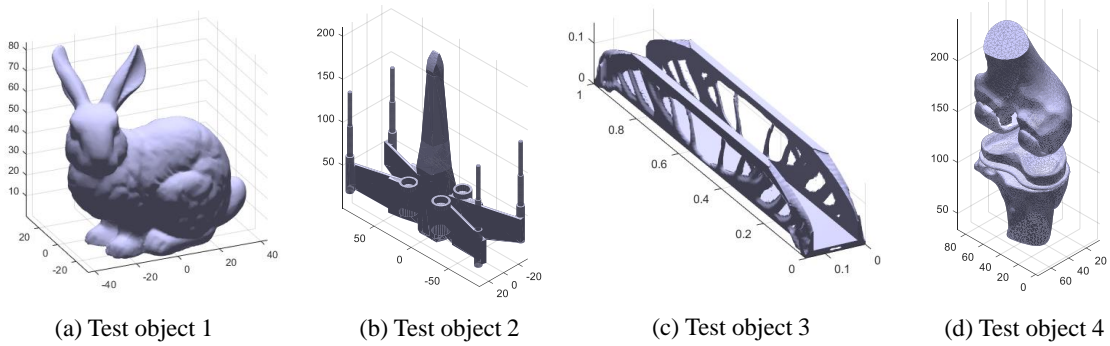


Figure 2.4: Experimental Geometries

Experiment 1 is designed to illustrate the sensitivity of the LOS distribution to pose, scale, and point cloud density. Based on the prior knowledge, the LOS distribution of a point cloud with respect to itself is expected to be scale and pose invariant and robust to point cloud density in quality inspection. A nominal point cloud is generated from the digital model for test object 1. This is then manipulated to simulate scans of rotated and scaled prints, and the resulting LOS distributions are compared to nominal distribution. LOS distributions of point clouds of dissimilar density are also compared.

Experiment 2 is designed to test the LOS inspection method on an object with parametric features. A geometrically correct copy of test object 2 is printed along with variations that serve as examples of errors. LOS shape inspection, inspection with a datum feature, and inspection without a datum feature are performed on each. Indexing disambiguation is also tested for four identical features.

Experiment 3 is designed to further implement the LOS inspection and feature re-indexing on a topology optimized object (Anonymity, 2019) as shown in Fig.2.4(c). A simulated defective sample is tested by following the procedure of LOS inspection without datum features as in *Experiment*

2.

Experiment 4 is designed to test the LOS inspection method on an object with non-parametric features. Variations of test object 3 are printed to represent good and bad parts, and LOS inspection is performed without a datum feature. Moreover, the proposed methods are applied to biomedical AM application in *Experiment 4* to illustrate its practical value and potential future application. In recent years, the demand of artificial organs or arthrosis continues to increase due to the lack of suitable donors. Different from objects for industrial purposes, biomedical parts usually consist of small components like vessels, scaffolds, bones and so on. The position accuracy of each component might impact the functionality of the final product. Therefore, a knee arthrosis shown in Fig.2.4(d) is designed as the third test object to inspect the shape and relative pose of two knee bones.

The LOS-D2 distribution was selected for all experiments to minimize computational expense. Physical artifacts for *Experiments 2* and *Experiment 4* were printed on a MakerGear FFF machine with 1.75mm PLA filament and scanned using a HDI 100 series 3D scanner from LMI Technologies. It is important to note that in *Experiments 2* and *Experiment 4*, no attempt was made to fixture or align the physical artifact or the resulting point cloud with respect to the nominal model.

2.5 Results

2.5.1 Experiment 1

The first experiment is designed to demonstrate the sensitivity of LOS distribution for a point cloud with respect to itself in terms of pose, scale and point cloud density, and to verify the advantage of LOS distribution in quality inspection. In this experiment, the entire test object is considered as a single feature. There are three sample point clouds in Fig.2.5(b)-(d) generated by simulation based on the given nominal point cloud which is downsampled with grid size equal to 2 as shown in Fig.2.5 (a). All three test objects 1a-1c are designed to simulate the situation in reality such that the scanned point cloud obtained from the 3D scanning 1) has different point cloud density from the nominal model, 2) has different pose from the nominal model, or 3) has different scale from the

nominal model. The LOS distribution comparison between each sample and the nominal model are shown in Fig.2.6. The threshold to claim an unacceptable sample is equal to 1.9796, which is established based 20 simulated models with a worst variability equal to 0.5 unit.

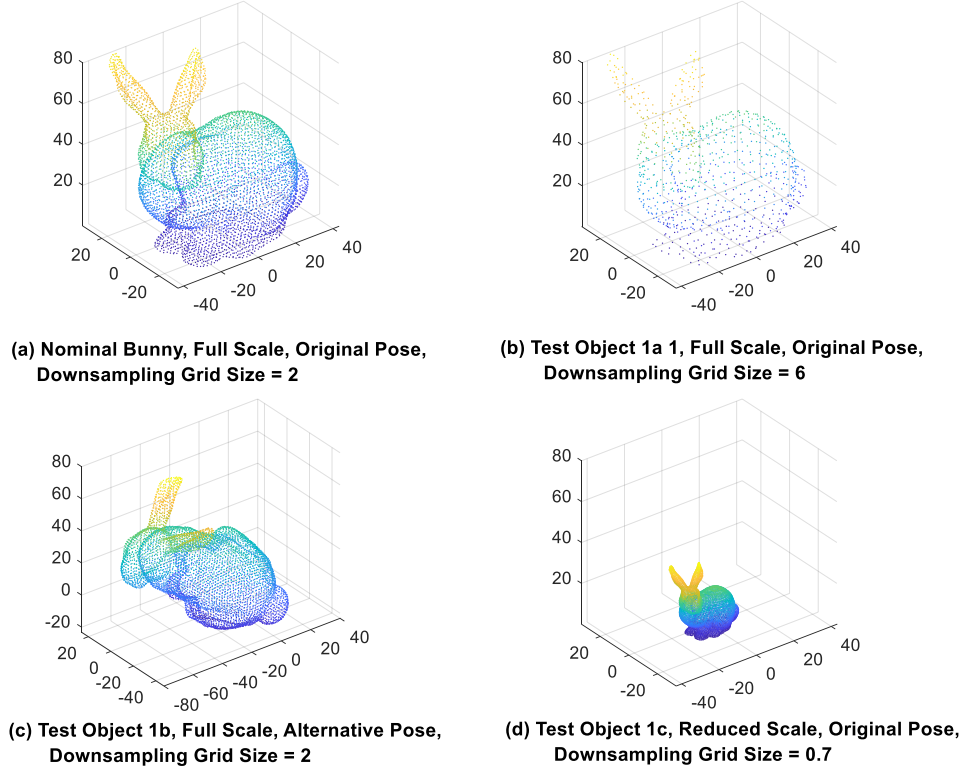


Figure 2.5: The point cloud associated with test object 1 was modified for scale, orientation, and point cloud density to test the sensitivity of the LOS distribution to these factors.

The first sample point cloud shown in Fig.2.5(b) is downsampled with grid size equal to 6, and thus it is less dense than the nominal point cloud. The K-S distance between their corresponding LOS distributions is 0.572. The K-S distance of the first sample is smaller than the threshold but larger than the other samples. This illustrates the importance of maintaining similar point cloud density between the nominal and scanned point clouds to reduce the variation observed in the K-S distance. In practice, point cloud density is determined by various inspection objectives (Colosimo et al., 2010; Huang et al., 2018). In some cases, sufficient density is required to capture high-spatial-frequency information of the point cloud; in other cases, low resolution is preferred considering the computational efficiency. The LOS distribution can be applied in both cases as long as Eq.(9) is satisfied to ensures that same level of details is maintained in sample and reference

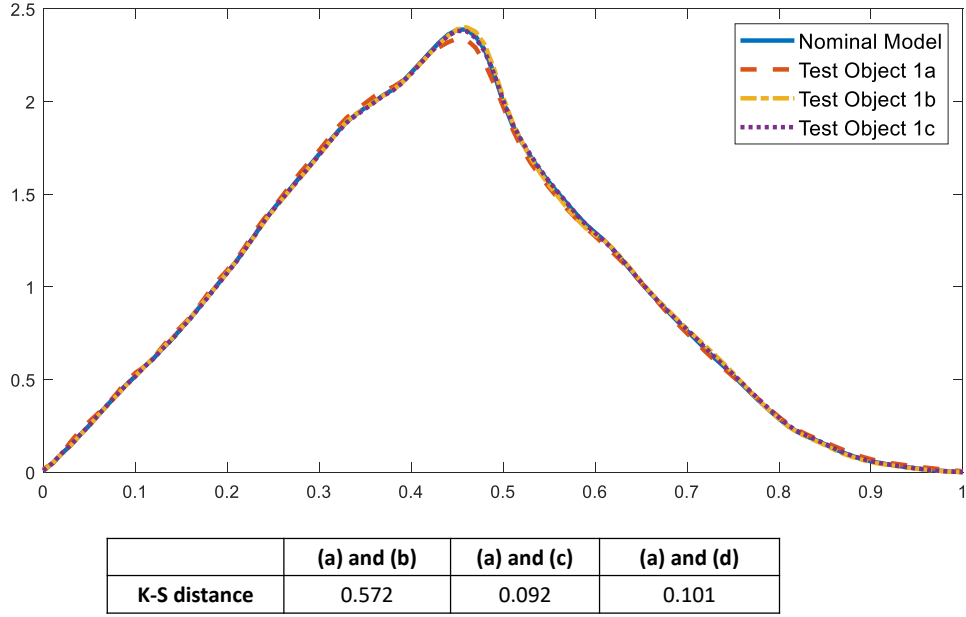


Figure 2.6: LOS distributions comparison for test object 1 are invariant to pose and scale, but are sensitive to point cloud density.

point clouds for quality inspection.

The second sample shown in Fig.2.5 (c) has the same scale and downsampling grid size as the nominal model, but its orientation is altered by rotating the object by 180, 160 and 90 degrees along the x,y, and z axes, respectively. The K-S distance between the LOS distribution of the second sample point cloud and nominal point cloud is 0.092, indicating that the LOS distribution is not sensitive to the pose of the part.

In Fig. 2.5 (d), the third sample part is scaled down to 30% of the nominal size and its grid size is set according to Eq. 2.9. The K-S distance between the LOS distribution and the nominal point cloud is 0.101. This indicates that the LOS distribution is scale-invariant as long as similar point cloud densities are maintained between parts.

The results of the second and the third sample indicate that the LOS distribution, as a shape profile, is invariant to the pose and scale of the shape. Each point cloud will have a unique LOS distribution, which is a piece-wise pdf and is no longer associated with a specific CCS, or a specific pose and scale of the part. Therefore, it is claimed that by the proposed inspection method based

on the LOS distribution, it does not require alignment.

2.5.2 Experiment 2

The second experiment is conducted on three test objects shown in Fig.2.7(a)-(c). Each test object contains four features of interest which are indicated in Fig.2.7(d). The point clouds of these features are obtained by k-means clustering and are indexed as shown in Fig.2.7(d). Test object 2a is printed from an unmodified CAD file. The test objects 2b and 2c were printed from modified CAD files to create shape and pose deviations that might occur during the printing process, as shown in Fig.2.7(b) and 2.7(c). Point clouds are downsampled with grid size equal to 1.8. The thresholds for each inspection process are calculated based on 20 simulation models with a worst variability equal to 0.1 unit.

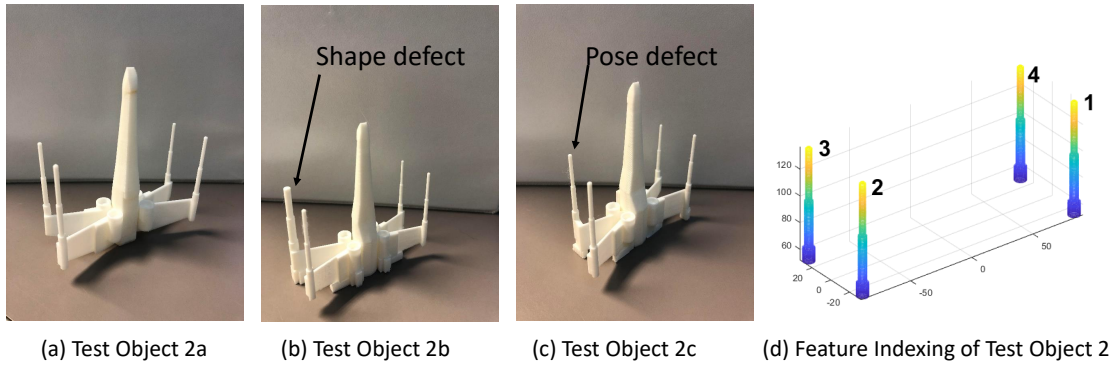


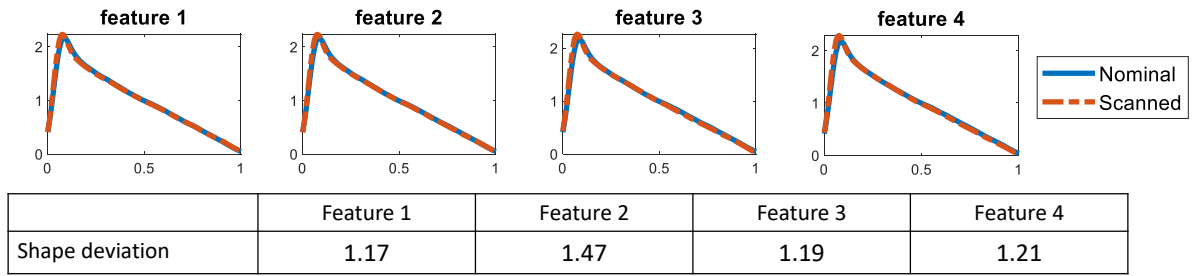
Figure 2.7: Objects printed for experiment 2(a-c); features of interest for experiment 2.

LOS Shape Inspection of Features. To test the effectiveness of the proposed method in shape inspection, the LOS distribution comparison is conducted on test objects 2a and 2b. This tests the methods' ability to detect the feature shape error present in test object 2b. The thresholds for shape inspection of each feature are presented in Table 2.3. The shape deviation of two test objects are shown in Fig.2.8. The results for test object 2a show that the LOS distributions generated from the sample part nearly overlap those based on the nominal model. Correspondingly, K-S distances of all four features are smaller than the thresholds indicating that all of them are acceptable. In contrast, in the results for test object 2b show that the shape deviation of feature #3 is signifi-

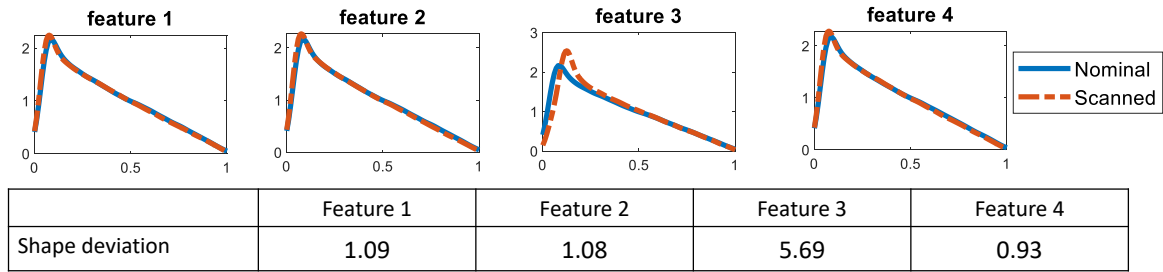
cantly larger than the threshold, indicating that this feature is not acceptable. It is worth noting that the potential application of LOS shape inspection in detecting non-homogeneous shape shrinkage/expansion is indicated in the result of test object 2b, which has a defective feature (feature #3) with abnormal shape expansion.

Table 2.3: Threshold of shape inspection for test object 2a and 2b.

	Feature 1	Feature 2	Feature 3	Feature 4
Threshold	1.471	1.496	1.472	1.515



(a) Shape Inspection Result of Test Object 2a



(b) Shape Inspection Result of Test Object 2b

Figure 2.8: Shape inspection results for test objects 2a and 2b detect the shape defect introduced in test object 2b.

LOS Inspection with Datum Features. LOS inspection with datum features is applied to test objects 2a and 2c. This tests the method's ability to detect the feature pose error in object 2c. The LOS deviation for each feature of interest is measured relative to the datum feature indicated in Fig.2.9. The thresholds for datum-based LOS inspection are presented in Table 2.4. The results of test objects 2a and 2c are shown in Fig.2.10. Sample and nominal LOS distributions nearly overlap in every case except for feature #3 of object 2c. The corresponding K-S distances confirm that the shape and/or pose of this feature is significantly larger than its corresponding threshold, and is

therefore not acceptable. To further narrow down the problem with this feature, a shape inspection is performed. Given that the shape inspection results for this feature are similar to those observed in Fig.2.8, one may conclude that pose is the primary issue with feature #3 of test object 2c.

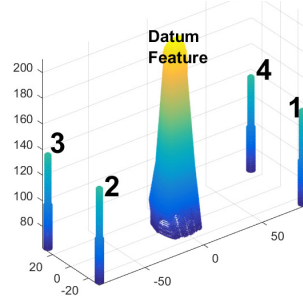
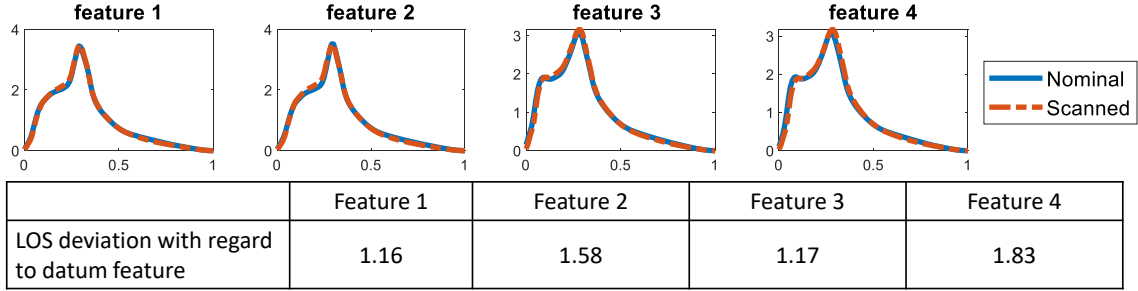


Figure 2.9: Datum feature and four target features.

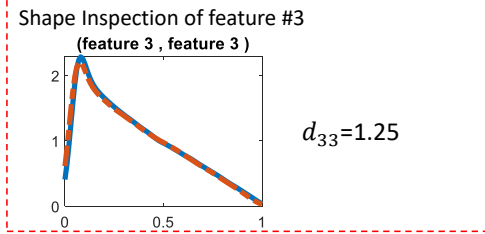
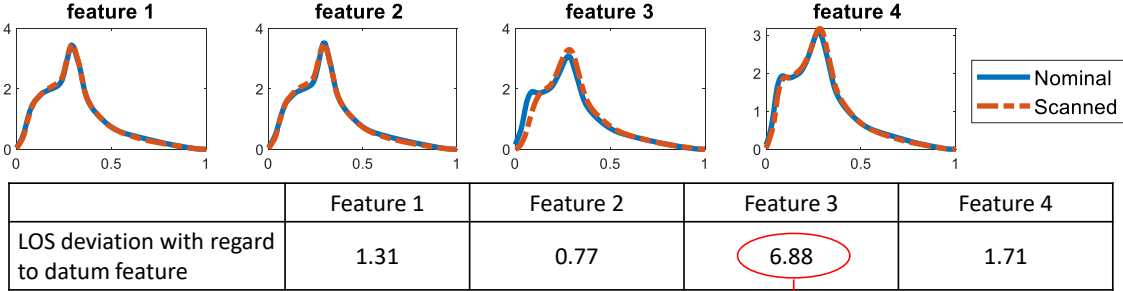
Table 2.4: Threshold of datum-based LOS inspection for test objects 2a and 2c.

	Feature 1	Feature 2	Feature 3	Feature 4
Threshold	2.224	2.399	2.092	2.269

LOS Inspection Without Datum Features. LOS inspection is also conducted on test objects 2a and 2c to test the method's ability to detect the feature pose error in object 2c without referencing a datum feature. For each pair of features, feature \mathcal{P}_i and feature \mathcal{P}_j for $i, j \in \{1, 2, 3, 4\}$, the nominal and sample LOS distributions are plotted and the corresponding inspection matrices are shown in Fig.2.11 for object 2a and in Fig.2.12 for object 2c. The full matrices are shown for completeness, but analysis is focused on the upper triangle due to their symmetry. The threshold for each entry in the K-S distance table are presented in Table 2.5. For object 2a, all nominal and sampled LOS distributions appear to overlap, and none of the corresponding K-S distances stand out. The same is true of the diagonal values for object 2c, indicating that feature shapes are acceptable; however, d_{13} , d_{23} and d_{34} clearly stand out, indicating that pose errors exist in the part. For this test case, the inspection matrix yields insight into what is wrong with the part. The interpretation is that features 1 & 3, 2 & 3, and 3 & 4 are not correctly positioned relative to each other. Given that all features except for 3 appear to be correctly positioned relative to one another, it is deduced that feature 3 is incorrectly positioned.



(a) LOS Inspection Result of Test Object 2a



(b) LOS Inspection Result of Test Object 2c

Figure 2.10: LOS inspection result for test objects 2a and 2c detect the pose defect introduced in test object 2c.

Table 2.5: Threshold of non-datum-based LOS inspection for test objects 2a and 2c.

Threshold	\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4
\mathcal{P}_1	1.470	1.809	2.234	1.838
\mathcal{P}_2	1.809	1.614	1.965	2.517
\mathcal{P}_3	2.234	1.965	1.426	2.438
\mathcal{P}_4	1.838	2.517	2.438	1.432

The results do highlight two weaknesses of LOS inspection without datum features. First, if multiple features are incorrectly positioned, the interpretation of which features are out of place might not be clear. Large values in the matrix would indicate pose problems within the part even if the specifics are not evident, but inspection against a datum feature offers a more definitive interpretation. Second, its computational expense scales in $\frac{n^2+n}{2}$, where n is the number of features. In contrast, inspection against a datum feature scales in n . For these reasons, inspection against a

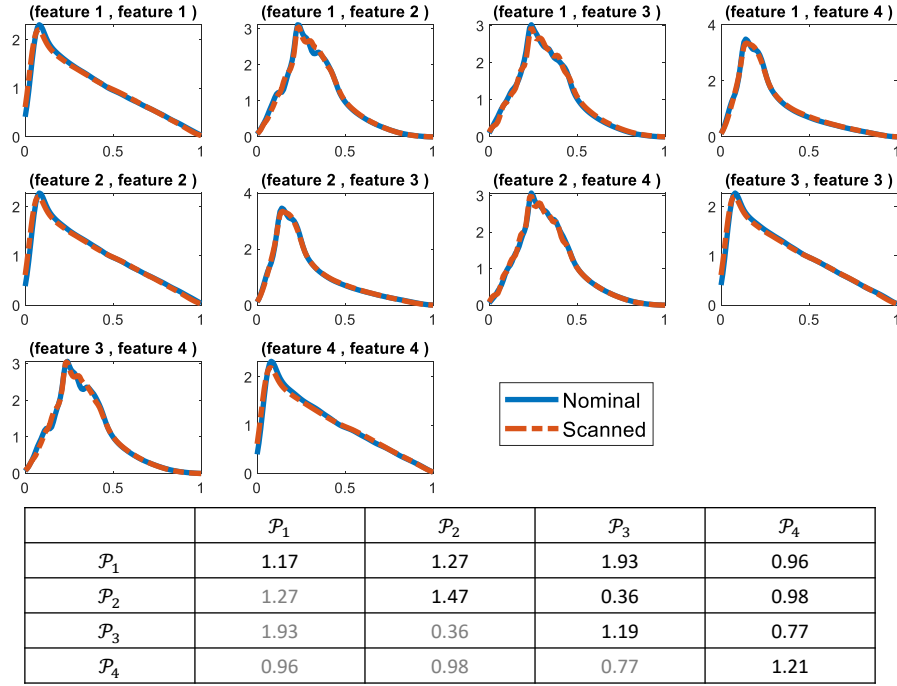


Figure 2.11: The LOS distribution comparison for test object 2a indicates no features out of place with respect to each other.

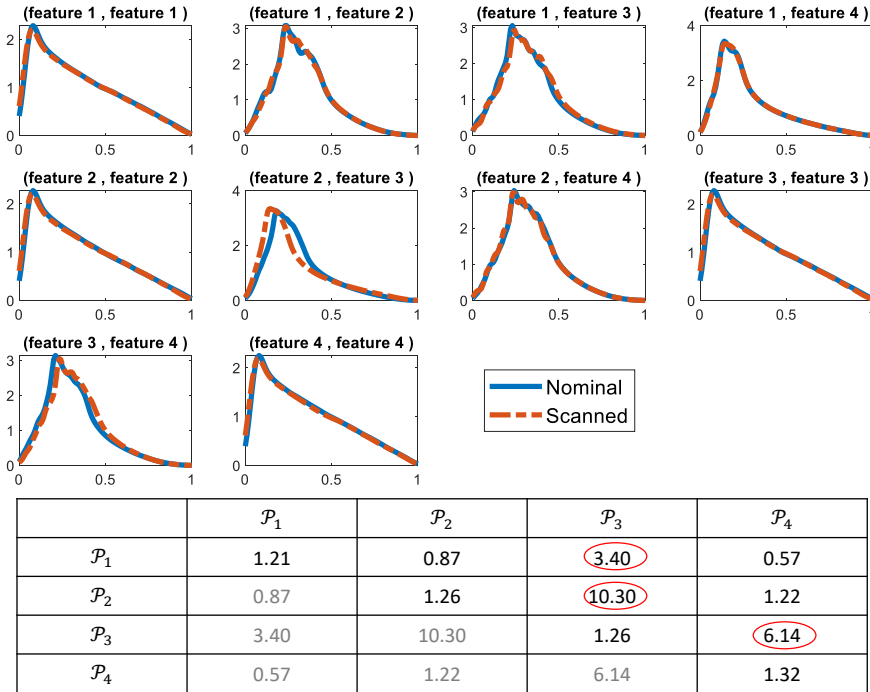


Figure 2.12: The LOS distribution comparison for test object 2c indicates pose problems with feature 3 relative to features 1, 2, and 4.

datum feature is preferred.

Feature Re-indexing. The ability to disambiguate indexing for similar features was tested on test objects 2a and 2c. Features in the sample point clouds were renumbered as shown in Fig.2.13. In the absence of unique features, one correct feature index must be known a priori; in this case, feature #2 is specified as being correctly indexed.

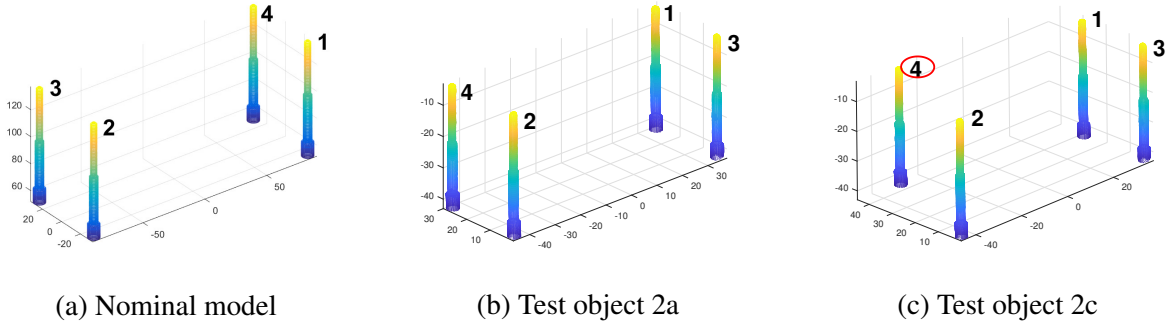
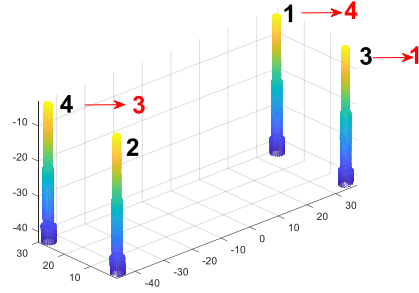


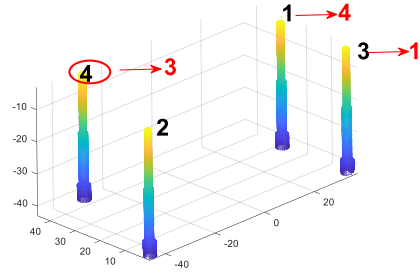
Figure 2.13: An example when feature indexing of nominal model and sample parts is inconsistent.



$i = 2$		$G_{il}^{(n)}$			
		$l = 1$	$l = 2$	$l = 3$	$l = 4$
$G_{ik}^{(s)}$	$k = 1$	1.70	17.62	24.05	0.98
	$k = 2$	15.87	1.47	13.98	16.37
	$k = 3$	1.27	15.45	21.89	1.91
	$k = 4$	22.66	13.99	0.36	23.27

Figure 2.14: Feature re-indexing was successful for test object 2a.

Referring to Eq.2.8, i represents this feature, $l \in \{1, 2, 3, 4\}$ represents the indices of features in the nominal point cloud, and $k \in \{1, 2, 3, 4\}$ represents those in the scanned point cloud. The results in Figs.2.14 and 2.15 show all combinations of K-S distances, $(\max |G_{ik}^{(s)} - G_{il}^{(n)}|)$, for the test parts. The correct figure indexes for the sampled parts are the values of k associated with the column-wise minimum. Taking test object 2a in Figs.2.14 as an example, feature #1 ($l = 1$) in the



$i = 2$		$G_{il}^{(n)}$			
		$l = 1$	$l = 2$	$l = 3$	$l = 4$
$G_{ik}^{(s)}$	$k = 1$	1.94	17.78	24.29	1.22
	$k = 2$	16.00	1.26	13.77	16.51
	$k = 3$	0.87	16.56	22.21	1.52
	$k = 4$	12.89	13.71	10.30	13.59

Figure 2.15: Feature re-indexing was successful for test object 2c despite the pose error associated with feature #3.

nominal model corresponds to feature #3 in the scanned model, feature #3 ($l = 3$) in the nominal model corresponds to feature #4 in the scanned model, and so forth. The resulting renumbering of the scanned features for both parts is graphically depicted in Figs.2.14 and 2.15. The results are correct, even in the presence of the feature pose present in test object 2c.

2.5.3 Experiment 3

The third experiment is conducted on a topology optimized object, which has seven features of interest which are created by k-means clustering. The feature indexing of the nominal model and the test object are shown in Fig.2.16 (a) and (b). The test object is created by simulation with one feature (feature #1 in the test object which is feature #4 in the nominal model) deviated from its original position. It is observed that the feature indexing is not consistent between the test object and the nominal model. Therefore, the features of the test object should be re-indexed before LOS inspection. It is worth noting that all features presented in Fig. 2.16 (c) have similar shape distributions, further demonstrating the effectiveness of the proposed methods in dealing with similar features. Point clouds are downsampled with grid size equal to 0.01. The thresholds

for each inspection process are calculated based on 20 simulation models with a worst variability equal to 0.0001 unit.

