

**Localized Heterogeneities in Metal Powder Bed Additive
Manufacturing – Investigating Process-Structure Relationships
Leveraging Scan Path Data**

A Dissertation Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

William Henry Halsey
December 2024

DEDICATION

For Breanna, Oliver, Henry, and Felicity. If I never had started this journey, I would not have you; because of you, I was determined to finish this journey.

ACKNOWLEDGEMENTS

I would like to give my profound thanks to Dr. Vincent Paquit for, among other things, serving as my technical advisor and for assisting in funding this work. Also, thanks to Dr. Russell Zaretski for serving as my committee chair and for your guidance during this Ph.D.

I would also like to thank all the great people that I have been privileged to work with at the Manufacturing Demonstration Facility. Finally, a sincere thank you to everyone who asked how my Ph.D. was going. I know that you asked because you cared.

Manuscripts contained herein have been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). Research was sponsored the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, Advanced Manufacturing Office.

PREFACE

This dissertation is to fulfill the requirements for a Ph.D. in Data Science and Engineering. The hope is that it provides an interesting and meaningful addition to the diverse, dynamic, and growing field of advanced manufacturing. The simultaneous confluence of new manufacturing process, better and cheaper sensors, increasingly powerful compute resources, and growing machine learning capabilities have opened opportunities for research that could lead to new materials, increased efficiency, faster design, and less waste in manufacturing. Nowhere is this more true than in additive manufacturing. The work contained herein offers just a glimpse of some of the possibilities that lie at the intersection of manufacturing and data science.

ABSTRACT

Before metal powder bed fusion additive manufacturing may be deployed on a wide scale, the reliability and repeatability of the process must be improved. Currently additively manufactured components suffer from mechanical anisotropy and often-poor performance due to the presence heterogeneities in microstructure and defects throughout the component. Work must be undertaken to help determine how the process itself contributes to the evolution of these heterogeneous structures. This is true for both electron beam and laser powder bed fusion technologies. The dynamism of the process is largely driven by the scan path that the heat source traverses to fuse feedstock into solid material - introducing nuanced thermal cycling and histories to different regions. The complexity of the process, time and space scales under consideration, and the increasing availability of data necessitate development of additional process-structure analyses to complement those in existence already. Two distinct tasks will be performed to explore localized heterogeneities and their connection to the powder bed process utilizing scan path data. The first task will investigate how the scan path itself may be manipulated in order to induce desired location specific microstructure in electron beam powder bed fusion utilizing genetic algorithm and concurrent greedy line search optimization algorithms. The second task will leverage scan path data to localize temporal process data. Specifically, the scan path data will be utilized to spatially map temporal data about gas flow and oxygen content of laser powder bed fusion builds. The data, in its new format, will then be used to predict the local prevalence of soot which has been shown to lead to mechanical anisotropy.

TABLE OF CONTENTS

INTRODUCTION	1
What is Metal Powder Bed Fusion Additive Manufacturing?	1
Additive Manufacturing.....	1
Metal Additive Manufacturing	1
Powder Bed Fusion	2
What are Spatially Localized Heterogeneities?	8
Localized Microstructure	8
Localized Flaws	10
What is a Scan Path?.....	12
What are Process-Structure Relationships?	15
Discussion	19
CHAPTER I Geometry-Independent Microstructure Optimization for Electron Beam	
Powder Bed Fusion	20
Abstract	21
Introduction.....	22
Methodology	24
Heat Transfer and Microstructure Model	25
Genetic Algorithm	28
Concurrent Greedy Line Search.....	36
Experiments and Results	45
50-50 Geometry	45
Oak Leaf Geometry.....	47
Analysis of Optimization	47
Discussion and Future Work.....	57
CHAPTER II SWAN: Software for Additive Investigation — Common Data	
Representations and Data Fusion.....	63
Abstract	64
Introduction and Background	65
Component Digital Twins for Certification and Qualification	67
Process Data.....	69
Spatial Mapping.....	72
Materials and Methods	75
Temporal, In-Situ Process Data	77
Trajectory Data	77
Spatial Mapping Algorithm	83
Results.....	88
Discussion	93
Common Data Representations and Complex Data Processing Pipelines.....	93
Time Complexity Analysis	94
Key Considerations for Spatial Mapping Operation.....	94
Extensions and Future Work.....	97
Conclusions.....	97

APPENDICES	99
APPENDIX A — Glossary of Scan Path Terms	99
APPENDIX B — Example Rasterized Spatio-Temporal Maps	100
CHAPTER III Localized Defect Detection from Spatially Mapped, In-Situ Process Data with Machine Learning	104
Abstract	105
Introduction	106
Materials and Methods	110
Data Sources and Descriptions	110
XGBoost Background	119
Methodology	124
Results	136
Model Performance	137
Feature Importance	139
Location-Based Model Performance	139
Discussion	139
Analysis	139
Future Work	143
Conclusion	144
CONCLUSION	146
REFERENCES	149
VITA	157

LIST OF TABLES

Table 1. Metal PBF machine and associated commercial technology names. Reproduced from Everton et al.[11].	5
Table 2. Sampling of literature investigating microstructural heterogeneity in L-PBF and EB-PBF as reported by Kok et al.[15].	9
Table 3. Enumeration of localized flaws in AM.	11
Table 4. Simulation parameters.	46
Table 5. Mean absolute pixel error for each optimization scheme for the 50-50 geometry, Figure 17.	48
Table 6. Mean absolute pixel error for each optimization scheme for the oak leaf geometry, Figure 18.	49
Table 7: Data that are available in the Arcam Q10 scan path files.	80
Table 8: Data that are available in Concept Laser M2 scan path files.	82
Table 9: Proposed scan path common data representation.	84
Table 10: Example rasterized spatio-temporal map for an Arcam Q10 trajectory.	100
Table 11: Example rasterized spatio-temporal map for a Concept Laser M2 trajectory.	102
Table 12: Description of in-situ process data used to detect defects for this analysis....	114
Table 13: Description of in-situ process data used to detect defects for this analysis....	118
Table 14: Description of a subset of XGBoost model parameters.	125
Table 15: Description of a subset of XGBoost model parameters.	135
Table 16: Description of a subset of XGBoost model parameters.	135
Table 17: Model performances. The top lines for each input feature set represents the aggregated performance across all four models and respective builds.	138

LIST OF FIGURES

Figure 1. List of AM processes. The first five, in gray, are applicable to metal AM. Reproduced from Sharrat[7].	3
Figure 2. Basic schematic of a PBF AM machine. Reproduced from Grasso and Colosimo[12].	6
Figure 3. Feature resolution of a) EB-PBF versus b) L-PBF. Reproduced from Dryepont et al.[9]	7
Figure 4. Examples of A) heterogeneities in grain size and b) heterogeneities in morphology and orientation. Reproduced from Yoder[30] and Antonysamy et al.[28] respectively.	11
Figure 5. Images of a), b), d) lack-of-fusion and c) keyholing porosity. Reproduced from Bartlett et al.[34]	13
Figure 6. Scan strategies used to determine heat source path in metal AM as seen in the X-Y plane (perpendicular to the build direction): a) unidirectional or concurrent fill, b) bi-directional, snaking, or countercurrent fill, c) island scanning, d) spot melting, e) spot melting contours with snaking fill and f) line melting contours with snaking fill. Reproduced from Sames et al.[6]	14
Figure 7. Example of two different scan paths, generated from the same scan strategy, for two layers of a single build with similar cross-sectional geometry. The authors note how the total melt duration for the lower right rectangles differs significantly between the two scan paths — over five seconds for one and less than a second for the other layer — resulting in significantly different thermal histories. Reproduced from Groeber et al.[17]	16
Figure 8. Process map for stainless steel and EB-PBF. Reproduced from Sames et al.[6]	18
Figure 9. Crossover scheme for spot path order in genetic algorithm.	34
Figure 10. Mutation scheme for spot path order in genetic algorithm.	37
Figure 11. Genetic algorithm.	37
Figure 12. Idealized relationship between spot time and equiaxed fraction.	39
Figure 13. Idealized loss function for points in the two different regions.	40
Figure 14. Concurrent greedy line search.	43
Figure 15. Grid of points to print. Changes to parameters of the scan path over these points will create different regions of microstructure.	43
Figure 16. Intent for microstructure regions for the two experiments. Black will be columnar, and white will be equiaxed.	44
Figure 17. Simulated microstructure of different scan path parameters for 50-50 geometry.	48
Figure 18. Simulated microstructure of different scan path parameters for leaf geometry.	48
Figure 19. Initial and final spot paths for the leaf geometry after genetic algorithm optimization. The color represents the position in the spot order – points that are dark blue are at the beginning, beige in the middle, bright red at the end.	49

Figure 20. Changes to scan path metrics for the leaf after genetic algorithm. The left half shows how the three different metrics (columns) change for each point from the first scan path (row 1) to the optimized scan path (row 2). The right side of the figure shows how the distributions of the metrics over the two predicted microstructure regions change.	54
Figure 21. Changes to scan path metrics for the leaf after concurrent greedy line search.	54
Figure 22. Changes to scan path metrics for the leaf after genetic algorithm then concurrent greedy line search.	55
Figure 23. The Mean Absolute Error for the best solution in each generation for the two geometries. Note that the MAE does not decrease monotonically because it only measures the accuracy of the predicted microstructure in comparison to the desired. The genetic algorithm, however, is optimizing a fitness function that factors both this error and the heuristic about distances between points in the scan path.	60
Figure 24. Data processing steps for building generalizable data pipelines for exploring and processing in-situ data from additive manufacturing using SWAN.	66
Figure 25. Illustration of a unified, voxelized, digital model of a component where data corresponding to a specific location in space is associated with a discrete voxel. ...	70
Figure 26. Examples of temporal, in-situ data captured from (top) a Concept Laser M2 machine and (bottom) an Arcam Q10 machine.	78
Figure 27. Example of trajectory data from an Arcam Q10 machine.	80
Figure 28. Example of trajectory data captured from a Concept Laser M2 machine with Left) non-melting laser motion included and Right) non-melting laser motion removed.	82
Figure 29. Graphical representation of the steps implemented to rasterize trajectory hatch lines.	86
Figure 30. Spatially mapped smoke detection counts for an Arcam Q10 machine with (left) no interpolation and (right) linear interpolation at a pixel resolution of 500 μ m.	89
Figure 31. Spatially mapped top flow rate for a Concept Laser M2 machine with (left) no interpolation and (right) linear interpolation at a pixel resolution of 500 μ m.	89
Figure 32. Spatially mapped photodiode data for a Concept Laser M2 machine at a pixel resolution of 100 μ m.	90
Figure 33. Change in rasterization time as a function of target pixel resolution.	91
Figure 34. Change in rasterization time as a function of number of pixels in target image space.	91
Figure 35. Change in rasterization time as a function of number of scan lines. Times are for a target pixel resolution of 100 μ m.	92
Figure 36. Left: geometry for build 1. Right: geometry for builds 2, 3, and 4. For builds 1 and 2 the gray and red regions represent where laser modules 1 and 2 are used respectively.	111
Figure 37. A plot of a single layer of log file data for 'FlowSensorTop' from build 1. The spikes show where the machine logs several records quickly in succession, while the	

timing between other records is greater. The black vertical lines indicate where the melt phase begins and ends for this layer.	114
Figure 38. Example of trajectory data from build 1 zoomed in on four components. The image shows the melt areas, skywriting, and jump lines, and the visualized data is are the raw digital intensity levels from the on-axis photodiode that was used to determine the component boundaries for spatial mapping.	116
Figure 39. Binary class labels for 'Soot' for each voxel of the build chamber in layer 10 of build 1 as predicted by Peregrine.	120
Figure 40. A sample layer from build 1 includes trajectory data, log data for 'FlowSensorTop', and the subsequent mapped data. The orange highlighted portion of the log data corresponds to the melt phase for the layer as indicated by the 'Exposure started for' and 'Exposure ended for' log file entries.	128
Figure 41. 3D plot of the build volume of spatially mapped process data for "Residual oxygen value in gas loop" with defects overlaid for build 2. Note that this image shows combined soot and debris defects [16], but for the actual analysis only soot was considered.	129
Figure 42. XGBoost feature importances for the twelve build and dataset combinations.	140
Figure 43. Visualization of balanced accuracy per component of each build for the 'Log' input feature set.	141
Figure 44. Spatial distribution of soot prevalence for the four builds.	141

INTRODUCTION

Before an exposition of the performed work begins, several points must be both asked and answered to establish the background and need for this work. The remainder of this section will outline the rationale for focusing this inquiry on metal powder bed fusion additive manufacturing, spatially localized heterogeneities, and scan path data for investigation as well as introduce relevant background information.

What is Metal Powder Bed Fusion Additive Manufacturing?

Additive Manufacturing

The first additive manufacturing (AM) system was developed over thirty years ago[1]. AM may be defined as "a process of joining materials to make objects from 3D model data, usually layer upon layer, as opposed to subtractive manufacturing methodologies"[2]. Possible positive impacts of these technologies include the ability to fabricate complex and intricate geometries and reduced waste of material. Additionally, the considerable flexibility of AM means that it introduces low to no switching costs when changing from manufacturing one geometry with a material to a different geometry or material.

Metal Additive Manufacturing

Metal AM, then, is a subset of AM processes that are suited for processing metals, and it has the potential to impact a wide swath of industries that include aerospace, biomedical, and nuclear[3]–[5]. This is due to the much-touted ability to fabricate components that

are geometrically impossible via other, traditional manufacturing means. Additionally, AM presents the unprecedented opportunity for producing small-batch or one-off, highly customized parts with lower overhead costs.

ASTM standard F2792 defines seven distinct categories of AM[2]. Of that set only a subset are relevant to metal AM[6], [7]. These categories are binder jetting, directed energy deposition (DED), powder bed fusion (PBF), sheet lamination, and material jetting as shown in Figure 1.

Powder Bed Fusion

This research will focus on PBF AM technologies which is defined as “an additive manufacturing process in which thermal energy selectively fuses regions of a powder bed”[2]. A review of metal AM by Sharratt[7] notes that several different terms are used when discussed PBF technologies including laser sintering (LS), selective laser sintering (SLS), direct metal laser sintering (DMLS), electron beam melting (EBM), and selective laser melting (SLM), but the main distinctions are the energy source and the means of consolidation. This work will focus on melt technologies and not sintering. As such, we may subdivide PBF into two additional categories based on the heat source that is used to fuse the powder together. The most common energy source is a laser[7]. The laser imparts energy to the metal powder through radiation[8], and we will denote these technologies as laser PBF (L-PBF). The other existing energy source is an electron beam. The electron beam melts the powder using kinetic energy[8] and will be denoted as

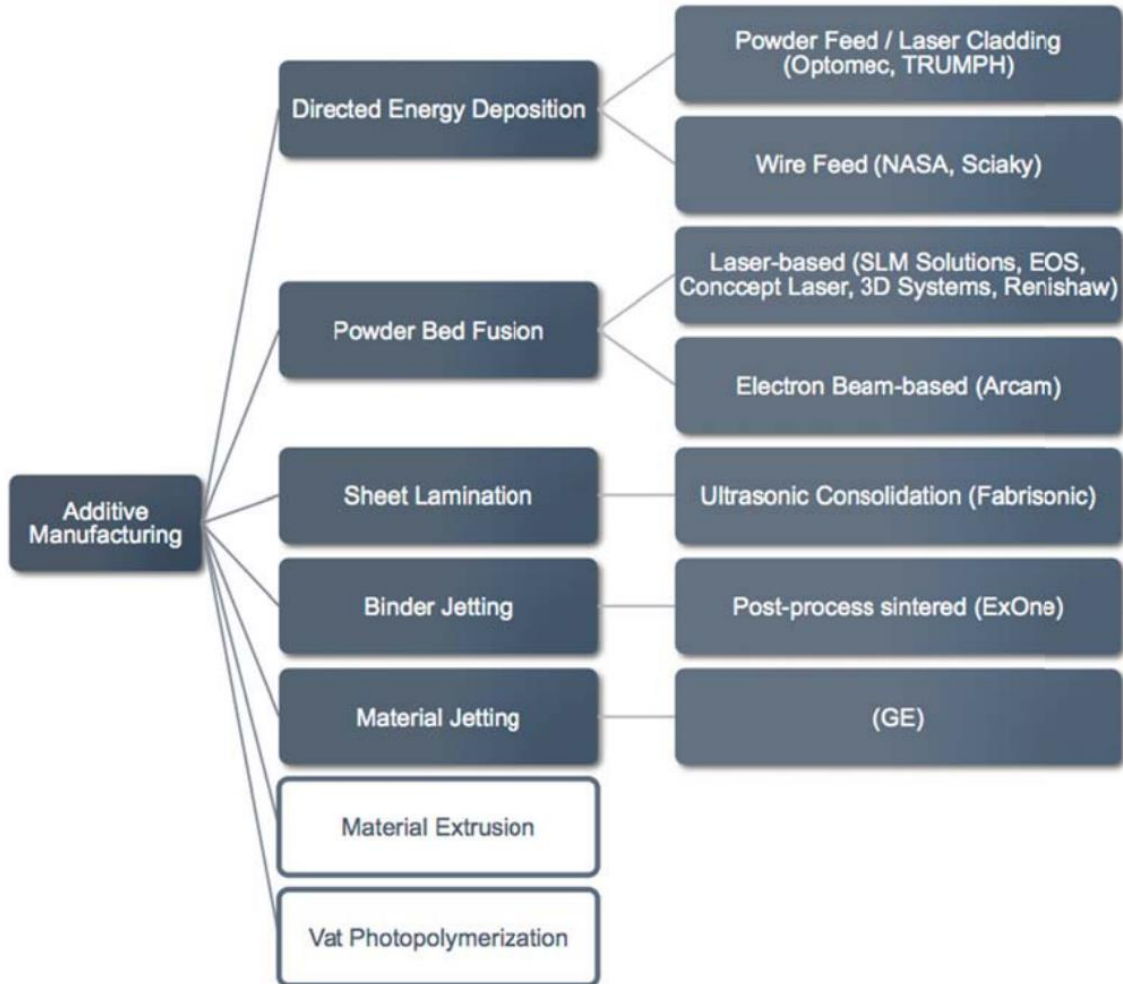


Figure 1. List of AM processes. The first five, in gray, are applicable to metal AM. Reproduced from Sharrat[7].

electron beam PBF (EB-PBF). Table 1 below lists all of the different powder bed systems.

Figure 2 shows a basic schematic of a metal PBF AM machine. The basic components are a build platform that descends as the build progresses from one layer to the next. This ensures that the build surface is a constant distance from the fixed heat source. PBF machines also have some sort of powder delivery system which includes a feedstock holding area and spreading or recoater mechanism. Finally, PBF machines have one or multiple energy sources - either a laser or an electron beam - that fuses layers of powder with the underlying substrate - either a build plate or previously fused material.

While PBF technologies are somewhat limited in the size of components that may be produced compared to other technologies, they excel in other important areas. L-PBF processes have been demonstrated to produce components with better feature resolution and surface finish compared to EB-PBF[9], [10]. Figure 3 shows the difference in features between the two technologies. On the other hand according to the manufacturers of Arcam EB-PBF machines, the most prevalent offering on the market, components fabricated with EB-PBF require fewer support structures and typically have lower residual stress due to the high processing temperatures[13]. Additionally, EB-PBF boasts considerably faster deflection speeds of the electron beam heat source which allows it to achieve faster build rates[14].

**Table 1. Metal PBF machine and associated commercial technology names.
Reproduced from Everton et al.[11]**

Term used in this article	Commercial Name	Machine Manufacturer
Laser-PBF	Direct metal laser sintering (DMLS)	EOS
	LaserCUSING	Concept Laser
	Selective laser melting (SLM)	Matsuura
	Direct metal production (DMP)	Phenix (3D systems)
	Selective laser melting (SLM)	Renishaw
	Selective laser melting (SLM)	Realizer
	Laser metal fusion (LMF)	Sisma Group
	Selective laser melting (SLM)	SLM Solutions
	Electron beam melting (EBM)	ARCAM
Electron beam-PBF	Electron beam melting (EBM)	ARCAM

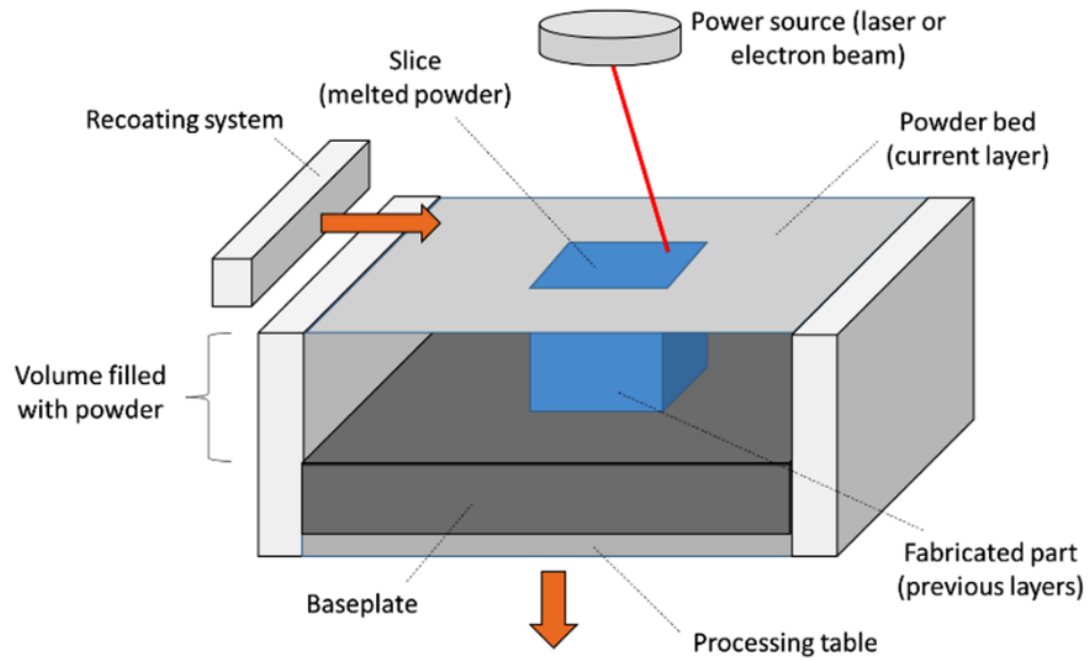


Figure 2. Basic schematic of a PBF AM machine. Reproduced from Grasso and Colosimo[12].

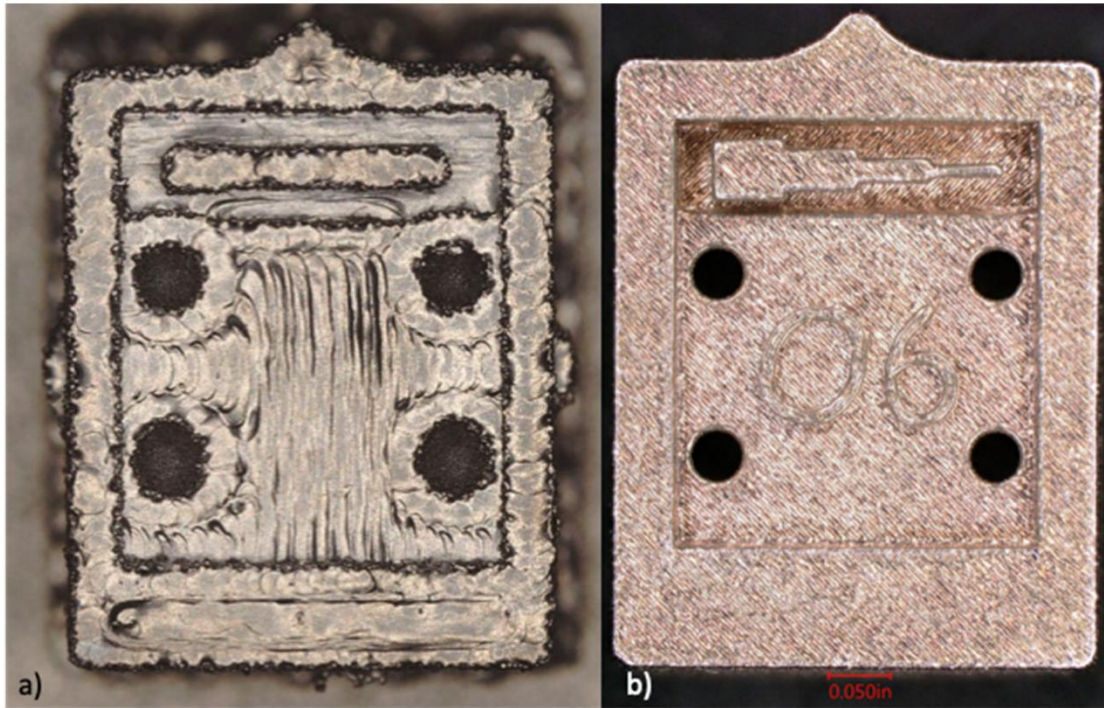


Figure 3. Feature resolution of a) EB-PBF versus b) L-PBF. Reproduced from Dryepondt et al.[9]

What are Spatially Localized Heterogeneities?

Despite the significant promise of these technologies, they have seen growing but still relatively little industrial penetration. It has been relegated in large part to prototyping instead of fabricating critical components. This is due to heterogeneities that plague components fabricated through the additive process and anisotropic properties[15]–[18] and a lack of standardized component certification and qualification processes means that extensive trial and error experimentation is often required[19], [20]. One key distinction is that anisotropy refers to material structures and properties that are orientation-dependent, while heterogeneity refers to non-uniform structure throughout a region of material. This work will focus on the latter because observed anisotropy is due, in part, to heterogeneities in PBF AM builds[17]. Heterogeneities may be found from the microstructural level all the way to flaws that are large enough to see with the naked eye.

Localized Microstructure

PBF demonstrates similarities to welding and specifically multipass welds; however, the PBF process offers more flexibility and can yield microstructures that differ from welding[16]. Microstructural heterogeneity of metal AM parts can arise from differences in morphology, size, orientation, and chemical composition of phases and grains and are material dependent and vary due to a combination of factors including processing parameters, location in the build, and thermal cycle histories[7], [15]. These factors affect the solidification conditions – specifically the thermal gradient and liquid-solid interface velocity of the melted region. Table 2 provides non-exhaustive set of examples of

Table 2. Sampling of literature investigating microstructural heterogeneity in L-PBF and EB-PBF as reported by Kok et al.[15]

	Grain Morphology	Grain Size	Composition
L-PBF	Ishimoto et al.[22], Hagihara et al.[23]	Amine et al.[24]	Xu et al.[25]
EB-PBF	Helmer et al.[26], Tan et al.[27], Antonyamy et al.[28]	Machry et al.[29], Yoder[30]	Tan et al.[31], Tan et al.[32], Sun et al.[33]

microstructural heterogeneities for both L-PBF and EB-PBF compiled from Kok et al[15], and Figure 4 shows examples of heterogeneities of grain size and grain morphology and orientation.

Localized Flaws

Flaws may broadly be defined as discontinuities in the material of AM fabricated components[11], [21]. They can lead to anisotropic mechanical properties and are often detrimental to the final performance capabilities of the component[15]. Their irregularity of occurrence and location complicate any potential component qualification process[19], [20]. A significant amount of research for metal PBF focuses on developing methods for detecting the occurrence of these defects both in-situ, while a component is being fabricated, as well as ex-situ once the build is completed[11]. Additional emphasis is placed on non-destructive evaluation (NDE) techniques to these ends as destructive techniques yield the current component useless and does not necessarily provide the requisite assurance that other subsequent parts will have the same defect locations and morphologies[7]. Several reviews survey the various localized flaws that may occur during the PBF process. They have reached something of a consensus on the following localized defects as shown in Table 3.

Since porosity figures prominently in the literature, we will discuss it in further detail.

Porosity is any location within a component that does not contain dense, fused material. It is often divided into three different subcategories based on the hypothesized causal

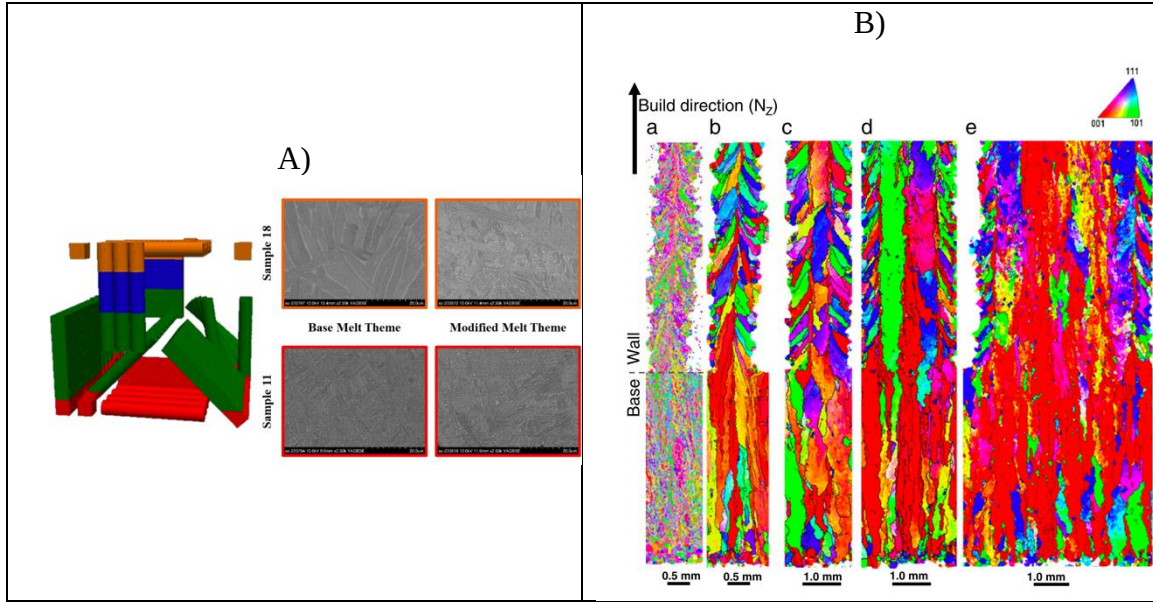


Figure 4. Examples of A) heterogeneities in grain size and b) heterogeneities in morphology and orientation. Reproduced from Yoder[30] and Antony et al.[28] respectively.

Table 3. Enumeration of localized flaws in AM.

	DeBroy et al.[16]	Kok et al.[15]	Sames et al.[6]	Everton et al.[11]	Sharrat[7]
Porosity	X	X	X	X	X
Lack-of-fusion	X	X		X	X
Cracking	X		X	X	
Balling			X	X	X
Surface Roughness	X				X
Geometrical Distortion			X		X

mechanism, and these mechanisms are largely process-based but may be powder-based. Gas porosity, specifically, may be powder based and is often a result of gas porosity that is present in the powder feedstock that is used in the build. However, in L-PBF inert gas that flows over the powder bed may become entrapped and also lead to spherical gas porosity. The other types of porosity are solely process induced. The second form of porosity is called lack-of-fusion porosity, and it occurs during the PBF process when the heat source fails to adequately penetrate into the layer of powder and fuse it to previously solidified layers. The third form is referred to as keyhole porosity and occurs when instabilities occur in deep melt pools that cause it to collapse and subsequently solidify with voids inside. Examples of lack-of-fusion and keyholing porosity are shown in Figure 5.

What is a Scan Path?

First, we must define what a scan strategy is. A scan strategy is the set of protocols and parameters that dictate how the heat source will fill in or melt material for a build. A scan path, then, is the actual path that the heat source follows during selective melting or deposition for lasers or electron beams that conforms to the scan strategy. The scan strategy for a given build may be adjusted by layer or by part.