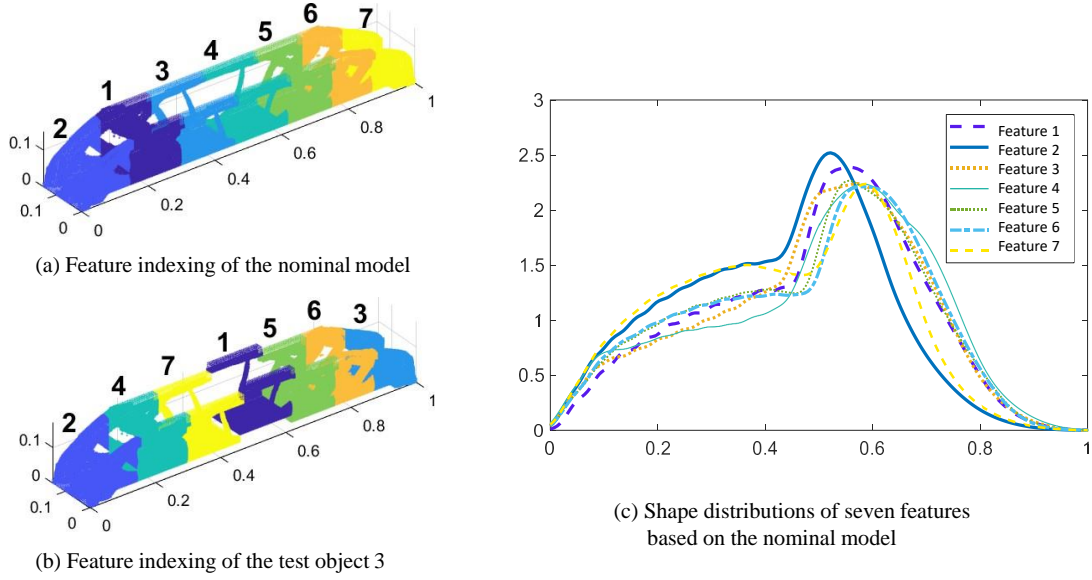
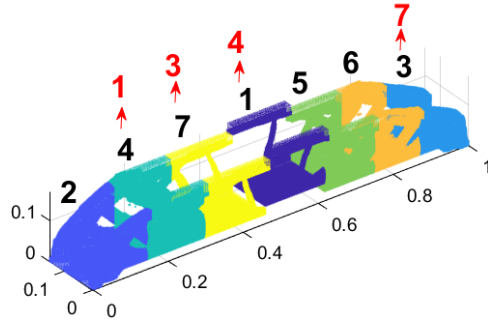


Figure 2.16: Test objects for Experiment 3; the shape distribution of each feature is compared in (c).

First, the index of the test object 3 should be adjusted based on the feature indexing of the nominal model. In this case, feature #2 is identified as the feature being correctly indexed by shape distribution comparisons referring to Eq. 2.7. Due to the similarity of shape distributions of the rest features, their re-indexing is performed by following the procedure as described in Experiment 2 for non-unique features. The re-indexed features of the test object 3 are shown in Fig.2.17 based on the similarity of each pair of LOS distributions. Next, the shape and position of each feature is examined by LOS inspection without datum features. The threshold for each entry in LOS inspection table is presented in Table 2.6. After the re-indexing, the pre-known defective feature is feature #4. Therefore, in the LOS inspection result shown in Table 2.7, large off-diagonal entries associated with feature #4, which are greater than the corresponding thresholds, are observed, indicating the existence of the pose problem of feature #4 relative to others.



		$G_{il}^{(n)}$						
$i = 2$		$l = 1$	$l = 2$	$l = 3$	$l = 4$	$l = 5$	$l = 6$	$l = 7$
$G_{ik}^{(s)}$	$k = 1$	9.73	8.51	2.70	0.28	3.80	3.11	4.43
	$k = 2$	7.51	0.17	10.75	7.16	9.71	8.71	9.90
	$k = 3$	13.76	10.02	4.13	4.99	1.21	2.19	0.01
	$k = 4$	0.05	7.38	12.31	8.83	13.46	12.20	13.72
	$k = 5$	13.55	9.89	3.49	4.79	0.07	1.64	1.27
	$k = 6$	12.20	8.83	4.52	3.62	1.67	0.10	2.26
	$k = 7$	12.23	10.83	0.13	3.70	3.54	4.60	4.12

Figure 2.17: Feature re-indexing was successful for test object 3.

Table 2.6: Threshold of LOS inspection for test object 3.

Threshold	\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{P}_5	\mathcal{P}_6	\mathcal{P}_7
\mathcal{P}_1	0.79	0.58	0.24	0.33	0.10	0.13	0.08
\mathcal{P}_2	0.58	0.38	0.68	0.57	0.74	0.91	0.76
\mathcal{P}_3	0.24	0.68	0.11	0.31	0.13	0.19	0.10
\mathcal{P}_4	0.33	0.57	0.31	0.44	0.24	0.31	0.53
\mathcal{P}_5	0.10	0.74	0.13	0.24	0.60	0.09	0.06
\mathcal{P}_6	0.13	0.91	0.19	0.31	0.09	0.37	0.35
\mathcal{P}_7	0.08	0.76	0.10	0.53	0.06	0.35	0.18

Table 2.7: LOS inspection result for the test object 3 indicating the pose problem with feature #4 relative to all other features.

	\mathcal{P}_1	\mathcal{P}_2	\mathcal{P}_3	\mathcal{P}_4	\mathcal{P}_5	\mathcal{P}_6	\mathcal{P}_7
\mathcal{P}_1	0.66	0.05	0.15	1.06	0.06	0.07	0.07
\mathcal{P}_2	0.05	0.15	0.11	1.29	0.08	0.11	0.01
\mathcal{P}_3	0.15	0.11	0.09	4.50	0.07	0.14	0.04
\mathcal{P}_4	1.06	1.29	4.50	0.41	5.77	1.14	1.08
\mathcal{P}_5	0.06	0.08	0.07	5.77	0.58	0.08	0.04
\mathcal{P}_6	0.07	0.11	0.14	1.14	0.08	0.35	0.31
\mathcal{P}_7	0.07	0.01	0.04	1.08	0.04	0.31	0.13

2.5.4 Experiment 4

Two bones joined make up the arthrosis shown in Fig.2.18. The lower half is considered as feature #1 while the upper half is feature #2. LOS inspection is conducted to inspect the shape and pose of these components. Test object 4a is printed with the intended geometry, while pose defects are deliberately introduced into test objects 4b and 4c. Point clouds and as-built artifacts are shown in Fig.2.18 for all three parts. All test objects are compared to the nominal model which is downsampled with grid size equal to 3. In this experiment, the thresholds to differentiate acceptable/unacceptable test object are established based on 20 simulated models with a worst variability equal to 0.3 unit. The threshold for each of the four entries in the K-S distance table are presented in Table 2.8.

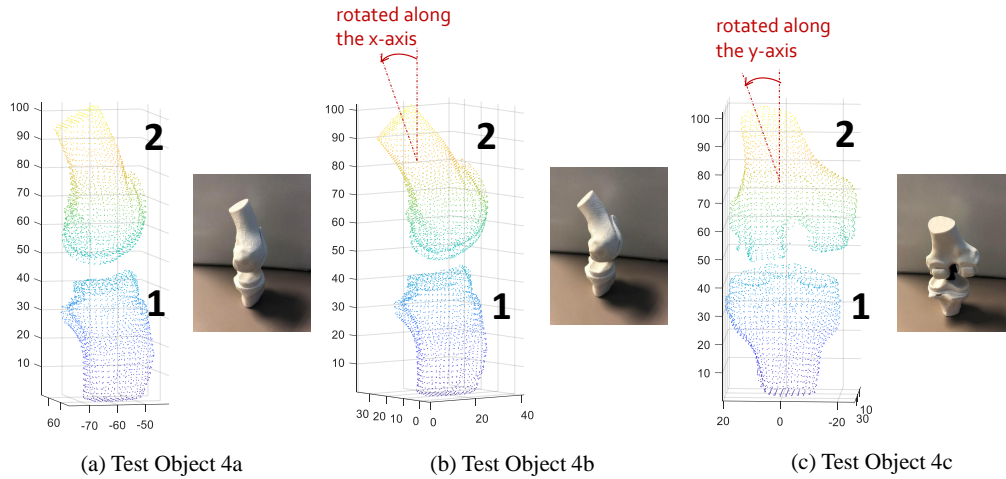


Figure 2.18: Three test objects were printed for Experiment 4: one unaltered and two with pose errors introduced.

Table 2.8: Threshold of LOS inspection for test object 4a, 4b and 4c

Threshold	\mathcal{P}_1	\mathcal{P}_2
\mathcal{P}_1	1.232	3.166
\mathcal{P}_2	3.166	1.164

The LOS inspection result and K-S distance values of three test objects are presented in Fig.2.19. It is observed that the test object 4a is accepted because all entries in the table are within their cor-

responding thresholds. Test object 4b is rejected due the large off-diagonal K-S distance value 4.21, which is greater than the corresponding threshold 3.166, indicating the existence of pose error of one feature relative to the other. Test object 4c is also unacceptable by following the similar rationale.

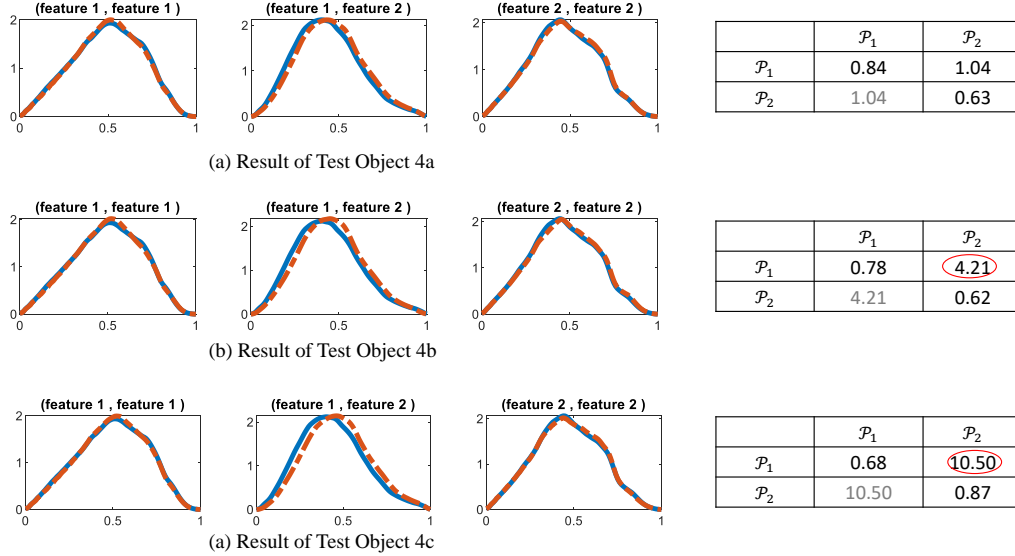


Figure 2.19: LOS inspection results for Experiment 4 indicate good shape fidelity for all test objects, but clearly differentiate the pose errors that exist in test objects 4b and 4c.

In addition, this experiment also gives insight that LOS distribution for a point cloud with respect to itself is pose-invariant. The diagonal entries in the table indicated the shape accuracy of a specific feature while the off-diagonal entries indicated the pose accuracy of a feature relative to another. From the diagonal entries associated with the shape of feature #2, it is observed that even the feature #2 of test objects 4b and 4c are altered, and the results of feature #2 shape inspection in the three tables are consistent in all three test objects. That is why we claim the shape inspection of LOS distribution is pose invariant.

2.6 Discussion

The results show that LOS distribution methods were able to differentiate between correctly and incorrectly printed features in physical trials. The results confirm that the method works for both parametric and nonparametric geometries, exhibits good pose and scale invariance, and that it can

resolve ambiguities caused by inconsistent feature indexing. It is worth noting that the process of establishing threshold based on simulation models will not increase the computational cost on a large scale. In most case, with reasonable sampling resolution, the threshold can be obtained within a minute. In addition, the simulation can be integrated with the inspection process in a "black box" for non-engineers. Therefore, the only input required from the users are scanned point cloud, the nominal model and the worst variability she/he can tolerate.

To evaluate the LOS distribution methods in the context of other more conventional methods, it is important to note that the former represents a fundamentally different, higher-level approach. Conventional methods focus on measuring the deviation of individual points from a CMM or 3D scanner. In the case of GD&T inspection, these points are typically classified as lying inside or outside of a volume that defines the tolerance for a given feature. In other schemes, the distance from each point to the ideal surface is quantified and analysis is based on this metric. In contrast, LOS inspection focuses on features as holistic entities represented by point clouds. It eliminates rigidly defined tolerance zones in favor of shape similarity measures. This alternative approach is not necessarily better or worse than conventional inspection methods; rather, its advantages and disadvantages must be considered with the specific inspection objectives in mind.

Key advantages of LOS inspection are related to datums and datum surfaces. Datums are conventionally established to serve as both a reference for measurement and to facilitate repeatable fixturing. As such, datum features are usually chosen to be simple parametric entities (planes, cylinders, etc.) and kept to the minimum required to meet 3-2-1 fixturing requirements. Although not strictly required, all critical part features usually reference a common set of datums to avoid refixturing during inspection. Such constraints are not a concern in LOS inspection because its insensitivity to the workpiece pose eliminate the need for fixture tooling. Each feature of interest can have unique datum(s), and since datums are represented via point clouds, they can be directly defined by complex combinations of parametric and/or nonparametric surfaces without resorting to implied datums. Time spent fixturing parts for inspection is eliminated, as is the cost and lead time associated with special fixturing. Features can also be inspected against datums that are of a

different scale. From the designer's perspective, datum specification becomes more intuitive and potentially more meaningful. This brings feature-based inspection to AM users who are not trained in the intricacies of GD&T and work outside of formal manufacturing environments.

LOS inspection is based on 3D scanning, which yields high point cloud densities in considerably less time than CMM. It can also leverage x-ray CT to inspect internal structure and other areas inaccessible via CMM such as deep holes, undercuts, and so forth. Conventional analysis of 3D scanned objects begins by aligning the measured point cloud with the nominal model, which poses problems. First, this cannot be done when the measured point cloud is scaled differently from the nominal model. Second, point cloud alignment is usually accomplished by least squares or some other global optimization algorithm. Reference features are not necessarily aligned correctly with the corresponding points in the measured point cloud, so inspection relative to these features is problematic. LOS inspection solves both problems.

The scale invariance of the LOS inspection method has an important application in direct-write AM processes where post-print processing can introduce significant shrinkage. Examples include thermoset polymer matrix composites (Pierson et al.) and ceramics (Rueschhoff et al., 2016; Costakis Jr et al., 2016). In these applications, 3D geometry is printed in the liquid state, relying on fluid yield stress to form a self-supporting structure. After printing, the entire structure is cured or fired, during which the part experiences significant shrinkage. This secondary curing process can be lengthy (16 hours in the case of Pierson et al.), so it is important to have confidence in the as-printed geometry, which is scaled by the shrinkage factor. Of course, implicit is the assumption of homogeneous shrinkage, which is valid in the example applications.

A fair assessment of LOS inspection also requires an examination of its disadvantages. The output of the LOS methods are distances between LOS distributions, which are in turn based on D2 distance, D3 area, D4 volume, or A3 angle shape vectors. This lacks the clear physical interpretation of pointwise physical distance to a true surface provided by conventional methods, although LOS inspection can offer more high-level insight. The advantages associated with the high-level nature of the method are tempered by the fact that LOS inspection also does not provide

the same level of detail as pointwise distance measurements. Small local anomalies in otherwise accurate features could be lost in the noise, and it does not provide the user with information as to what is wrong with features that are flagged as inaccurate. This can be mitigated somewhat via finer feature segmentation, but at higher computational cost. Finally, LOS inspection relies on 3D scanning technology that has yet to match the precision and reliability of CMM. This gap is expected to gradually close, though, as the technology advances.

These advantages and disadvantages suggest an application-specific tradeoff between LOS inspection and more conventional methods, but it is not necessarily an either-or choice. LOS inspection could be used in combination with conventional methods to provide supplemental information. For 3D scanned parts of similar scale, the same point clouds could be analyzed using both LOS inspection and pointwise deviation to yield a more complete evaluation of as-built geometry. LOS inspection could also be used as a screening tool in conjunction with CMM/GD&T. In this role, LOS inspection could screen for suspected bad parts, thereby reducing the number of parts sent to CMM. It could also identify specific features of concern, the focusing CMM inspection resources on the features most likely to be out of tolerance.

2.7 Conclusion

This work proposes a point-cloud-based inspection method based on the novel LOS distribution, which is a generalization of the shape distribution concept used in object recognition. LOS distributions are probability density functions that encode both shape and pose information based on geometric relationships between points in two point clouds. The LOS distribution between features of interest is constructed from both the scanned artifact and the nominal model, and the difference between the distributions is used as a quality metric. Inspection via LOS distributions permits feature-based inspection with respect to form, position, and orientation with or without datum surfaces. Although the method can be applied to parametric geometries in traditional manufacturing, it is particularly well suited to inspection of nonparametric geometries in the high-mix, low-volume production environment typical of AM. The method is tested on several physical AM

builds, and the results show that the method is able to reliably differentiate between acceptable and unacceptable features.

References

- Ameta, G., Lipman, R., Moylan, S., and Witherell, P. “Investigating the role of geometric dimensioning and tolerancing in additive manufacturing”. *Journal of Mechanical Design*, 137(11): 111401, 2015a.
- Ameta, G., Witherell, P., Moylan, S., and Lipman, R. “Tolerance specification and related issues for additively manufactured products”. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, page V01AT02A027. American Society of Mechanical Engineers, 2015b.
- Anonymity. “Top 10 Best Low Cost 3D Scanners (2018 Update)”. URL <https://www.3dnatives.com/en/top-10-low-cost-3d-scanners280320174/>, November 2017a.
- Anonymity. “Downsampling a pointcloud using a voxelgrid filter”. URL http://pointclouds.org/documentation/tutorials/voxel_grid.php, May 2017b.
- Anonymity. “Topology optimized bridge”. URL <https://www.thingiverse.com/thing:3491813>, March 2019.
- Ben-Shabat, Y., Lindenbaum, M., and Fischer, A. “3d point cloud classification and segmentation using 3d modified fisher vector representation for convolutional neural networks”. *arXiv preprint arXiv:1711.08241*, 2017.
- Besl, P. J. and McKay, N. D. “Method for registration of 3-d shapes”. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.
- Brackett, D., Ashcroft, I., and Hague, R. “Topology optimization for additive manufacturing”. In *Proceedings of the Solid Freeform Fabrication Symposium, Austin, TX*, volume 1, pages 348–362. S, 2011.
- Charles, R. Q., Su, H., Kaichun, M., and Guibas, L. J. “Pointnet: Deep learning on point sets for 3d classification and segmentation”. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 77–85. IEEE, 2017.
- Choi, J.-Y., Choi, J.-H., Kim, N.-K., Kim, Y., Lee, J.-K., Kim, M.-K., Lee, J.-H., and Kim, M.-J. “Analysis of errors in medical rapid prototyping models”. *International Journal of Oral and Maxillofacial Surgery*, 31(1):23–32, 2002.
- Ciocca, L., De Crescenzo, F., Fantini, M., and Scotti, R. “Cad/cam and rapid prototyped scaffold construction for bone regenerative medicine and surgical transfer of virtual planning: a pilot study”. *Computerized Medical Imaging and Graphics*, 33(1):58–62, 2009.

- Colosimo, B. M., Moroni, G., and Petrò, S. “A tolerance interval based criterion for optimizing discrete point sampling strategies”. *Precision Engineering*, 34(4):745–754, 2010.
- Colosimo, B. M., Cicorella, P., Pacella, M., and Blaco, M. “From profile to surface monitoring: Spc for cylindrical surfaces via gaussian processes”. *Journal of Quality Technology*, 46(2): 95–113, 2014.
- Colosimo, B. M., Pacella, M., and Senin, N. “Multisensor data fusion via gaussian process models for dimensional and geometric verification”. *Precision Engineering*, 40:199–213, 2015.
- Costakis Jr, W. J., Rueschhoff, L. M., Diaz-Cano, A. I., Youngblood, J. P., and Trice, R. W. “Additive manufacturing of boron carbide via continuous filament direct ink writing of aqueous ceramic suspensions”. *Journal of the European Ceramic Society*, 36(14):3249–3256, 2016.
- Del Castillo, E., Colosimo, B. M., and Tajbakhsh, S. D. “Geodesic gaussian processes for the parametric reconstruction of a free-form surface”. *Technometrics*, 57(1):87–99, 2015.
- Eckart, B., Kim, K., and Kautz, J. “Hgmr: Hierarchical gaussian mixtures for adaptive 3d registration”. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 705–721, 2018.
- El-Katatny, I., Masood, S., and Morsi, Y. “Error analysis of fdm fabricated medical replicas”. *Rapid Prototyping Journal*, 16(1):36–43, 2010.
- Germani, M., Raffaeli, R., and Mazzoli, A. “A method for performance evaluation of re/rp systems in dentistry”. *Rapid Prototyping Journal*, 16(5):345–355, 2010.
- Grilli, E., Menna, F., and Remondino, F. “A review of point clouds segmentation and classification algorithms”. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:339–344, 2017.
- Hong-Seok, P. and Mani, T. U. “Development of an inspection system for defect detection in pressed parts using laser scanned data”. *Procedia Engineering*, 69:931–936, 2014.
- Huang, D., Du, S., Li, G., Zhao, C., and Deng, Y. “Detection and monitoring of defects on three-dimensional curved surfaces based on high-density point cloud data”. *Precision Engineering*, 53:79–95, 2018.
- Jahani, S., Kontar, R., Veeramani, D., and Zhou, S. “Statistical monitoring of multiple profiles simultaneously using gaussian processes”. *Quality and Reliability Engineering International*, In press, 2018.
- Jin, Y., Plott, J., Chen, R., Wensman, J., and Shih, A. “Additive manufacturing of custom orthoses and prostheses—a review”. *Procedia CIRP*, 36:199–204, 2015.
- Juneja, M., Thakur, N., Kumar, D., Gupta, A., Bajwa, B., and Jindal, P. “Accuracy in dental surgical guide fabrication using different 3-d printing techniques”. *Additive Manufacturing*, 22: 243–255, 2018.

- Kazmi, I. K., You, L., and Zhang, J. J. “A survey of 2d and 3d shape descriptors”. In *Computer Graphics, Imaging and Visualization (cgiv), 2013 10th International Conference*, pages 1–10. IEEE, 2013.
- Li, X. and Guskov, I. “Multiscale features for approximate alignment of point-based surfaces.”. In *Symposium on geometry processing*, volume 255, page 217. Citeseer, 2005.
- Lu, X., Yao, J., Tu, J., Li, K., Li, L., and Liu, Y. “Pairwise linkage for point cloud segmentation.”. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 3(3):201–208, 2016.
- Magnusson, M. *The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection*. PhD thesis, Örebro universitet, 2009.
- Mateus, D., Horaud, R. P., Knossow, D., Cuzzolin, F., and Boyer, E. “Articulated shape matching using laplacian eigenfunctions and unsupervised point registration”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008. URL <http://perception.inrialpes.fr/Publications/2008/MHKCB08>.
- Meadows, J. D. *Geometric dimensioning and tolerancing: applications, analysis & measurement (per ASME Y14. 5-2009)*. James D. Meadows & Associates, 2009.
- Meakin, J. R., Shepherd, D. E. T., and Hukins, D. W. L. “Fused deposition models from ct scans”. *The British Journal of Radiology*, 77(918):504–507, 2004.
- Minetola, P. “The importance of a correct alignment in contactless inspection of additive manufactured parts”. *International Journal of Precision Engineering and Manufacturing*, 13(2): 211–218, 2012.
- Monika Mahto, B. S. “3D opportunity for electronics”. URL <https://www2.deloitte.com/insights/us/en/focus/3d-opportunity/additive-manufacturing-3d-printed-electronics.html>, May 2017.
- Murphy, S. V. and Atala, A. “3d bioprinting of tissues and organs”. *Nature Biotechnology*, 32(8): 773, 2014.
- Myronenko, A. and Song, X. “Point set registration: Coherent point drift”. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010.
- Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. “Shape distributions”. *ACM Transactions on Graphics (TOG)*, 21(4):807–832, 2002.
- Pierson, H., Celik, E., Abbott, A., De Jarnette, H., Gutierrez, L. S., Johnson, K., Koerner, H., and Baur, J. “Mechanical properties of printed epoxy-carbon fiber composites”. *Experimental Mechanics*, pages 1–15.
- Pierson, H. A. and Chivukula, B. “Process–property relationships for fused filament fabrication on preexisting polymer substrates”. *Journal of Manufacturing Science and Engineering*, 140(8): 084501–084507, 2018.

- Puzicha, J., Buhmann, J. M., Rubner, Y., and Tomasi, C. “Empirical evaluation of dissimilarity measures for color and texture”. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1165–1172. IEEE, 1999.
- Rao, P. K., Kong, Z., Duty, C. E., Smith, R. J., Kunc, V., and Love, L. J. “Assessment of dimensional integrity and spatial defect localization in additive manufacturing using spectral graph theory”. *Journal of Manufacturing Science and Engineering*, 138(5):051007–051019, 2016.
- Rueschhoff, L., Costakis, W., Michie, M., Youngblood, J., and Trice, R. “Additive manufacturing of dense ceramic parts via direct ink writing of aqueous alumina suspensions”. *International Journal of Applied Ceramic Technology*, 13(5):821–830, 2016.
- Savio, E., De Chiffre, L., and Schmitt, R. “Metrology of freeform shaped parts”. *CIRP Annals*, 56(2):810–835, 2007.
- Shi, Q., Xi, N., Chen, Y., and Sheng, W. “Registration of point clouds for 3d shape inspection”. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 235–240. IEEE, 2006.
- Silva, D. N., De Oliveira, M. G., Meurer, E., Meurer, M. I., da Silva, J. V. L., and Santa-Bárbara, A. “Dimensional error in selective laser sintering and 3d-printing of models for craniomaxillary anatomy reconstruction”. *Journal of Cranio-Maxillofacial Surgery*, 36(8):443–449, 2008.
- Song, X., Pan, Y., and Chen, Y. “Development of a low-cost parallel kinematic machine for multidirectional additive manufacturing”. *Journal of Manufacturing Science and Engineering*, 137(2):021005–021018, 2015.
- Standard, I. “1101: 2004 geometrical product specifications (gps)–geometrical tolerancing–tolerances of form, orientation, location and run-out”. *International Organization for Standardization, Geneva*, 2004.
- Steinberg, D. M. “Industrial statistics: The challenges and the research”. *Quality Engineering*, 28(1):45–59, 2016.
- Tangelder, J. W. and Veltkamp, R. C. “A survey of content based 3d shape retrieval methods”. In *Shape Modeling Applications, 2004. Proceedings*, pages 145–156. IEEE, 2004.
- Turk, G. and Levoy, M. “The stanford bunny”, 2005.
- Yamaguchi, F. *Curves and surfaces in computer aided geometric design*. Springer Science & Business Media, 2012.
- Zhang, C., Lei, Y., Zhang, L., and Chen, N. “Modeling tunnel profile in the presence of coordinate errors: A gaussian process-based approach”. *IIE Transactions*, 49(11):1065–1077, 2017.
- Zhang, X., Li, G., Xiong, Y., and He, F. “3d mesh segmentation using mean-shifted curvature”. In *International Conference on Geometric Modeling and Processing*, pages 465–474. Springer, 2008.

Zhu, Z., Anwer, N., Huang, Q., and Mathieu, L. “Machine learning in tolerancing for additive manufacturing”. *CIRP Annals*, 67(1):157–160, 2018.

3 Multi-resolution In-plane Alignment and Error Quantification

Additive manufacturing has shown its capability in producing complex geometries. Due to the additive nature, the in-situ layer-wise inspection of geometric accuracy is essential to making AM reach its full potential. This work proposes a novel automated in-plane alignment and error quantification framework to distinguish the fabrication, measurement, and alignment errors in additive manufacturing. In this work, a multi-resolution framework based on wavelet decomposition is proposed to automatically align two-dimensional point clouds via a polar coordinate representation and then to differentiate errors from different sources based on a randomized complete block design (RCBD) approach. In addition, a two-step optimization model is proposed to find the best configuration of the multi-resolution framework. The proposed framework can not only distinguish errors attributed to different sources but also evaluate the performance and consistency of alignment results under different levels of details. A sample part with different featured layers, including a simple free-form layer, a defective layer, and a layer with internal features, is used to illustrate the effectiveness and efficiency of the proposed framework. The proposed alignment method outperforms the widely used Iterative Closest Point algorithm. This work fills a research gap of state-of-the-art studies by automatically quantifying different types of error inherent in manufacturing, measuring, and part alignment.

3.1 Introduction

Additive manufacturing (AM) has shown its significant advantages over traditional subtractive and forming manufacturing processes for offering agile production of complex geometries (Gibson et al., 2014). Nevertheless, to make AM reach its full potential, geometric quality inspection is indispensable, especially in high-value applications such as biological structure substitutes in medical areas (Jin et al., 2015) and airfoils in aerospace industry (Liu et al., 2017) where component failures are not tolerated. Due to the layer-wise additive nature of AM processes, the layer-by-layer in-situ inspection and monitoring (rather than offline quality inspection) are desired in resolving failures on the current layer or out-of-control process variation as soon as they are recognized before the

errors get accumulated and affect the subsequent layers.

The geometric accuracy of each layer is usually referred to as in-plane (x-y plane) shape deviation (Cheng et al., 2018; Wang et al., 2017; Luan and Huang, 2017). Usually, the shape of the current layer can be described by tracing thousands of points along the specified boundary using optical/vision measurement equipment or by slicing a thin layer of 3D point cloud obtained from 3D scanning. The output of measurement can be considered as a two-dimensional point cloud, which must be aligned with the corresponding base-truth CAD model prior to evaluating the geometric accuracy of the layer. Common alignment methods coming with a point cloud processing software package rely on least-square fitting algorithms to find the optimal rigid transformation that minimizes the sum of squared distances between the measured alignment points and their target reference points. However, the performance of least-square fitting in aligning two-dimensional point clouds is worth to be examined. Moreover, in practice, the dimensional accuracy is associated with datum features from a Geometric Dimensioning and Tolerancing (GD&T) perspective, but the least-square fitting methods are not able to automatically align point clouds based on such datum features without first performing a global rough alignment. A phase-correction-based alignment method is proposed by Jin et al., (2019) for datum-based alignment. This method is adopted and improved in this work to align a point cloud with different levels of details maintained, and to evaluate the alignment performance in terms of alignment error and consistency.

Based on the aligned point cloud, the point-wise deviation, as a common metric for evaluating dimensional accuracy, is calculated for the currently inspected layer. It is usually presented with a deviation color map (e.g., Rao et al., 2016) or a deviation profile or using polar coordinates (e.g., Song et al., 2014). It is worth pointing out that the variation observed in the point-wise deviation is the result of systematic error, measurement error, and alignment error. The systematic error mainly results from a persistent issue, such as an inappropriate design or out-of-control fabrication process parameter, and leads to a consistent error, which is the one to trigger an alarm informing engineers to remove the error during in-situ monitoring. The measurement error is usually referred to as random error, which is unavoidable due to the limited resolution or precision of the mea-

asuring equipment. The alignment error provides a measure of how well a particular point cloud alignment/registration method performs in aligning the sample layer to the nominal model. The performance of alignment might not only be affected by the algorithm itself, but also varies by using different levels of detailed point cloud information. The systematic error and random error can usually be differentiated by measuring the sample multiple times and then using Analysis of Variance (ANOVA) or Average and Range method. However, there is a gap in existing research regarding how best to identify the alignment error and distinguish three error terms by an instant scan during in-situ monitoring. In this work, a detailed review is provided to summarize the existing deviation modeling and error quantification methods in additive manufacturing along with common point cloud alignment methodologies.

In addition, after resolving the impact of alignment and random errors, the change point of systematic error distribution is desired to be identified as early as possible so that engineers can be informed to resolve the errors from the change point. Change point is the time instant when the distribution of a random process changes. In the in-situ inspection and monitoring of AM processes, the systematic error calculated based on the multi-resolution framework provides real-time information about when (at which layer) a systematic deviation occurs. This problem can be solved via quickest change detection (QCD), which is often applied in detecting changes in a time series as quickly as possible under a probability of false alarm (PFA) constraint. Usually, the distribution before the change point can be obtained based on the normal operating condition of the system. However, the post-change distribution might not be readily available due to the unexpected nature of the fabrication variations. Existing QCD methods can be roughly classified into Bayesian and Minimax (non-Bayesian) methods. The Bayesian procedures are applied when the prior knowledge of the change point is available. A well-known procedure is the Shiryaev method (Shiryaev, 1963) which aims at minimizing the average detection delay (ADD) subject to an upper bound on the probability of failure (POF). If the prior probability of change point is unknown, the non-Bayesian methods, such as the cumulative sum (CUSUM) (Page, 1954) method and the Shiryaev-Roberts (SR) procedure (Roberts, 1966), are applied to minimize the delay considering

a lower bound of Average Run Length (ARL). The limitation of these works is that only a binary hypothesis test (single pre-change and post-change model) is considered. In Lai (1998) and Tartakovsky (2004), methods of the non-Bayesian QCD formulation with multiple post-change models have been discussed. However, those studies on QCD did not consider the objective of ADD for QCD problems with multiple post-change models under a non-Bayesian setting. In this work, a QCD method for minimizing the ADD (Nath and Wu, 2018) under a non-Bayesian setting is integrated with the multi-resolution framework to accomplish a more comprehensive online inspection scheme for AM processes.

Regarding methodological contributions, this work proposes a multi-resolution framework for automatically aligning a layer to its nominal shape with different levels of details and distinguishing the errors due to fabrication, measurement, and alignment. In addition, this research is among the first attempts that provide an efficient and effective solution to find the change point (layer) in layer-by-layer inspection of an AM process with regard to in-plane systematic deviation. Shapes are modeled using a polar coordinate representation/profile, which is decomposed into different levels of components via wavelet decomposition. Then, an alignment method based on phase correction is conducted on different wavelet components. After alignment, the randomized complete block design (RCBD) approach (Montgomery, 2013) is applied to quantify the manufacturing (systematic) error, alignment error, and random error based on the deviation profile. In addition, a two-step optimization model is proposed to optimize the efficiency and effectiveness of the proposed framework. This work mainly focuses on illustrating the multi-resolution framework; therefore, the procedure of datum-based alignment, which has been elaborated upon in Jin et al. (2019), will not be repeated in this work. Three layers of a 3D printed human heart are considered as examples to demonstrate the effectiveness of proposed methods. The first two layers are designed to show the procedure of newly proposed multi-resolution framework. The third layer is chosen as an example for showing capability of the proposed method with datum-based alignment to deal with complex layers which have internal or concave features.

3.2 Literature Review

3.2.1 Point Cloud Alignment

The goal of an alignment or registration process is to transform a model from the internal coordinate system of a measurement device into a global coordinate system that the user desires. Usually, it keeps the reference (nominal) model fixed and lets the source point cloud obtained from a 3D scanner to best match the reference by certain transformation. Existing registration algorithms can be classified coarsely into rigid and non-rigid approaches. Rigid approaches assume a rigid environment such that the transformation can be modeled using only 6 degrees of freedom (DOF). Non-rigid methods, on the other hand, allow for a higher number of DOF in order to cope with non-linear or partial stretching or shrinking of an object. Because alignment is a pre-processing step in quality inspection, rigid approaches are often preferred for aligning the scanned point cloud to the reference model. Detailed reviews of rigid registration are provided by Bellekens et al. (2014), Bellekens et al. (2015), Tam et al. (2013), and Markelj et al. (2012). In the literature, most applications employ either a simple Singular Value Decomposition (SVD) or Principal Component Analysis (PCA) for registration, or use a more advanced iterative scheme based on the Iterative Closest Point (ICP) algorithm. In general, most rigid alignment methods, including ICP, rely on least-squares fitting algorithms to find the optimal rigid transformation. In ICP, the error of alignment (or the distance between the two point clouds) is minimized by iteratively revising the transformation matrix that aligns the two point clouds.

Unfortunately, limited work has been done on how these registration methods work under GD&T specifications. Some commercial software/hardware providers (e.g., BuildIT Construction, 2019) offer alignment options based on a given feature/datum, which is manually specified by users. Since specific alignment algorithms used by a particular software/hardware provider are usually confidential, useful information is difficult to obtain online. Indeed, the most popular alignment options include the best fit alignment, feature-based alignment, and degrees-of-freedom alignment (Woodward et al., 2018). Among the three, the feature-based alignment can

align two point clouds based on some important point sets specified by users, but it requires a global least-squares fitting before performing feature-based local adjustment. In practice, it is possible that defects on the sample part might deteriorate the alignment accuracy and cause point cloud misalignment, which might impact the accuracy of local adjustment and establish an even worse alignment.

The alignment methods proposed in this work are motivated by the phase correction approach (Wolberg et al., 2000) for image registration. The main idea of phase correction is to find the phase shift by representing the original image in the polar coordinate system. The proposed methods have been initially studied by Jin et al. (2019), which can directly align the reference shape to the sample point cloud without conducting a rough pre-alignment or manual intervention. In this work, the alignment idea is advanced by using a multi-resolution alignment methodology based on the wavelet components to verify the alignment consistency under different levels of resolution. The wavelet decomposition is a common technique that applied in signal and image processing to investigate the information with different level of details. For point cloud data, this technique is usually applied for point cloud denoising (Huang et al., 2009, Rosman et al., 2013), and surface decomposition/ reconstruction (Manson, et al., 2008, Chen et al., 2011, Jakovljevic et al., 2015). Torre-Ferrero et al. (2008) proposed a 3D registration method based on wavelet decomposition. However, the wavelet decomposition is applied on the circon image, which is a shape presentation for 3D point cloud, rather than on the point cloud directly. Also, it did not consider the alignment at different resolutions. To the best of our knowledge, this is the first attempt in solving such complex alignment problems under a multi-resolution framework to hierarchically evaluate the alignment performance and analyze the variation from different sources.

3.2.2 Deviation Modeling and Quantification in AM processes

In previous studies, most shape deviation modeling approaches are proposed under the context of design compensation by either changing the CAD, STL, or slicing techniques.

Tong et al. (2003) and Tong et al. (2008) proposed parametric error models to map the effects

of all errors in STL and FDM processes into “virtual” parameters without explicit meanings. The coefficients of the models that are considered as a global measure of accuracy can be estimated through regression of the measurement data represented in a Cartesian Coordinate System (CCS). In a different direction, a series of works (Huang et al., 2014, Song et al., 2014, Huang et al., 2015, Sabbaghi et al., 2016, Wang et al., 2017) on predictive modeling and compensation have been conducted by modeling the deviation in a Polar Coordinate System (PCS). In this representation, the Cartesian coordinates (x, y) of a manufactured shape are transformed into polar coordinates $(\theta, r(\theta))$, where θ is the angle of a point as defined with respect to coordinate axes printed on the product, and $r(\theta)$ is the observed radius for θ . The desired shape is represented by the nominal radius set $r_0(\cdot)$. The deviation at $\theta \in [0, 2\pi]$ is then defined as

$$\Delta r(\theta, r_0(\cdot)) = r(\theta) - r_0(\theta) \quad (3.1)$$

This deviation modeling approach is adopted in this work. In particular, the shape profile represented with the polar coordinates are considered as an observed “signal” of each layer for wavelet analysis.

Different from modeling error based on experimental data, other kinds of error modeling approaches in AM (e.g., Navangul et al., 2013; Zha et al., 2015) are proposed to model the chordal error from the tessellation, which is computed as the distance between the STL facet and the CAD surface. In medical applications (Arrieta et al., 2012, Pinto et al., 2015), the geometric error is evaluated both globally using volumetric intersection indexes calculated over segmented Computed Tomography scans and locally by applying a 3D surface map with a color code that allows for differentiating regions for which the model is overestimated, underestimated, or correctly estimated.

In contrast, relatively few studies have been conducted on how to distinguish the errors from different sources. In Song et al. (2014), the errors are categorized into the extruder positioning error before material deposition and the processing error occurred during the printing process or

after material deposition. Zhu et al. (2017) proposed a deviation modeling approach from a Design for AM (DfAM) perspective. In their work, the systematic deviation is considered as the in-plane deviation between the actual printed shape and the nominal shape of a 2D layer, and the random errors are generated by translating the contour points with a given distance derived from a random field theory. However, none of these studies have considered how to differentiate the systematic, alignment, and random errors from the deviation profile generated by a single scan of a layer. The methodological contribution of this work is to propose a novel multi-resolution alignment and inspection framework to fill this research gap.

3.3 Method

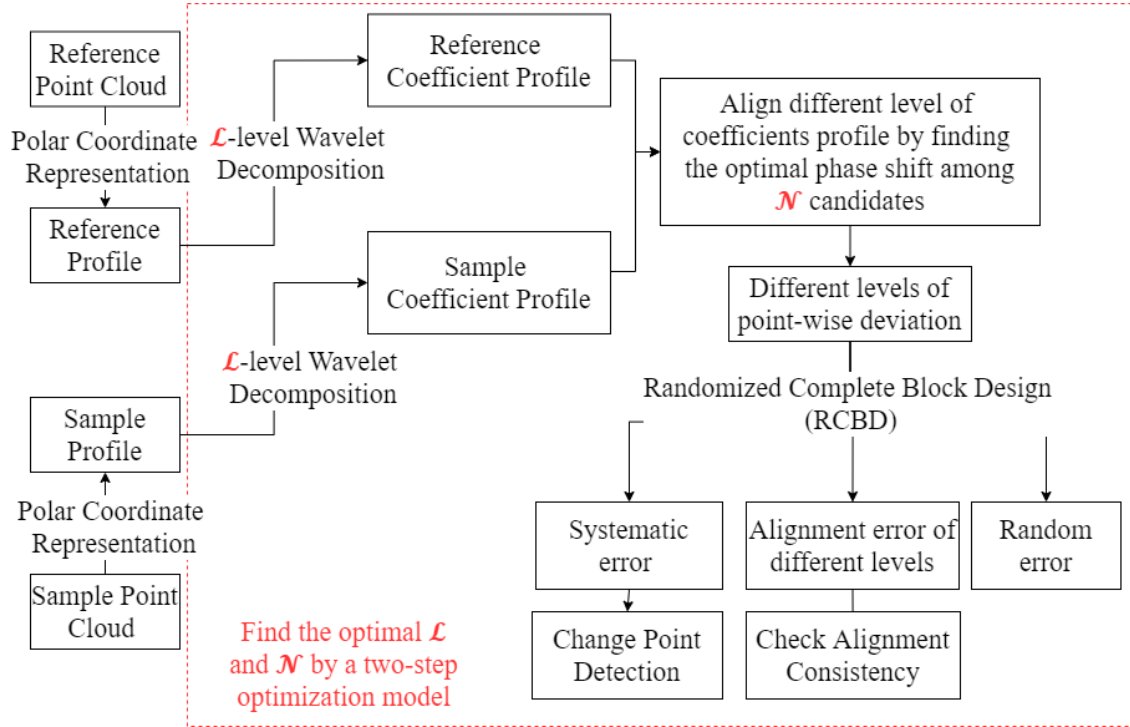


Figure 3.1: An overview of the automatic multi-resolution alignment and inspection framework.

The proposed automatic multi-resolution alignment and inspection framework is illustrated in Figure 3.1. The sample point cloud is obtained by 3D scanning while the reference (nominal) point cloud is generated from the STL file. First, both the sample point cloud and reference (nominal) point cloud of a given layer are represented in a shape profile with polar coordinates. Then, \mathcal{L} -

level wavelet decomposition is conducted on the two profiles, respectively, to obtain \mathcal{L} levels of detail for each profile. Next, each level of detail coefficients of the sample profile is aligned to the corresponding detail coefficients of the reference profile by finding the optimal phase shift among the \mathcal{N} best candidates. Note that the proposed alignment methods are proposed based on the assumption that the sample point cloud and reference point cloud have already been aligned along the z-axis. The randomized complete block design (RCBD) is then conducted on the point-wise deviation calculated based on different levels of alignment results to distinguish the errors from different sources. To improve the alignment effectiveness and efficiency, a two-step optimization model is proposed to find the optimal level of wavelet decomposition and the maximum number of candidates to be used in the alignment process. Based on the systematic error obtained from RCBD, a QCD method is implemented to identify the layer where the earliest systematic error distribution changes during the printing process. The details of these steps will be explained next.

3.3.1 Shape Profile

A shape profile is a piecewise polynomial function, $r(\theta)$, estimated based on the polar coordinates (Jin et al., 2019). Given a point (x, y) from the point cloud, the corresponding polar coordinates (θ, r) can be expressed as

$$\begin{aligned}\theta &= \text{atan2}(y - y_0, x - x_0) \\ r &= \sqrt{(x - x_0)^2 + (y - y_0)^2}\end{aligned}\tag{3.2}$$

where (x_0, y_0) is the centroid of the 2D shape. The angle θ of a point should be within $[-\pi, \pi]$, and its corresponding radius should be normalized into a range of $[0, 1]$ to eliminate scaling effects. Based on the angles and radius obtained from the point clouds, the corresponding piecewise polynomial function can be estimated. An example is provided in Figure 3.2 to show the representations of a 2D point cloud in the CCS, PCS, and shape profile. In order to establish future point-wise correspondence between the reference point cloud and sample point cloud, the same query set Θ is applied on both reference and sample profiles to generate the corresponding radii. For any $\theta \in \Theta$, the reference radius is denoted as $r_0(\theta)$ and the observed radius in the sample is

denoted as $r(\theta)$. The discretized shape profiles using the query set Θ are denoted as \mathbf{r}_j^0 and \mathbf{r}_j for the reference model and sample model, respectively. The reason for applying the same query set is to ensure same number of points are sampled with the same resolution in both sample shape profile and reference shape profile.

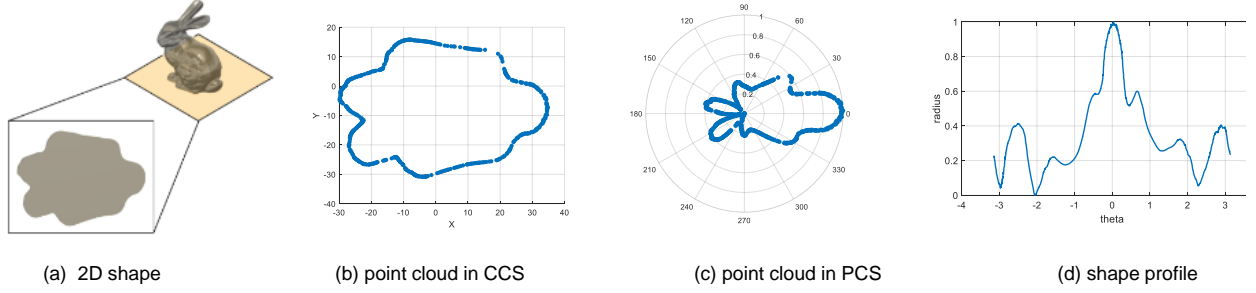


Figure 3.2: An example of shape representation in CCS, PCS, and shape profile.

3.3.2 Wavelet Decomposition

Wavelet decomposition (Mallat, 1989) is a common technique usually applied in signal or image processing to learn the details of information at different resolutions. Like a signal or image, point cloud data also contains local variations at different spatial frequencies. With a multi-resolution representation using wavelet decomposition, the shape profile can be interpreted hierarchically with different levels of detail. In this work, a shape profile can be considered as a “signal”, in which angle is analogous to time, and radius is analogous to amplitude. Therefore, the original shape profile \mathbf{r}_j , as shown in Figure 3.3, can be decomposed into an approximation coefficient vector \mathbf{r}_{j-1} and a detail coefficient vector \mathbf{d}_{j-1} by down-sampling the signal after passing through a high-pass filter HiD and a low-pass filter LoD, respectively. The approximation coefficients are a lower resolution representation of the original profile, and the detail coefficients are defined as the difference of information between the original profile and approximation at resolution $j - 1$.

Technically, the two wavelet components can be obtained by

$$r_j(\theta) = \sum_{k \in \mathbb{Z}} r_{j-1}(k) \phi_{j,k}(\theta - k) + \sum_{k \in \mathbb{Z}} d_{j-1}(k) \psi_{j,k}(\theta) \quad (3.3)$$

$$r_{j-1}(k) = \langle r_j(\theta) \phi_{j,k}(\theta) \rangle, \quad d_{j-1}(k) = \langle r_j(\theta), \psi_{j,k}(\theta) \rangle$$

where $\phi_{j,k}(\theta - k)$ is the scaling function, and $\psi_{j,k}(\theta)$ is the wavelet function. The approximation r_{j-1} can be further decomposed into two lower level components by following the similar procedure. It is worth noting that the index j or $j - 1$ denotes only the detail level the coefficient vector contains. For instance, if 4-level wavelet decomposition is performed on a shape profile r_j , then the level 1 to level 4 detail coefficient vectors are denoted as d_{j-1} , d_{j-2} , d_{j-3} , and d_{j-4} .

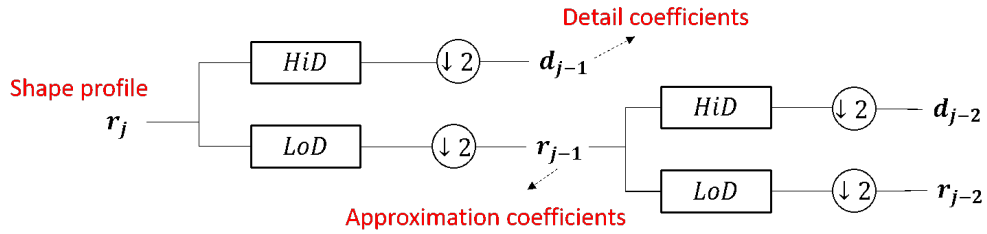


Figure 3.3: Wavelet decomposition of shape profile.

Considering the tradeoff between the alignment performance and computational efficiency, both wavelet levels and wavelet functions need to be optimized. In practice, the maximum level of wavelet decomposition can be determined using

$$\mathcal{L}_{\max} = \lfloor \log_2 \left(\frac{n}{n_w} - 1 \right) \rfloor \quad (3.4)$$

where n is the length of the signal, n_w gives the length of the decomposition filter associated with the selected mother wavelet. It is worth pointing out that the optimal $\mathcal{L} \in \{1, 2, \dots, \mathcal{L}_{\max}\}$ can be obtained using an optimization model illustrated in Section 3.3.5.

3.3.3 Multi-Resolution Alignment based on Coefficient Profile

The detail coefficient vector obtained from the wavelet decomposition is named as the coefficient profile. If a \mathcal{L} -level wavelet decomposition is performed on the original profile, there are \mathcal{L} coefficient profiles to represent the original profile with different spatial frequencies. Let \mathbf{d}_{j-l}^0 denotes the nominal coefficient profile while \mathbf{d}_{j-l} represents the sample coefficient profile at l^{th} level of resolution. The main idea of alignment based on the coefficient profile is to find the optimal phase shift for the sample coefficient profile \mathbf{d}_{j-l} in the corresponding reference coefficient profile \mathbf{d}_{j-l}^0 at each level l , $l \in \{1, 2, \dots, \mathcal{L}\}$. The alignment based on the sample and reference coefficient profiles at different resolutions can be accomplished in a three-step procedure.

First, for each level of sample coefficient profile \mathbf{d}_{j-l} , find \mathcal{N} best matches in its corresponding extended reference coefficients profile $\mathbf{d}_{j-l}^{'0}$ by similarity search (Hetland, 2009), and calculate the corresponding \mathcal{N} phase shifts. The phase shift at level l based on the i^{th} best match is denoted as $\Delta\theta_{li}$, $l \in \{1, 2, \dots, \mathcal{L}\}$ and $i \in \{1, 2, \dots, \mathcal{N}\}$. It is worth noting that the reference coefficient profile is extended with one cycle on the left and one cycle on the right to ensure that at least one best match can be found for the sample coefficients profile. An example of finding 5 best matches in the 2nd level of the sample coefficient profile of the example shape from Figure 3.2 is provided in Figure 3.4. The horizontal axis shown in Figure 3.4 represent the indices of θ s. The corresponding θ values can be obtained from the query set with a given index. Each of the 5 matches has a phase shift, which indicates how much the sample coefficient profile needs to be moved along the θ axis. A positive θ indicates a movement to the right, while a negative value indicates a shift to the left.

Second, for each level l , find the optimal phase shift $\Delta\theta_l^*$ among all the candidates. Specifically, each $\Delta\theta_{li} \in \Delta\theta(l \cdot)$ is evaluated by calculating the total absolute point-wise deviations based on the aligned sample and reference shape profiles using $\Delta\theta_{li}$. The optimal phase shift $\Delta\theta_l^*$ at the l^{th} level of resolution is determined by

$$\Delta\theta_i^* = \arg \min_{j \in \{1, 2, \dots, \mathcal{N}\}} \sum_{k=1}^n |\Delta r^{(l,j)}(\theta_k)| \quad for \quad \forall i \in \{1, 2, \dots, \mathcal{L}\} \quad (3.5)$$

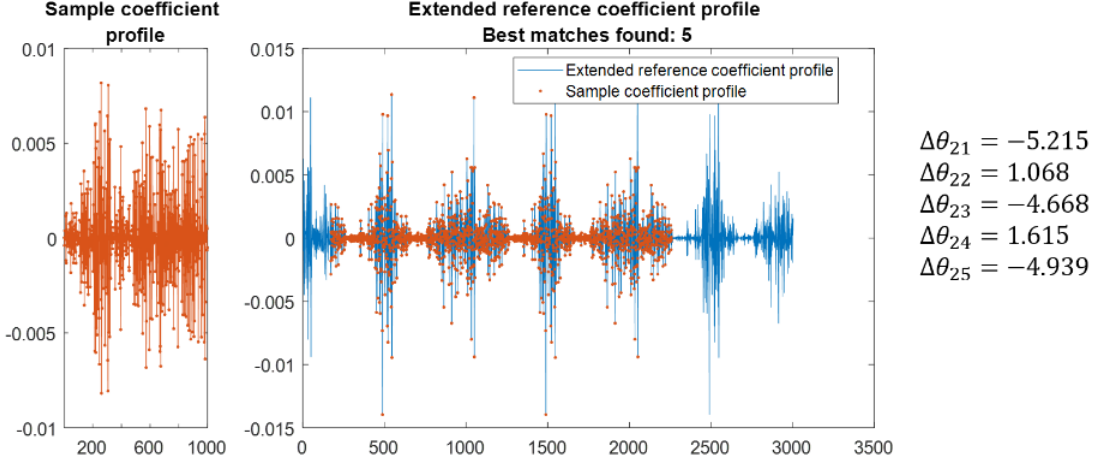


Figure 3.4: An example of finding best matches for the 2nd level of the sample coefficient profile in the extended reference coefficient profile.

where $\Delta r^{(l,j)}(\theta_k)$ is the deviation at angle $\theta_k \in \Theta$ based on the phase shift $\Delta\theta_{li}$, and n is the length of the query set Θ . Continuing with the example shown in the first step, the evaluation of each candidate shift in $\Delta\theta_2$ is shown in Figure 3.5 along with the aligned shape profiles. Among all shift values in $\Delta\theta_2$, $\Delta\theta_{23}$ and $\Delta\theta_{24}$ achieve the smallest error among the 5 candidates. In this case, we can arbitrarily select either $\Delta\theta_{23}$ or $\Delta\theta_{24}$ as the optimal phase shift for the 2nd level of alignment.

Third, align the l^{th} level of coefficient profiles $(\mathbf{d}_{j-l}^0, \mathbf{d}_{j-l})$ and original shape profiles $(\mathbf{r}_j^0, \mathbf{r}_j)$ using the optimal phase shift $\Delta\theta_l^*$. Based on the optimal phase shift of each level obtained from the previous step, the 4-level multi-resolution alignment result of the example shape is shown in Figure 3.6.

Similar alignment procedures can be conducted based only on the segments associated with datum features or a collection of user-defined features. The detail steps and applications are elaborated upon in Jin et al. (2019).

3.3.4 Deviation quantification

According to the alignment shape profiles at each level l , the deviation at each point (angle) k can be calculated and denoted as v_{lk} . One challenge of deviation quantification using the proposed

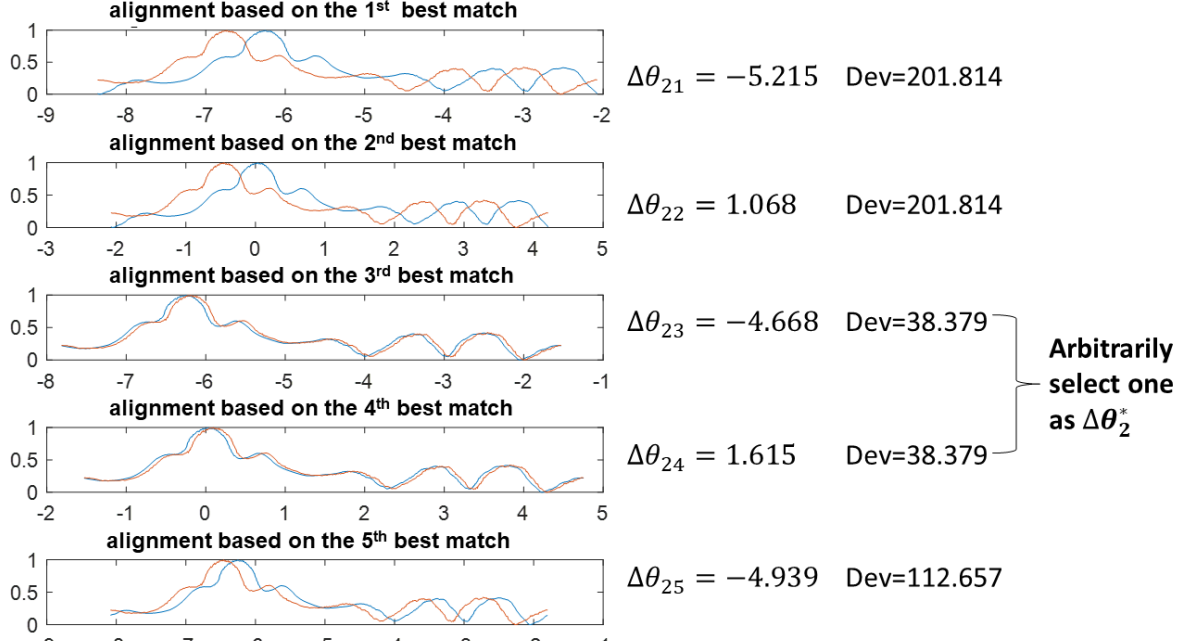


Figure 3.5: Find the optimal phase shift for the 2nd level of the coefficient profile.

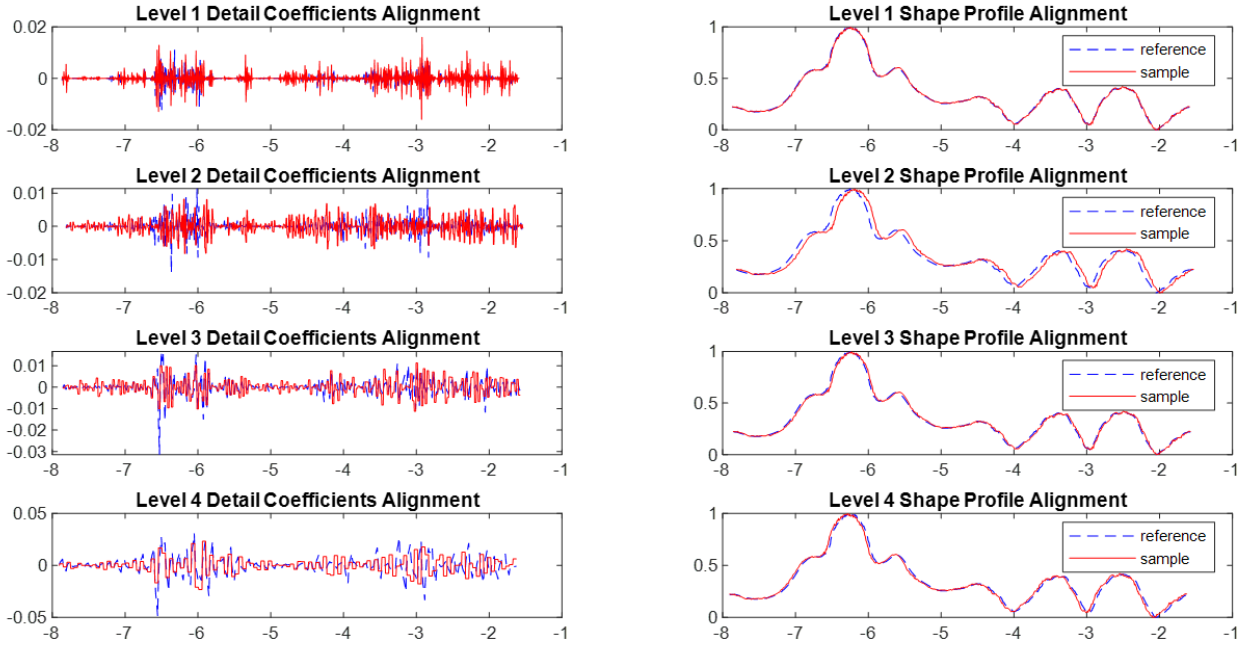


Figure 3.6: Multi-resolution alignment of detail coefficient profiles and shape profiles.

multi-resolution alignment approach is in checking the consistency of quantification results at different levels. In particular, it is likely that the values of deviation at different locations (i.e., angles) are quite different. To resolve this issue by “noise reduction” (Montgomery, 2013), RCBD is ap-

plied on the point-wise deviation matrix to learn the sources of variance. In this work, each level of alignment is considered as a treatment while each sample point (angle) is considered as a block as demonstrated in Table 3.1. Given the point cloud with size n , m points are sampled from the n points. The m sample points can be points associated with the critical features, or can be obtained via uniform sampling to eliminate bias.

Table 3.1: Point-wise Deviation Based on the Multi-resolution Alignment

		Block 1	Block 2	Block 3	Block m	
Sample point Level		1	2	3	m	
Treatment 1	1	v_{11}	v_{12}	v_{13}	v_{1m}	$\mu_{1\cdot}^v$
Treatment 2	2	v_{21}	v_{22}	v_{23}	v_{2m}	$\mu_{2\cdot}^v$
Treatment 3	3	v_{31}	v_{32}	v_{33}	v_{3m}	$\mu_{3\cdot}^v$

Treatment \mathcal{L}	\mathcal{L}						$\mu_{\mathcal{L}\cdot}^v$
		$\mu_{\cdot 1}^v$	$\mu_{\cdot 2}^v$	$\mu_{\cdot 3}^v$	$\mu_{\cdot m}^v$		$\mu_{\cdot\cdot}^v$

To check the consistency of alignment at different resolutions, a hypothesis test shown in Eq. 3.6 is conducted on the mean values of \mathcal{L} treatments (rows), $\mu_{1\cdot}^v, \mu_{2\cdot}^v, \dots, \mu_{\mathcal{L}\cdot}^v$.

$$\begin{aligned}
 H_0 : \quad & \mu_{1\cdot}^v = \mu_{2\cdot}^v = \dots = \mu_{\mathcal{L}\cdot}^v \\
 H_1 : \quad & \text{at least one "equal" is not satisfied}
 \end{aligned} \tag{3.6}$$

If the alignment results of different levels are consistent, a p-value greater than the pre-determined significant level should be observed (*i.e.*, fail to reject the null hypothesis). Inconsistency of alignment would indicate that the deviation measured in Table 3.1 not only results from the systematic or measurement error but from the alignment. Therefore, more candidates (best matches) should be evaluated in the alignment procedure. The stop criterion of stopping adding candidates is elaborated in section 3.3.5. If the inconsistency still exist after the stop criterion is satisfied, then it is

high likely that the defects on the layer has affected the alignment accuracy. As a result, a large systematic error and alignment error might be observed in the result.