Figure 6 shows several of the more widely deployed scan strategies that are used in metal PBF AM. Raster scan strategies, both with and without contours, are often used for both L-PBF and EB-PBF[6]. The island scanning strategy is used primarily in L-PBF and is

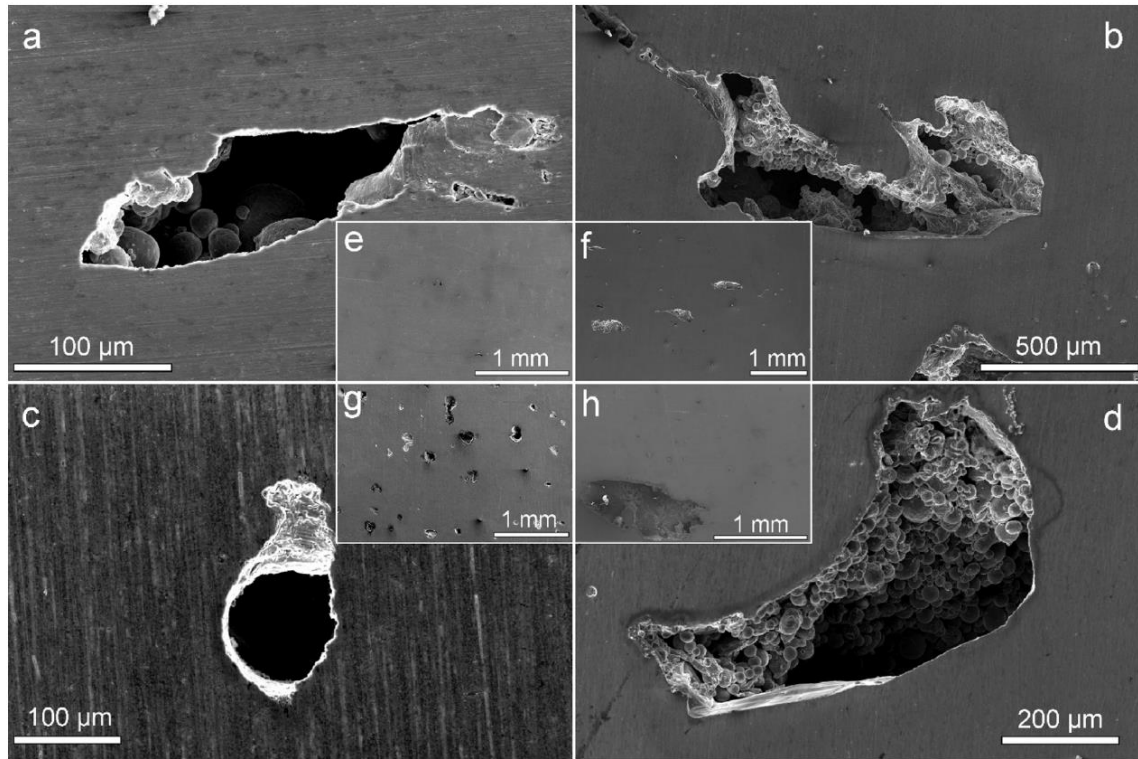


Figure 5. Images of a), b), d) lack-of-fusion and c) keyholing porosity. Reproduced from Bartlett et al.[34]

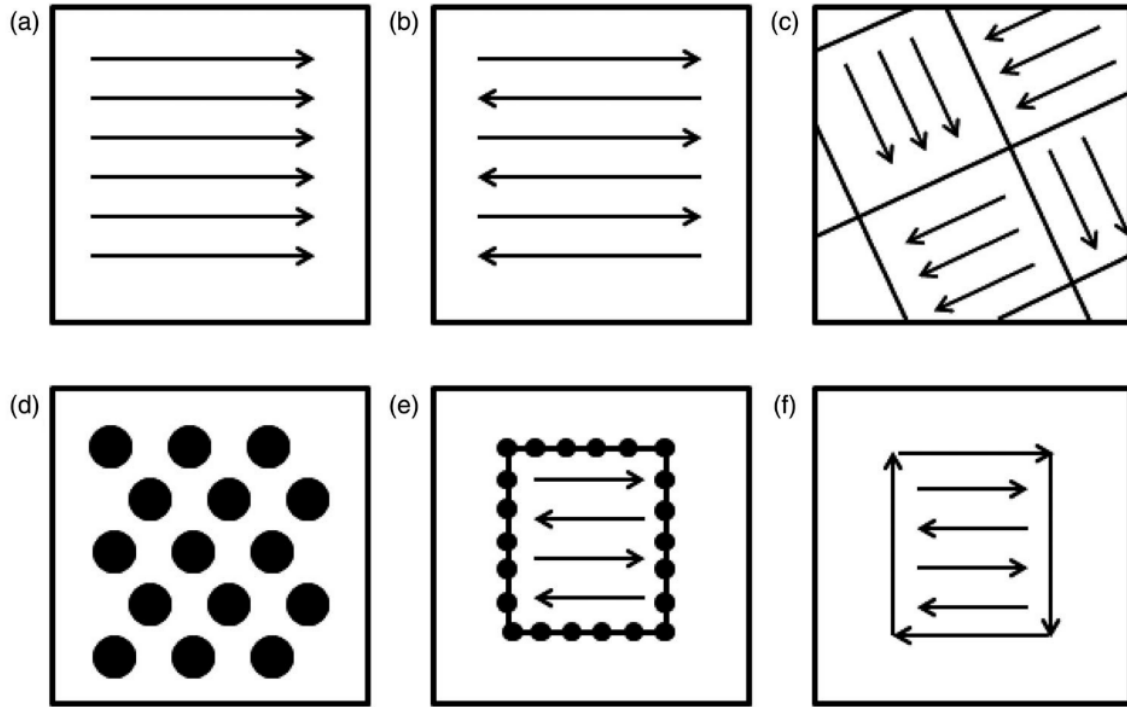


Figure 6. Scan strategies used to determine heat source path in metal AM as seen in the X-Y plane (perpendicular to the build direction): a) unidirectional or concurrent fill, b) bi-directional, snaking, or countercurrent fill, c) island scanning, d) spot melting, e) spot melting contours with snaking fill and f) line melting contours with snaking fill. Reproduced from Sames et al.[6]

employed in order to reduce residual stresses within the part during fabrication[35]. Finally, the spot melt – also referred to as point melt – is used primarily in EB-PBF because of the higher deflection rates of the electron beam that are possible[36]. Additionally, the selection of a scan strategy directly affects selection of other processing parameters; the power and velocity of the beam must be optimized for a given scan strategy[6].

The scan path for a layer, when speed information is available, provides the spatio-temporal history for the fabrication of a layer. Scan paths for proximal layers may be inspected to determine how the melt differed. A single scan strategy may produce significantly different scan paths from one layer to the next as seen in Figure 7. The intersection of and interactions between scan path, geometry, and other processing parameters contribute significant complexity to the PBF AM.

What are Process-Structure Relationships?

Intuitively, process-structure relationships refer to the coupling of the additive process and the resulting structures — both microstructure and flaws — that form in the final, as-fabricated component. It is part of a larger concept — process-structure-properties relationships — that is of paramount interest. As previously noted the heterogeneous structures that evolve in an additively manufactured component leads to anisotropic mechanical properties that precludes its use in critical applications[16]. One reason for the localized variability is the immense complexity that underlies the PBF processes. The

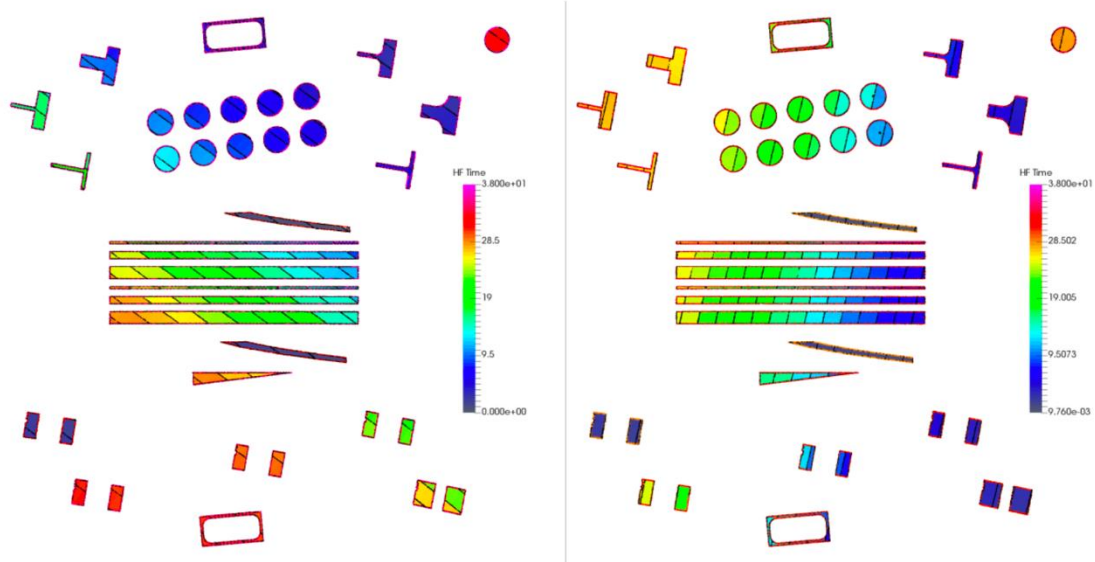


Figure 7. Example of two different scan paths, generated from the same scan strategy, for two layers of a single build with similar cross-sectional geometry. The authors note how the total melt duration for the lower right rectangles differs significantly between the two scan paths — over five seconds for one and less than a second for the other layer — resulting in significantly different thermal histories. Reproduced from Groeber et al.[17]

combination of processing parameters, part geometry, material, and the many simultaneous phenomena in the PBF process leads to oft-times seemingly stochastic defects both in the powder bed and in the part itself[6]. This warrants further investigation into process-structure relationships.

Equation 1.

$$E \left(\frac{J}{mm^3} \right) = \frac{P(W)}{V \left(\frac{mm}{s} \right) \cdot h(\mu m) \cdot t(\mu m)}$$

One approach of investigating process-structure relationships has been relating a metric called "energy density" to resulting structures[37]. Equation 1 shows how energy density is calculated for raster scan paths - where P is beam power, V is beam velocity, h is hatch spacing, and t is layer thickness. An alternative to relating simple energy density to outcomes is the creation of process maps. This approach seeks to enumerate process outcomes in terms of process variables[6], [38]. Most commonly, this method seeks to link outcomes to beam power, P , and speed, V , and gives a more granular view about how changes in these values affects process outcomes and are commonly used in welding[6]. Figure 8 shows an example process map. However, the speed and power needed to achieve certain outcomes may be sensitive to several factors such as scan strategy and geometry[6].

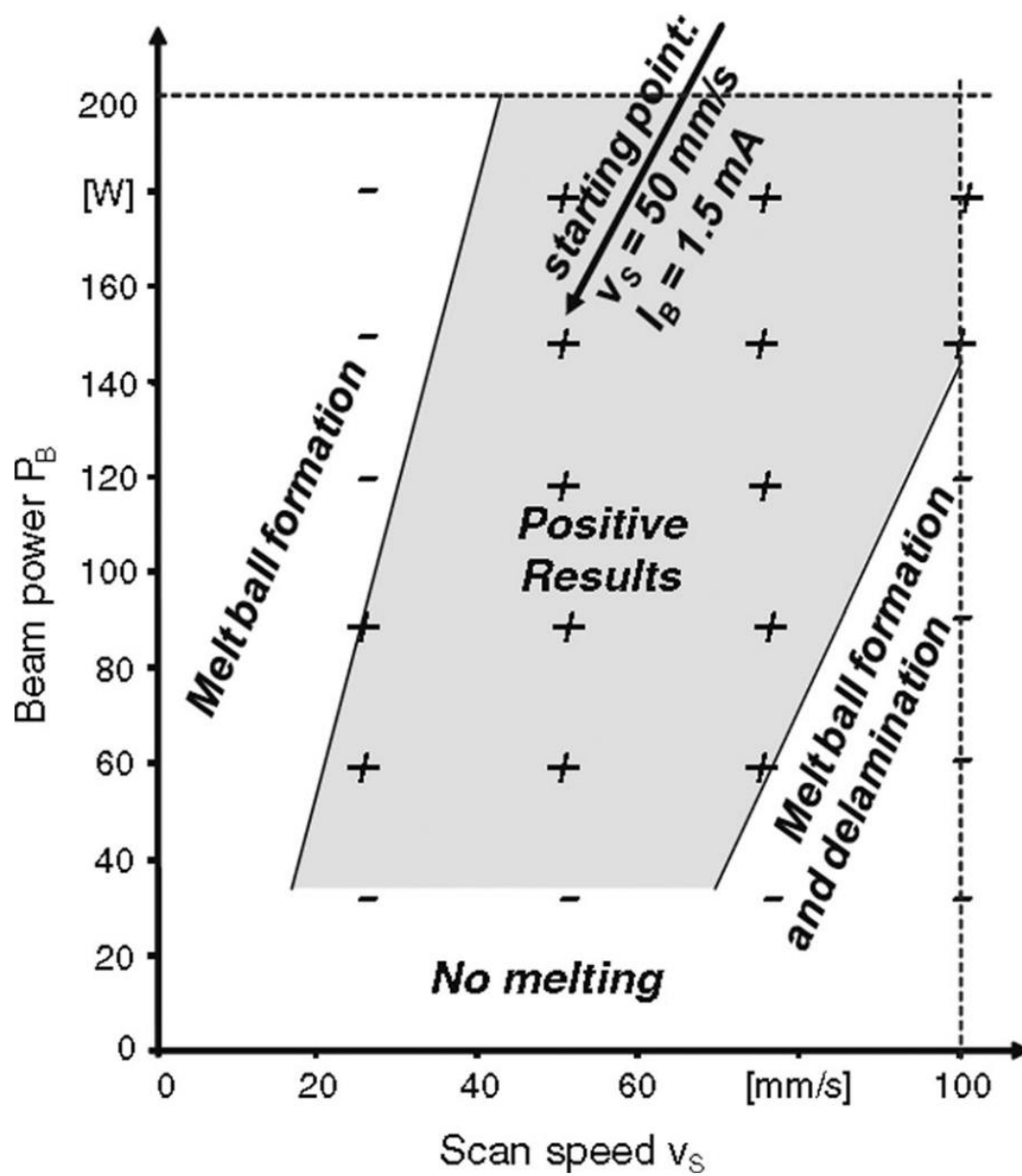


Figure 8. Process map for stainless steel and EB-PBF. Reproduced from Sames et al.[6]

Discussion

The survey of the literature that has been undertaken demonstrates several key and ongoing and themes for metal PBF AM.

1. Metal PBF AM is a promising set of technologies that still have room for improvement to reduce mechanical anisotropy and inconsistent quality in fabricated components.
2. The inconsistent quality is due, in large part, to localized heterogeneities in microstructure and flaws throughout fabricated components.
3. Investigating process-structure relationships is necessary to understand how the complex AM process affects the evolution of these structures.
4. The scan path is an important key to understanding this inherently spatio-temporal process as well as the corresponding dynamic interactions that introduce significant process complexity.

As such, this work will focus on investigating process-structure relationships — specifically the process scan path and how it may relate to localized heterogeneities in metal PBF AM.

CHAPTER I
GEOMETRY-INDEPENDENT MICROSTRUCTURE OPTIMIZATION FOR
ELECTRON BEAM POWDER BED FUSION

A version of this chapter was originally published by William Halsey, James Ferguson, Alex Plotkowski, Ryan Dehoff, Vincent Paquit:

William Halsey, James Ferguson, Alex Plotkowski, Ryan Dehoff, Vincent Paquit. “Geometry-Independent Microstructure Optimization for Electron Beam Powder Bed Fusion.” *Additive Manufacturing*. 35(2020): 101354.

The content of this chapter is the same as the original manuscript except for two instances. Figure 11 and Figure 14 were labeled as “Algorithm 1” and “Algorithm 2” in the original article. The change herein was made to more closely follow the formatting conventions of this institution. Additionally, some terms have been changed to follow conventions adopted for this dissertation, ie. EBM changed to EB-PBF.

The student was responsible for ideation, implementation, data collection, analysis, and manuscript writing. Mr. James Ferguson assisted in ideation and programming — specifically developing the code for interfacing with the heat transfer model. Dr. Alex Plotkowski developed the analytical heat transfer model that was employed to simulate and evaluate the scan paths as well as writing the manuscript portion pertaining to the heat transfer model. Dr. Ryan Dehoff served as the EB-PBF subject matter expert, and Dr. Vincent Paquit assisted in funding this work. All coauthor reviewed the manuscript.

Abstract

The ability to manipulate the formation of microstructure is one of the potential advantages of additive manufacturing. Yet, the additive manufacturing process is riddled with complex interactions between processing parameters, geometry, and spot melting sequence which makes the task of controlling microstructure challenging. This paper

proposes a method to solve the microstructure texture optimization problem by leveraging established optimization techniques and increasingly fast and accurate analytical heat transfer models. Respectively, we apply two different optimization algorithms, genetic algorithm and concurrent greedy line search, in order to induce arbitrarily shaped microstructure regions, and we use a heat transfer model to evaluate the fitness of solutions in an iterative approach that is far more efficient than pure trial and error. The objective of the optimization algorithm is to control local thermal conditions — hence microstructure formation — during the melting process by manipulating the order and the dwell time of all the points of the infill pattern sequence. We then analyze the optimized sequence to determine important path characteristics that affect the microstructure formation. Through examples, we show that relative order and frequency, as measured by weighted average and variance of spot order, are important factors that influence the final characteristics.

Introduction

Metal additive manufacturing has the potential to enable unprecedented customization of produced parts. The process allows for the fabrication of parts that are unattainable with traditional manufacturing techniques — not only geometry but also control of structural and mechanical properties of the part. One way to achieve control of the mechanical properties is to control the texture of the microstructure. The relationship between physical properties and microstructure is well known from other techniques and continues to be studied extensively for metal additive manufacturing[15].

In metal additive processes, the processing parameters have been shown to have a marked effect on microstructure and structural properties across a variety of materials and metal additive technologies using traditional raster and island scans. Simchi and Pohl[39] demonstrate how processing parameter affect microstructure in Iron powder; H. Gu et al.[37] for 17-4PH stainless steel; Arisoy et al.[40] for Inconel 625; Narra et al.[41] for AlSi10Mg; and several papers have demonstrated a relationship for Ti-Al6-V4[42], [43]. Additionally, significant amounts of work have been undertaken to establish relationships between process parameters and microstructure for Inconel 718[44]–[47].

The ultimate goal would be to tailor the texture type within the component; however, translating the findings from previous research to obtain site-specific control of texture for arbitrary regions of a component are neither clear nor intuitive. Dehoff et al.[36] was able to demonstrate local texture control, but the exact parameters were chosen in a manual trial-and-error process. Furthermore, the findings are not readily applicable to other patterns or geometries. Antonysamy et al.[28] shows that geometry alone can have significant effects on the resulting microstructure, therefore each new geometry — whether physical cross-section or desired microstructural region — requires unique processing parameters, and the endeavor remains an art instead of a science. Although, the advent of better process modeling[48] allows for more opportunities to leverage compute resources to optimize processing and scan parameters for microstructure for

domains larger than melt tracks and comprised of unique and complex microstructure regions.

The paper will be structured as follows. First, section *Methodology* will discuss the algorithms and methods that we use to optimize microstructure, and section *Experiments and Results* will enumerate the specific experimental settings and outline the results. Then, we analyze salient factors that correlate with microstructure in section *Analysis of Optimization*. Finally, section *Discussion and Future Work* will contain a discussion of the work and outline possible future work that can extend this effort.

Methodology

For these experiments we chose to optimize scan path parameters for the Arcam electron beam powder bed fusion systems. We chose electron beam technologies because the high deflection rate of the beam allows for more complex local thermal histories and a greater ability to manipulate the formation of local microstructure. Additionally, the model requires a target alloy as input in order to simulate the melt process. So, as part of this analysis and as a necessary input into the heat transfer model, we choose Inconel 718 as our target material. Furthermore, work has been previously undertaken investigating the relationship between processing parameters and the columnar to equiaxed transition and attempts to control which texture forms in bulk regions of Inconel718[49]. Additionally, that work leverages less traditional spot melt scan patterns which was also utilized for

these experiments. We expect that the findings of this work would be useful for the task of texture control for other materials.

In order to utilize the optimization algorithms we require a way to evaluate the quality of different solutions. To this end we employ an analytical heat transfer model that simulates a print path and calculates the thermal gradient and solid-liquid interface velocity for cells within a geometry. It also outputs a value on a scale between 0 and 1 that indicates the predicted percentage of equiaxed grains – values close to zero indicate a low prevalence of equiaxed grains, and values close to one indicate a high prevalence. As such, when evaluating different spot paths we use the value 1 as the target value where equiaxed microstructure is desired and a target value of 0 where columnar microstructure is desired.

Heat Transfer and Microstructure Model

To predict microstructure evolution in response to the scan path, heat transfer effects must first be considered. If non-linear effects such as latent heat evolution, fluid convection, vaporization, and radiation are neglected, the governing partial differential equation may be reduced to an analytically tractable form. Here, the solution proposed by Nguyen et al.[50] for the transient temperature T around a moving volumetric Gaussian heat source at time t was used:

Equation 2.

$$T - T_0 = \frac{2\eta Q}{\rho c(\pi/3)^{3/2}} \int_0^t \frac{\exp\left(\frac{-3x(t')^2}{\psi_x} - \frac{3y(t')^2}{\psi_y} - \frac{3z(t')^2}{\psi_z}\right) dt'}{\sqrt{\psi_x \psi_y \psi_z}}$$

where

Equation 3.

$$\psi_i = 12\alpha(t - t') + \sigma_i^2 \text{ for } i = x, y, z$$

and T_0 is the initial temperature, Q the heat source power, η the absorption efficiency, ρ density, and c specific heat. The shape of the beam is defined by widths σ_i , and the coordinates of the point of interest $x(t') = (x, y, z)$ are defined relative the transient location of the heat source. Because the scan path is defined by a set of piece-wise functions, the integral in Equation 2 can not be evaluated analytically. Therefore, an adaptive Gaussian quadrature scheme was used to efficiently calculate the thermal field numerically. To relate the thermal conditions to the solidification microstructure, the resultant thermal gradient and solid-liquid interface velocity were extracted from temperature history of each point in the simulation domain at the time that its temperature fell from above to below the alloy liquidus temperature. The thermal gradients in each direction G_x , G_y , and G_z were calculated by analytically taking the respective partial derivatives of Equation 2. The resultant thermal gradient is then $\bar{G} = \sqrt{G_x^2 + G_y^2 + G_z^2}$.

The cooling rate, $\dot{T}$, was calculated using a linear interpolation in time. The solid-liquid interface velocity, V_{s-l} , may then be simply calculated[51]:

Equation 4.

$$V_{s-l} = \frac{\bar{G}}{\dot{T}}$$

The preferential selection of a specific grain structure morphology was predicted using the columnar-to-equiaxed transition model proposed by Gaumann et al.[52], who extended the original work by Hunt[53] to non-equilibrium conditions similar to welding and additive manufacturing. The model relates the fraction of equiaxed grains to the local thermal conditions by considering the competition between the advancing columnar dendrites and nucleating equiaxed grains. Under high thermal gradient conditions in which the nucleation undercooling may be safely neglected, the relationship between the thermal conditions and the equiaxed grain fraction, ϕ , is

Equation 5.

$$\frac{\bar{G}^n}{V_{s-l}} = a \left\{ \sqrt[3]{\frac{1}{n+1} \frac{-4\pi N_0}{3\ln[1-\phi]}} \right\}^n$$

where a and n are material parameters relating to a simplified model of dendrite growth, and N_0 is the volume density of activated nucleation sites. Based on previous work in additive manufacturing of Ni-base superalloys, values of $a = 1.25 \times 10^6$,

$n=3$, and $N_0 = 3 \times 10^{13}$. For the present work, the exact values of these parameters are not critical. Yet, for practical application of this work, calibration of this microstructural model, and the nucleation density in particular, is necessary[54].

Genetic Algorithm

Raghavan et al.[55] experimentally demonstrated that local preheat temperature is the most important parameter for controlling texture. This parameter can not be directly controlled in the EB-PBF process, but the local preheat of a point may be manipulated by changing the order in which points are printed. As a point is printed it heats up the region of material adjacent to it. This local heating effect, in conjunction with the time delay before adjacent points are actually melted, influence the microstructure of those adjacent points. However, the relationship between a particular spot order and the subsequent local thermal histories that result is highly non-linear and inherently complex.

This task falls into a category of problems called combinatorial optimization. We employ a genetic algorithm because they are well-suited to find a solution to such combinatorial problems. A genetic algorithm is an evolutionary algorithm that seeks to find an optimal solution by emulating evolutionary processes[56]. They have been leveraged for many other manufacturing tasks such as milling[57], turning[58], [59], and cutting[60] of Inconel 718. Additionally, a genetic algorithm has been leveraged to control microstructure in arc welded Inconel 625[61]. One contribution of this work is to show

how genetic algorithm can be used to optimize point order spot melt infill patterns to manipulate microstructure development.

Genetic algorithms have three operational components — a fitness function, crossover, and mutation operations — that enables them to find good solutions for complex optimization problems. First, the fitness function is used to evaluate and score a population of possible solutions. This score is used to rank solutions from best to worst. The success of a genetic algorithm can hinge on the ability of the fitness function to accurately and quantitatively capture what is a good solution. The fitness function must not just evaluate the performance of the solution but also help push the search to regions where good solutions are more likely to be found. Ultimately, the genetic algorithm seeks to generate a solution that maximizes the fitness function. To this end we chose to use a two-part fitness function to accomplish these ends shown in Equation 6. Borrowing genetics terminology, the first term of the fitness function aims to evaluate the phenotype of the solution — the quality of the microstructure that results from the path as predicted by the simulation. The second term of the fitness function serves to quantify the genotype of the specimen — inherent properties of the path that may affect its and offsprings' resulting microstructure but are not directly observable from the simulation.

Equation 6.

$$F_{\text{tot}}(s) = F_{\text{sim}}(s) + F_{\text{heu}}(s)$$

The first term, F_{sim} , in the fitness function evaluates a specimen, or potential spot ordering, based on the predicted microstructure for that path from the model in relation to the desired microstructure. This fitness for the specimen is the sum of the fitness for each point in the path for a given specimen, P_s , as shown in Equation 7.

Equation 7.

$$F_{sim}(s) = \sum_{p=1}^{P_s} f_{sim}(p)$$

For each point the fitness is calculated as shown in Equation 8, where ϕ_{target} is 0 for points where columnar microstructure is desired and 1 for equiaxed points, and ϕ_p is the predicted equiaxed fraction from the model for an individual point in the spot order.

Equation 8.

$$f_{sim}(p) = -|\phi_{target} - \phi_p|$$

Equation 9 is a further simplification of the fitness function for the two distinct regions of microstructure. So, the fitness from this equation will always be negative, yet as the predicted microstructure for a point in the path becomes more similar to the target microstructure for the region the contributed fitness will grow closer to zero.

Equation 9.

$$f_{\text{sim}}(p) = \begin{cases} -\phi_p & \text{if } p \in \text{columnar} \\ \phi_p - 1 & \text{if } p \in \text{equiaxed} \end{cases}$$

The second term, F_{heu} , evaluates a characteristic of the path itself, as opposed to the error of the microstructure, in order to help determine if this solution will lead to more fit solutions. Again, this term is the sum of the fitness for each point in the path beginning with the second spot in the path, seen in Equation 10.

Equation 10.

$$F_{\text{heu}}(s) = \sum_{p=2}^{P_s} f_{\text{heu}}(p, p_{\text{prev}})$$

Raghavan et al.[55] has shown that increased local preheat temperature leads to equiaxed microstructure. One way to increase local preheat temperature is to print points that are physically close together at approximately the same time which will create larger melt pools and lower thermal gradients. This means that any points that are near each other in space should be close together in the spot order to induce more equiaxed microstructure and far apart in the spot order to produce more columnar microstructure. Equation 11 shows the general form for this fitness term. It serves to evaluate whether the current point is a suitable distance from the point that was melted immediately before it, and this will serve to move the genetic algorithm towards candidate solutions with more ideal local preheat temperature characteristics.

Equation 11.

$$f_{\text{heu}}(p, p_{\text{prev}}) = - \left| 1 - \frac{1}{1 + \|p - p_{\text{prev}}\|} \right|$$

Equation 12 is the simplification for the two distinct microstructure regions. A point that lies in the columnar region is more greatly penalized if it is physically close to the preceding point in the spot path; conversely a point that lies in the equiaxed region is more greatly penalized if it is physically far away from the preceding point. Similarly to f_{sim} this portion of the fitness function is bounded between -1 and 0 for every point in the path, and as a point more closely conforms to the heuristic penalty to the fitness will approach 0.

Equation 12.

$$f_{\text{heu}}(p, p_{\text{prev}}) = \begin{cases} -\frac{1}{1 + \|p - p_{\text{prev}}\|} & \text{if } p \in \text{columnar} \\ \frac{1}{1 + \|p - p_{\text{prev}}\|} - 1 & \text{if } p \in \text{equiaxed} \end{cases}$$

Together these equations serve to penalize a potential solution if it has poor simulated results or if it violates a known heuristic. We chose to weight the two fitness terms equally when evaluating the total fitness of a possible solution. Since the fitness terms serve to penalize ill-suited solutions, the aim of the genetic algorithm is to generate a solution that maximizes the fitness function. The choice of fitness function creates a fitness surface for possible solutions that may contain local optima that genetic

algorithms may be susceptible to. However, the mutation operation, discussed later, can serve to move candidate solutions out of local optima.

Genetic algorithms have a crossover operation that takes chunks of parameters from two existing solutions and combines them in a way that produces more fit solutions. This operation serves to allow the population of solutions to traverse the solution space of possible orderings quickly across subsequent generations while preserving large sections of previously successful paths. Since the goal is to find an optimal ordering of points, this task belongs to a class of problems commonly referred to as Traveling Salesman problems. When using genetic algorithms to solve this class of problems the crossover operation must produce new solutions that are still valid traversals of all the points. Many different crossover operations for this class of problems have been developed. We chose to employ two simple methods in conjunction to create subsequent specimens. One child was produced using an ordered crossover (OX) scheme[56] between the two parents – specifically Linear Ordered Crossover (LOX)[62]. The other child was produced using a Position-Based Crossover (POS or PBX) scheme[63]. These two crossover schemes were chosen because individually, each crossover scheme serves to only produce one offspring; however, leveraging them in conjunction allows us to create two new solutions to test without the need to performing the crossover operation twice with each set of parents in order to create the new solutions. Additionally, the structures of these two crossover schemes are compatible such that it allows for this hybrid approach. Figure 9 shows how our hybrid crossover operation is implemented.

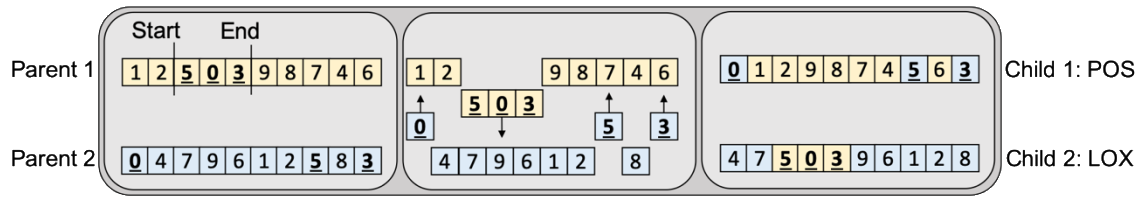


Figure 9. Crossover scheme for spot path order in genetic algorithm.

An initial crossover point for the crossover operation was chosen with a probability of 50 percent. Once a first crossover point was found the end point was chosen randomly from the points between the first crossover point and the end of the path. Additional crossover points were chosen with 50 percent probability beginning with the point after the previous end point. This process was repeated until the end of the spot path was reached, so several crossover operations could occur between the two specimens. The crossover operation allows future offspring to receive large portions of a scan path from the parents. This operation, then, introduces coarse-grain perturbations to candidate solutions by shuffling large chunks of the scan paths while allowing child solutions to exploit intact good sections of scan path that are inherited. A 50 percent crossover rate ensures that there is good mixing between parent solutions to produce children. Less than 50 percent would tend towards replicating the parent solutions and greater than 50 percent would tend towards introducing more changes with fewer intact portions being passed down from the parents.

Additionally, genetic algorithms employ a mutation operator that introduces more fine-grain perturbations to previous solutions. These mutations allow for searching the solution space guided by previous successful solutions that have been able to persist in the population through the generations. We implemented mutations as a swap of adjoining spots in the spot path. The probability that a mutation would occur was 75 percent for every point to help ensure a more diverse set of solutions were explored in the

solution space. This operation is shown in Figure 10. Multiple mutations are allowed for each specimen.

The genetic algorithm is enumerated in Figure 11. Our implementation of the algorithm has three distinct groups within the population that are each treated slightly differently. Out of the entire population, S of size N , the least fit half are eliminated and not allowed to contribute to generating new solutions; the most fit half are allowed to propagate. Additionally, this implementation has E specimens, the elites, that are the most fit individuals within a generation. They remain unchanged through the crossover and mutation phases to ensure that the overall best solutions do not get worse over time. For this implementation to work the size of E must be less than $N/2$. Also, in the genetic algorithm G is the number of generations to run, s is a single specimen in S , $F_{tot}(s)$ is the fitness score for s , $F_{sim}(s)$ is the fitness for s evaluated from the simulation, and $F_{heu}(s)$ is the fitness for s evaluated from a heuristic. For this work we use a population size, N , of 119, the number of generations, G , was 250, and the elite size, E , was 5.

Concurrent Greedy Line Search

Raghavan et al.[55] also experimentally shows that spot time is the second most important parameter for controlling microstructure. Additionally, they experimentally demonstrate the relationship between dwell time and microstructure texture. Namely, longer spot times tend to induce more equiaxed grain growth; whereas shorter spot times yield more columnar grains. Thus, the aim of this optimization is to find an appropriate

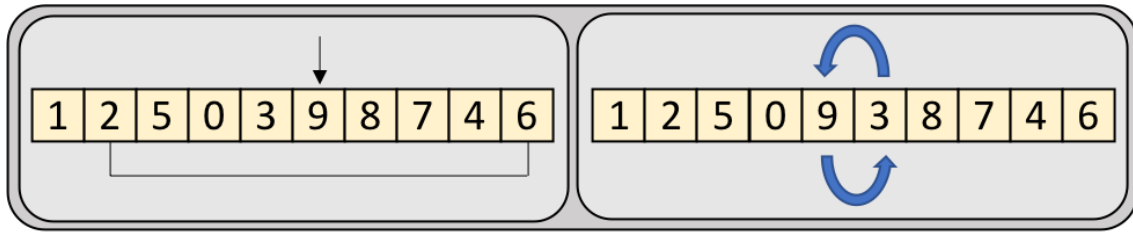


Figure 10. Mutation scheme for spot path order in genetic algorithm.

1: 2: 3: 4: 5: 6: 7: 8: 9: 10: 11: 12: 13: 13: 14: 15: 16: 17: 18:	Generate population S of size N for G generations do for $s \in S$ do Calculate $F_{sim}(s)$ and $F_{heu}(s)$ $F_{tot}(s) = F_{sim}(s) + F_{heu}(s)$ end for Determine E most fit $s \in S$ Eliminate $N/2$ least fit $s \in S$ Duplicate $N/2$ most fit $s \in S$ Create random pairings s_a, s_b for $s_a, s_b \in S$ and $s_a \notin E$ do $s_a, s_b \leftarrow \text{Crossover}(s_a, s_b)$ $s_a \leftarrow \text{Mutate}(s_a)$ $s_b \leftarrow \text{Mutate}(s_b)$ end for end for Calculate $F_{sim}(s)$ and $F_{heu}(s)$ $F_{tot}(s) = F_{sim}(s) + F_{heu}(s)$ Return single most fit $s \in S$
--	---

Figure 11. Genetic algorithm.

spot time for each individual point that will produce the desired overall microstructure regions as an aggregate.