Moreover, by following the procedure of ANOVA, the general systematic error, the alignment error at the i^{th} level, and random error at each sample point can be calculated:

$$\text{Systematic error: } e_s = \mu_{..}^v = \left(\sum_{i=1}^{\mathcal{L}} \sum_{j=1}^m v_{ij} \right) / \mathcal{L}m. \quad (3.7)$$

$$\text{Alignment error for level } i : e_a^{(i)} = \mu_{i.}^v - \mu_{..}^v, \quad i \in \{1, 2, \dots, \mathcal{L}\}. \quad (3.8)$$

$$\text{Random error for each point of level } i \text{ at block } j : e_r^{(i,j)} = v_{ij} - \mu_{i.}^v, \quad i \in \{1, 2, \dots, \mathcal{L}\}, \quad j \in \{1, 2, \dots, m\} \quad (3.9)$$

3.3.5 Optimal setting for point cloud alignment

As illustrated in section 3.3.3, the performance of alignment can be improved by searching the optimal level of wavelet decomposition \mathcal{L} and the maximum number of best phase shifts \mathcal{N} when aligning the sample and reference coefficient profiles. Given \mathcal{N} and \mathcal{L} are values, the optimization model of the first step for determining the best phase shift at each level can be expressed as

$$z_1(\mathcal{N}, \mathcal{L}) = \min \sum_{l \leq \mathcal{L}} \sum_{i \leq \mathcal{N}} \left(\sum_{\theta \in \Theta_s}^m \Delta r^{(l,i)}(\theta_k) \right) x_{li} \quad (3.10)$$

$$s.t. \quad \sum_i x_{li} = 1, \quad \forall l \in \{1, 2, \dots, \mathcal{L}\} \quad (3.11)$$

$$x_{li} \in \{0, 1\} \quad \forall l \in \{1, 2, \dots, \mathcal{L}\}, \forall i \in \{1, 2, \dots, \mathcal{N}\} \quad (3.12)$$

where x_{li} is the decision variable ($x_{li} = 1$, if the l^{th} level of decomposition uses i^{th} match; $x_{li} = 0$, otherwise). The total point-wise deviation $\sum_{\theta \in \Theta_s} \Delta r^{(i,j)}(\theta)$ depends on the sample points set Θ_s . With different sampling strategies, the objective value may vary. The second-step optimization is to minimize the sum of the objective value from the first step and the computational cost to

determine the optimal values of \mathcal{L} and \mathcal{N} .

$$\min \quad \alpha z_1(\mathcal{N}, \mathcal{L}) + \beta(\mathcal{N}\tau + \mathcal{L}\omega) \quad (3.13)$$

$$s.t. \quad \mathcal{N}, \mathcal{L} \in \mathbb{Z}^+, 3 \leq \mathcal{L} \leq \mathcal{L}_{\max} \quad (3.14)$$

Note that the decision variables \mathcal{N} and \mathcal{L} are integers, where \mathcal{L} should at least be 3 to accomplish the multi-resolution framework and be less than \mathcal{L}_{\max} given by Eq 3.4. The unit computation costs for wavelet decomposition and alignment are denoted as ω and τ , respectively. The penalty coefficients α and β should be adjusted by considering the magnitude of deviation. In this work, the above two-step optimization model is solved by a heuristic algorithm. For each pair of values of \mathcal{N} and \mathcal{L} , the integer programming problem $z_1(\mathcal{N}, \mathcal{L})$ is solved directly. If the improvement of the objective value is less than the defined threshold δ , the algorithm stops and reports the solution; otherwise, another iteration is performed.

Algorithm 1:

Input: A sequence of $\mathcal{N}_k \in \mathbb{Z}^+$ and $\mathcal{L}_k \in \mathbb{Z}^+, 3 \leq \mathcal{L}_k \leq \mathcal{L}_{\max}$, stop criterion ζ , which is equal to the improvement of the objective value ;

- 1 $k \leftarrow 0$;
- 2 **repeat**
- 3 Generate sample query set Θ_s and calculate corresponding $\Delta r^{(l,i)}(\theta)$ for $\theta \in \Theta_s$;
- 4 Solve the integer programming problem $z_1(\mathcal{N}_k, \mathcal{L}_k)$ and find the best solution x_{li} ;
- 5 $k \leftarrow k + 1$;

until ζ is not satisfied;

3.3.6 Quickest Change Point Detection

Given a specific decision rule δ for determining the change point, the QCD problem based on the systematic error can be formulated as:

$$\min ADD(\delta)$$

$$s.t. PFA(\delta) \leq \alpha \quad (3.15)$$

where $PFA(\delta) = P(\hat{l} < l | \mathbf{R}_n^X)$ and $ADD(\delta) = E[|\hat{l} - l| > l]$. Here we denote the true change point (layer) as l and the estimated change point as \hat{l} based on the decision rule δ . The mapping from an observation to the change is denoted as \mathbf{R}_n^X , for which $\delta(\mathbf{R}_n^X) = \hat{l}$.

Let $\mathbf{X}_n, n = 1, 2, \dots$ be a sequence of observed systematic errors at moment n until the printing stops. Then, the probability density function (pdf) before ($n < l$) is denoted as $f_{0,n}(X_n | X^{1:n-1})$, and let the pdf after the change point ($n > l$) be $f_{i,n}(X_n | X^{1:n-1})$ based on the i^{th} post-change model where $i = 1, 2, \dots, M$ with $M < \infty$. A typical change point scenario is demonstrated in Figure 4. At any moment n , a decision is made based on the following hypothesis test:

$$\begin{aligned} H_0 : l < n \\ H_1 : l \geq n \end{aligned} \quad (3.16)$$

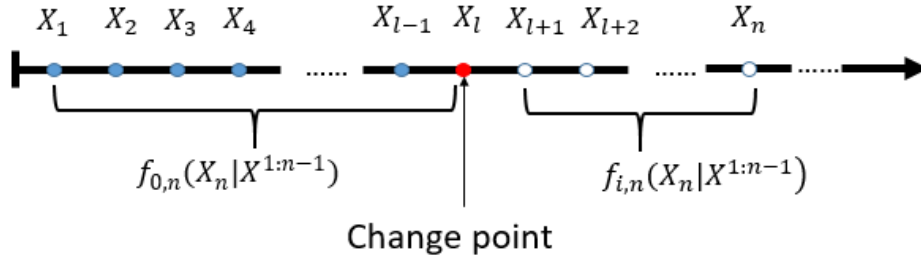


Figure 3.7: Demonstration of change point in a time series.

In practice, due to unpredictable variations in fabrication, both the prior probability of change point $P(l = k) = \pi_k, k = 1, 2, \dots$ and the post-change models are unknown. Based on the derivation in Page (1954), Lai (1998), and Nath and Wu (2018), the change point can be identified according to the following theorem.

Theorem 1. For a given PFA upper bound α , the change point is detected using $\hat{l} = \inf\{n > l : \Lambda(n) \geq \frac{M\bar{l}}{\alpha}\}$, where $\Lambda(n) = \sum_{i=1}^M \Lambda_i(n)$ is the sum of the Shiryaev-Roberts (SR) statistic

$\Lambda_i(n)$, and $\bar{l} = \sum_{k=1}^{\infty} k\pi_k$ is the mean over the prior distribution of change point.

The SR statistic $\Lambda_i(n)$ for detecting a change corresponding to the i^{th} post-change model can be calculated by:

$$\Lambda_i(n) = \sum_{k=1}^n \exp[Z_i^{k:n}] \quad (3.17)$$

where

$$Z_i^{k:n} = \sum_{t=k}^n \log \frac{f_{i,t}(X_t|X^{1:t-1})}{f_{0,t}(X_t|X^{1:t-1})}$$

which can be obtained by the likelihood ratio test on the given hypotheses. The detection rule given in Theorem 1 is an extension of SR procedure, which is adapted to detect a change point with multiple post-change models (Nath and Wu, 2018). Accordingly, the upper bound of false alarm rate is $\frac{\alpha}{\bar{l}}$. Based on the theorem, the mean \bar{l} can be calculated based on the prior distribution of all possible positions $k, k = 1, 2, \dots$. In the non-Bayesian setting, the prior probability of the change point π_k is unknown. However, it can be estimated by:

$$\pi_k = \frac{h_0 X_k}{\sum_{k=1}^n h_0(X_k)} \quad (3.18)$$

where $h_0(X_k)$ is the probability that X_k belongs to the empirical distribution of systematic error measured before moment n under the normal system condition.

A simulated example using a time series of 100 observations following a normal distribution is provided in Figure 3.8 to illustrate the QCD process. It is assumed that the distribution change happens at the 51st observation. The observation before the change point follows the normal distribution $N(0, 0.1^2)$ while the observations after the change point are from the normal distribution $N(0.4, 0.2^2)$. The QCD is performed by considering a single post-change model and multiple post-change models, respectively. The tolerance α on the PFA is set to be 0.05.

In the case with the single post-change model, it is assumed that the post-change model is following $N(0.4, 0.2^2)$. With this single post-change model, $M = 1$ and the detection statistic can be calculated directly from Eq.(3.17). The result of QCD based on the single post-change is demonstrated in Figure 3.9. It is observed that the detection statistic exceeds the threshold,

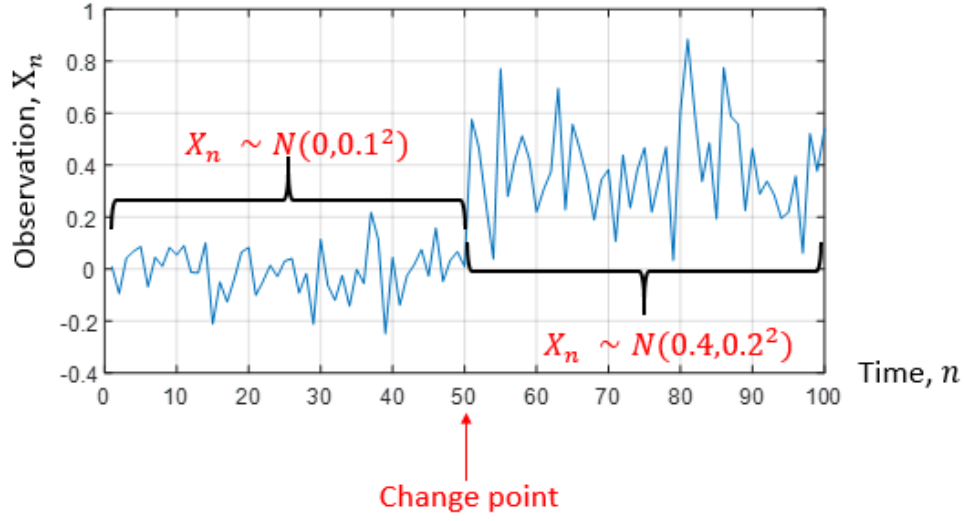


Figure 3.8: Overview of simulation data.

$\frac{M\bar{l}}{\alpha} = \frac{\bar{l}}{\alpha} = 1.004 \times 10^5$, at the 53rd observation. Therefore, in this case, the detection delay is equal to 2.

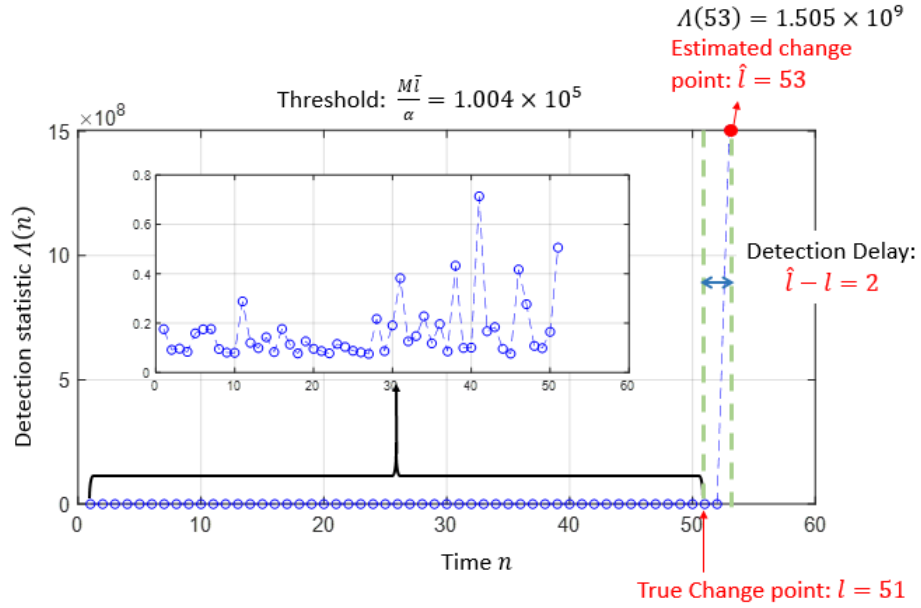


Figure 3.9: The QCD result based on single post-change model.

When considering multiple post-change models ($M=3$) following $N(0.38, 0.22^2)$, $N(0.42, 0.18^2)$ and $N(0.4, 0.2^2)$, a similar detection procedure can be performed based on Theorem 1. Note that

two candidate post-change models are slightly different from the true post-change distribution to simulate the inaccuracy of parameter estimation in practice. The result of considering multiple post-change models is presented in Figure 3.10 that shows the same estimation of change point.

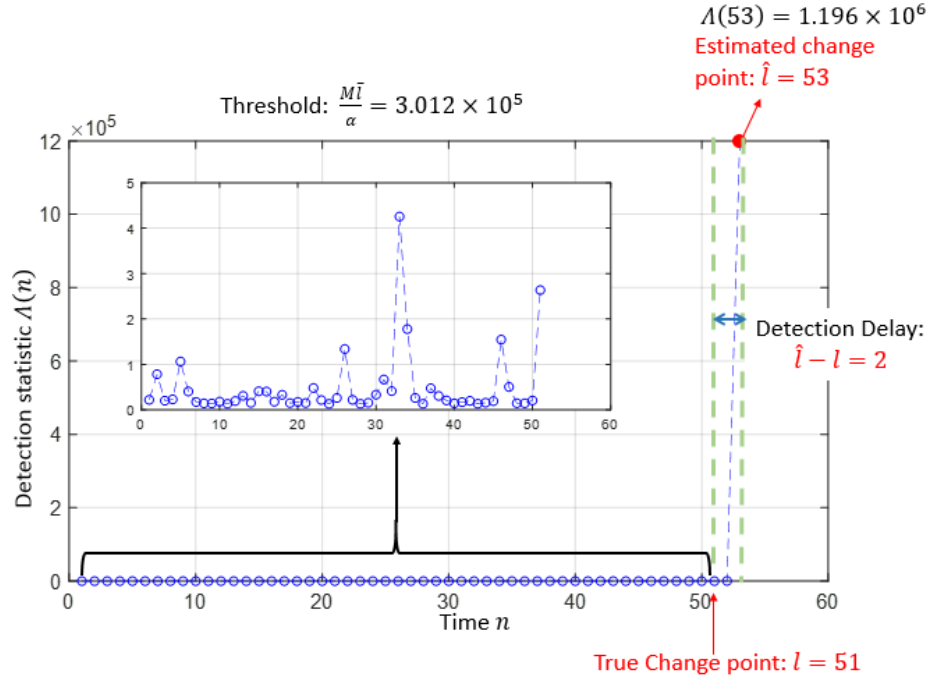


Figure 3.10: The QCD result based on multiple post-change model.

To further investigate the sensitivity of the QCD method, numerical experiments are conducted by considering different distribution parameters for the single post-change model. Specifically, given $N(0, \sigma^2)$ as the distribution before the change point, the data after the change point is generated from $N(1, \sigma^2)$ with $\sigma \in [0.01, 2]$. The ADD is measured and plotted against σ in Figure 3.11 by considering the single post-change model following $N(1, \sigma^2)$ as well. It is observed that the performance of QCD is much more unstable with larger ADD variation as the variance of the data increases. It can be explained by the fact that with a large variance, the distribution before and after the change point cannot be well separated, and therefore, the likelihood ratio test might not work effectively in calculating the SR statistics. The variation of ADD can also be observed when changing the distribution parameters of the post-change model as shown in Figure 3.12.

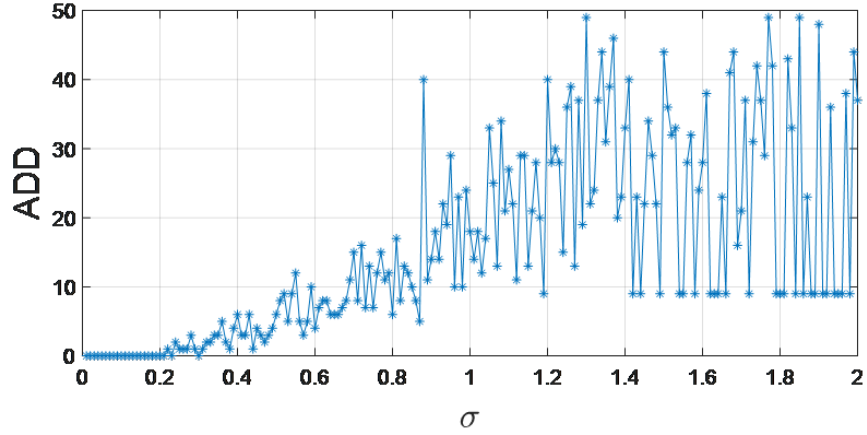


Figure 3.11: The plot of ADD under different variance.

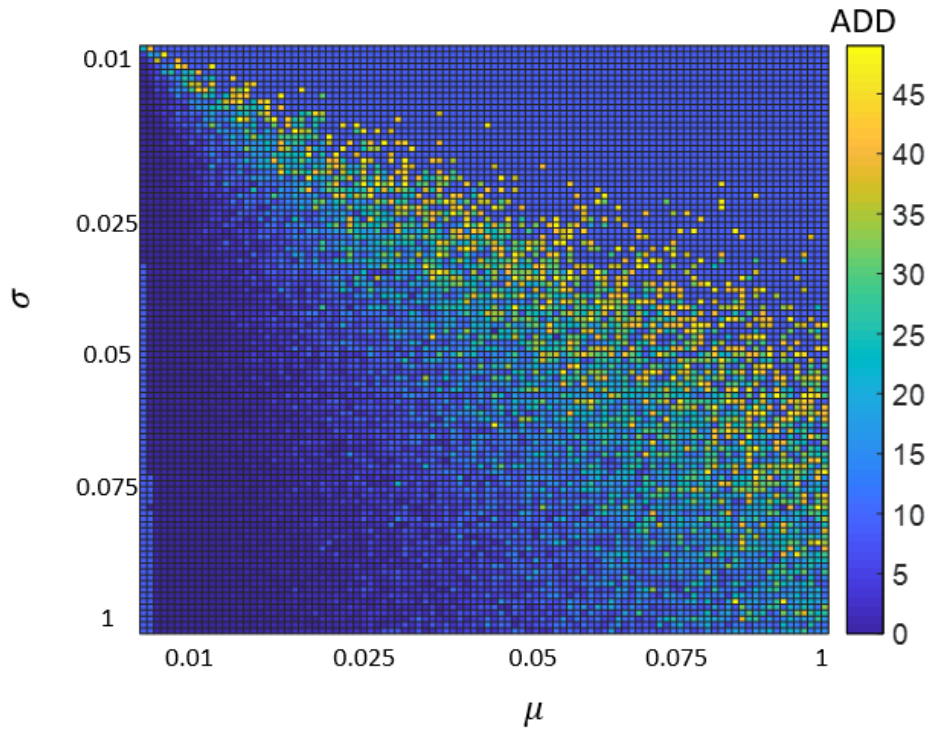


Figure 3.12: Performance of QCD under different post-change distribution $N(\mu, \sigma^2)$.

3.4 Experiment

To validate the proposed framework, a human heart shown in Figure 3.13(a) was printed with MakerGear FFF machine with 1.75mm PLA filament (see Figure 3.13(b)), and scanned by Nikon X TH 225 ST μ CT scanner (see Figure 3.13(c)). The output of CT scanning was a TIFF image stack, which was converted to a point cloud by ImageJ (see Figure 3.13(d)). The proposed multi-

resolution framework is implemented with Matlab on a desktop with Intel core i7 CPU.

In this work, the layer-wise in-plane deviation inspection is simulated by observing X-Y coordinates of a thin slice of the 3D point cloud. It is assumed that the printed sample and the nominal model have been aligned along the z-axis. The layer thickness values of the nominal point cloud and the scanned point cloud are determined based on their scales to ensure that both models have the same number of layers.

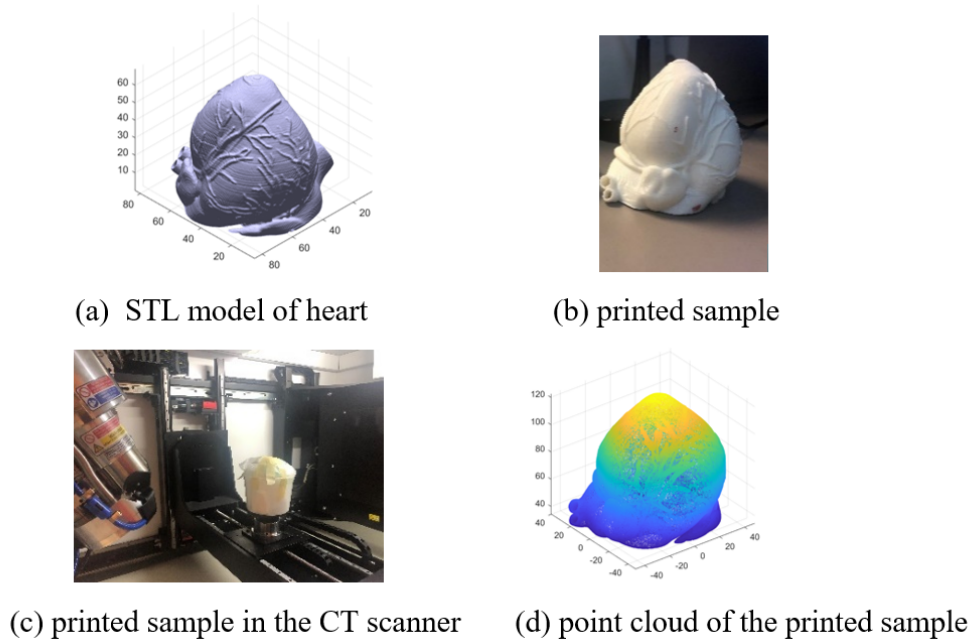


Figure 3.13: Experiment setup.

In the experiment, three sample layers (see Figure 3.14) are to be inspected using the multi-resolution framework. The 1st layer is a simple conforming free-form layer used for comparing the proposed alignment method to the default ICP methods provided in Matlab. The 2nd layer is a defective layer used for demonstrating the effectiveness of the error quantification method. The 3rd layer has complex internal features showing the positions of key tissues of a human heart. This layer is chosen to illustrate how to accommodate our methods on inspecting layers without convexity constraints. Moreover, an additional experiment is conducted on 100 layers from the upper half of the human heart for illustrating the QCD of systematic error.

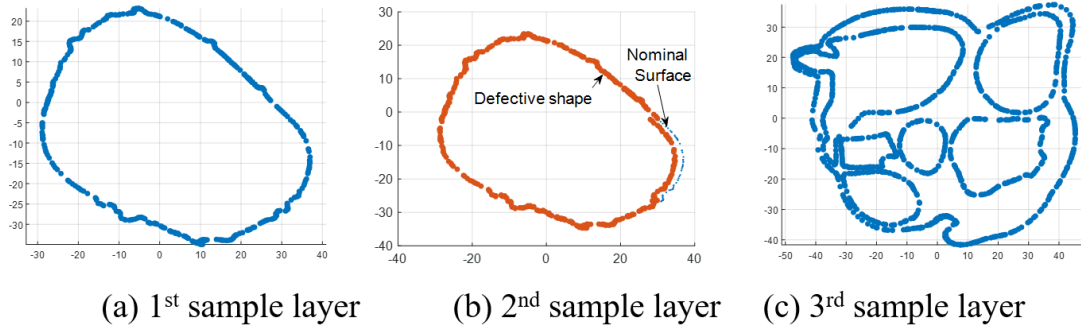


Figure 3.14: Three sample layers.

3.4.1 The 1st layer — a simple layer

The 1st layer is known as a conforming layer, whose shape profile along with its corresponding reference shape profile are shown in Figure 3.15. To achieve better alignment, the alignment parameters, \mathcal{N} and \mathcal{L} , are first optimized using Algorithm 1. The upper bound of wavelet decomposition level \mathcal{L}_{\max} is 8 for this shape profile. The penalty coefficient α of the objective value and β of the computational cost are equal to 2 and 0.1, respectively, in this experiment. The result of the optimization is shown in Figure 3.16, suggesting that 4-level wavelet decomposition should be conducted on the original shape profiles. For each level of alignment, the best phase shift for aligning the sample coefficient profile to the reference coefficient profile is found among 25 candidates. Then, the coefficient profiles, as well as the shape profiles, can be aligned based on the optimal phase shift of each level (see Figure 3.17).

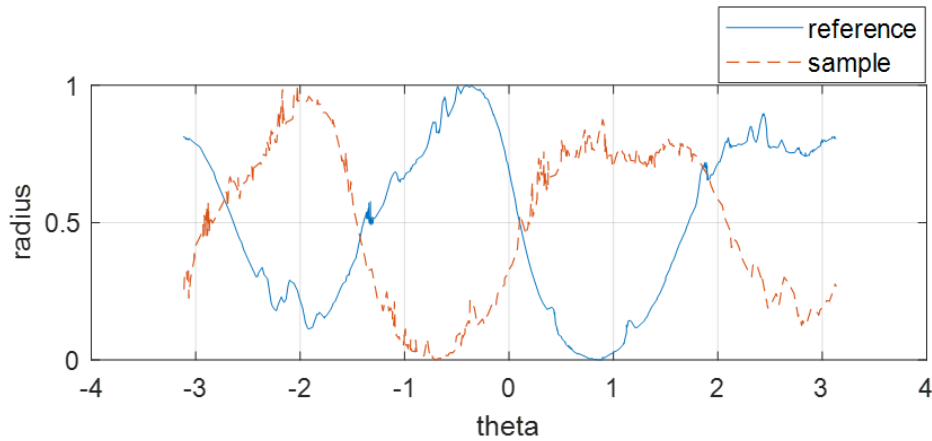


Figure 3.15: Shape profile of the 1st layer.

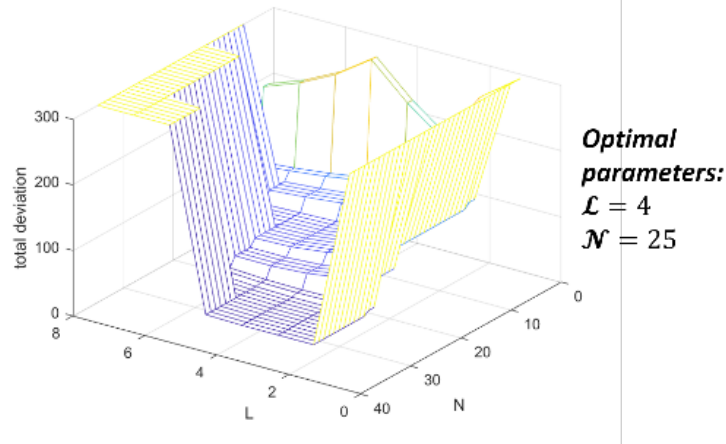


Figure 3.16: Optimization of alignment parameters for the 1st layer.

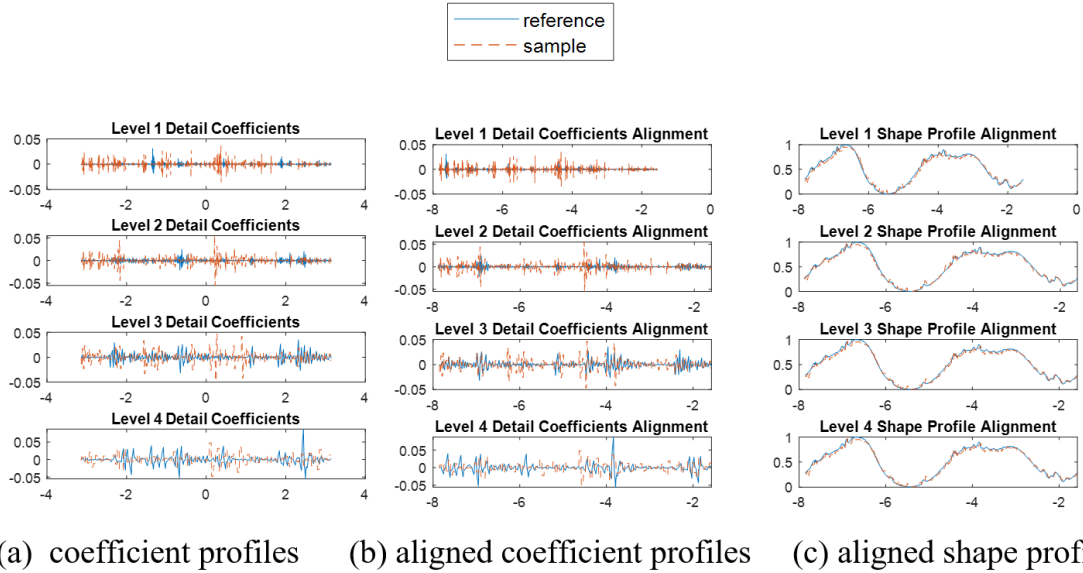


Figure 3.17: 4-level multi-resolution alignment results.

Next, the aligned shape profiles are converted back to the point cloud in CCS, and the result is compared with the alignment of ICP. Figure 3.18 shows that the proposed multi-resolution alignment method results in smaller root mean square errors (RMSE) than ICP at all four levels. Clearly, the proposed alignment method outperforms ICP in aligning the two-dimensional point cloud.

It is also worth mentioning that the proposed alignment method is comparatively more computationally efficient than the iterative alignment algorithms (e.g., ICP). The wavelet decomposition, alignment, and deviation quantification are not time-consuming which would not take longer than

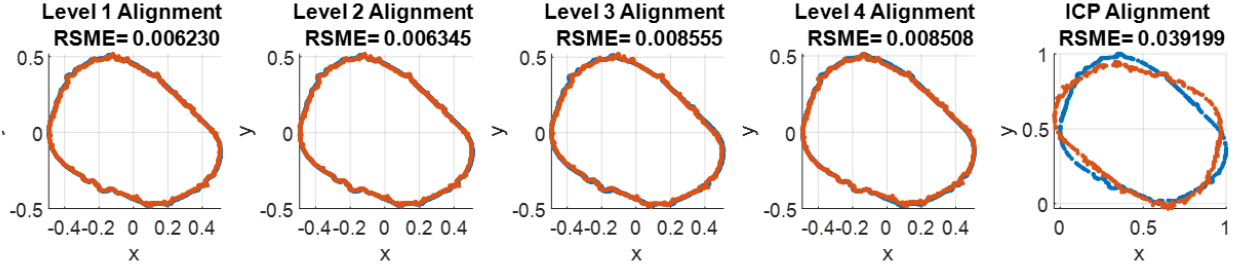


Figure 3.18: Alignment comparison with ICP.

1 second in total. The potential bottleneck occurs at the alignment optimization procedure. As \mathcal{N} increases, the computational time will gradually increase. In this experiment, it cost 2.35 seconds to obtain the optimal alignment at $\mathcal{N} = 25$. The trade-off between the performance and computational time is important, however, not a significant issue in this proposed multi-resolution framework.

3.4.2 The 2nd layer — a defective layer

The 2nd sample layer, as shown in Figure 14, has a defective area, which is created manually for validating the error quantification capability of the proposed method. The optimal alignment parameters for this layer are $\mathcal{N} = 19$ and $\mathcal{L} = 3$. The alignment profiles based on the optimal setting are presented in Figure 3.20.

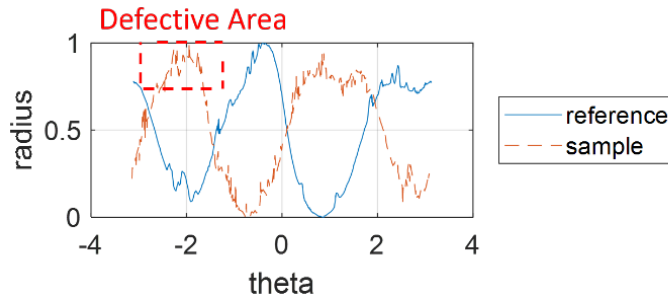


Figure 3.19: Shape profile of the 2nd layer.

The point-wise deviations are then calculated based on the alignment at each level. It is observed in Figure 3.21 that the deviation values are small. To reduce the effect of sample size on the result of ANOVA, only a certain number of points are sampled from the population. Again, each

level of alignment is considered as a treatment and each sample point (θ) is denoted as a block. It is worth mentioning that two sampling strategies can be implemented according to a user-defined inspection requirement.

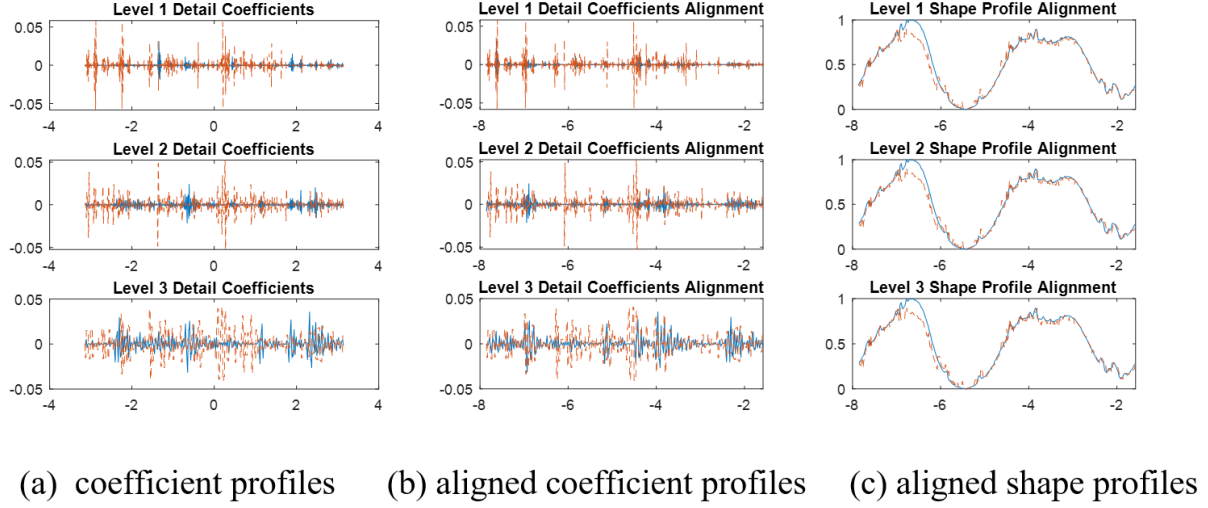


Figure 3.20: 3-level multi-resolution alignment results.

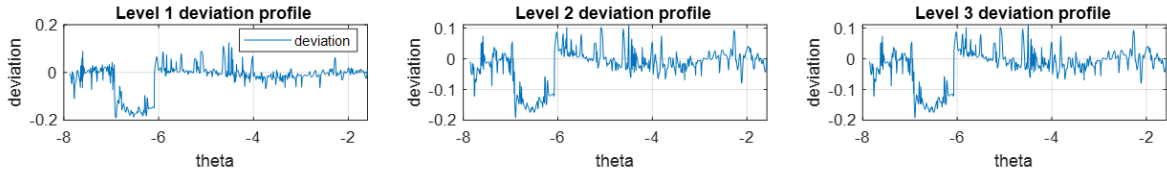


Figure 3.21: Deviation profiles.

If a global inspection or alignment consistency check is required, the sample points can be generated uniformly from the population. An example based on 100 regularly sampled points is provided in Table 3.2 and Figure 16. The result of the hypothesis test on the alignment consistency is indicated in the ANOVA table as shown in Table 3.2. Clearly, the p-value is larger than the significant level 0.05, indicating that different levels of alignment are consistent. Based on Eqs 3.7-3.9, the systematic error, alignment error of each level, and random error at each point are calculated and demonstrated in Figure 3.22. The magnitude of each error term is calculated based on the standardized polar coordinates. Therefore, all the errors should be scaled back to their original domain for interpretation in practice.

Table 3.2: Result of ANOVA

Source	DF	Adj SS	Adj MS	F-Value	P-Value
treatment	2	0.00036	0.000180	1.50	0.225
block	99	1.07970	0.010906	91.14	0.000
Error	198	0.02369	0.000120		
Total	299	1.10376			

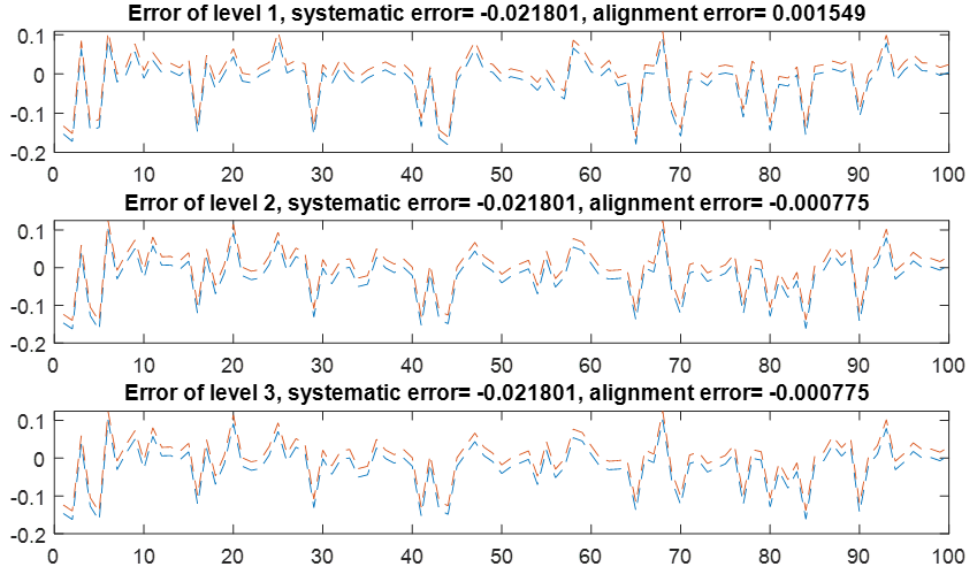


Figure 3.22: Result of error quantification based on 100 uniformly sampled points.

If a local inspection is required, the points are sampled only from the concerned area defined by users. In this experiment, there are 57 points sampled around the defective area. The error quantification results regarding the systematic, alignment, and random errors are shown in Figure 3.23. One can see that the local systematic error is larger than what is observed in the global inspection. In addition, the alignment error is much smaller than the other two error terms in both global and local inspections.

Note that the errors measured in both global and local inspection are based on the standardized coordinates. Therefore, in practice, all the errors should be scaled up based on the size of the printed sample by multiplying the error with $\max_{\theta \in \Theta} r(\theta) - \min_{\theta \in \Theta} r(\theta)$. In this case, the local systematic error is scaled up as -1.64 unit of the sample size, which is close to the designed

deviation -1.8 unit as we introduced to the layer. This supports the claim that the proposed multi-resolution alignment method can work effectively even on aligning defective layers.

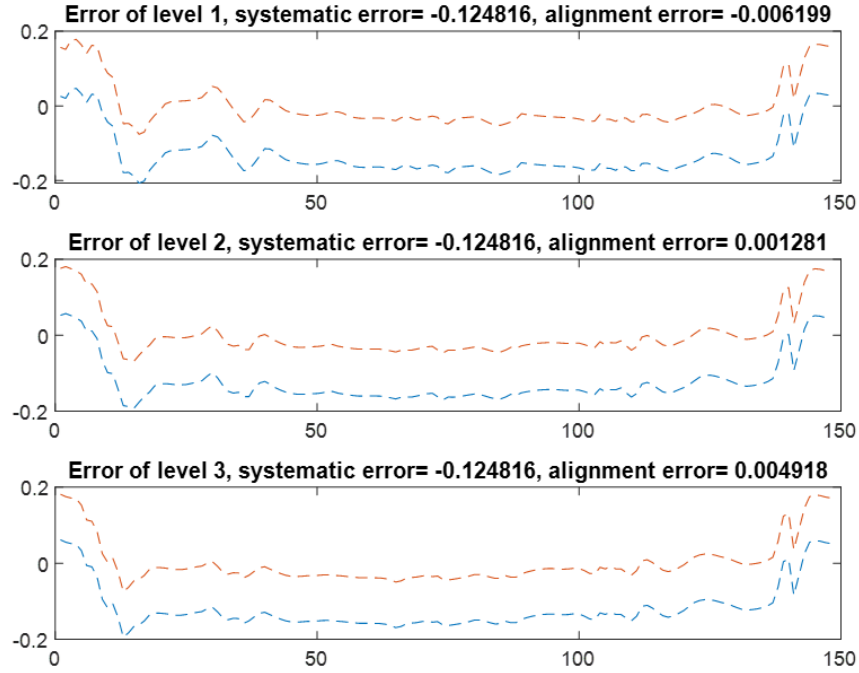


Figure 3.23: Result of error quantification based on 57 points around the defective area.

3.4.3 The 3rd layer — a layer with key internal features

The 3rd sample layer is much more complex than the previous two layers. It contains 6 internal features and 1 outer contour shape. The features, as shown in Figure 3.24, can be separated by spectral clustering (Zelnik-Manor and Perona, 2005). In this case, the outer contour shape (cluster 1) is considered as a datum feature for multi-resolution alignment. Therefore, the shape profile of cluster 1 (see Figure 3.25) is decomposed into 3 levels of detail coefficients. The detail coefficients, datum shape profile, and original complex shape are then aligned using $\mathcal{N} = 25$ and $\mathcal{L} = 3$ (see Figure 3.26). All features are aligned based on the phase shift of the datum.

Based on the aligned shapes, each feature (cluster) is inspected separately by following individual error quantification procedures. The systematic error and alignment error of each feature at different resolution levels are shown in Table 3.3. It can be observed that most clusters, except for

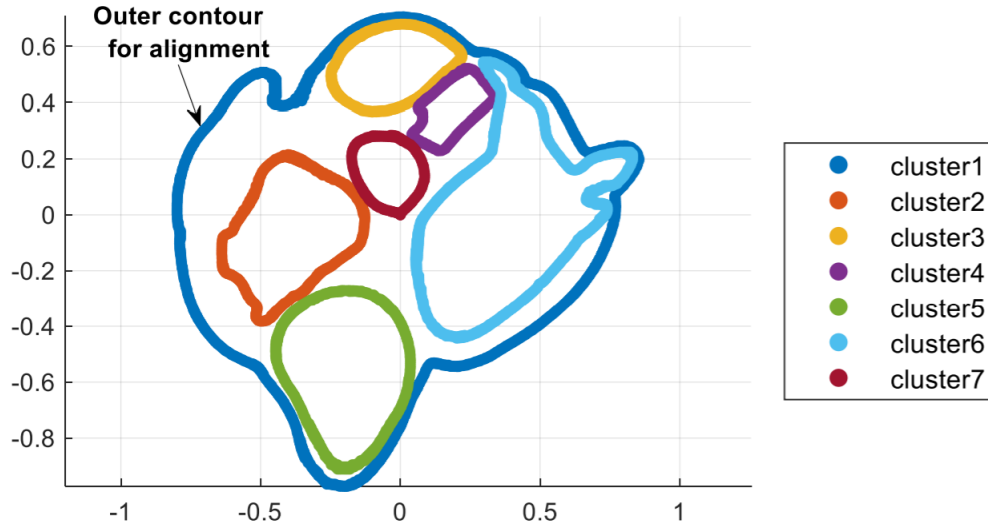


Figure 3.24: Feature segmentation by spectral clustering.

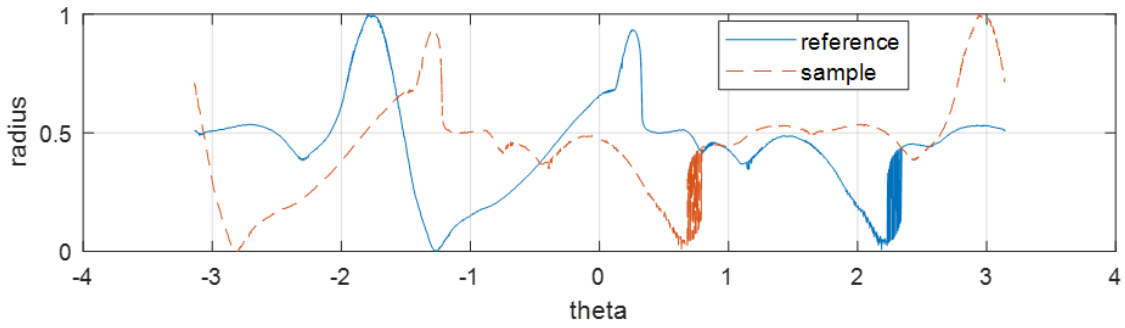


Figure 3.25: Shape profile of the outer contour shape.

cluster 1, have achieved satisfactory accuracy.

Table 3.3: Error Quantification of Each Feature

	Systematic error	Alignment error (level1)	Alignment error (level2)	Alignment error (level3)
Cluster 1	0.0376	-0.0372	0.0727	-0.0355
Cluster 2	-0.0030	-0.0090	0.0039	0.0051
Cluster 3	0.0074	0.0001	0.0010	-0.0011
Cluster 4	-0.0340	0.0232	-0.0470	0.0237
Cluster 5	0.0081	-0.0001	-0.0011	0.0012
Cluster 6	0.0014	0.0270	-0.0174	-0.0095
Cluster 7	-0.0115	-0.0068	0.0046	0.0022

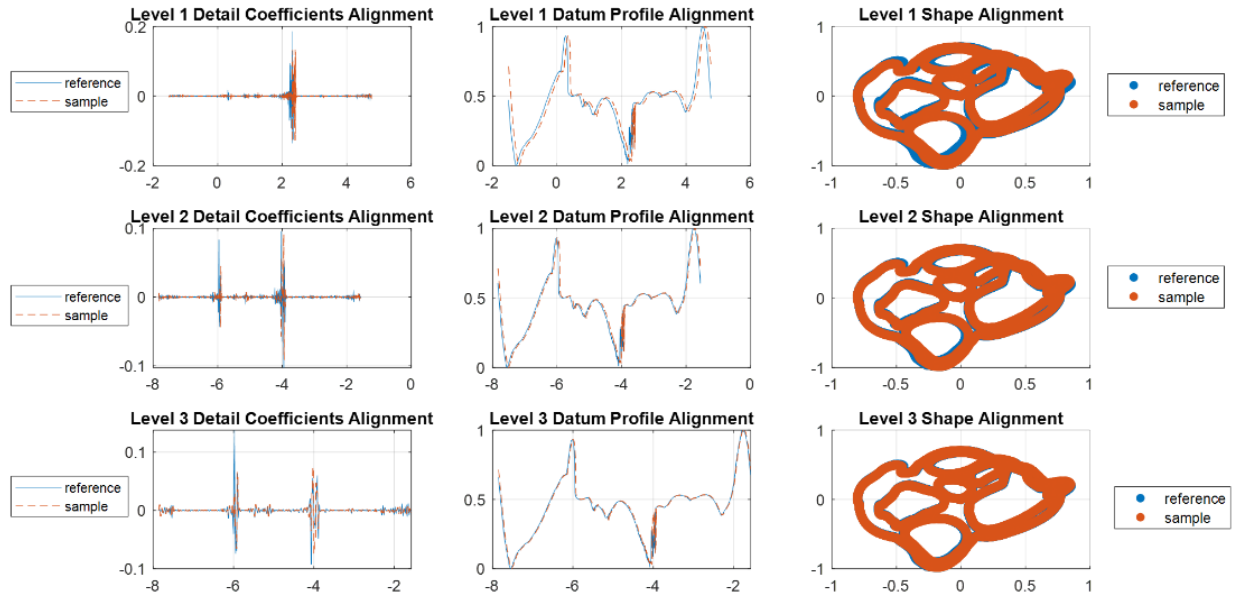


Figure 3.26: 3-level multi-resolution alignment results based on the datum feature.

This is one noticeable advantage of the proposed methods that it has the ability in handling complex layers with internal features by conducting deviation quantification in a separate and adaptive fashion. In this work, the outer contour of the complex is considered as the datum feature for alignment. In practice, any feature can be considered as the datum, and the rest features can be aligned based on the alignment of the datum. This capability allows the proposed framework to be involved in boarder applications associated with layer-wise inspection of complex geometries especially in biomedical and aerospace industries.

3.4.4 A case study of QCD method

To demonstrate the procedure of QCD under the multi-resolution framework, layer 75 to layer 175 from the upper half of the human heart are considered as sample layers to be inspected. A defective area is added to the simulated sample part, where the deviation starts at layer 120 and ends at layer 175 as shown in Figure 3.27.

The systematic error of each layer is calculated under the multi-resolution framework based on 100 uniformly sampled points. Figure 3.28 shows that there is an obvious change in the systematic

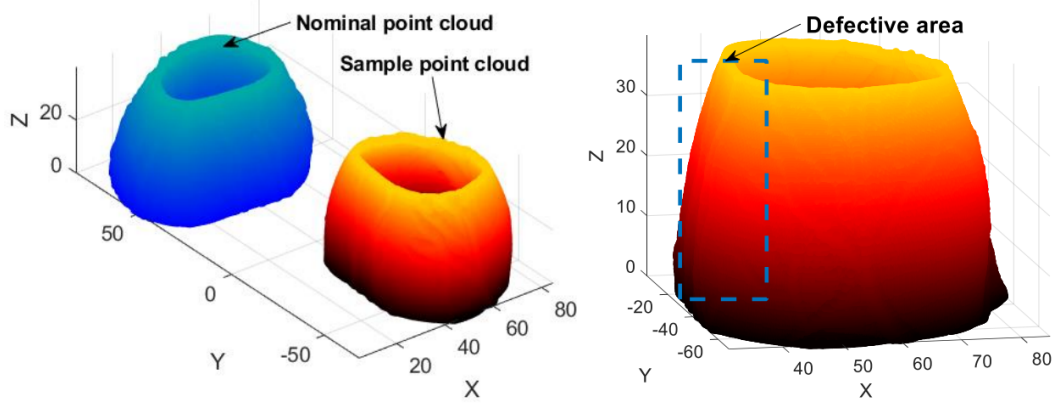


Figure 3.27: Sample layers from the upper half of the human heart

error distribution. If the change point detection method works effectively, the value of \hat{l} close to 120 can be observed at a post-change model that has a mean value close to the magnitude of the mean of systematic error after layer 120 (equal 1 in this designed defective sample). The ADD calculated based on the post-change model with different mean values μ_d is demonstrated in Figure 3.29. It is observed that a minimum ADD equal to 1 is achieved at $\mu_d \in [0.3, 1.6]$

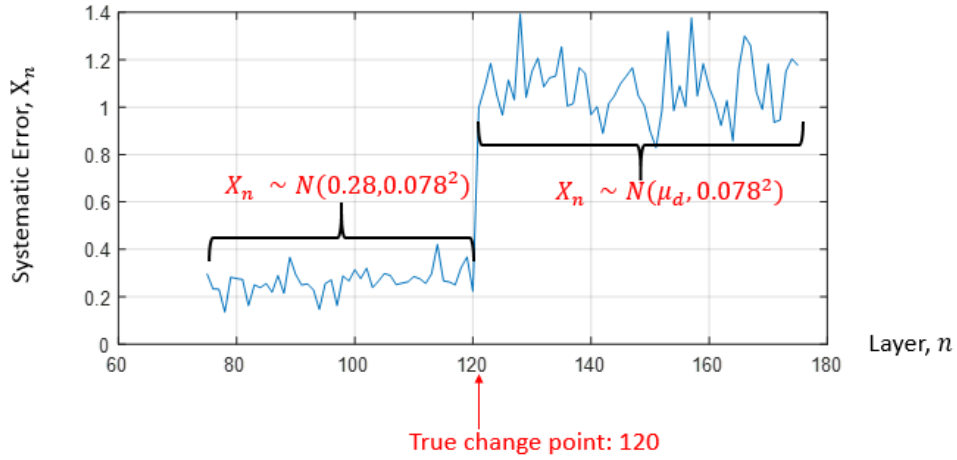


Figure 3.28: Systematic error from layer 75 to layer 175

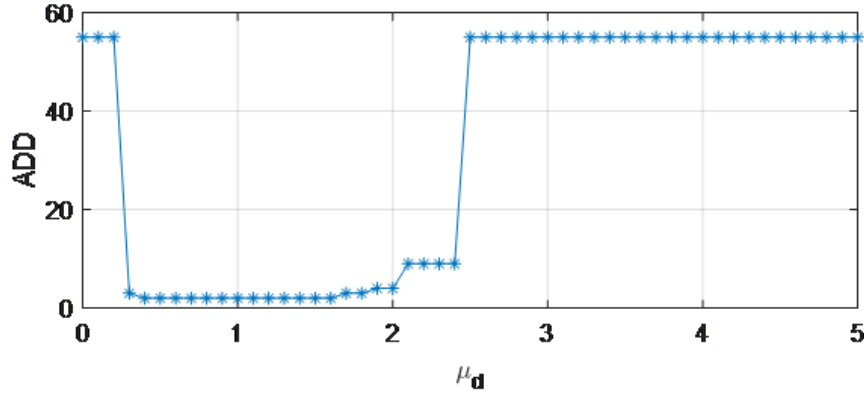


Figure 3.29: The ADD under different mean of post-change model

3.5 Conclusion

This work proposed a multi-resolution in-plane alignment and inspection scheme in additive manufacturing. The proposed framework is able to not only distinguish errors generated from different sources but also evaluate the performance and consistency of alignment when varying the level of details. Three example layers of a human heart are used to illustrate the superior performance of the proposed framework over existing methods. Moreover, one of the most important capabilities of the proposed framework is that it can deal with complex layers with internal features by conducting deviation quantification in a separate and adaptive fashion. Regarding on-line applications, the proposed framework can be integrated into an online inspection and monitoring system in additive manufacturing to generate alarms when nonconformities are observed in the current working layer.

References

- Arrieta, C., Uribe, S., Ramos-Grez, J., Vargas, A., Irarrazaval, P., Parot, V., and Tejos, C.. “Quantitative assessments of geometric errors for rapid prototyping in medical applications”. *Rapid Prototyping Journal*, 18(6):431-442, 2012.
- Bellekens, B., Spruyt, V., Berkvens, R., Penne, R., Weyn, M.. “A benchmark survey of rigid 3D point cloud registration algorithm”. *International Journal on Advances in Intelligent Systems*, 8(1):118–127, 2015.
- Bellekens, B., Spruyt, V., Berkvens, R. and Weyn, M.. “A survey of rigid 3d pointcloud registration algorithms”. In *AMBIENT 2014: the Fourth International Conference on Ambient Computing*,

Applications, Services and Technologies IARA, pages 8-13, 2014.

BuildIT Construction, Module 8.1. “Align scans to CAD - point based alignment with BuildIT Construction”, URL https://knowledge.faro.com/Software/BuildIT/BuildIT_Construction/08.1-Module-Align_Scans_to_CAD_-_Point-Based_Alignment_with_BuildIT_Construction, 2018.

Chen, G., and Maggioni, M.. ”Multiscale geometric dictionaries for point-cloud data”. In *International Conference on Sampling Theory and Applications (SampTA)*, 4(6), 2011.

Cheng, L., Wang, A., Tsung, F.. “A prediction and compensation scheme for in-plane shape deviation of additive manufacturing with information on process parameters”. *IIE Transactions*, 50(5):394–406, 2018.

Fantoni, S., Castellani, U. and Fusiello, A.. “Accurate and automatic alignment of range surfaces”. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference*, IEEE, pages 73–80, 2012.

Gibson, I., Rosen, D.W., Stucker, B. *Additive manufacturing technologies*, Springer, New York, 2014.

Hetland, M. L.. “The basic principles of metric indexing”. In *Swarm intelligence for multi-objective problems in data mining*, Springer, pages. 199-232, 2009.

Huang, W., Li, Y., Wen, P. and Wu, X.. “Algorithm for 3D point cloud denoising”. In *2009 Third International Conference on Genetic and Evolutionary Computing*, IEEE, pages 574-577, 2009.

Huang, Q., Nouri, H., Xu, K., Chen, Y., Sosina, S., and Dasgupta, T.. “Predictive modeling of geometric deviations of 3D printed products - a unified modeling approach for cylindrical and polygon shapes in ”. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, pages 25-30, 2009.

Huang, Q., Zhang, J., Sabbaghi, A., and Dasgupta, T.. “Optimal offline compensation of shape shrinkage for three-dimensional printing processes”. *IIE Transactions*, 47(5):431-441, 2015.

Huang, Q., Nouri, H., Xu, K., Chen, Y., Sosina, S., and Dasgupta, T.. “Statistical predictive modeling and compensation of geometric deviations of three-dimensional printed products”. *Journal of Manufacturing Science and Engineering*, 136(6):061008, 2014.

Sabbaghi, A., and Huang, Q.. “Predictive model building across different process conditions and shapes in 3D printing”. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, pages 774-779, 2016.

Jakovljevic, Z., Puzovic, R. and Pajic, M.. “Recognition of planar segments in point cloud based on wavelet transform”. *IEEE Transactions on Industrial Informatics*, 11(2):342-352, 2015-.

Javaid, M. and Haleem, A. “Additive manufacturing applications in medical cases: A literature based review”. *Alexandria Journal of Medicine*, 54(4):411–422, 2018.

Jin, Y., Liao, H., Pierson, H.. “Automatic Feature-Based Point Cloud Alignment and Inspection”. *Procedia Manufacturing*, 39:484-492, 2019.

- Jin, Y., Plott, J., Chen, R., Wensman, J., Shih, A.. “Additive manufacturing of custom orthoses and prostheses - A review”. *Procedia CIRP*, 36: 199–204, 2015.
- Lai, T.L.. “Information bounds and quick detection of parameter changes in stochastic systems”. *IEEE Transactions on Information Theory*, 44(7):2917-2929, 1998.
- Liu, R., Wang, Z., Sparks, T., Liou, F., and Newkirk, J.. “Aerospace applications of laser additive manufacturing”. In *Laser Additive Manufacturing*, Woodhead Publishing, pages 351-371, 2017.
- Luan, H. and Huang, Q.. “Prescriptive modeling and compensation of in-plane shape deformation for 3-D printed freeform products”. *IEEE Transactions on Automation Science and Engineering*, 14(1):73–82, 2017.
- Mallat, S. G.. “A theory for multiresolution signal decomposition: the wavelet representation”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7:674-693, 1989.
- Manson, J., Petrova, G. and Schaefer, S.. “Streaming surface reconstruction using wavelets”. *Computer Graphics Forum*, 27(5):1411-1420, 2008.
- Markelj, P., Toma, D., Likar, B. and Pernuš, F.. “A review of 3D/2D registration methods for image-guided interventions”. *Medical image analysis*, 16(3):642–661, 2012.
- Montgomery, D. C.. *Design and Analysis of Experiments*, 8th Ed., John Wiley & Sons, 2013.
- Nath, S. and Wu, J.. “November. Bayesian Quickest Change Point Detection with Multiple Candidates of Post-Change Models”. In *IEEE Global Conference on Signal and Information Processing*, IEEE, pages 51-55, 2018.
- Navangul, G., Paul, R., and Anand, S. “Error minimization in layered manufacturing parts by stereolithography file modification using a vertex translation algorithm”. *Journal of Manufacturing Science and Engineering*, 135(3):031006, 2013.
- Page, E.S.. “Continuous inspection schemes”. *Biometrika*, 41(1/2):100-115, 1954.
- Pinto, J. M., Arrieta, C., Andia, M. E., Uribe, S., Ramos-Grez, J., Vargas, A., and Tejos, C.. “Sensitivity analysis of geometric errors in additive manufacturing medical models”. *Medical Engineering and Physics*, 37(3):328-334, 2015.
- Rao, P. K., Kong, Z., Duty, C. E., Smith, R. J., Kunc, V., and Love, L. J.. “Assessment of dimensional integrity and spatial defect localization in additive manufacturing using spectral graph theory”. *Journal of Manufacturing Science and Engineering*, 138(5):051007, 2016.
- Roberts, S.W.. “A comparison of some control chart procedures”. *Technometrics*, 8(3):411-430, 1966.
- Rosman, G., Dubrovina, A., and Kimmel, R.. “Patch-Collaborative Spectral Point-Cloud Denoising”. *Computer Graphics Forum*, 32(8):1-12, 2013.

- Rueckert, D., Sonoda, L.I., Hayes, C., Hill, D.L., Leach, M.O., Hawkes, D.J.. “Nonrigid registration using free-form deformations: application to breast MR images”. *IEEE transactions on medical imaging*, 18(8):712–721, 1999.
- Savio, E., Chiffre, L.D., Schmitt, R. (2007), “Metrology of freeform shaped parts”, *CIRP annals*, Vol. 56 No. 2, pp. 810–835.
- Segal, A., Haehnel, D. and Thrun, S.. “Generalized-icp.”. *Robotics: science and systems*, 2:35, 2009.
- Shiryayev, A.N.. “On optimum methods in quickest detection problems”, *Theory of Probability Its Applications*, 8(1):22-46, 1963.
- Song, S., Wang, A., Huang, Q., and Tsung, F.. “Shape deviation modeling for fused deposition modeling processes”. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, IEEE, pages 758-763, 2014.
- Tam, G.K., Cheng, Z.Q., Lai, Y.K., Langbein, F.C., Liu, Y., Marshall, D., Martin, R.R., Sun, X.F. and Rosin, P.L.. “Registration of 3D point clouds and meshes: a survey from rigid to nonrigid”. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013.
- Tartakovsky, A.G. and Veeravalli, V.V.. “Change-point detection in multichannel and distributed systems”. *Applied Sequential Methodologies: Real-World Examples with Data Analysis*, 173:339-370, 2004.
- Torre-Ferrero, C., S. Robla, E. G. Sarabia, and J. R. Llata. ”A Coarse-to-Fine Algorithm for 3 D Registration Based on Wavelet Decomposition”. In *Proceedings of the 12th WSEAS international conference on Systems*, WSEAS, pages 763-770, 2008.
- Wang, A., Song, S., Huang, Q., Tsung, F.. “In-plane shape-deviation modeling and compensation for fused deposition modeling processes”. *IEEE Transactions on Automation Science and Engineering*, 14(2):968–976, 2017.
- Wolberg, G., and Zokai, S.. “Robust image registration using log-polar transform”. In *Proceedings 2000 International Conference on Image Processing*, volume 1, pages 493-496, IEEE, 2000.
- Zelnik-Manor, L., and Perona, P. ”Self-tuning spectral clustering”. *Advances in Neural Information Processing Systems*, 1601-1608, 2005.
- Zha, W., and Anand, S.. “Geometric approaches to input file modification for part quality improvement in additive manufacturing”. *Journal of Manufacturing Processes*, 20:465-477, 2015.
- Zhu, Z., Anwer, N., and Mathieu, L.. “Deviation modeling and shape transformation in Design for Additive Manufacturing”. *Procedia CIRP*, 100(60):211-216, 2017.

4 Optimization of a Concurrent AM Process

Fused filament fabrication, like most layer-wise additive manufacturing processes, is hindered by low production rate and scalability issues. Concurrent fused filament fabrication mitigates these disadvantages by distributing the processing of each layer amongst multiple extruders working in parallel. The objective of this work is to develop a general toolpath allocation and scheduling methodology to achieve this. Breaks in toolpath that are inherently created by slicing software for single-extruder machines are used to form sub-paths, and the assignment of these to available extruders is formulated as a scheduling problem with collision constraints. A formal optimization model is presented, and two novel heuristics are developed to obtain approximate solutions. Three case studies demonstrate the application of these algorithms and compare their relative performance with respect to fabrication time and computational cost. In simulations with three extruders, layer printing times were reduced by as much as 60% compared to single-extruder machines. The proposed heuristics also exceeded the performance of two baseline toolpath scheduling algorithms by as much as 45%. Two key layer characteristics were found to influence heuristic performance, and the advantages and disadvantages of each algorithm are discussed in the context of these characteristics.