However, in the EB-PBF process setting the spot time or dwell time for individual spots is not possible. Only a single global spot time may be set as a parameter. The beam will deposit energy at every point for this fixed amount of time. Therefore, in order to tailor spot times for individual points we developed a method where a single small global spot time was set. Then, the spot time for a single point may be changed by including it multiple times in immediate succession in the path. In practice this leads to the heat source dwelling on the point for a longer duration – an integer multiple of the global base spot time.

The optimization task then becomes determining, for a given spot order, how many times to print each point in immediate succession in order to yield the global desired microstructure. For this task a global base spot time of 0.05 ms was chosen, and we constrained the number of times a point could be printed between 2 and 10. This yielded a discrete range of possible spot times between 0.1 and 0.5 ms with 0.05 ms increments, shown in Figure 12 and Figure 13. Dictating upper and lower bounds for the multipliers is necessary because it ensures that the optimization can not push the values indefinitely to zero and positive infinity which is not physically possible for the machine to achieve.

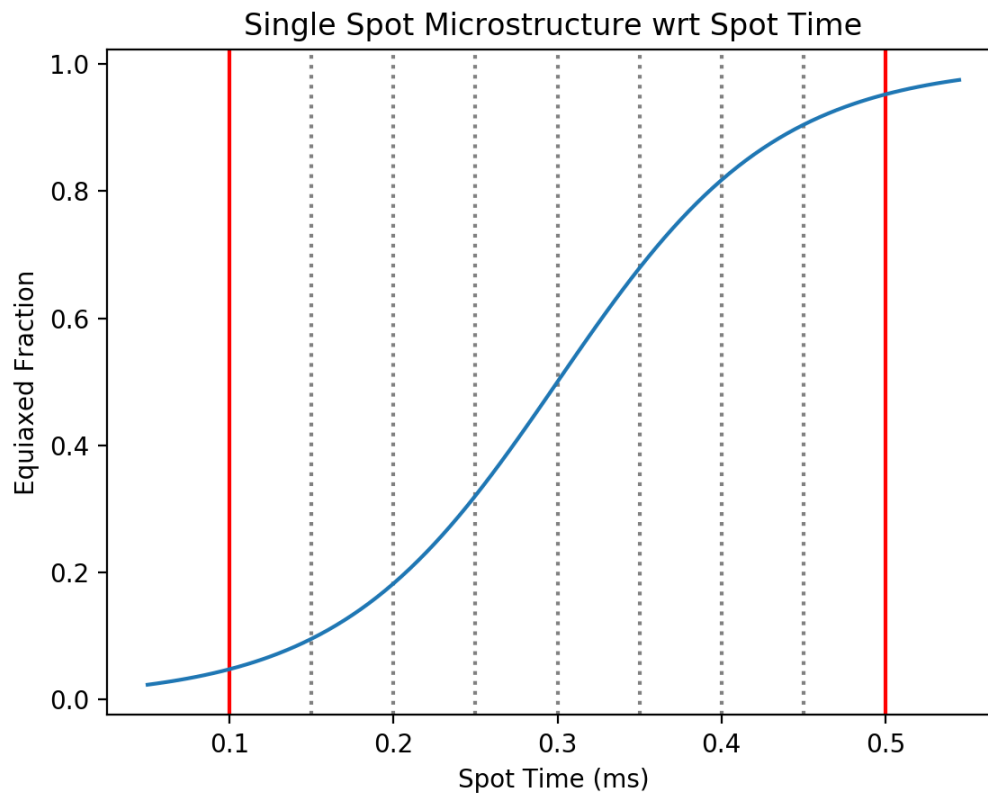


Figure 12. Idealized relationship between spot time and equiaxed fraction.

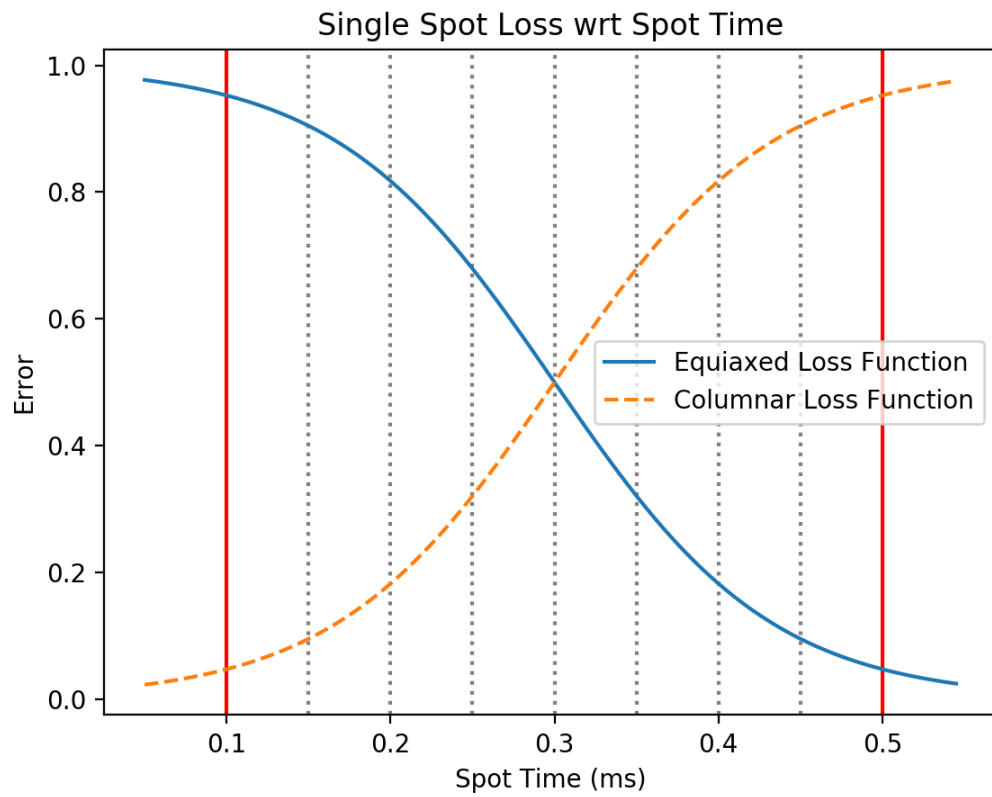


Figure 13. Idealized loss function for points in the two different regions.

Additionally, these bounds keep the simulations within the same regime as those explored by Raghavan et al.[55]. In this regime we can assume a strict monotonic relationship between spot time and equiaxed fraction.

To this end we employed a concurrent greedy line search, and we adopted an important simplification in order to make the optimization easier. During the update phase of the algorithm we disregard the interactions that occur between points. This simplification allows us to increase or decrease the multiplier for every point concurrently during each iteration of the algorithm. This is in contrast to an algorithm that would rerun the simulation with any change to any multiplier of a point in the scan path. So, for a given iteration of the algorithm, the predicted microstructure from the model is evaluated for each point, and it is determined whether to print that point a greater or lesser number of times in succession for the next iteration of the algorithm. Hence the optimization is concurrent and greedy because the points are updated simultaneously without regard to how neighboring points are changing in that iteration. Only after updating all of the multipliers do we rerun the simulation in order to get feedback regarding how changing the various multipliers affects the microstructure in the various regions of the component. Again, we assume that the relationship between spot time and microstructure for a single point is strictly monotonic based on the empirical evidence gathered in [55], and this assumed relationship for our set of experiments is shown in Figure 12.

Typically, optimizations require calculating the error between predicted and desired results and deriving a gradient in order to update the parameters. Yet while the magnitude of the gradient is not known for our update rule, the direction of the gradient is known for both regions of microstructure if the relationship is assumed to be strictly monotonic. As a result, Figure 13 shows how the error or loss for points in the equiaxed microstructure region would decrease with increasing spot time and how error for points in the columnar microstructure region decreases with decreasing spot time. We can derive a simple loss shown in Equation 13 which will provide the direction of the update. Since the domain is discrete, the update simply entails increasing or decreasing the multiplier for a single point by one in whichever direction is indicated by the sign of the loss. This update rule, and the entire algorithm is shown in Figure 14.

Equation 13.

$$\text{Error}_p = \phi_{\text{target}} - \phi_p$$

For each experiment we employed the same initial set of candidate points that filled a 15x10 mm rectangle - shown in Figure 15. The points were laid out on a hexagonal grid with 200 micron spacing between each point. Each solution also began with a spot order that was a random ordering of these 4,559 points. We ran our optimization algorithms for two geometries each with two distinct regions of microstructure. Figure 16 shows these two geometries. The black regions indicate where columnar microstructure is desired; the white regions show where equiaxed is desired. With these two geometries we can see

```

1:      Initialize global base spot time and multipliers,  $M$ , for  $p \in P$ 
2:      while  $M$  changing do
3:          Simulate path
4:          for  $p \in P$  do
5:               $Error_p = \phi_{target} - \phi_p$ 
6:               $M_p \leftarrow M_p + \text{sgn}(Error_p)$ 
7:          end for
8:      end while

```

Figure 14. Concurrent greedy line search.

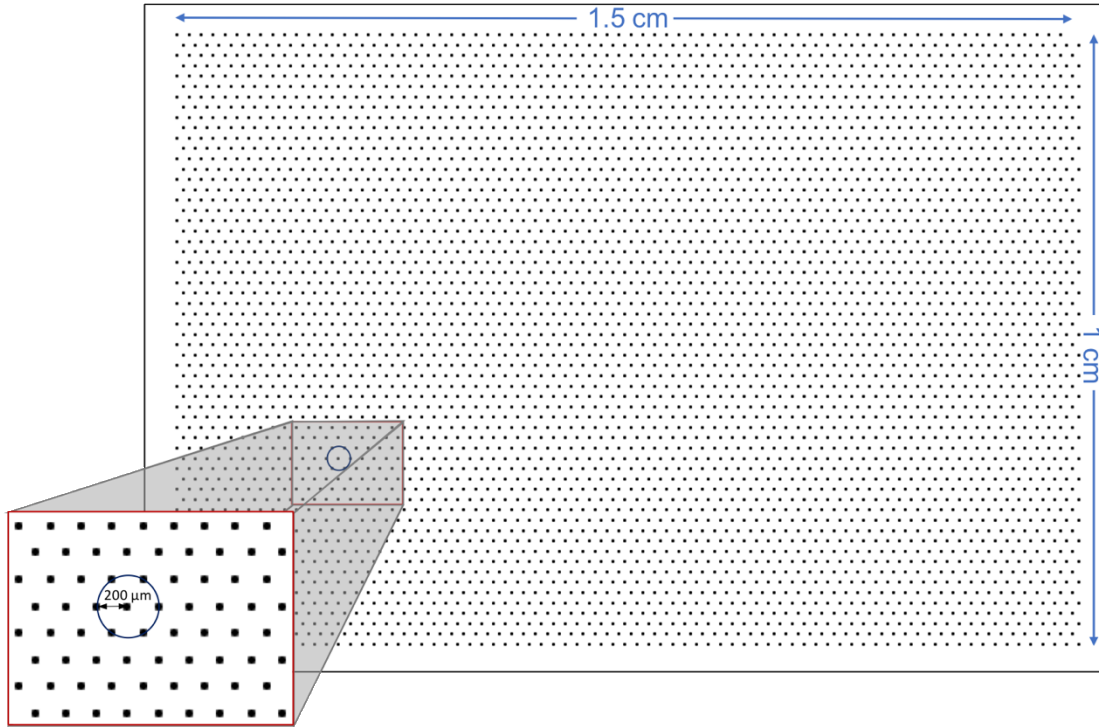


Figure 15. Grid of points to print. Changes to parameters of the scan path over these points will create different regions of microstructure.



(a) 50-50 geometry



(b) Oak leaf geometry

Figure 16. Intent for microstructure regions for the two experiments. Black will be columnar, and white will be equiaxed.

how well these optimization techniques work for different types of microstructure regions. For simulating the solutions we used the values enumerated in Table 4. Additionally, for the genetic algorithm a global spot time of 0.23 ms was chosen because it was observed to yield a reasonable mix of simulated equiaxed and columnar regions for the print area given a random path.

Experiments and Results

For each geometry we run three sets of optimizations: genetic algorithm alone, concurrent greedy line search alone, and concurrent greedy line search on the result of genetic algorithm. We refer to the third optimization as “hybrid”. Also, for comparison purposes we use mean absolute pixel error to report the quality of the different solutions calculated as shown in Equation 14.

Equation 14.

$$\text{MAE} = \frac{\sum^P |\phi_{\text{target}} - \phi_p|}{P}$$

50-50 Geometry

A 50-50 pattern, shown in Figure 16(a), was chosen to simply test the ability of the methods to utilize modeled data and manipulate spot path parameters in order to optimize microstructure — a proof of concept. This geometry is simple and has no complex boundary between the two different regions; however, it is difficult because the desired equiaxed region is not in the center of the part. The equiaxed region is off to the side of

Table 4. Simulation parameters.

Variable	Value
Beam Size X	10 μm
Beam Size Y	10 μm
Beam Size Z	100 μm
Absorption Efficiency	80%
Beam Power	80 W
Latent Heat	On
Material Properties	Inconel 718

the geometry where boundary conditions typically induce more columnar microstructure. Figure 17 shows the simulated results for the first and last iteration for each of the three optimizations, and Table 5 enumerates the mean absolute pixel errors for the final solutions.

Oak Leaf Geometry

The oak leaf pattern, shown in Figure 16(b), was chosen to test the flexibility and generalizability of the optimization algorithms. Its equiaxed region is in the center, but the boundary between the two different regions is much more complex, with an interlacing of the two regions around the lobes and margins of the leaf. Figure 18 shows the simulated results for the first and last iteration for each of the three optimizations, and Table 6 enumerates the mean absolute pixel errors for the final solutions.

Analysis of Optimization

Initial visual inspection of the resulting scan paths does not readily confer insights about what path features correlate with the resulting microstructure. Figure 19 shows the initial spot path from the first generation of the genetic algorithm, (a), and the final order from the last generation of the genetic algorithm, (b), for the oak leaf geometry. Again, simply observing the raw scan path order offers little information that differentiates the initial and final scan paths. So, in order to understand and quantify the differences between the different scan strategies, we developed three metrics to evaluate certain intrinsic properties for each point based on the scan path itself. After running the experiments we

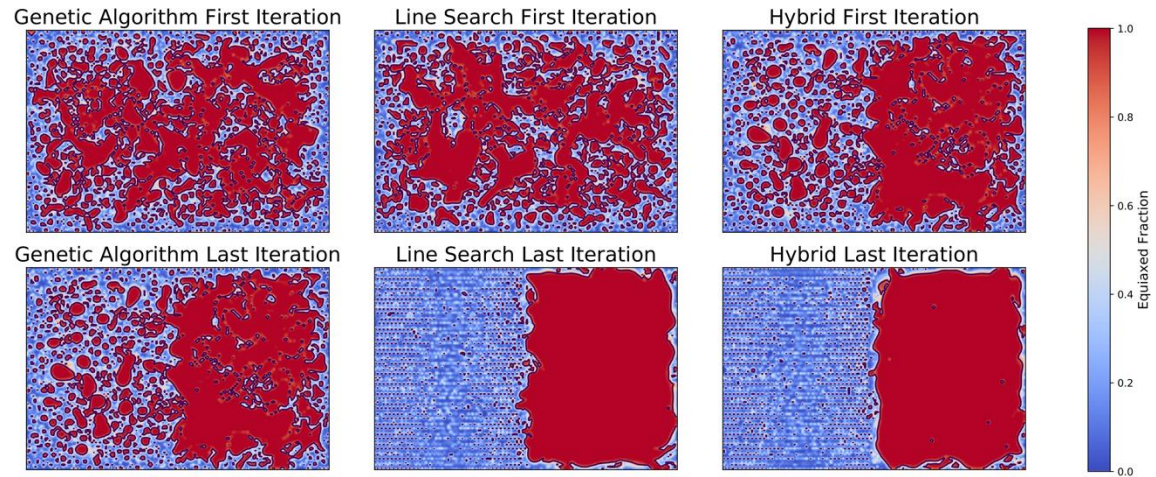


Figure 17. Simulated microstructure of different scan path parameters for 50-50 geometry.

Table 5. Mean absolute pixel error for each optimization scheme for the 50-50 geometry, Figure 17.