4.1 Introduction

Fused Filament Fabrication (FFF) is an additive manufacturing (AM) process based on extrusion, whereby polymer filament is deposited by a moving extruder to form layers (Gibson et al., 2014). As with AM in general, it excels at producing complex geometric forms without the need for part-specific tooling. Compared to other AM processes, it has high material utilization, and it is relatively low-maintenance, safe, inexpensive, and environmentally benign (Gao et al., 2015). As one of the most popular AM techniques, FFF has also been shown to have good potential for industrial applications, such as automobile prototyping and rapid tooling (“3D Printed Shelby Cobra,” n.d.). In addition, reinforcing fibers may be added to the feedstock to increase strength (Hawley et al., 1984), and FFF for carbon-fiber composites based on thermosetting polymers has

been demonstrated (Compton et al., 2014).

The origin of FFF lies in producing small but complex geometries. However, with demand from industries such as aerospace, architecture, and automotive for the low-volume fabrication of large products, there is a strong motivation to scale up the process (Vurpillat, 2016; “BetAbram,” n.d.; Severson, 2015; “Erectorbot,” n.d.). Nevertheless, upscaling the FFF process is hindered by limited fabrication speed. Increasing layer thickness can speed up the process, but this sacrifices resolution and increases stairstepping errors (Kim et al., 2008). This limitation forces manufacturers and customers to choose between long cycle times and extremely poor surface finishes.

To improve the scalability of FFF, a significant decrease in workpiece cycle time without degrading its resolution is needed. The most straightforward method adopted in subtractive manufacturing to reduce cycle time is to decompose the task into several subtasks and assign each to different machines working in parallel. In an analogous approach, this research proposes that FFF machines could be equipped with multiple extruders working collaboratively on each layer. In such a concurrent multi-extruder AM system, each extruder would be independently driven, and each would be able to reach the entire workspace in order to maximize flexibility. Existing FFF planning algorithms already divide the toolpath for each layer into several sub-paths, including perimeters for outlining the layer and multiple rasters for interior in-filling (Jin et al., 2015), and these may be used to subdivide the task of printing the layer among the extruders. A key advantage to this approach is that it works with any existing toolpath planning software, effectively decoupling path planning from path scheduling. This allows manufacturers to implement the heuristics developed herein using any software that makes toolpath code available to the user and allows developers of proprietary systems to do the same without the need to adapt or redevelop their toolpath planning software. Further, risk for end users is mitigated by using tried and true toolpath planning software that they already trust to produce quality parts with predictable properties.

There are two major issues that must be resolved before such a concurrent multi-extruder system may be developed. First, since all the extruders share the same workspace, a collision-avoidance strategy must be developed. Second, the makespan (i.e., the time to complete one layer)

of the multi-extruder system is highly dependent on the assignment and sequencing of sub-paths among extruders. Further, these problems are intertwined: Sub-path allocation and scheduling can influence the potential for collision, and collision avoidance will place constraints on sub-path scheduling. To address the two issues, a method for finding the optimal schedule that will prevent collisions while achieving the minimum makespan is of special interest. In addition, this method should be computationally efficient so as not to introduce unnecessary delays prior to printing, between layers, or within a layer.

In this paper, the first issue is solved by a collision check strategy based on the area swept by each extruder over a given time interval. The second issue is addressed by formulating an optimization model for the multi-extruder scheduling problem, with the objective of minimizing the makespan subject to collision constraints. Since the optimization problem is NP-hard, two heuristic algorithms are proposed. The first method features a novel combination of collision avoidance strategies with the Longest Processing Time First (LPT) scheduling algorithm. The second is an entirely novel method, called the Minimum Overlapped Area First (MOA) algorithm, which reduces the likelihood of collision by prioritizing concurrent scheduling of sub-paths having minimal mutual interference. The performance of these heuristics is measured and compared in terms of makespan the computational time through three case studies with layers possessing different characteristics. In these case studies, layer printing time was reduced by as much as 60%. The remainder of this paper is organized as follows: Section 4.2 provides a brief literature review on current multi-extruder FFF applications, collision avoidance in robotic motion planning, and existing parallel scheduling algorithms. In Section 4.3, the optimization model, heuristic methods, and collision avoidance strategies are described in detail. In Section 4.4, three case studies are provided to demonstrate the application of the algorithms and compare their performance. Finally, Section 4.5 discusses the conclusions drawn from the case studies and suggests directions for future research.

4.2 Literature Review

While many FFF machines possess multiple extruders, in the vast majority of cases the extruders are fixed on the same moving platform, meaning they are not capable of independent motion. Such machines are usually designed for multi-material and/or multicolor printing (Pax, 2013; Schumacher et al., 2014). This arrangement can also be used to implement synchronous printing, whereby multiple copies of the same part are created simultaneously on the same platform to speed up the fabrication for batch production (“Cultivate3d,” n.d.). This approach limits part size to a fraction of the build platform and is clearly not a solution for one-off production, so it is of limited use in addressing the fabrication speed problem in large-scale 3D printing. By contrast, concurrent multi-extruder FFF involves extruders which are capable of independent motion and controlled so as to cooperatively complete each layer. One notable example is Project Escher (“Autodesk,” n.d.; Miller, 2016) announced by Autodesk in 2016. Project Escher, however represents but one of the numerous kinematic arrangements that could be chosen for the implementation of concurrent FFF, and its control is simplified considerably by the fact that the extruders may not overlap in the X-Y plane. There is a need for a general treatment of the problem, published in the open academic literature, to drive innovation and advancement of concurrent FFF fabrication technology.

The two key challenges in concurrent FFF are avoiding collision and efficiently allocating tasks between extruders. Fortunately, the literature is rich with studies on collision avoidance and parallel process scheduling, which provides a strong theoretical and technical starting point for formulating the optimization problem and developing computationally efficient heuristics to arrive at a good solution. Collision avoidance is covered extensively in the robotics literature since it is one of the major tasks in multi-agent planning (Arbib and Rossi, 2000; Wu et al., 2002; Klee et al., 2015). The main purpose is to find a collision-free trajectory that achieves the optimization goal in terms of the total distance or travel time, scheduled arrival time, and time delay (Jeevamalar and Ramabalan, 2012). The literature on multi-robot collision avoidance mainly focuses on one of three aspects: obstacle avoidance for mobile robots, collision-free trajectory planning for manipulators, or collision avoidance for multiple robot systems. The collision detection techniques

applied in the first two aspects mainly involve sensing and navigating the environment (David et al., 2016; Pandey and Parhi, 2016; Almasri et al., 2016; Chong et al., 2009), graphic mapping (Švestka and Overmars, 1998; Yang et al., 2010; O'Donnell and Lozano-Pérez, 1989; Lee and Lee, 1987), and mathematical modeling and algorithm development (Ata and Myo, 2006; Hasan and Gavrilova, 2010; Daachi et al., 2012). However, there is often more flexibility in the trajectory of robots, whereas an FFF extruder must not vary from its path or speed while extruding. In addition, there is limited work on how task allocation and scheduling among robots dynamically affects the collision check procedure. Babu et al. (2006) invented a cluster tool to allow multiple robots to serve a plurality of processing stations without collision. Each robot's trajectory forms an envelope and the process is collision free if all envelopes do not overlap. However, the robots have to sense their location and motion in real time in order to find a collision-free path, which is infeasible in the FFF process where extrusion cannot be interrupted mid-path. Nevertheless, the envelope idea provided by Babu's team inspired the proposed swept-area collision check method proposed in this paper. Jose et al. (2016) used a genetic algorithm for task allocation and an A* algorithm for path planning in a multi-robot system. Unlike the paths that are predetermined by the slicing software in AM, the path generated by the A* algorithm can adapt to different allocation solutions. Since the positions of the robots are fixed when they perform their intended tasks, the collision check is only for the rapid travel motion based on the instantaneous coordinates of the robots in a pre-defined map; a collision is indicated if any two of the robots have the same position coordinates at a particular time. Unlike a multi-robot system, the operation of the FFF process is piecewise continuous, and each sub-path should be considered as an indivisible task. In essence, the nature of the FFF process leads the present work to be formulated as a scheduling problem with collision constraints rather than as a path planning problem, which is dominant in the robotics literature.

The general objective of a parallel scheduling problem is to minimize the makespan (C_{max}), which is defined as the elapsed time between starting and finishing one production unit (Pindeo, 2012). Let $P_m || C_{max}$ denote such a scheduling problem with m parallel machines. It has been

shown that, in general, this is an NP-hard problem (Garey and Johnson, 1978), and this fact has inspired the development of different heuristics, including branch-and-bound enumeration (Azizoglu and Kirca, 1999; van de Velde, 1993; Belkaid et al., 2012) and local neighborhood search for reassignment (Brucker et al., 1996; Brucker et al., 1997), as well as approximations using list scheduling (LS) (Granham, 1969), bin-packing base (Coffman et al., 1978), and linear programming (Lenstra et al., 1990; Potts, 1985). Among all the heuristics, LS is particularly powerful for solving on-line scheduling problems (Chen et al., 1998). If the jobs are listed in descending order of processing time, then this is known as the longest processing time first (LPT) algorithm (Granham, 1969; Coffman et al., 1978). Compared to the optimal condition (OPT), LPT can achieve the worst-case ratio of $C_{max}(\text{OPT})/C_{max}(\text{LPT})=4/3-1/(3m)$ (Coffman et al., 1978). Because of its proven performance, LPT will serve as the starting point for the development of the algorithms presented herein.

Implicit in the previously mentioned scheduling algorithms is the assumption that the machines/processes are independent, yet this is not the case in the concurrent FFF method envisioned herein; collision between extruders working in close proximity is a very real possibility. As for incorporating interdependency constraints into scheduling problems, Miura et al. (2002) proposed a method of assigning and scheduling detection tasks among parallel working cameras in a dynamic environment. Assignments were evaluated by considering the number of tracked persons and the cameras' attention distribution in the given specific space. Instead of solving the problem of physical collisions between equipment, the method proposed in Miura's study is intended to solve resource utilization conflicts. Since collisions in the multi-extruder process mainly result from the physical motion of the extruders, the methods proposed in Miura's work are not well-suited to the present problem. Li et al. (2015) provided a scheduling model for a robotic flow-shop with multiple robots serving parallel machines. However, the overlap indicated in this study is the crossover of robots on the same track. The crossover can be avoided by determining the optimal operation cycle of each robot. Considering the comparatively flexible movement of extruders and uncertain cycle time of printing tasks, this method still cannot address the collision between ex-

truders. Based on this brief review of related literature, an adequate solution to the problem of parallel scheduling for multiple extruders with collision avoidance does not presently exist. This research addresses this problem and makes the following contributions to the literature base:

1. A formal optimization model for minimizing the makespan under collision avoidance constraints is proposed.
2. A novel modification of the LPT algorithm incorporating collision avoidance is presented as one method for finding a near-optimal solution.
3. A novel, and potentially more computationally efficient algorithm, Minimum Overlapped Area First (MOA), is presented.

To the best of the authors' knowledge, this research is the first attempt to develop an integrated method of the collision checking and parallel scheduling for the FFF process.

4.3 Method

4.3.1 Assumptions

The model and algorithms presented in this section were developed under the following assumptions:

1. Sub-paths are pre-known and derived from single-extruder toolpath discontinuities. Tool-path generated by existing AM slicing software for single-extruder operations commonly includes printing pauses while the extruder rapid travels to a new position within the layer. These pauses define the start and end points of sub-paths which may be assigned to multiple extruders. This approach allows users to utilize existing slicing and toolpath generation tools, even if they are proprietary. Precisely replicating the printing discontinuities that exist in single-extruder operations also means that workpiece quality should be similar, thereby easing the process qualification process.

2. Perimeter sub-paths must precede raster sub-paths. In order to have better surface finish and dimensional accuracy, perimeters are conventionally printed before the infill rasters. This research maintains this convention.

Additionally, several simplifying assumptions have been made so as not to distract from demonstrating the key aspects of the algorithms. Any could be removed by relatively straightforward methods.

1. The kinematic chain to which the extruders are attached is neglected. In reality, the specific mechanism(s) used to manipulate the extruders may introduce additional opportunities for collision. This could be accounted for with a more complex collision checking algorithm that includes all rigid bodies supporting the extruder, though such an algorithm would lack generality since it is tied to a specific printer geometry. In the case of mobile extruder bases (Milkert, 2014) or kinematically redundant manipulators (Chiaverini et al., 2008), the collision check algorithm developed herein is directly applicable since the kinematic chain that supports the extruder can be repositioned and/or reconfigured in real time.
2. Translational velocity of the extruders is constant for a given layer; that is to say that the two dimensional position of the extruder is given by $x = x_0 + f\Delta t$, where f is a feed rate vector of constant magnitude and Δt is the collision check time period (discussed in detail in section). Accelerations due to starting, stopping, and changing the direction of the extruder are neglected since these are typically very high, making them a relatively minor factor in computing extruder trajectories. If accelerations are indeed significant, extruder position may be evaluated in a piecewise fashion using $x = x_0 + \frac{1}{2}a \Delta t$ for sections of the trajectory subject to acceleration.
3. Rapid travel is neglected. Rapid travel time is typically very short compared to the time that the extruders are actually engaged, and it is assumed that a collision-free rapid travel path may be found using path planning algorithms from the robotics literature, including those mentioned in the literature review. Rapid travel time between all sub-paths may be computed

at the outset and treated as a delay between sub-paths, which is discussed in detail in section 4.3.6.

4.3.2 Brief Description of Three Cases

Two major factors will influence the process makespan: sub-path length variability and the probability of collision. A large makespan may result from an unbalanced schedule due to the large variance in sub-path lengths or delay introduced to prevent collisions. To illustrate these cases, three layers (Figure 4.1) with different features are considered. The first is a simple layer presented for the purpose of demonstrating the algorithms. The second layer consists of sub-paths with large variance in length, while the third one has sub-paths with nearly uniform length. These two cases are used to compare the performance and relative merits of the proposed heuristics with respect to makespan and computational time.

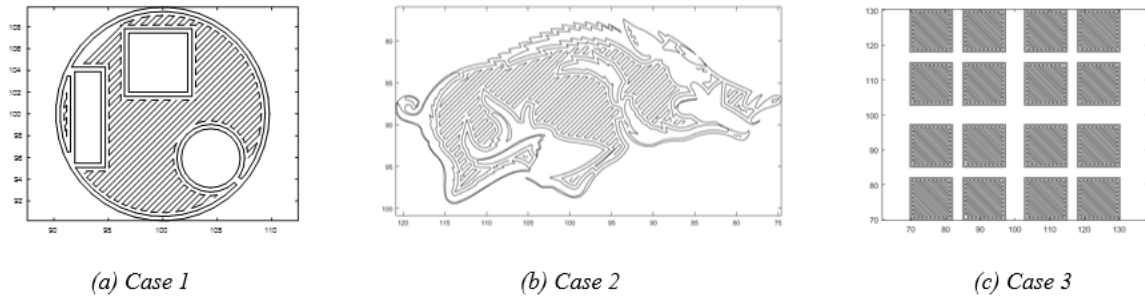


Figure 4.1: The overview of three different layers

4.3.3 Baseline Methods

Intuition suggests that dividing each layer into contiguous, and inasmuch as possible, equal regions – one for each available extruder – would be a simple but reasonable approach to sub-path allocation. Two versions of this approach are offered as baselines for evaluating the performance of the LPT and MOA heuristics proposed herein. The first simply divides the bounding rectangle of the layer into equal areas without regard to the actual layer geometry. This naïve scheduling (NS) algorithm is simple to automate and extremely computationally efficient. Of course, considering

the actual layer geometry in establishing the contiguous regions should provide a better solution in most circumstances. Assuming that human intuition would be almost as good or possibly better than any image analysis technique for this purpose, the second algorithm relies upon a human to establish regions with simple boundaries. Automating this human segmented scheduling (HSS) algorithm is of course impractical, but beating its performance serves as a formidable challenge for the proposed heuristics. The following procedure is applicable for both baseline methods: (1) Initial boundaries are established. Let z_i represent region i for $i = 1, 2, \dots, m$, where m is the number of extruders. (2) Sub-paths that lie entirely within a given region are first assigned to that region. Then the total length of sub-paths assigned to each region, denoted as L_i , is calculated. (3) The remaining sub-paths, which span multiple regions, are then assigned to the region with smaller L_i . The only difference between the NS and HSS methods is in the method of establishing the initial regional boundaries. For the three layers provided in section 4.3.2, the regions generated by the NS algorithm is shown in Figure 4.2 and those of the HSS algorithm are shown in Figure 4.2. After the regions are established, each extruder starts printing the assigned sub-paths from left to right (i.e., the sub-path with the leftmost start point is scheduled first). Whenever a potential collision is detected (the collision check method will be described in section 4.3.7), the candidate sub-path on one extruder will wait until conflicting sub-paths on other extruders have completed.

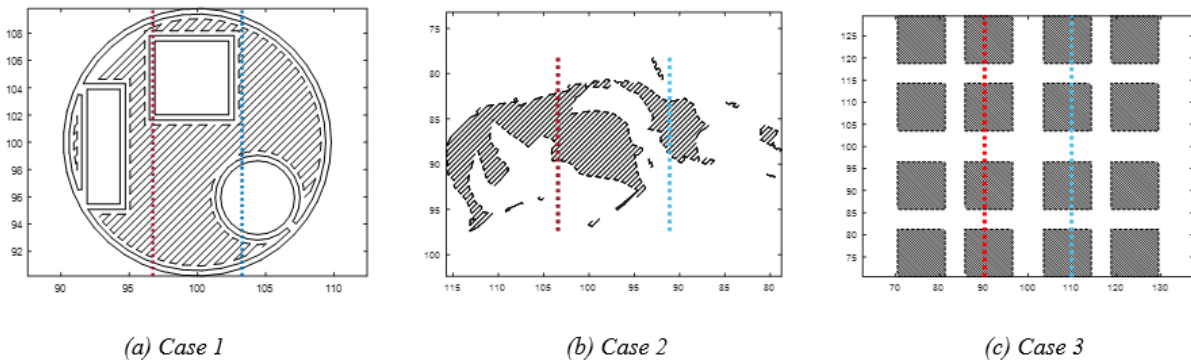


Figure 4.2: Region division in the NS method (boundaries represented by dashed lines)

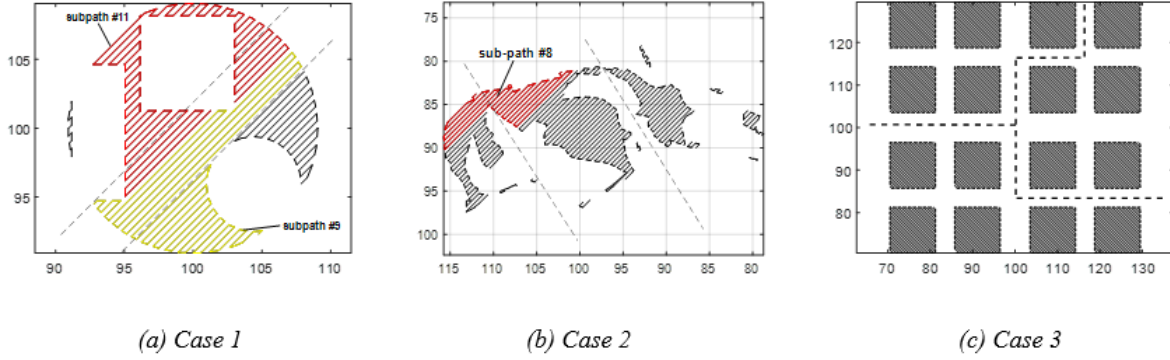


Figure 4.3: Region division in the HSS method (perimeter sub-paths omitted for clarity). (a) Sub-path #11 and #9 are two raster sub-paths of case 1. Both of them span two regions. Sub-path #11 is assigned to the left region, and sub-path #9 is assigned to the middle region. (b) Sub-path #8 is a raster sub-paths of case 2, which spans two regions and is assigned to the left region. (c) No sub-paths span multiple regions.

4.3.4 Optimization model

An optimization model can be formulated as a parallel scheduling problem that minimizes the makespan of the process subject to collision-avoidance constraints. A nomenclature for this model is given in Table 4.1. The parallel scheduling problem can be formulated as

$$\min C_{max} \quad (4.1)$$

$$s.t. \sum_{j \in J} p_j x_{ij} \leq C_{max} \quad for \quad \forall i \in M \quad (4.2)$$

$$S \equiv \{(j, k) : \chi_{ij} + \chi_{ik} > 1, \quad for \quad \forall i \in M\} \quad (4.3)$$

$$D \equiv \{(j, k) : x_{ij} + x_{ik} \leq 1, \quad for \quad \forall i \in M\} \quad (4.4)$$

$$ST_k = \sum_j a_{jk} p_j ET_k = \sum_j a_{jk} p_j + p_k \quad for \quad (j, k) \in S \quad (4.5)$$

$$C \equiv \{(j, k) : (j, k) \in D | ST_k \in [ST_j, ET_j] \quad or \quad ST_j \in [ST_k, ET_k]\} \quad (4.6)$$

$$CST_{jk} = \max\{ST_j, ST_k\} \quad for \quad (j, k) \in C \quad (4.7)$$

$$CET_{jk} = \min\{ET_j, ET_k\} \quad for \quad (j, k) \in C \quad (4.8)$$

$$\sum_g^{G_{jk}} \varphi_{jkg} = 0 \quad for \quad G_{jk} = \frac{CET_{jk} - CST_{jk}}{dt} \quad and \quad (j, k) \in C \quad (4.9)$$

$$\sum_{i \in M} x_{ij} = 1 \quad \text{for } \forall j \in J \quad (4.10)$$

$$a_{jk} + a_{kj} \leq 1 \quad \text{for } j, k \in S \quad (4.11)$$

where C_{max} is the makespan, which should be longer than the completion time of any individual extruder. p_j is the processing time for sub-path j . Two collections of decision variables are defined for this problem: x_{ij} and a_{jk} . Note that for any a_{jk} , sub-path j does not necessarily immediately precede sub-path k .

Collision checking is accomplished by considering the area swept out by each extruder over a given time period. In order to refine the sub-path set for collision checking, we further define two sets: S and D , where set S includes pairs of sub-paths assigned to the same extruder and set D contains those on different extruders. The start time ST_k of sub-path k equals the total processing time of sub-paths ahead of k on the same extruder, and its end time ET_k can be obtained by adding the processing time of sub-path k to ST_k . If the processing period for any pair of sub-paths (j, k) in set D has an overlapped period $[CST_{jk}, CET_{jk}]$, a collision check is performed by dividing the time period into G_{jk} time slots, where G_{jk} is determined according to a predefined time interval dt . If a potential collision between sub-path j and k is identified, the collision sign φ_{jkg} for any time slot g in this time period equals 1. The detailed collision check process is illustrated in section 4.3.7. Additionally, perimeter sub-paths should be printed prior to infill raster sub-paths. Thus, the minimum makespan for printing perimeter sub-paths is first calculated by including all the perimeter sub-paths in the set J . Let C_P denote the completion time of all perimeter sub-paths, and let the set J equal to a set R that includes all the raster sub-paths and modify equation (2) as $C_P + \sum_{j \in R} p_j x_{ij}$ for $\forall i \in M$.

4.3.5 Heuristic Algorithms

To solve the NP-hard parallel scheduling problem, two collision-free heuristic scheduling algorithms are developed that make the problem tractable. The first one combines the longest processing time first (LPT) with collision avoidance constraints; the second algorithm, minimum

Table 4.1: Symbols Used in the Optimization Model

Set	
J	A given set of sub-paths. $j, k \in J$
M	A given set of available extruders. $i \in M$
S	Include pairs of sub-paths (j, k) assigned to the same extruder. It will change with the decision variable.
D	Include pairs of sub-paths (j, k) assigned to the different extruder. It will change with the decision variable.
C	Include pairs of sub-paths (j, k) that need collision check. It requires $(j, k) \in D$ and the processing periods of sub-path j and sub-path k overlap in the time domain.
Variable	
x_{ij}, x_{ik}	$x_{ij} = \{1, \text{if sub-path } j \text{ is assigned to extruder } i; 0, \text{otherwise}\}, j \in J, i \in M$ $x_{ik} = \{1, \text{if sub-path } k \text{ is assigned to extruder } i; 0, \text{otherwise}\} k \in J, i \in M$
a_{jk}	$a_{jk} = \{1, \text{if sub-path } j \text{ is assigned before sub-path } k; 0, \text{otherwise}\}$. Note that for any a_{jk} , sub-path j does not necessarily immediately precede sub-path k .
ST_k	The start time sub-path k , which is equal to the total processing time of sub-paths j ahead of k on the same extruder. $(j, k) \in S$
ET_k	The end time sub-path k , which can be obtained by adding processing time of sub-path k to the start time sub-path k .
CST_{jk}	The start time of overlapped period for $(j, k) \in C$
CET_{jk}	The end time of overlapped period for $(j, k) \in C$
G_{jk}	The number of time slot for collision check, which is obtained by divide the length of the overlapped period with a given time interval dt
φ_{jkg}	The collision sign for any pair of sub-paths $(j, k) \in C$ during any time slot g , where $g = 1, 2, \dots, G_{jk}$. $\varphi_{jkg} = \{1, \text{if a potential collision is identified between sub-path } j \text{ and } k \text{ during time slot } g; 0, \text{otherwise}\}$.
C_{max}	Makespan of the printing given set of sub-paths, which should be longer than the completion time of any individual extruder.
Parameter	
p_j, p_k	Process time of sub-path j, k , where $j, k \in J$
dt	A given time period for determining the size of the time slot for collision check

overlapped area first (MOA), is a novel method for assessing the potential for collision between sub-paths and attempting to simultaneously schedule sub-paths that have minimal potential for collision. Both of the heuristic algorithms are generally applicable as long as all the sub-paths are pre-known.

For better surface finish and dimensional accuracy, all perimeter sub-paths should be completed before starting the infill raster sub-paths. Thus, for each layer, the scheduling of perimeters and

rasters can be considered as two separate, sequential problems.

Longest Processing Time First (LPT) In solving a regular parallel scheduling problem $P_m || C_{max}$, the conventional LPT algorithm assigns the m longest jobs to m machines at $t = 0$. After that, the longest job among those not yet processed is assigned to each machine as the machine becomes free (Pindeo, 2012). In this research, the LPT algorithm was modified by adding a collision checking step (details provided in Section 4.3.7). Figure 4.4 shows a flow chart of the algorithm, described as follows: The modified LPT algorithm starts by assigning the longest sub-path to the first extruder. Then it searches the remaining sub-paths in descending order of processing time until a sub-path is found that will not cause a collision with those currently being processed. This newly found sub-path is assigned to a free extruder, and the process is repeated until all extruders have been assigned a sub-path. Assuming that the number of sub-paths exceeds the number of extruders, the earliest time at which an extruder will complete its assigned sub-path and become free is computed. The algorithm then searches for the longest unassigned path that will not generate a collision if scheduled at that time. This process repeats until no other sub-paths can be scheduled for parallel processing, and a single extruder completes the remaining sub-paths for the layer. Since perimeters must be completed before rasters, the modified LPT algorithm is applied twice: first for perimeters, then for rasters. It is worth pointing out that the modified LPT algorithm requires repeated collision checks, which can become computationally expensive for some layer geometries. This weakness motivated the development of an alternative algorithm.

Minimum Overlapped Area First (MOA) In general, the potential for collision between any given pair of sub-paths is related to the intersection of the areas swept out by the extruders while completing those sub-paths. A smaller overlap indicates a lower probability of collision during concurrent processing, and no overlap means that no collision will occur. The general principle behind MOA is to concurrently schedule sub-paths with minimal overlap. This algorithm can reduce the number of collision checks in some layer geometries, and lower the computational cost accordingly. A flowchart for the MOA algorithm is shown in Figure 4.5. Initially, the total

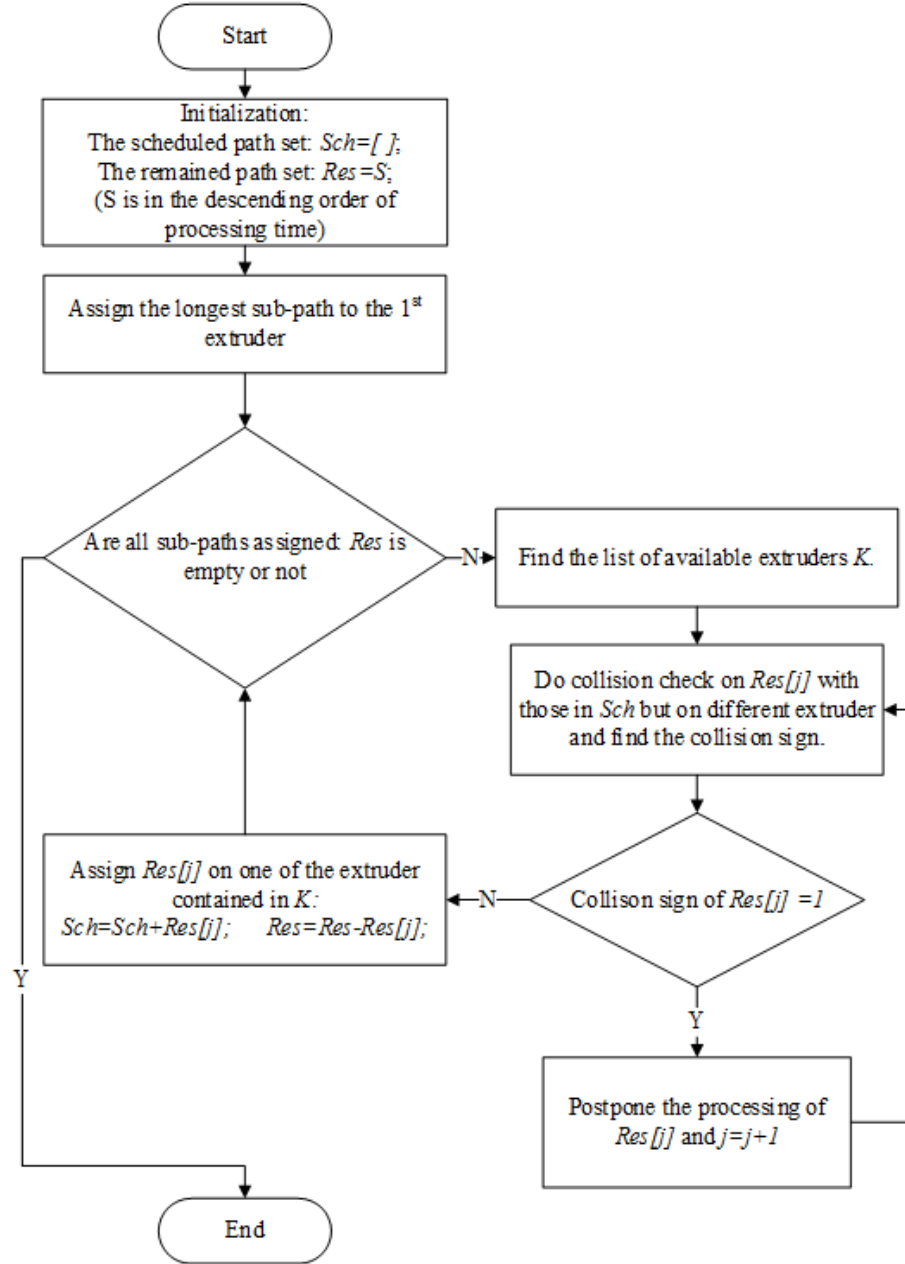


Figure 4.4: LPT algorithm flow chart

overlapped area between each sub-path and all other sub-paths is computed, and the one with the highest total overlapped area is scheduled on the first extruder. This has the effect of scheduling the most troublesome sub-path as early as possible, which was found to improve algorithm performance. Subsequent sub-paths are assigned by first ranking them in order of overlapped area with the sub-paths currently in progress, then searching these from the lowest to the highest for

a sub-path that will not produce a collision. The calculation of overlapped areas for all possible sub-path pairs is accomplished in a preprocessing step that precedes the implementation of the MOA algorithm. Each pairwise overlap calculation involves three steps: First, the convex hull of each sub-path is constructed (Figure 4.6a). Then each convex hull is offset by the size of extruder (Figure 4.6b). Lastly, the overlapped area for each possible pair of offset convex hulls is calculated (Figure 4.6c). It is worth pointing out that these calculations need not be repeated in executing the algorithm. The total overlapped area (with all other sub-paths) for use in the initial assignment is derived from these results. As with the LPT algorithm, perimeter sub-paths need to be completed before rastering the infill can begin. The MOA heuristic was found to perform poorly when scheduling perimeters due to a very high incidence of large overlapping areas of the convex hulls. As a result, perimeter scheduling is always accomplished via the LPT algorithm, even when MOA is used for the rasters.

4.3.6 Adaptive Algorithm Modification

In both the LPT and MOA algorithms, it is quite possible that as a given extruder becomes free, all remaining unscheduled sub-paths will cause collision if immediately scheduled. In other words, the collision sign of $Res[j]$ in Figure 4.4 (LPT) or $MOA[j]$ Figure 4.5 (MOA) for any $j \in Res$ will be 1. Figure 4.7 shows an example: Extruder 2 finishes sub-path A and is ready to accept a new sub-path, but scheduling any of the sub-paths in the set of those that remain, $Res = \{1, 2, 3\}$, would result in one or more collisions with B and C, already in process. In the basic algorithms described thus far, Extruder 2 would be idle until the completion of C, at which time the algorithm would evaluate whether any of the remaining sub-paths could be scheduled on Extruders 2 and/or 3. If not, none of the remaining sub-paths could be scheduled until the completion of B, leaving both 2 and 3 idle. In practice, it was found that a number of incompatible sub-paths were frequently scheduled sequentially on a single extruder toward the completion of the layer. This left the remaining extruders idle for an extended period of time, partially defeating the purpose of having multiple extruders.

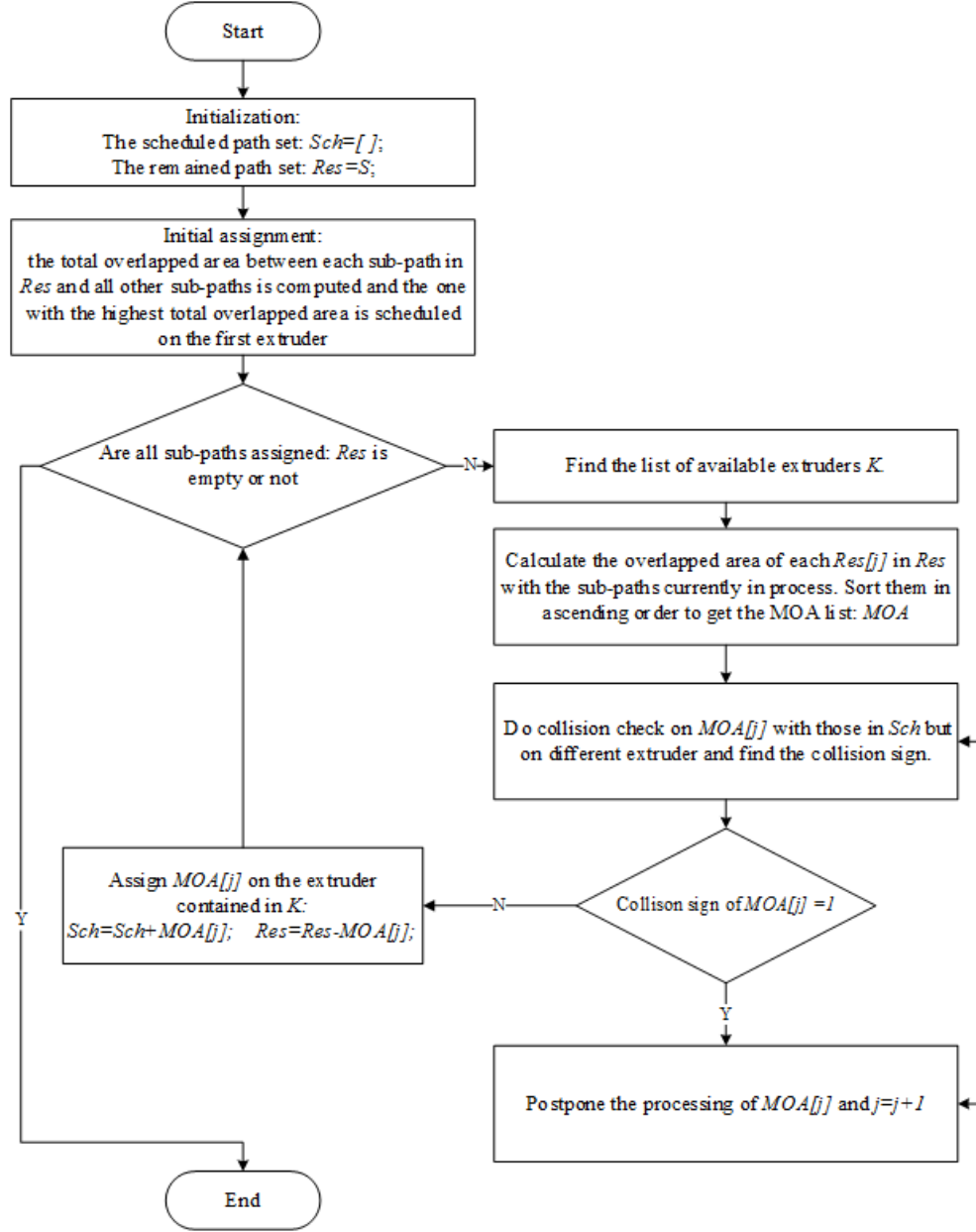


Figure 4.5: MOA algorithm flow chart

This weakness can often be rectified by shifting the scheduling of a candidate sub-path in time so as to avoid collision. This is done by defining an adaptive delay interval, ΔT . Instead of waiting for the completion of a sub-path, the algorithms evaluate whether each remaining sub-path can be scheduled at time $n\Delta T$ for $n = 1, 2, 3, \dots$ without causing collision, and the sub-path requiring minimum delay is scheduled. Returning to the example in Figure 4.7, each sub-path in

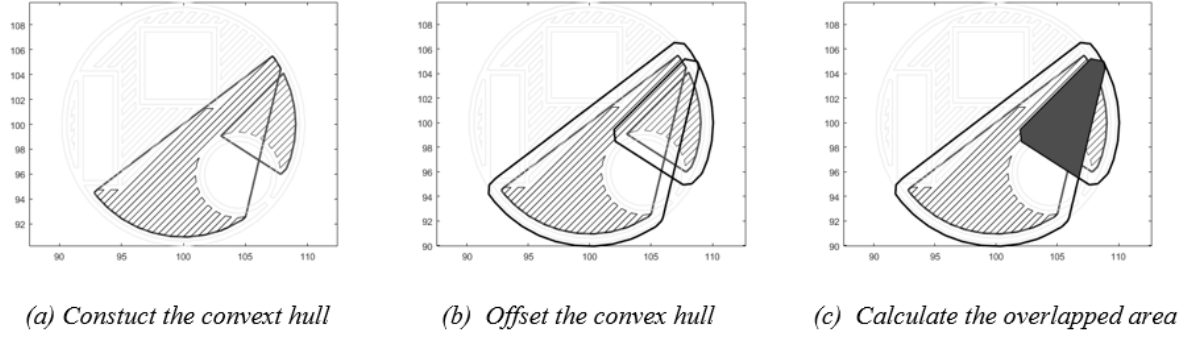


Figure 4.6: Computing overlapped area for two sub-paths

$Res = \{1, 2, 3\}$ is tested at $\Delta T, 2\Delta T, 3\Delta T$, etc. to determine the earliest time at which it can be scheduled without collision. It is found that 3 can be scheduled after a delay of $2\Delta T$, while sub-paths 1 and 2 require delays of $7\Delta T$ and $4\Delta T$, respectively. Adaptive delay significantly increases the number of collision checks required for each algorithm. Thus, the magnitude of ΔT represents a tradeoff between minimizing makespan and the computational load of the algorithms.

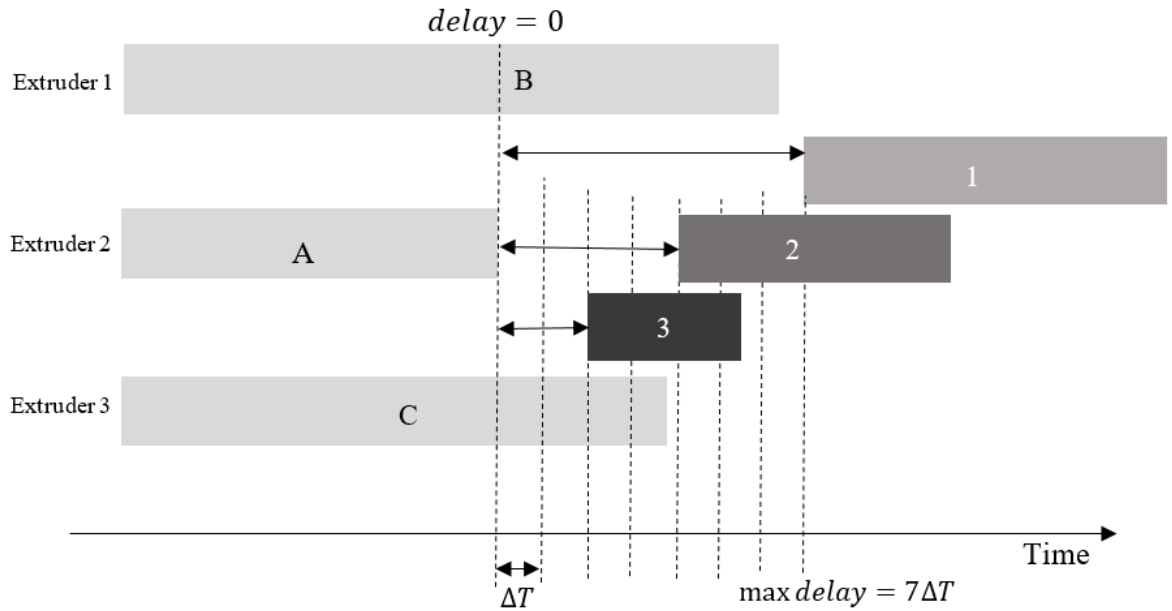


Figure 4.7: Example of finding the minimum delay

4.3.7 Collision Check Algorithm

Collision checking is accomplished by considering the area swept out by each extruder over a given time period Δt . In general, this area is a polygon whose specific geometric features are determined by the size and trajectory of the extruder. A non-zero intersection between these areas for any two sub-paths is taken to mean that a collision exists between those sub-paths during time period Δt . Of course, this is not precisely true – in actuality, a non-zero intersection indicates the existence of potential collision – but if Δt is small, the chances of a false positive is also small. There is a tradeoff, though, in that a smaller Δt increases the number of collision check operations for a given sub-path, which in turn increases computational cost. Further, finding the intersection of general polygons, especially concave polygons which are frequently encountered in this context, can be computationally expensive. For this reason, the area is extended to be the bounding rectangle (Figure 4.8) of the actual swept area for the purpose of computing collisions. Again, this decision represents an attempt to balance computational efficiency versus false positives.

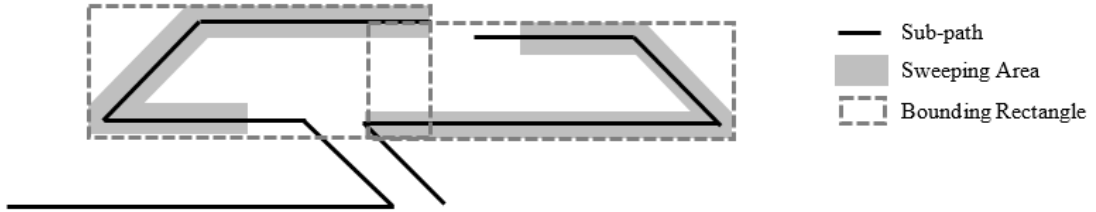


Figure 4.8: False positive example for collision detection. The shadowed area is the true area swept by the extruders.

The procedure for computing the swept area includes three steps:

Step 1: Identify the inspection period. Let $st_{j_1 j_2}$ denote the start time of the inspection time period for any pair of parallel processed sub-paths j_1 and j_2 , which equals the latest start time of the two sub-paths. Let $ed_{j_1 j_2}$ denote the end time, which is the earliest completion time between sub-paths j_1 and j_2 . An example is shown in Figure 4.9: Extruder 2 is free and ready to accept sub-path A while the other extruders are processing sub-paths B and C. The inspection time period of sub-paths A and B is $[st_{AB}, ed_{AB}]$, and that of sub-paths A and C is $[st_{AC}, ed_{AC}]$.

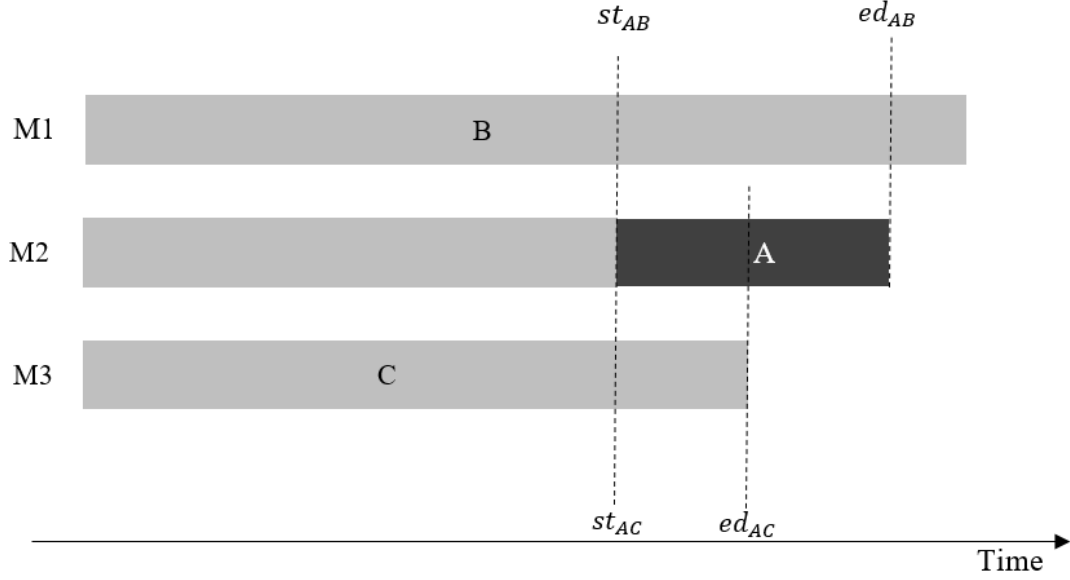


Figure 4.9: Example of identifying inspection time period

Step 2: Divide the inspection time period into several time intervals and find the corresponding path segments that belong to each interval. Letting Δt be the length of each time interval, the inspection time period $[st, ed]$ can be divided into

$$\begin{aligned}
 T_1 : [st, st + \Delta t] & T_2 : [st + \Delta t, st + 2\Delta t] \dots \\
 T_i : [st + (i - 1)\Delta t, st + i\Delta t] & \dots T_N : [st + (N - 1)\Delta t, ed]
 \end{aligned} \tag{4.12}$$

where $N = \lceil \frac{ed_{j_1j_2} - st_{j_1j_2}}{\Delta t} \rceil$. Given sub-path j with G line segments in total and a constant printing speed ν , the corresponding path segment that belongs to any T_i can be found using the procedure described by the following pseudo code. In addition, an example is given in Figure 4.10.

Algorithm 1 Collision Check Algorithm

```
% Let  $D_g$  denotes the length of segment  $g$ ,  $(SX_g, SY_g)$  denotes its start point, and  $(EX_g, EY_g)$ 
denotes its end point.  $g = 1, 2, \dots, G$ 
% Assume the start printing time of sub-path  $j$  is  $ST_j$ 
% Sweeping area of  $T_1$ :
% The printed length of line segment before  $st$  is  $\Delta DS = (st - ST_j)\nu$ 
 $g = 1$ ;
while  $\Delta DS > D_g$  do
     $\Delta DS = \Delta DS - D_g$ ;
     $g + 1 \rightarrow g$ ;
% When iteration ends,  $g = gs$ . Calculate the start point  $P$  of sweeping area
 $X_P = SX_g + \Delta DS \frac{EX_g - SX_g}{D_g}$ ;
 $Y_P = SY_g + \Delta DS \frac{EY_g - SY_g}{D_g}$ ;
 $D_g = \text{dist}[(X_P, Y_P), (EX_g, EY_g)]$ ;
% the length of segment to be covered in  $T_1$  is  $\Delta D = \Delta t \cdot \nu$ 
for  $i = 1 : N$  do
    while  $\Delta D > D_g$  do
         $\Delta D = \Delta D - D_g$ ;
        Add segment  $g$  to set  $T_1$ ;
         $g + 1 \rightarrow g$ ;
    % calculate the end point  $Q$ 
     $X_Q = SX_g + \Delta D \frac{EX_g - SX_g}{D_g}$ ;
     $Y_Q = SY_g + \Delta D \frac{EY_g - SY_g}{D_g}$ ;
     $D_g = \text{dist}[(X_Q, Y_Q), (EX_g, EY_g)]$ ; % remaining length of segment  $g$ 
    %  $Q$  becomes the new start point
     $(X_Q, Y_Q) \rightarrow (X_P, Y_P)$ ;
```

In the example, P is the start point of line segment in interval T_1 , which is on segment $g : S_g \rightarrow E_g$. The remaining length is $D_g = \text{dist}(P, E_g)$. After the first iteration, $\Delta D = \Delta t \cdot \nu - D_g - D_{g+1}$ and $g = g + 2$. Thus, end point Q is on segment $g + 2 : S_{g+2} \rightarrow E_{g+2}$. The coordinates of Q would be $(X_Q = SX_{g+2} + \Delta D \frac{EX_{g+2} - SX_{g+2}}{D_{g+2}}, Y_Q = SY_{g+2} + \Delta D \frac{EY_{g+2} - SY_{g+2}}{D_{g+2}})$. Line segment in T_1 should include $P \rightarrow E_g, S_{g+1} \rightarrow E_{g+1}$ and $S_{g+2} \rightarrow Q$.

Step 3: Build the bounding rectangle to form the extended sweeping area. In step 2, all the line segments that belong to any interval T_i are found. The boundary of the bounding rectangle should be the extreme points of the line segments, offset outward by half of the extruder dimension. For example, Figure 4.11. shows the continued example. The extreme points are indicated by red dots. If the extruder dimension is $2r$, then the upper right boundary of the bounding rectangle would be

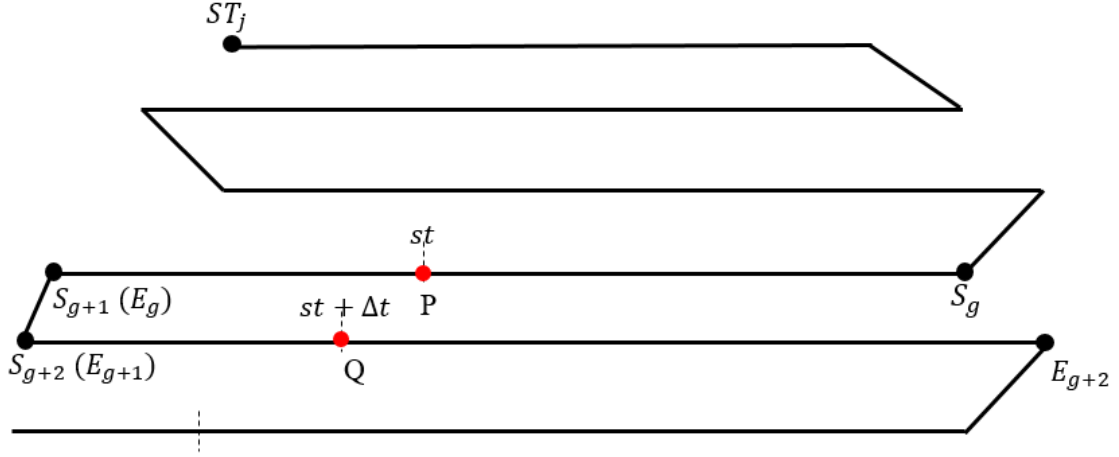


Figure 4.10: Example of identifying inspection time period

$X_P + r$ and $Y_P + r$, respectively. The collision inspection for each pair of sub-paths tests whether the bounding rectangles for time interval T_i intersect. Intersection detection is similar to the third preprocessing step for calculating the overlapped area in the MOA method.

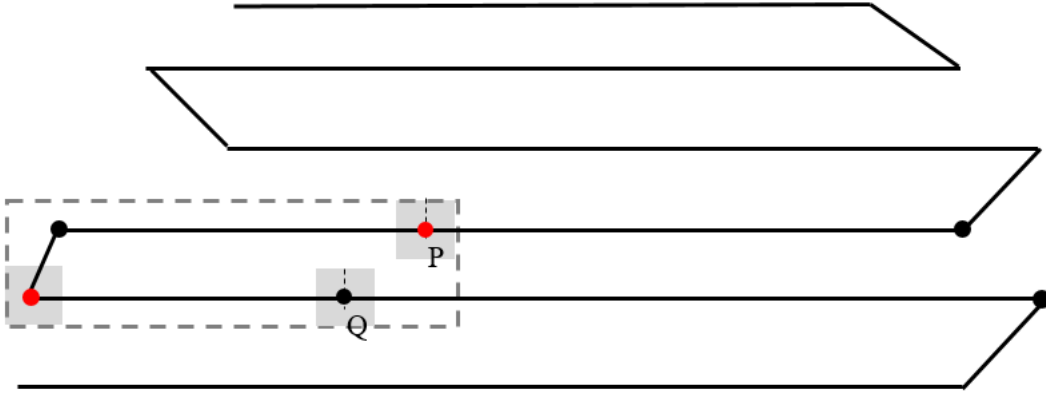


Figure 4.11: Finding the bounding rectangle to form the extended sweeping area

4.4 Result and Discussion

The proposed heuristics are implemented in Matlab on the three layers described in Section 4.3.2 and their performance is compared to the two baseline algorithms. The first case (Section 4.4.1) is used for demonstrating the LPT and MOA algorithms as well as the methods for comparing their performance. The second and third cases (Sections 4.4.2 and 4.4.3) are presented to com-

pare the relative merits of the proposed heuristics in handling layers with divergent (Case 2) or uniform (Case 3) lengths of sub-paths. In these studies, the sub-paths (including both perimeters and rasters) are generated by the same slicer. The printing speed is 2 units/sec, and the time interval Δt for collision checking is 0.1 sec. For comparing the performance of the algorithms, the computational time and makespan (time to print one layer) are of primary interest. In particular, the computational time is calculated based on a desktop computer with an Intel core i7 CPU. In addition, the total delay introduced into the process (as per Section 4.3.6) is also presented, as it adversely affects the computational time.

It is worth pointing out that since MOA is applied only to raster sub-paths, only the raster sub-paths are considered in the second and third cases for performance comparison. For a conceptual investigation, different ratios of extruder size E to layer size W are also considered in the two cases when comparing the performance of the scheduling algorithms. As this ratio increases, finding a collision-free scheduling solution becomes more difficult. For illustration purposes, the size of extruder is changed while the layer size, defined as the layer's minimum side length W , is held constant. However, in practice, with an extruder of fixed size, the ratio varies as the layer size changes.

4.4.1 Case 1

The first layer is a 20 unit \times 20 unit circle that includes some concave features. The slicer identified 7 perimeter paths and divided the infill raster into 4 sub-paths (i.e., 11 paths in total as shown in Figure 4.12). The width of the extruder's bounding rectangle E is 2 units.

Under the given configuration, the LPT and MOA method was applied for up to six extruders, with the resulting makespans and computational times presented Table 4.2. While makespan is reduced by concurrent processing, the improvement asymptotically reaches a limit as the number of extruders is increased. Meanwhile, computational time tends to increase with the number of extruders with a few exceptions. Without declaring an optimal number, which would involve some situationally dependent weighting between the two parameters, three extruders appears to

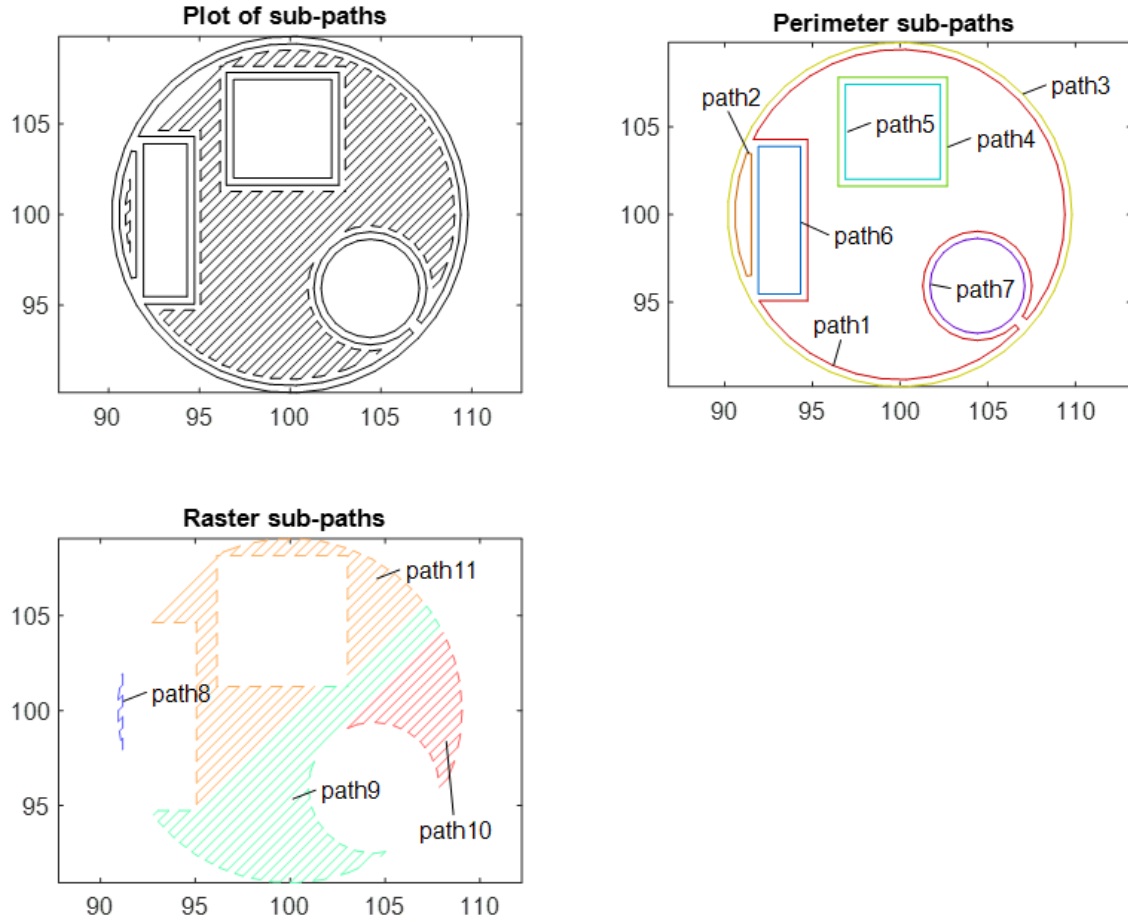


Figure 4.12: Sub-paths of the layer (where axis values indicate the position of layer on the workspace)

be a reasonable compromise for case 1. Similar analyses for cases 2 and 3 indicate three or four extruders. For the purposes of this paper, a three-extruder configuration is adopted throughout to facilitate comparison between algorithms and cases.

Table 4.2: Performance of LPT and MOA for Case 1 vs Number of Extruders)

Number of extruders		1	2	3	4	5	6
Makespan (s)	LPT	318	168	135	131	131	131
	MOA	318	176	135	131	131	131
Computational Time (s)	LPT	0	0.325	2.61	6.81	6.74	6.99
	MOA	0	1.99	1.47	4.24	4.44	4.68

Figure 4.13 shows the path allocations and scheduling results for the three extruders. For bet-

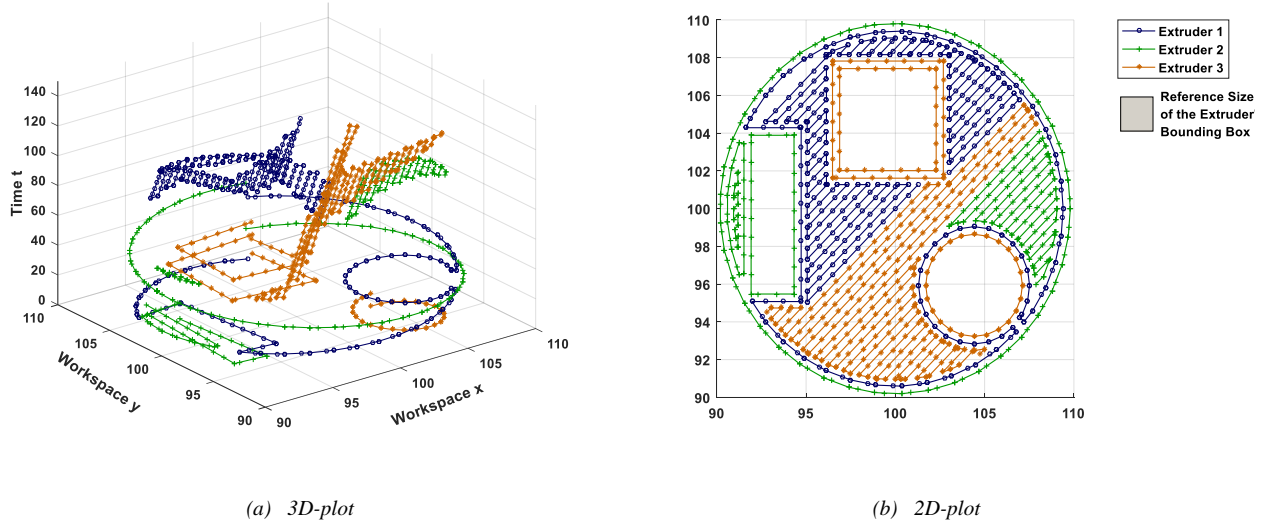


Figure 4.13: Graphic illustration of the scheduling result from the LPT heuristic (a) The toolpath path is shown in time (vertical axis) indicating no intersection between extruder trajectories. (b) The relative size of the extruder's bounding rectangle to the layer is provided.

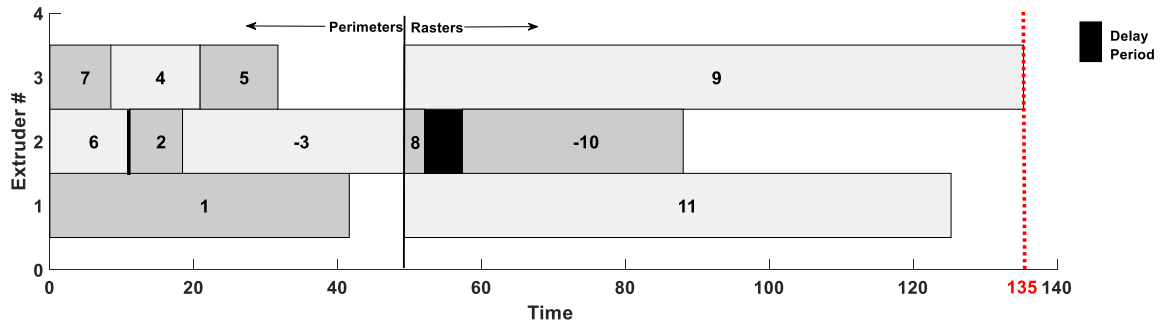
ter visualization, both two-dimensional and three-dimensional plots are provided to illustrate the path distribution and progression of each extruder over time. Clearly, the collision-free LPT algorithm effectively avoids collisions between the three extruders in this concurrent AM environment. Similar results for the MOA and both baseline methods may be obtained similarly.