Method Description	Mean Absolute Error
Genetic Algorithm	0.354
Concurrent Greedy Line Search	0.154
Hybrid	0.152

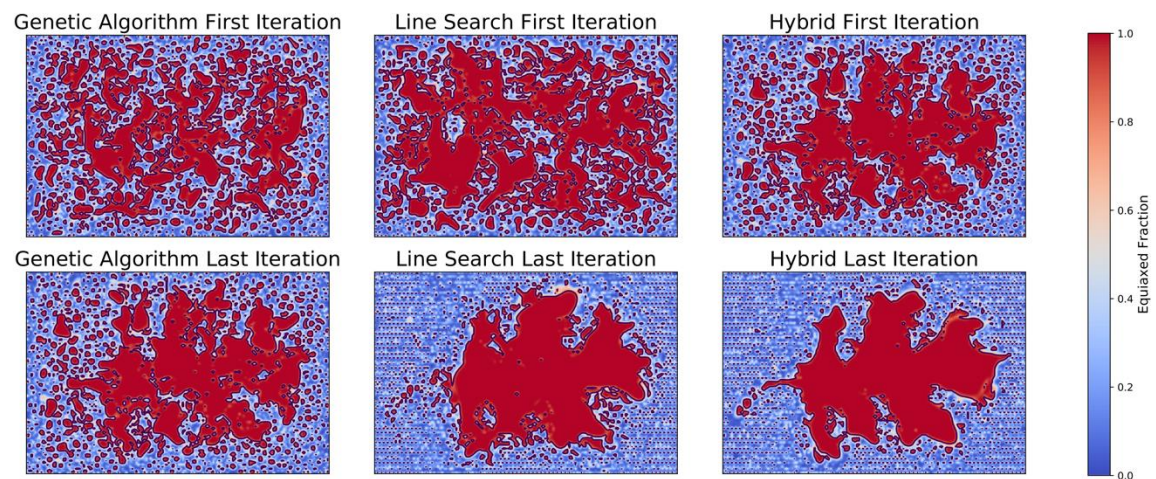


Figure 18. Simulated microstructure of different scan path parameters for leaf geometry.

Table 6. Mean absolute pixel error for each optimization scheme for the oak leaf geometry, Figure 18.

Method Description	Mean Absolute Error
Genetic Algorithm	0.339
Concurrent Greedy Line Search	0.251
Hybrid	0.220

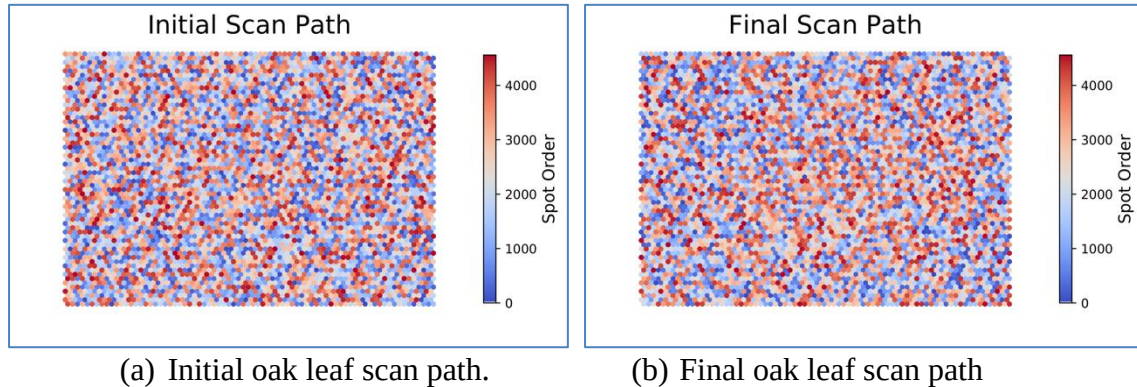


Figure 19. Initial and final spot paths for the leaf geometry after genetic algorithm optimization. The color represents the position in the spot order – points that are dark blue are at the beginning, beige in the middle, bright red at the end.

can analyze the changes between the initial poor solutions and the final optimized solutions in order to determine how these intrinsic path characteristics differ. Then we can explore how these path properties relate to the control of local material characteristics of the simulated part. The metrics quantitatively compare individual points across different scan orders and spot times based on their average position in time in the spot order, variation or deviation in spot order position from neighbors, and an indicator of the relative cumulative energy for a point during the simulated melt.

These three metrics involve a weighting scheme in order to account for the effects that surrounding points have on each individual point. The interaction between points is based on the complex relationship across space and time, and the intuition is that points that lie closer together in space and time have a greater effect on each other. The spatial component of the weight is calculated using standard Euclidean distance. The temporal portion of the weight is based on the conduction diffusion equation

Equation 15.

$$\delta = \sqrt{\alpha(t - t_0)} ,$$

where $(t-t_0)$ is the amount of elapsed time since the energy was deposited, α is the thermal diffusivity of the material, and δ represents the radius of diffusion. So, we combined these two equations into a single spatio-temporal distance metric as

Equation 16.

$$D = \sqrt{|x|^2 + \delta^2} .$$

The weight matrix is then calculated as

Equation 17.

$$w = \frac{1}{D} \text{ where } D > 0 .$$

Then, the average moment in time that a given point was printed is calculated as shown in Equation 18, where w_i is the weight between the current point and point i and t_i is the time that point i is printed in the overall order. This metric serves to highlight regions of points that are printed, on average, towards the beginning, middle, or end of the melt.

Equation 18.

$$\mu_t = \frac{\sum_i w_i t_i}{\sum_i w_i}$$

Additionally, we calculate a weighted variance of position in the spot order for every point in the path. The intuition for this calculation is that it serves to show regions that have relatively low, average, and high variation in spot order. We can infer from this information which regions have a high, medium, and low energy density deposition, respectively. Put another way this metric shows which regions were exposed to energy

from the beam at high, medium, and low frequencies. This calculation is enumerated in Equation 19, where w_i is the weight between the current point and point i , t_i is the time that point i is printed in the overall order, and μ_{ti} is the average time in the spot order that point i is printed as calculated in Equation 18.

Equation 19.

$$s_t = \frac{\sum_i w_i (t_i - \mu_{ti})}{\sum_i w_i - \bar{w}_1}$$

Finally, we developed one additional metric that we termed “cumulative energy”. This calculation serves to demonstrate the proportions of total energy that was deposited at each point coupled with the contribution of energy from neighboring points. This metric allows quick inspection of how changing spot time and spot order affect the the relative amount of energy deposited at a given point. The calculated “energy” for a point is the beam on time for a spot. Again, we use the spatio-temporal weighting scheme so the energy from neighboring points that are melted far apart in time contribute less to each other's cumulative energy. This metric is shown in Equation 20, where w_i is the weight between the current point and point i and Δt_i is the beam duration for spot i .

Equation 20.

$$e_t = \sum_i w_i \Delta t_i$$

We can visualize these values for each point in a scan path, see Figure 20(a) through Figure 22(a). From this we get interesting information. First, it allows us to observe spatial trends. The weighted average can show regions that are printed toward the beginning or end of the melt, for instance. The weighted variance shows regions where the beam visits at higher and lower frequencies. Likewise, cumulative energy shows where the beam deposits relatively more or less energy over the course of the melt. Additionally, it allows us to observe how these spatial trends and regions of metrics change in light of the optimization that was performed to produce the desired microstructure regions - we can see how the metrics change in order to induce desired microstructure.

Additionally, we can look at these metrics in relation to the actual predicted microstructure. This gives insight into the possible differences between predicted microstructure regions for different scan paths and different optimizations. To this end we use an equiaxed fraction value of 0.8 to divide the predicted microstructure into two distinct regions: less than 0.8 is columnar and greater is equiaxed. Then we can see how the values of these metrics are distributed for the different microstructures for the different scan paths - see Figure 20(b) through Figure 22(b).

Figure 20 through Figure 22 graphically show these metrics for the beginning and end scan paths for each of the different optimization methods. In the plot for genetic algorithm, Figure 20(a), the charts on the far left show that the cumulative energy does

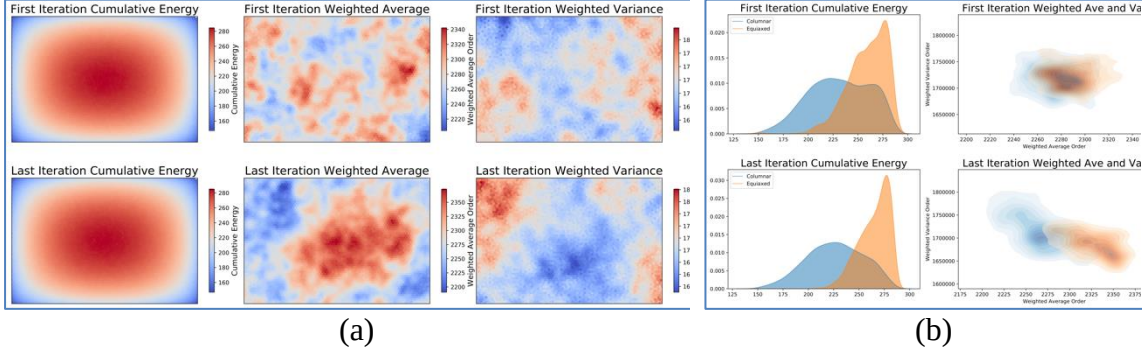


Figure 20. Changes to scan path metrics for the leaf after genetic algorithm. The left half shows how the three different metrics (columns) change for each point from the first scan path (row 1) to the optimized scan path (row 2). The right side of the figure shows how the distributions of the metrics over the two predicted microstructure regions change.

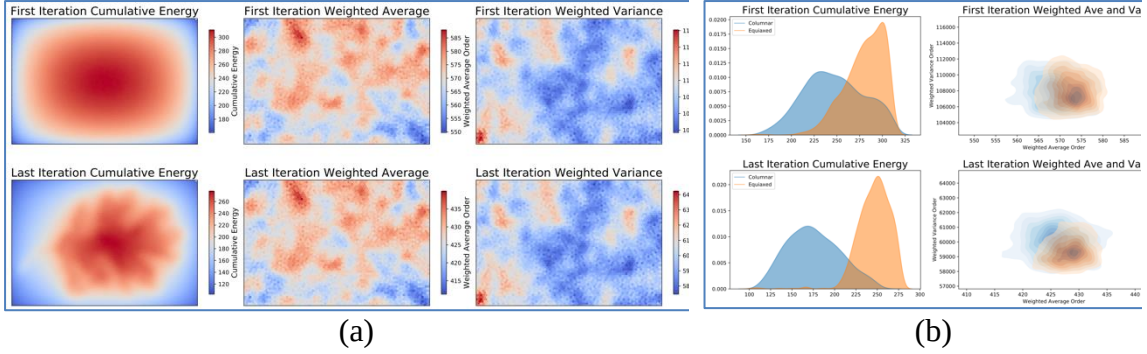


Figure 21. Changes to scan path metrics for the leaf after concurrent greedy line search.

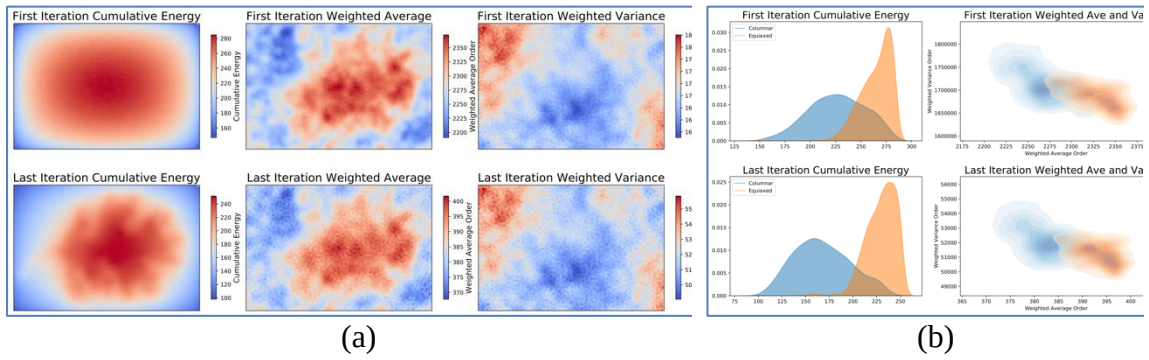


Figure 22. Changes to scan path metrics for the leaf after genetic algorithm then concurrent greedy line search.

not noticeably change as a result of changing the spot order; however, one may note that the optimized spot order exhibits some interesting properties for average spot order and variance of spot order. Indeed, the genetic algorithm generated a solution where the desired equiaxed regions are printed towards the end of the order on average and also have a relatively lower variance compared to the desired columnar regions. This aligns with the empirical evidence that higher local preheat temperature leads to more equiaxed grain formation.

Figure 20 shows how initially predicted columnar and equiaxed regions are almost equally likely to occur across calculated values of weighted average and variance of spot order. However, after the optimization, predicted columnar regions appear preferentially in regions that are printed toward the beginning of the melt and with a higher variance, and predicted equiaxed regions appear preferentially in regions that are printed toward the end of the melt and with a lower variance. So, we see that unlike the initial random scan path, the occurrences of the two different predicted microstructures in the final optimized scan path have diverged with respect to weighted average and weighted variance of scan order. So while the different textures of microstructure initially form with little preference to average order and frequency of the scan path, the scan path can be manipulated such that it influences how microstructures form preferentially depending on intrinsic characteristics of the path. Also, some divergence is seen between the two regions for cumulative energy from the initial to final spot orderings.

Figure 21 shows the metrics for the initial and final scan paths for the line search algorithm optimization of spot time. This method appears to only affect the cumulative energy metric. It also has a noticeable effect on the absolute values for the average and variance of spot order metrics. However, unlike the spot order optimization, the two regions of predicted microstructure show little to no divergence in weighted average and variance, shown in Figure 21(b). Additionally, the divergence in cumulative energy between the two regions is much more prominent than in the spot order optimization.

Finally, the hybrid optimization method, where spot order was optimized first with genetic algorithm followed by spot time optimized by line search is shown in Figure 22. This approach shows a seeming superposition of the effects of both of the individual optimizations. One can observe a large differentiation in cumulative energy, weighted average, and weighted variance for the two microstructure regions.

Discussion and Future Work

This work has demonstrated two different optimization techniques that can be leveraged during the design process in order to facilitate the control of microstructure. With the genetic algorithm we found that within the space of possible point orders there are some orders that are better suited for producing desired arbitrary microstructure regions. Also, a genetic algorithm is absolutely necessary in order to discover such a path because even the smallest geometry has a possible solution space that is far too large to fully enumerate. For example, for our genetic algorithm we simulated approximately 30,000

different scan paths in order to find one that reasonably approximated the desired microstructure regions. For reference, a geometry with only eight points has a solution space of 40,320 possible orderings. This is because the search space is a permutation of possible point orderings; and therefore, it increases as a factorial of the number of points. Nonetheless, the genetic algorithm approach is slow — taking about five days to complete the 250 generations and yield a solution.

The concurrent greedy line search algorithm, on the other hand, yielded seemingly better results in a fraction of the time. This is because the dominating factor for determining the run time of the various algorithms is the number of times the heat transfer model is run because its compute time eclipses that of the actual optimization code. Equation 21 shows the number of times that the genetic algorithm calls the simulation code.

Equation 21.

$$O\left(\frac{GP}{J}M\right)$$

So the magnitude of the genetic algorithm run time is a function of the model run time multiplied by the number of generations, G , and the population size, P , and divided by the number of jobs or batches that the population can be divided into and run in parallel. For our experiments the number of generations was 250, with a population size of 119, and split into 7 jobs. With a 1.8 minute execution time for the model for our problem size

this puts the total run time for the genetic algorithm right in the neighborhood of our previously stated five days.

However, it may not be necessary to run the genetic algorithm as long as we did in order to find a reasonable solution that could be the basis of the hybrid optimization. Figure 23 shows the error curves of the two geometries over the generations. While the error curve is steadily decreasing over the generations, it does so at a decreasing rate. As such, a specific MAE value could be chosen that could be the basis of an early stopping scheme. That scan path could then be used as the input to the concurrent greedy line search similar to the hybrid approach.

In contrast to the long run time of the genetic algorithm and its many calls to the heat transfer model, Equation 22 shows the number of times that the concurrent greedy line search calls the simulations code.

Equation 22.

$$O(I M)$$

Again, the run time is a function of the model execution time multiplied by the number of iterations. The line search algorithm would terminate after seven to eleven iterations for a total worse case execution time of approximately 20 minutes. The execution time for the hybrid approach, then, is just the sum of the previous two. As such this method, in

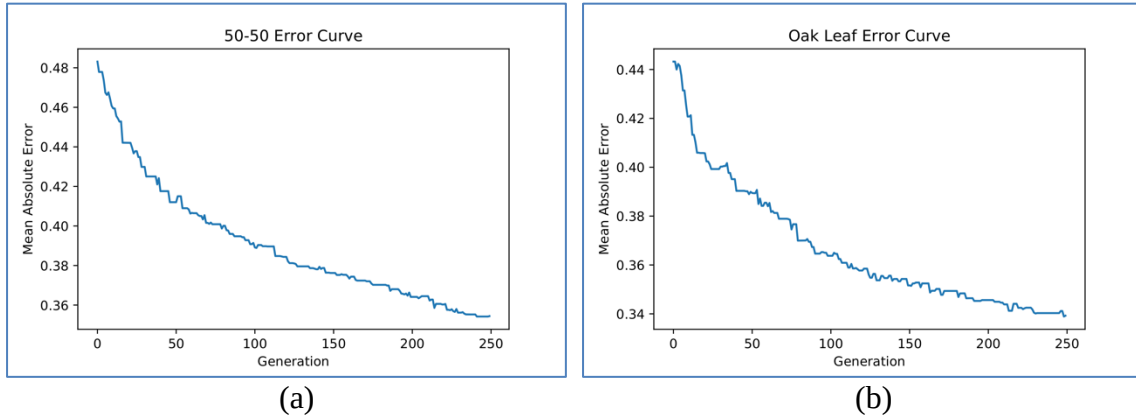


Figure 23. The Mean Absolute Error for the best solution in each generation for the two geometries. Note that the MAE does not decrease monotonically because it only measures the accuracy of the predicted microstructure in comparison to the desired. The genetic algorithm, however, is optimizing a fitness function that factors both this error and the heuristic about distances between points in the scan path.

isolation, is seemingly one to pursue for quick, useful results. However, this more crude method offers less granularity for control. Ultimately, we found that the best results stemmed from employing both of these optimization techniques in unison, and early stopping of the genetic algorithm may provide adequate results for use in the hybrid approach.