Figure 4.14 shows the scheduling results of all four algorithms visualized in the time domain in order to explicitly show their differences (negative signs indicate the paths were reversed to avoid collision). As previously mentioned, all perimeters should be completed before rastering starts, and MOA is applicable only to raster sub-paths, so differences between the LPT and MOA exit only for sub-paths 8–11. The detailed results of scheduling performance for the four algorithms are shown in Table 4.3.

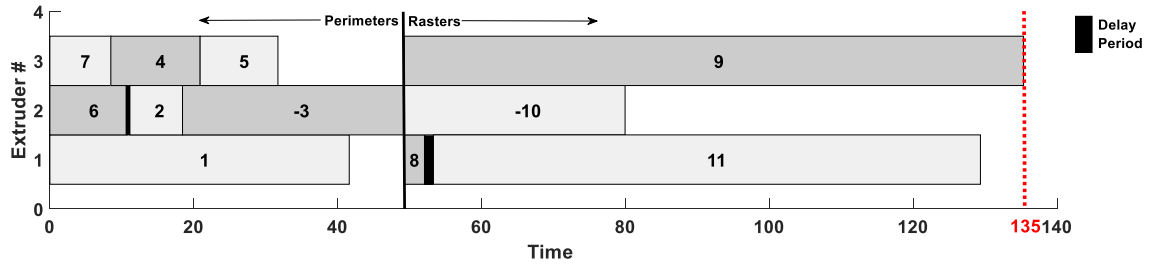
Table 4.3: Performance Comparison among LPT, MOA, NS and HSS for the Simple Layer

	LPT	MOA	NS	HSS
Cmax (sec)	135	135	245	266
Total delay in perimeter (sec)	0.3	0.3	54.0	64.8
Total delay in raster (sec)	5.2	1.2	73.2	83.2
Computational time (sec)	2.90	1.47	0.56	0.38

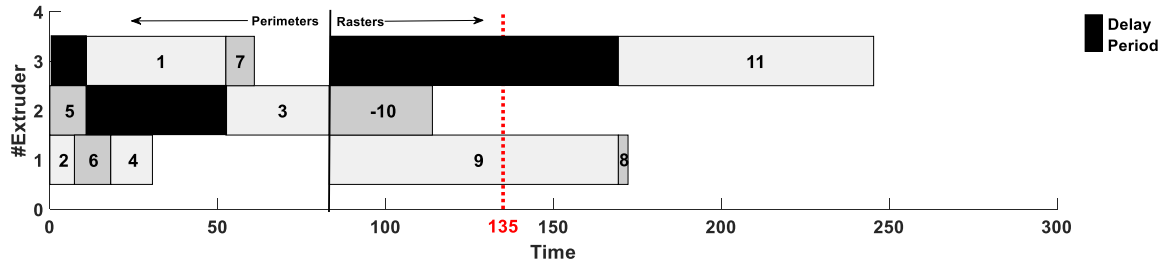
Algorithm performance may be evaluated by comparing to both the single-extruder makespan



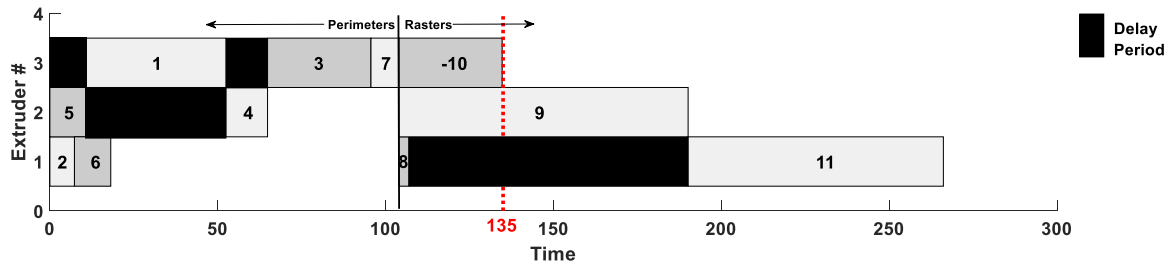
(a) LPT result



(b) MOA result



(c) NS result (the dashed line shows the C_{max} of LPT and MOA)



(d) HSS result (the dashed line shows the C_{max} of LPT and MOA)

Figure 4.14: Result of LPT, MOA, NS and HSS for 3-extruder scheduling. Negative signs indicates that the path was reversed to avoid collision. Black indicates delay introduced to avoid collision.

and the theoretical lower bound of the multi-extruder makespan (single-extruder makespan divided by the number of extruders). For case 1, the single-extruder makespan is 318 s, and the theoretical lower bound is 106 s. Thus, the results represent a 56% improvement over the single-extruder makespan and are 24% slower than the theoretical minimum.

Total delay, defined as the cumulative length of all delay periods on all extruders, is reported in Table 4.3 as measure of mutual interference between extruders. Delay results from the algorithms' search for a future time at which to schedule an unassigned sub-path. This search involves numerous collision checks, potentially causing a significant increase in computational time. Total delay time assists in differentiating between long computational times are the result of a large layer versus those caused by an overly crowded environment. More delay time in a layer does not necessarily imply an increase makespan, which may seem counterintuitive at first glance. A simple example can be found in Figure 4.14 (a) and (b): although the two schedules have different total delay times, their makespan is identical since the delay time for extruder 1 is identical.

4.4.2 Case 2

Case 2 (Figure 4.15) consists of 17 raster sub-paths with high variability in length. The minimum side size W of the layer is 25 units. The width of the extruder's bounding rectangle E is incrementally increased from 4 units to 12 units to evaluate the algorithms as collisions become more likely. The performance measures of the algorithms are presented in Figure 4.16. By means of comparison, the single-extruder makespan for this layer is 265 seconds.

In this case, the variance in length is significant, so when the ratio of extruder size to layer size is small, LPT performs better than MOA in minimizing the makespan. In other words, the dominating factor that influences the makespan in this case is the sub-path lengths. The longer makespan of MOA is attributed to the unbalanced schedule where some long sub-paths are scheduled at the end of the process. However, as the E/W ratio increases, LPT loses its advantage as additional opportunities for collision introduce long delays and long computational time becomes the dominating factor. As a result, the performance of MOA becomes much better than LPT, not

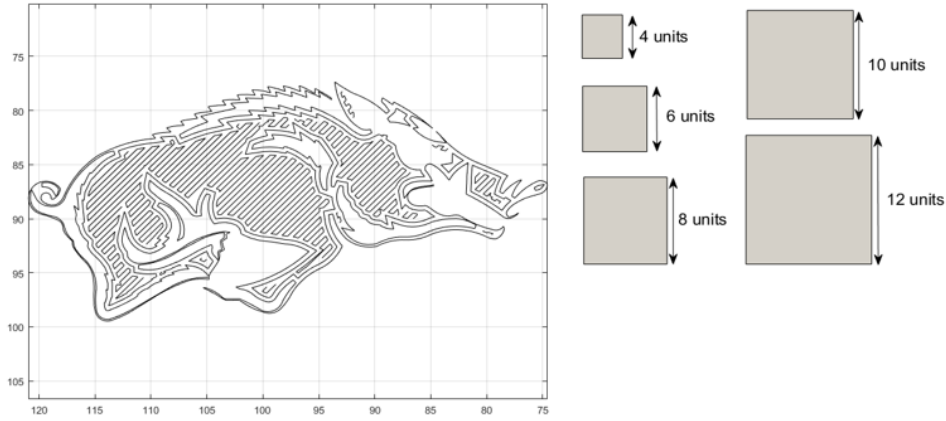


Figure 4.15: The layer overview of Case 2.

only in minimizing the makespan, but also in reducing the computational time. Both LPT and MOA outperform the baseline methods with respect to makespan under all conditions.

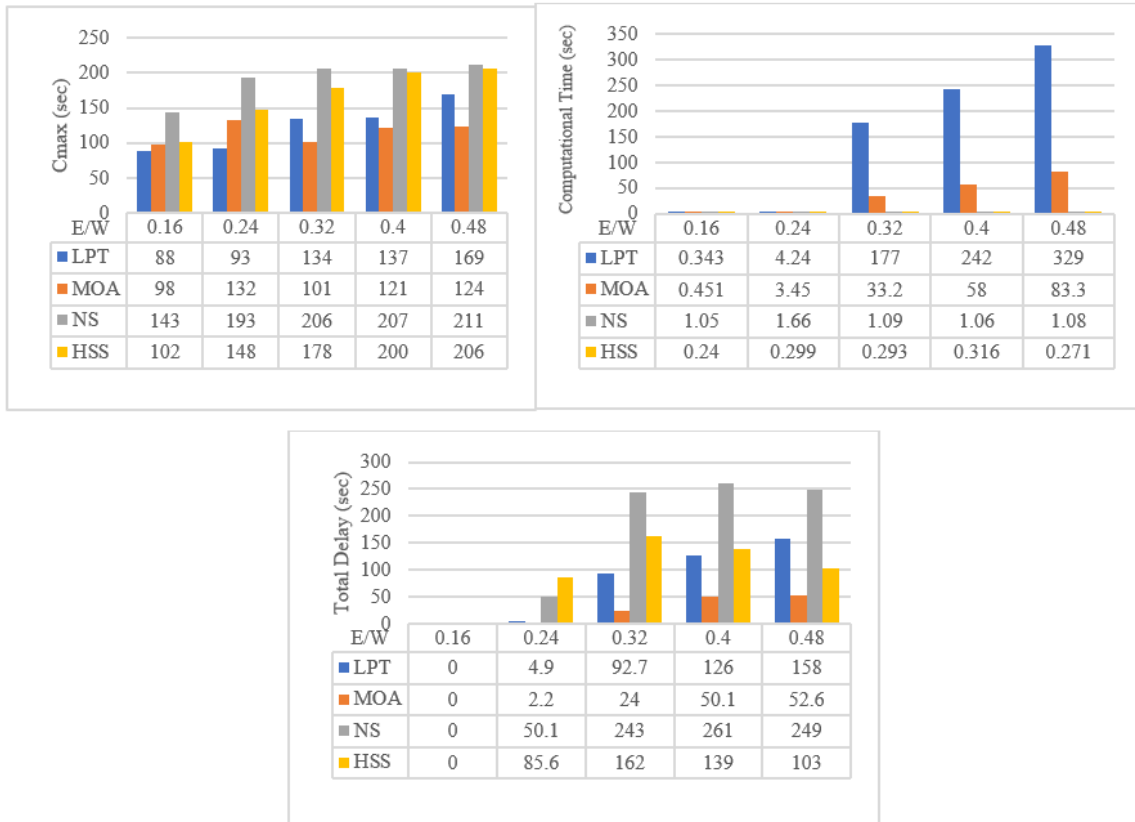


Figure 4.16: Performance Comparison among LPT, MOA, NS, HSS for Case 2

The single-extruder makespan for case 2 is 264 s and the theoretical minimum is 88 s. The

results show a decrease from the single-extruder makespan of between 53% and 67%, and are 0% to 41% slower than the theoretical minimum.

4.4.3 Case 3

The third case study considers the layer shown in Figure 4.17. Sixteen square blocks lead to a small variance in sub-path lengths. The minimum side size W of the layer is 60 units. In order to study the performance of the algorithms as the probability of collisions increases, the width of the extruder is incrementally increased from 10 units to 24 units. The performance of the two heuristics is summarized in Figure 4.18. By means of comparison, the single-extruder makespan for this layer is 2358 seconds.

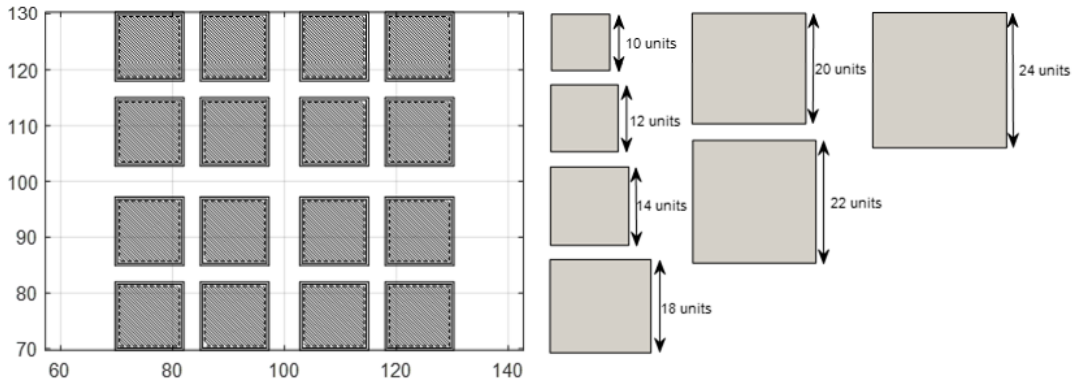


Figure 4.17: The layer overview of Case 3

In this case, all the sub-paths have almost the same length, with variance less than 1 unit. Figure 18 shows that LPT loses its advantage and becomes inferior as the ratio E/W increases, while on the other hand, the performance of MOA is quite stable with a makespan of 884 seconds. In addition, the computational times of both LPT and MOA increase as the ratio increases, but the increase in LPT is much more significant than that of MOA.

It is notable that even though the baseline methods underperform LPT and MOA in minimizing the makespan as the ratio E/W increases, the difference is not as significant as it was in the first two cases. This is mainly due to the equal length of the sub-paths. In this case, the maximum possible

delay is the fabrication time of one square block (sub-path). As the delay of LPT increases, the makespan will approach that of the baseline method.

The single-extruder makespan for case 3 is 2358 s and the theoretical minimum is 786 s. The results show a decrease from the single-extruder makespan of 63%, and are 12% slower than the theoretical minimum.

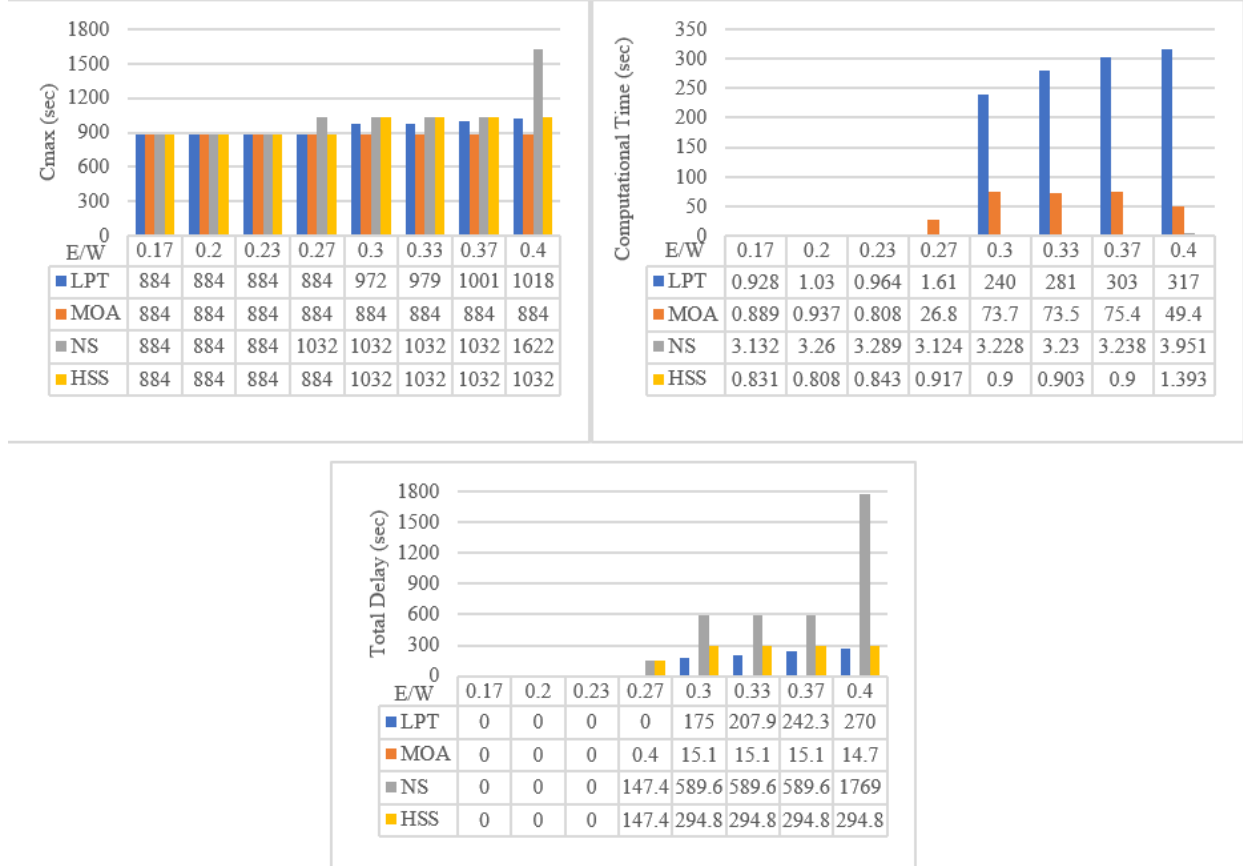


Figure 4.18: Performance Comparison among LPT, MOA, NS and HSS for Case 3

4.4.4 Discussion

The case studies show that the proposed LPT and MOA algorithms reduce fabrication time significantly as compared to the two baseline scheduling approaches, though this performance advantage comes with a computational cost. Given that fabrication time usually far exceeds digital preprocessing time, and that adding computational resources is usually cheaper than adding fabrication

resources, this represents an attractive tradeoff.

The case studies also suggest that two key layer characteristics determine which heuristic will perform better: layer size relative to extruder size (E/W ratio) and the variability of sub-path processing times. Case 2 indicates that the LPT algorithm tends to perform better when the layer is large relative to the extruder size and there is a wide variety of sub-path processing times. These circumstances minimize the number of potential collisions and increase the probability of finding a compatible sub-path that may be scheduled immediately, thus achieving the proven efficiency of the LPT algorithm with reasonable computational cost. If the layer is small relative to the extruder, as in Cases 2 and 3 where the E/W ratio is large, or if the variability in sub-path lengths is low, as in Case 3, extruder utilization and computing time both suffer under LPT. This is due to the fact that LPT does not consider the probability of collision between two paths, and it can expend considerable computing resources in a brute-force effort to schedule the next-largest sub-path, even if it has extensive interference with those already scheduled. The results of Cases 2 and 3 further indicate that the MOA algorithm is more effective and computationally efficient when the potential for collision is high due to a small layer or highly uniform sub-path lengths. This is attributed to the fact that the MOA algorithm gives preference to scheduling concurrent paths with low probability of collision, thereby avoiding excessive searching for minimum delays. But since the MOA algorithm does not consider the processing time for each sub-path, it can postpone long sub-paths toward the end. With fewer unscheduled sub-paths to pick from, finding one that can be concurrently scheduled with this long sub-path can be difficult or impossible, thus increasing makespan and/or computational load.

Based on these results, a hybrid heuristic was developed with the hope of achieving superior performance by combining the best attributes of both algorithms. This algorithm defaults to LPT, thereby assigning higher priority to longer sub-paths. When the algorithm is unable to immediately schedule the next-longest sub-path, it temporarily switches to MOA instead of spending excessive time searching for an acceptable delay. Unfortunately, the hybrid algorithm did not show a performance advantage in the above case studies, and in some instances led to inferior performance.

Both heuristics potentially add time to the digital preproduction process, and the practicality of this should be addressed. The processing times reported herein are based on executing the algorithms in Matlab, and they could be significantly reduced by optimizing the code and switching to a compiled language. It is also possible to parallelize the scheduling and production activities by computing the tool path schedule for layer i while printing layer $(i-1)$.

4.5 Conclusion

The objective of this research was to create a general toolpath allocation and scheduling methodology to facilitate parallel scheduling of independently-operating extruders for concurrent FFF. This challenge was formulated as a scheduling problem with collision constraints, and an optimization model was developed to minimize fabrication time. Two heuristics were proposed to find approximate but efficient solutions to this otherwise NP-hard problem. One heuristic is a novel modification of the well-established LPT scheduling algorithm, adapted to handle collision prevention. The other (MOA) is a novel algorithm that inherently attempts to concurrently schedule sub-paths with low mutual interference. The application, effectiveness, and computational cost of these heuristics have been demonstrated for three AM layers, and the relative merits of each has been examined and compared with two baseline scheduling approaches.

An advantage of approaching the allocation of toolpath as a scheduling problem is that it decouples toolpath allocation from toolpath planning. The heuristics developed in this paper work with any single-extruder toolpath planning algorithm. This allows AM equipment and software manufacturers to integrate the heuristics directly with their own proprietary toolpath generation code without modification. End users who can access AM G-code (as with the popular Marlin firmware, for example) can implement the heuristics even if they are using closed-source toolpath planning software. These heuristics will also not need to be modified or updated to accommodate new toolpath planning algorithms as they are developed.

Concurrent extruder scheduling can significantly decrease the cycle time for FFF and mitigate its scalability issues. Fabrication times for the layers considered in this paper were reduced by

37% to 66% using three extruders. For large-scale 3D printing, where there is ample space to accommodate multiple extruders, significant reduction in fabrication time is achievable. At present, fabricating large parts with a single extruder implies that layer thickness must scale with part size to keep printing time reasonable. The resulting surface finish on a workpiece the size of an automobile, for example, is so poor that the required post-process finishing operations offset most, and sometimes all, of the benefits of AM. In concurrent FFF, however, the number of extruders can scale with part size, thereby allowing reasonable cycle times with far better layer resolution. Combined with current research on FFF for fiber-reinforced and thermosetting materials, AM of large carbon fiber reinforced polymer parts such as aircraft components or boat hulls is conceivable.

References

- Almasri, M. M., Alajlan, A. M., and Elleithy, K. M. “Trajectory planning and collision avoidance algorithm for mobile robotics system”. *IEEE Sensors Journal*, 16(12):5021-5028, 2016.
- Arbib, C., and Rossi, F. “Optimal resource assignment through negotiation in a multi-agent manufacturing system”. *IIE Transactions*, 32(10):963-974, 2000.
- Ata, A., and Myo, T. “Collision-free trajectory planning for manipulators using generalized pattern search”. *International Journal of Simulation Modelling*, 5(4):145-154, 2006.
- Autodesk. “Project Escher”, URL <http://projectescher.com/>, 2016.
- Azizoglu, M. E. R. A. L., and Kirca, O. M. E. R.. “Scheduling jobs on unrelated parallel machines to minimize regular total cost functions”. *IIE Transactions*, 31(2):153-159, 1999.
- Babu, V., Kaplan, R., and Ishikawa, T. *Optimized Cluster Tool Transfer Process and Collision Avoidance Design*. U.S. Patent Application No. 11/338/323, 2006.
- Belkaid, F., Maliki, F., and Boudahri, F. “A branch and bound algorithm to minimize makespan on identical parallel machines with consumable resources”. *Advances in Mechanical and Electronic Engineering*, 176:217-221, 2012.
- BetAbram. “The first 3D house printers manufacturer”, URL <http://betabram.com/>, 2016.
- Brucker, P., Hurink, J., and Werner, F. “Improving local search heuristics for some scheduling problems—I”. *Discrete Applied Mathematics*, 65(1):97-122, 1996.
- Brucker, P., Hurink, J., and Werner, F. “Improving local search heuristics for some scheduling problems”. *Part II, Discrete Applied Mathematics*, 72(1):47-69, 1997.
- Chen, B., Potts, C.N., and Woeginger, G.J. *Handbook of Combinatorial Optimization*, Springer, pages 1493-1641, 1998.

- Chiaverini, S., Oriolo, G., and Walker, I. D.. “Kinematically redundant manipulators”. *Springer handbook of robotics*, pages 245-268, 2008.
- Chong, J., Ong, S., and Nee, A. “Robot programming using augmented reality: an interactive method for planning collision-free paths”. *Robotics and Computer-Integrated Manufacturing*, 25(3): 689-701, 2009.
- Coffman, J., Edward G, Garey, M. R., and Johnson, D. S. ”An application of bin-packing to multiprocessor scheduling”. *SIAM Journal on Computing*, 7(1):1-17, 1978.
- Compton, B. G., and Lewis, J. A. “3D-printing of lightweight cellular composites”. *Advanced Materials*, 26(34):5930-5935, 2014.
- Cultivate3d. “The Beast”, URL <http://www.cultivate3d.com/>.
- Daachi, B., Madani, T., and Benallegue, A.. “Adaptive neural controller for redundant robot manipulators and collision avoidance with mobile obstacles”. *Neurocomputing*, 79:50-60, 2012.
- David, J., Valencia, R., and Iagnemma, K. “Task assignment and trajectory planning in dynamic environments for multiple vehicles”. In *RSS 2016 Workshop on Task and Motion Planning*, Ann Arbor, Michigan, 2016.
- Erectorbot. “Large format 3D printers”. URL <http://www.erectorbot.com/>, 2016.
- Gao, W., Zhang, Y., and Ramanujan, D. “The status, challenges, and future of additive manufacturing in engineering”. *Computer-Aided Design*, 69:65-89, 2015.
- Garey, M. R., and Johnson, D. S. “Strong NP-completeness results: motivation, examples, and implications”, *Journal of the ACM (JACM)*, 25(3):499-508, 1978.
- Gibson, I., Rosen, D.W., and Stucker, B. *Additive Manufacturing Technologies - 3D Printing, Rapid Prototyping, and Direct Digital Manufacturing*, Springer, 2014.
- Graham, R. L. “Bounds on multiprocessing timing anomalies”, *SIAM Journal on Applied Mathematics*, 17(2):416-429, 1969.
- Hasan, M., and Gavrilova, M. L. “A geometric approach to drill path collision avoidance”. In *2010 International Symposium on Voronoi Diagrams in Science and Engineering*, IEEE, pages 244-253, 2010.
- Hawley, R. C. *Method of Manufacturing a Composite Reinforcing Structure*. U.S. Patent No. 4/439/387, 1984.
- Jeevamalar, J., and Ramabalan, S. “Optimal trajectory planning for autonomous robots - a review”. *International Conference On Advances In Engineering, Science And Management*, IEEE, pages 269-275, 2012.
- Jin, Y., He, Y., and Xue, G. “A parallel-based path generation method for fused Deposition modeling”. *The International Journal of Advanced Manufacturing Technology*, 77(5-8): 927-937, 2015.

- Jose, K., and Pratihari, D. K.. “Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods”. *Robotics and Autonomous Systems*, 80:34-42, 2016.
- Kim, G., and Oh, Y. “A benchmark study on rapid prototyping processes and machines: quantitative comparisons of mechanical properties, accuracy, roughness, speed, and material cost”. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 222(2):201-215, 2008.
- Klee, S. D., Gemignani, G., and Nardi, D. “Multi-robot task acquisition through sparse coordination”. *2015 International Conference on Intelligent Robots and Systems*, IEEE, pages 2823-2828, 2015.
- Lee, B. H., and Lee, C. G. “Collision-free motion planning of two robots”. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(1):21-32, 1987.
- Lenstra, J. K., Shmoys, D. B., and Tardos, É. “Approximation algorithms for scheduling unrelated parallel machines”. *Mathematical Programming*, 46(1-3):259-271, 1990.
- Li, X., Chan, F. T., and Chung, S. “Optimal multi-degree cyclic scheduling of multiple robots without overlapping in robotic flowshops with parallel machines”. *Journal of Manufacturing Systems*, 36:62-75, 2015.
- Milkert, H.. “3&Dbot Mobile 3D Printer Has No Print Volume Limitations”. URL <https://3dprint.com/15508/3dbot-mobile-3d-printer/>, 2014
- Miller, R. “Autodesk Looks to Future of 3D Printing with Project Escher”. URL <https://techcrunch.com/2016/04/10/autodesk-looks-to-future-of-3d-printing-with-project-escher/>, 2016.
- Miura, J., and Shirai, Y.. “Parallel scheduling of planning and action for realizing an efficient and reactive robotic system”. *7th International Conference on Control, Automation, Robotics and Vision*, pages 246-251, 2002.
- O’Donnell, P. A., and Lozano-Pérez, T. “Deadlock-free and collision-free coordination of two robot manipulators”, *Proceeding of the IEEE International Conference on Robotics & Automation (ICRA)*, volume 89, pages 484-489, 1989.
- Oak Ridge National Laboratory. “3D Printed Shelby Cobra”, Retrieved from <http://web.ornl.gov/sci/manufacturing/shelby/>, 2016.
- Pandey, A., and Parhi, D. R. “Multiple mobile robots navigation and obstacle avoidance using minimum rule based ANFIS network controller in the cluttered environment”. *International Journal of Advanced Robot Automn*, 1(1):1-11, 2016.
- Pax, C. E. *Multi-Extruder*, U.S. Patent No. 8/512/024, 2013.
- Pindeo, M.. *Scheduling: Theory, Algorithms, and Systems*, Springer, 2012.
- Potts, C. N. “Analysis of a linear programming heuristic for scheduling unrelated parallel machines”. *Discrete Applied Mathematics*, 10(2):155-164, 1985.

- Schumacher, N. D., Patterson, N. J., and Harris, K. L. *3D printer and printhead unit with multiple filaments*, U.S. Patent No. 8/827/684, 2014.
- Sevenson, B. “Shanghai-based WinSun 3D Prints 6-Story Apartment Building and an Incredible Home”, URL <https://3dprint.com/38144/3d-printed-apartment-building/>, 2015.
- Švestka, P., and Overmars, M. H.. “Coordinated path planning for multiple robots”. *Robotics and Autonomous Systems*, 23(3):125-152, 1998.
- van de Velde, and Steef L. “Duality-based algorithms for scheduling unrelated parallel machines”. *ORSA Journal on Computing*, 5(2):192-205, 1993.
- Vurpillat, J., “3D Demonstrators Designed for Bigger, Lighter Auto and Aerospace Parts”. URL <http://blog.stratasys.com/2016/08/24/infinite-build-robotic-composite-3d-demonstrator/>, 2016.
- Wu, S. H., Fuh, J. Y., and Nee, A. Y. C.. “Concurrent process planning and scheduling in distributed virtual manufacturing”. *IIE Transactions*, 34(1):77-89, 2002.
- Yang, X., Wang, H., and Zhang, C.. “A method for mapping the boundaries of collision-free reachable workspaces”. *Mechanism and Machine Theory*, 45(7):1024-1033, 2010.

5 Summary

Additive manufacturing (AM) has attracted increasing attention and interest of different industrial sectors for its ability of producing complex geometries outside of traditional production environments. However, there are still some technical challenges that need to be solved to reach AM's full potential. In this dissertation, three important and challenging issues in terms of geometric quality inspection and process efficiency are discussed in detail. Different methodologies applying knowledge of computer vision, statistics, and optimization modeling are proposed in this dissertation to fill several research gaps in offline inspection, online inspection, and concurrent 3D printing.

Among the studies with regard to geometric inspection, this dissertation is the first attempt to develop a fast, feature-based geometric offline inspection technique that does not require alignment before inspection and can be implemented by users without specialized training in inspection according to geometric dimensioning and tolerancing conventions. In addition, this dissertation fills a research gap of state-of-the-art studies regarding multi-resolution analysis of point cloud data for automatically quantifying different types of error inherent in manufacturing, measuring, and part alignment under an online inspection paradigm. Furthermore, this dissertation is among the pioneer studies that first propose and investigate the multi-extruder concurrent AM processes to improve fabrication efficiency while maintaining geometric quality. The contributions of this dissertation have been revealed from both theoretical and practical perspectives based on simulation and physical experiments.

In the future, the methods and models proposed in this dissertation can be integrated into physical AM systems to inspect, monitor, and optimize the process based on real-time feedback. Although the proposed methodologies are discussed under AM application in this dissertation, they can be easily accommodated to other manufacturing processes which are faced with similar technical challenges of the geometric inspection and process efficiency improvement.