Additionally, the scan path metrics proposed in this work present a rough framework for quantitatively comparing the characteristics of different possible scan paths. These metrics may be used to more fully understand path properties, and when leveraged with models for microstructure or others, be employed to understand relationships between scan path properties and resulting material properties.

We foresee several tasks that could be performed to build upon this work. This framework may be deployed with different models that simulate the relationship between scan path and microstructure or it could even be leveraged with other models that predict other mechanical properties based on scan paths for optimization of those properties.

With increased model scalability will come the possibility of performing spot path optimizations on larger cross sections and eventually across multiple layers. Further work will also include optimization of other parameters including spot spacing, spot density, and spot location. Relaxing the rigid grid of point will lead to greater customization and control of resulting microstructure.

Additionally, once well quantified and understood, effort should be put forth towards creating more complex, heuristic-based spot in-fill patterns for controlling features of the final part that are based on the results of the genetic algorithm. This will allow insights from the genetic algorithm to be leveraged for new geometries without requiring the full algorithm to be run again. The scan path metrics may be used as the basis for determining the heuristics to be used in order to optimize properties for new geometries.

CHAPTER II
SWAN: SOFTWARE FOR ADDITIVE INVESTIGATION — COMMON DATA
REPRESENTATIONS AND DATA FUSION

A version of this chapter has been submitted for publication by William Halsey, Luke Scime, Michael Sprayberry, Zack Snow, Vincent Paquit:

William Halsey, Luke Scime, Michael Sprayberry, Zack Snow, Vincent Paquit.
“SWAN: Software for Additive Investigation — Common Data Representations and Data Fusion.”

The student was responsible for ideation, implementation, and manuscript writing. Dr.

Luke Scime served as subject matter expert for Concept Laser M2 data. Dr. Michael

Sprayberry served as subject matter expert for Arcam Q10 data. Dr. Zack Snow and Dr.

Luke Scime assisted in code verification. Dr. Vincent Paquit assisted in funding this

work. All coauthor reviewed the manuscript.

Abstract

One of the exciting aspects of additive manufacturing is the ability to produce small batches or individual components; however, realizing this feat will require new processes for component qualification. This could be accomplished with a richly expressed digital twin of the component. The data that would comprise such a digital twin are of different modalities — both spatial and temporal — and methods for leveraging them in conjunction must be developed. Additionally, building generalizable data processing pipelines is confounded by vendor specific file types for analogous data. We introduce SWAN which is a growing Python library. It contains definitions of abstract common data representations for analogous data types that are produced during additive manufacturing as well as data parsers to read and transform vendor-specific data to the corresponding common data representation. SWAN also aims to capture a growing set of data processing routines and pipelines that may be built upon the common data

representations and therefore applied to any machine that has appropriate data parsers as shown in Figure 24.

These key ideas will be demonstrated through a representative case study — spatial mapping of temporal process data leveraging scan path information. This capability will be demonstrated for two distinct powder bed fusion technologies — an Arcam Q10 and a Concept Laser M2 representing electron beam powder bed fusion (EB-PBF) and laser powder bed fusion (L-PBF) respectively. In this context, the common data representations for temporal process data and scan path data will be discussed and rationalized, the data processing routine to perform the spatial mapping will be described and discussed, and time complexity studies will be performed and analyzed.

Introduction and Background

Additive manufacturing (AM), also commonly known as 3D printing, is a relatively new manufacturing paradigm that has garnered increasing attention since its inception. The process consists of selective deposition or fusion of material in an iterative fashion — often layer by layer — until the target geometry has been achieved. AM has the potential to be an enabling driver toward more complex component design and customizability, less material waste, more production flexibility, and less overhead for small batches of components.

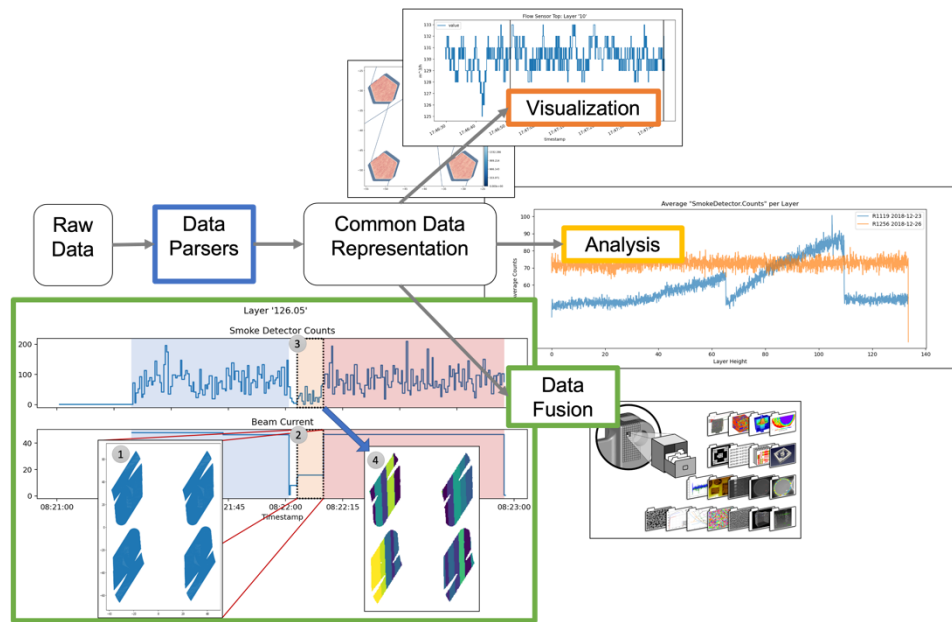


Figure 24. Data processing steps for building generalizable data pipelines for exploring and processing in-situ data from additive manufacturing using SWAN.

Due to these unique characteristics of AM processes, industries such as automotive, aerospace, and energy are interested in developing and leveraging AM technologies further — especially metal AM — for fabrication of geometrically complex components for small production runs. These industries also require strict tolerances and robust certification and qualification of components in their various applications. Currently, these needs have exposed a shortcoming of metal AM technologies — the variability of material and mechanical properties across machines, components, and even layers or regions of an individual component. However, the incremental nature of AM also offers opportunities for unprecedented monitoring of the fabrication process that can be directly linked to locations within a component[7]. As such, one goal major goal in the metal AM research community is to leverage process data and develop models to link variations in the AM process to the resulting material structures and flaws and ultimately to the final mechanical properties[16], [64]–[66].

Component Digital Twins for Certification and Qualification

The metal AM process is full of variability even when using nominally robust processing and scan path parameters[6] (refer to APPENDIX A — Glossary of Scan Path Terms). The variability in the AM process leads to variability in the mechanical performance of fabricate components and stems from interactions between the powder, heat source, shield gas flow, other machine subsystems, and the environment. Scime et al.[67] discusses how likely use cases for AM entail small batch sizes coupled with the variability of the process leads to the need for instance qualification. Instance

qualification will demand data related to each individual fabricated component as opposed to traditional qualification methods that rely on aggregated data. An instance-specific dataset from which additional discoveries may be made about a physical object or system is referred to as digital twin[68], [69]. Currently, much of the research that uses the term digital twin heavily focuses on process modeling[70]–[73] and does not address the possibilities of collecting process data to create a digital twin of the manufactured component for analysis to assess component quality.

To these ends, another area of investigation is exploring in situ process monitoring and data collection methods that would allow for the detection of flaws in an AM builds[7], [11]. Sharrat[7] describes several monitoring and qualification methodologies that may be leveraged. The first is to monitor the stability of the manufacturing process using machine health monitoring sensors for any disruptions in the build environment as they may indicate flaws. Another approach is to observe each fabricated layer using cameras and identify any flaws from collected layer-wise images. The accumulation of these detected flaws yields a three-dimensional flaw map that may assist in qualification[7]. Data for assessing process stability is typically some form of temporal data, while data for layer-wise flaw detection is often image data. An instance-specific digital twin will contain both types of data; however, they are not trivially utilizable in conjunction for qualification. With an appropriate transformation to convert the temporal process data into the spatial domain, both sets of data could be leveraged together for more accurate and localized flaw detection. Groeber and Jackson[74] introduce software and data

representation for high dimensional data for microstructure characterization and modeling called Dream.3D. They outline how spatial data from AM builds may be structured and stored in discrete voxels, among other representations. Figure 25 shows a visual representation of the proposed voxelized digital twin data paradigm.

Process Data

AM machines have the potential to produce rich multimodal datasets that may be leveraged for instance qualification. Although data from AM machines are often in vendor-specific formats, the data produced is often analogous across the various makes and models of machines. Such analogs include temporal process data, spatial image data, and scan path data.

Temporal process data

Many metal AM machines have integrated sensors that monitor the various subsystems and actuators during the fabrication of a component. Key points to monitor are the material deposition system, the heat source system, and the systems that are responsible for maintaining the build environment. Yoder et al.[30] gives an outline of how the data from the subsystems may correlate with component quality for the EB-PBF process. The authors monitor substrate temperature, vacuum pressure, recoater feedback, electron beam current, and sensors that observe smoking events — when unsintered and electrostatically charged powder is ejected from the powder bed into the build chamber[75]. L-PBF systems must monitor inert gas flow and oxygen content in the build

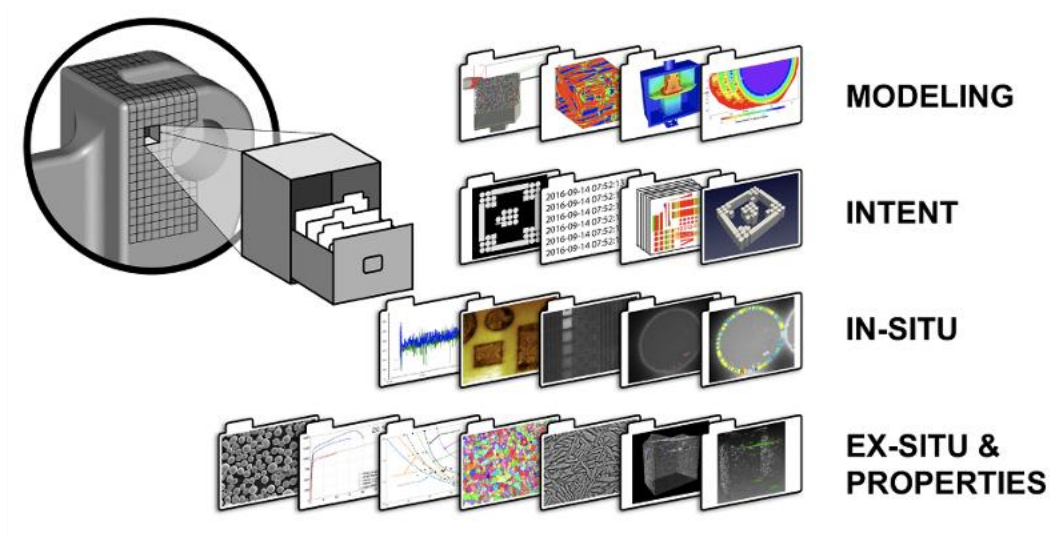


Figure 25. Illustration of a unified, voxelized, digital model of a component where data corresponding to a specific location in space is associated with a discrete voxel.

chamber as well[12]. Other temporal data is sometimes collected to monitor the process, including on-axis photodiode data and acoustic data[7]. Note that the temporal resolution of these data can vary widely. Additionally, some of the data is recorded asynchronously — that is, it is not recorded at a set frequency or regular interval.

Spatial image data

This type of data consists of images that are captured during key points during layer fabrication. Often an image is taken after material deposition, and another is taken upon completion of the melt. Many different sensor configurations have been studied varying in wavelengths detected — from visible light to long-wave infrared — and camera resolution[11].

Imaging data is an increasingly common input for flaw detection algorithms. Importantly, these data consist of discrete measurements over the spatial domain — pixels when discussing simple images or voxels when discussing the volume of data for an entire build. Spatial data stored in pixels or voxels is referred to as a raster representation. Image data is common to AM and many other applications, and many mature software libraries and packages exist to handle this data. Across software libraries the data representations for image data are similar and relatively trivial to manipulate, extend, or integrate into data processing pipelines. As such this type of data is not a primary focus for SWAN and will not be discussed further.

Scan path data

In addition to defining some key concepts in this section, a glossary of terms associated with scan path data is given in APPENDIX A — Glossary of Scan Path Terms. Powder bed fusion AM machines operate by moving a heat source across a bed of metal powder to melt and fuse the material within and across layers. The specific path that the heat source traverses across the powder bed to create the fused geometry for a layer is called a scan path or trajectory. A layer's scan path is generated by the AM systems' software and is a function of the geometrical cross section, the scan strategy, and corresponding parameters, as well as other machine parameters. Note the distinction between the scan strategy — which dictates characteristics of the space filling trajectory across all layers — and the actual scan path that represents the path travelled by the heat source for a specific layer. Scan path data contain temporal orderings of spatial locations. Thus, this data may be leveraged to map temporal data to spatial locations or the inverse operation to assign spatial characteristics to approximate time stamps — although the latter task is arguably more speculative. As the fusion of material in the powder bed AM process has complex spatiotemporal interactions, scan path data may be useful to understand these interactions, their connection to cross-sectional geometry, relation to effects on material structure, and resulting mechanical properties.

Spatial Mapping

One way to leverage time series and natively spatial data together is to employ a spatial mapping operation to convert the temporal data into the spatial domain. Having both sets

of data in the same domain will allow for investigating correlations between temporal signatures and localized flaws. So, a need exists for a robust data transformation operation — or spatial mapping — that will allow for leveraging in-situ, temporal data more intuitively. Several studies have been conducted that use temporal data for flaw detection, and many of these works require a spatial mapping step. Examples of several spatial mapping methods and applications are outlined here.

Clijsters et al.[76] describes a "mapping algorithm" to transfer data from a time-domain into a position-domain representation. The authors spatially mapped photodiode intensity data and melt pool features derived from an on-axis near IR camera. They map the data into a raster representation i.e., an image. Importantly, several key assumptions for the work are (1) that the temporal data is already temporally aligned with the scan trajectory and (2) that the temporal data is sampled at a sufficiently fast rate to achieve the final spatial resolution that they require.

Vandone et al.[77] describes a data fusion approach to combine melt pool features and statistics derived from an on-axis melt pool camera, laser power and velocity captured in the trajectory data, and three dimensional surface scans and porosity information from microscopy that are collected ex situ. They use the trajectory data in two ways. First, they use it to temporally register the melt pool data. Second, they use it to perform an inverse operation - a temporal mapping - to take the off-line data and register them such that they may be indexed by a time stamp. They accomplish this second task by spatially

registering the trajectory to the spatial data and then using the trajectory data as a "bridging function". However, this work does not expand this methodology beyond single track experiments.

Petrich et al.[78] spatially map natively temporal acoustic data, multispectral data, and other features derived from the trajectory data. Build plate coordinates, defined by the trajectory data, served as the reference coordinate frame for spatial registration. The multispectral data is recorded in sync with the scan path information and thus is temporally aligned with the trajectory data, but they use cross-correlation to temporally align the acoustic data. The data is then in a point cloud format and the rasterization is accomplished by assigning pixel-wise values based on the nearest point in the point cloud. Like Clijsters et al.[76], this approach assumes that the temporal data is sampled at a sufficiently fast rate to achieve the final spatial resolution that they require.

Kozjek et al.[79] introduces a data fusion method for combining disparate data sources from L-PBF AM. The method that they propose is not exactly spatial mapping because the data they fuse already have associated spatial coordinates. However, their method seeks to spatially align on-axis photodiode data and ex-situ x-ray computed tomography (CT) data with the machine input tool path using iterative closest point-clouds (ICP) registration. They further leverage the relative sample times of the machine input tool path and on-axis photodiode data to refine the spatial alignment of these data modalities.

They also propose a method for organizing and storing the fused data for efficient data access and analysis.

These works report the necessary steps required for mapping and fusing temporal data with other spatial raster data: (1) temporal registration with the scan trajectory data and (2) rasterization to transform the data into an image-like form. They also demonstrate some key assumptions that may not hold for all temporal process data. For example, temporal data that is sampled concurrently with the trajectory data may or may not require temporal registration and that step might be skipped. However, one should not simply assume that concurrently captured data is inherently temporally registered. A second assumption is related to the format of the trajectory data and sampling rate of the temporal data that do not hold - namely that both are regularly sampled and at a high enough rate to meet the spatial resolution needs. This is often not true with the temporal process data collected in machine log files, and below we will discuss some examples in more detail; however, a more general spatial mapping method is needed to handle all of the different temporal data streams that are available.

Materials and Methods

Significant variability in raw data representations and formats exist across AM machine Original Equipment Manufacturers (OEM). The variability complicates building complex data pipelines and analyses that are usable for more than a subset of AM machines. Thus, the introduction of common data representations facilitates data processing pipelines and

transformations, like the spatial mapping procedures we will demonstrate, that will be extensible to other AM machines. All that will be required are data parsers that will transform raw machine data to conform to the common data representation. Note that the common data representations outlined herein are conceptual data representations and not rigid implementation specifications about data types, etc.

The spatial mapping operation requires two sets of data from the AM process – temporal data that is to be mapped and trajectory data that contains location and timing data for the heat source. As a demonstration, this work focuses on one type of temporal data - namely data from machine build logs. This was chosen due to the common availability of such data and because machine log data illuminate specific nuances of spatial mapping task since this data is often sampled at much lower frequencies or at irregular intervals unlike other on-axis sensor data. Spatially mapping these data will create a spatial representation of, and facilitate better comprehension about, build environment changes and how they align with the heat source trajectory and correlate with localized variation in material properties. We will discuss the results from spatial mapping for two metal powder bed AM machine — Arcam Q10 and Concept Laser M2. Additionally, we will show spatial mapping results of on-axis photodiode data to demonstrate the applicability of this method to other forms of time series data that are collected in-situ.

Temporal, In-Situ Process Data

The Arcam Q10 and Concept Laser M2 both have sensors that monitor and record the status and operation of their subsystems during a build. These data are stored in a text-based log file, and each record contains, in some form, a time stamp, variable or sensor name, and a corresponding value. Figure 26 provides an illustration for this type of data for the Concept Laser M2 and Arcam Q10 machines respectively. From this data one has access to a temporal view of the build or build environment. Importantly, the log files also contain timestamps that correspond to the start of the melt phase for each layer.

The essential components of temporal data for a given process variable are a series of timestamps and corresponding sensor readings. Theoretically, any data that has these components will be spatially mappable. Note, as shown in Figure 26, we adopt the convention of imputing a constant value between successive sampled values producing a stepwise time series.

Trajectory Data

The trajectory data for a given layer contains information about the location and timing of the heat source during the melt phase. The format of the raw data is often different between different OEMs; however, some work has been undertaken to create a standardized data format to communicate with machines[80]. Further, we must note that workflows for each machine may produce at least two distinct sets of trajectory data — commanded trajectory and measured trajectory. For the proposed task, using the

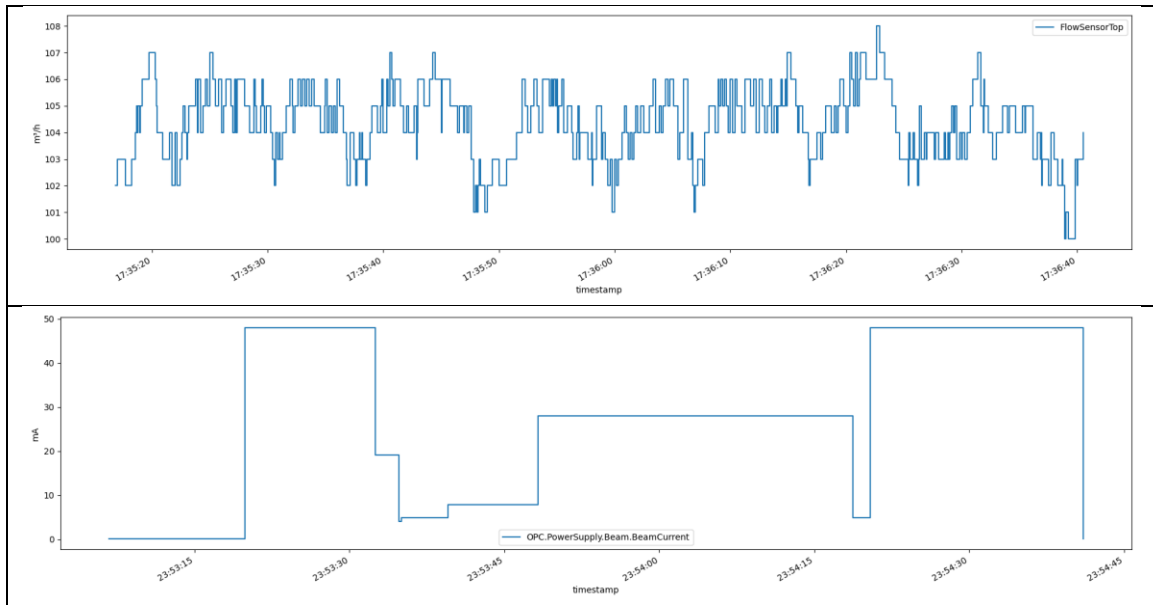


Figure 26. Examples of temporal, in-situ data captured from (top) a Concept Laser M2 machine and (bottom) an Arcam Q10 machine.

measured trajectory is ideal because it will enable to most accurate localization of temporal events. However, utilizing intended trajectory data may not be a fruitless endeavor, and either type may be suitable to utilize for spatial mapping and data fusion, with varying degrees of accuracy, provided that the raw data can be transformed into the proposed common data representation. Trajectory data from Arcam Q10 machines are not typically output by the machine software, and it exists in a binary format. However, the "hatch2D" trajectory format from OpenVectorFormat[80] has some similarities and can be useful to better understand the data format that will also be described here.

Arcam Q10 machine

The trajectory data for the Arcam Q10 machine is generated by the machine prior to printing and represents the intent for the layer. Note that these files do not contain contour information. The trajectory data contains the x and y locations for start and end points of a scan line as well as speed information. From this information the temporal duration of each scan line and of the total melt time may be derived. Table 7 shows a list of a subset of features available in the Arcam Q10 scan path files, and Figure 27 shows a plot of the scan lines for a single layer.

Concept Laser M2 machine

The trajectory data for the Concept Laser M2 machine is collected from a National Instrument PXIe data acquisition module at a sampling rate of 40 KHz. Each sample contains the x and y location of the heat source at that moment. Using the distance

Table 7: Data that are available in the Arcam Q10 scan path files.

Feature	Description
P1.X	x location of the start point
P1.Y	y location of the start point
P2.X	x location of the end point
P2.Y	y location of the end point
Speed	Linear velocity of the electron beam in <i>m/s</i>

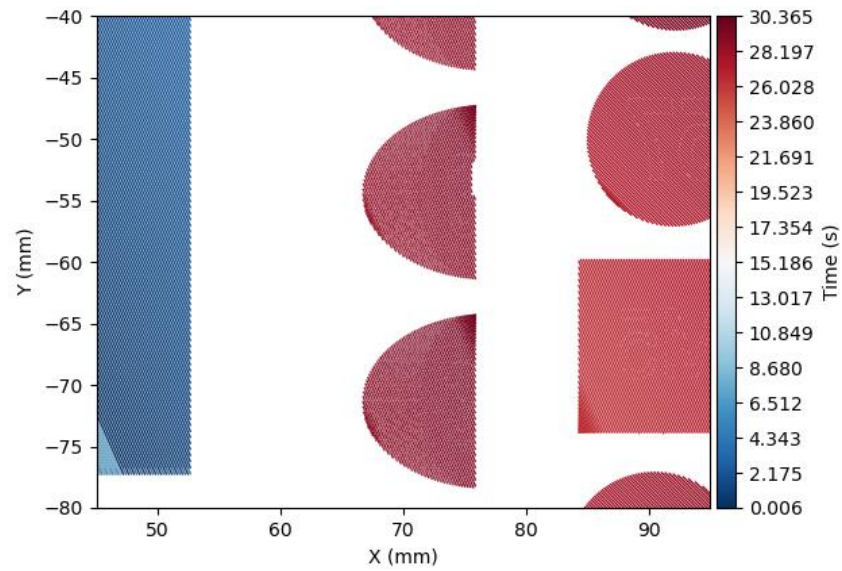


Figure 27. Example of trajectory data from an Arcam Q10 machine.

between consecutive sampled locations and the sampling rate, the speed of each scan line and of the total melt time may be derived. Additionally, the scan path data contains the features described in Table 8, and Figure 28 shows a plot of the scan lines for a single layer.

Common data representation for trajectory data

As previously described, the trajectory data may be presented or structured in several different ways depending on how the data is generated. Trajectory data, in general, represents a complex piece-wise parametric function that relates time, t , to location, (x, y) . A simple sampling of coordinates at a constant rate might not be sufficient for several scenarios — largely related to the EB-PBF process. For example, EB-PBF systems have a large range of possible speeds at which the beam can move. Simple sampling of scan paths where this occurs would lead to spatial under-sampling in regions where the beam is moving fast and over-samplings where the beam is moving more slowly. Additionally, the EB-PBF process is capable of using point or spot melt strategies where the beam dwells in a single location for a period of time. The dwell time does not need to be strictly constant across all multiple melt phases for a single layer.

It is possible to handle all these scenarios if a robust common data representation is chosen for trajectory data. The most flexible data format is similar to the former structure from the OpenVectorFormat "hatch2D" representation where each scan is represented by both a start and end point. For effective spatiotemporal transformation, each hatch must

Table 8: Data that are available in Concept Laser M2 scan path files.

Feature	Description
X	x location of the sampled point
Y	y location of the sampled point
Area	<i>Derived metric related to the detected melt pool size</i>
Intensity	<i>On-axis photodiode intensity</i>
Parameter	Indicator of which parameter set is being used

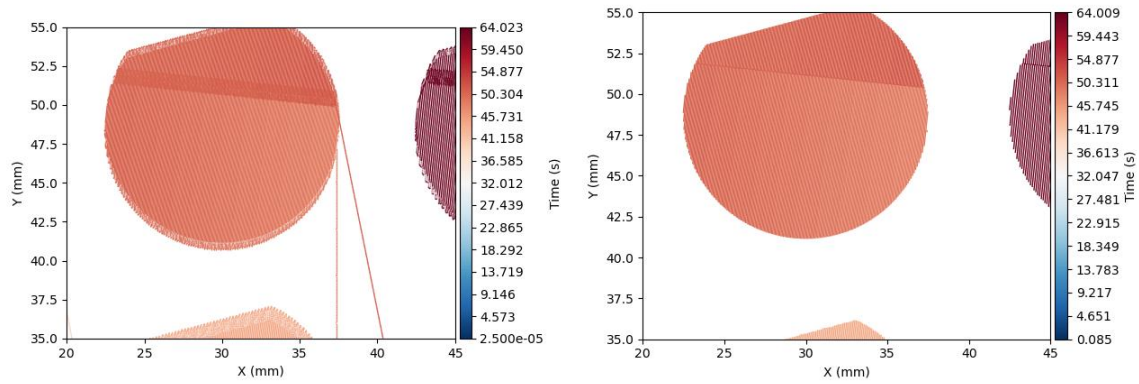


Figure 28. Example of trajectory data captured from a Concept Laser M2 machine with Left) non-melting laser motion included and Right) non-melting laser motion removed.

have duration information. The proposed common data representation for trajectory data is shown in Table 9. Adopting this form will allow any spatial mapping operation to be applicable to both technologies and all print strategies that are known thus far. From these features other necessary and useful features may also be derived such as speed, distance, and relative time since the start of the melt.

Example of converting sampled trajectory data to the common data representation

As noted, the data from the Concept Laser M2 machine contains sampled x and y coordinates at a nominally fixed rate of 40 kHz. The conversion of this data to the chosen common data representation would occur in two steps. First, sampled locations are converted into a series of line segments. Each sample location serves as the start point of a segment. The end point, then, is simply the next sampled location. A list of N sampled locations becomes a series of $N - 1$ subscans. Second, with a known sampling rate, the duration of each subscan is trivially derived. If the laser state (i.e., on or off) is known these subscans can be consolidated into atomic hatches thereby significantly reducing the size of the representation.

Spatial Mapping Algorithm

Once the temporal and trajectory data have been parsed into the specified common data representations, the temporal data may be spatially mapped. The spatial mapping operation consists of two distinct steps: (1) rasterization – that is, the data will be mapped to discrete voxels throughout the component volume at a specified spatial resolution and

Table 9: Proposed scan path common data representation.

Feature	Description
X1	x location of the start point
Y1	y location of the end point
X2	x location of the end point
Y2	y location of the end point
Δt	Temporal duration of the sub scan

(2) temporal registration. This will allow for correlation of the mapped temporal data to other spatial data — in-situ images for example.

Rasterization

To complete this task, we require a fast method for converting a vector representation of the trajectory data, represented by coordinates of the start and end locations, into a rasterized representation of the line connecting the start and end points of the subscan. This is a common task in computer graphics, and one of the original methods for determining the set of pixels that best approximates a line is called the Bresenham algorithm[81]. A fast Python implementation of the Bresenham algorithm exists that utilizes vectorized operations in Numpy[82], and it requires pixel locations of the start and end of each line as input. Figure 29 shows the change of coordinate transformation that is used to convert the cartesian coordinates of a point on the build plate coordinate system to an approximate pixel location; it also shows the output of the Bresenham algorithm of the best set of pixels that approximate a line given a start and end pixel.

The basic algorithm for creating the spatial map then is:

For each scan line

1. Find containing voxel of start location
2. Find containing voxel of end location
3. `Pixels = Bresenham(start_pix, end_pix)`
4. `Duration per pix = scan duration / number of pixels`

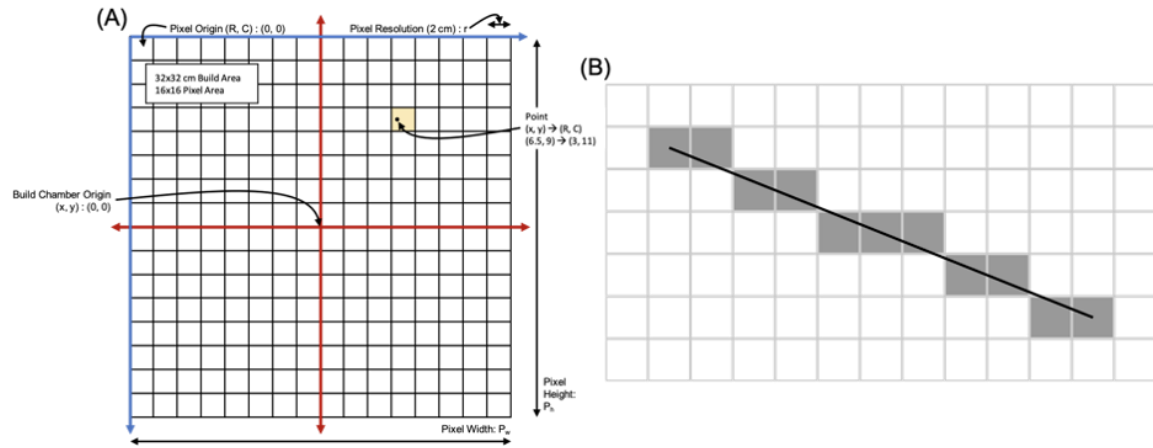


Figure 29. Graphical representation of the steps implemented to rasterize trajectory hatch lines.

5. Calculate relative time that the heat source enters each pixel

This spatial map is an array that contains the time that the heat source intersects each traversed pixel relative to the start of the melt. All that remains is to determine the absolute time that the melt began. That time may be used to index into the temporal data for any variable and retrieve the corresponding value using any desired interpolation procedure.

Several representative images after the rasterization step are shown in APPENDIX B — Example Rasterized Spatio-Temporal Maps. They show the output without and with utilizing a naive hole filling algorithm. This is necessary for pixels that are not intersected by a theoretically, infinitesimally thin scan line when the hatch spacing is greater than the pixel width.

Temporal registration

The trajectory data for these machines only provide timestamps for each location that are relative to the start of each melt phase for a given layer. The log data, however, have absolute, real timestamps. To align these data, we identify signatures in the machine log files that indicate the beginning of each melt phase. The relative time stamps in the trajectory data then represent offsets from the absolute beginning of the melt for a given layer.

Assumptions and rationalizations

This approach requires several assumptions. First, the signature used from the machine log files to identify the beginning of the melt phase is assumed to be accurate. Incorrect temporal registration will lead to spatially mis-mapped data, which would throw off any possible correlations between the mapped data and localized characteristics. The Arcam machine will be particularly susceptible to this as the beam traverses over the powder bed at significantly faster speeds. However, if a constant interpolation scheme is used when indexing into the time series, then the low sampling rate of the machine log data, relative to the speed of the heat source, will mean that most voxels will contain the correctly mapped value. Additionally, this initial implementation will not account for any temporal delays that may exist between a change in a machine log value and its possible subsequent effect at the powder bed — changes in speed of the inert gas that flows over the powder bed for L-PBF that is sensed at an inlet or outlet for example.

Results

Several representative images from spatially mapping process data are shown below in Figure 30 through Figure 32. Figure 33 through Figure 35 show how the rasterization time changes with changing pixel resolution, number of pixels, and number of scan lines respectively.

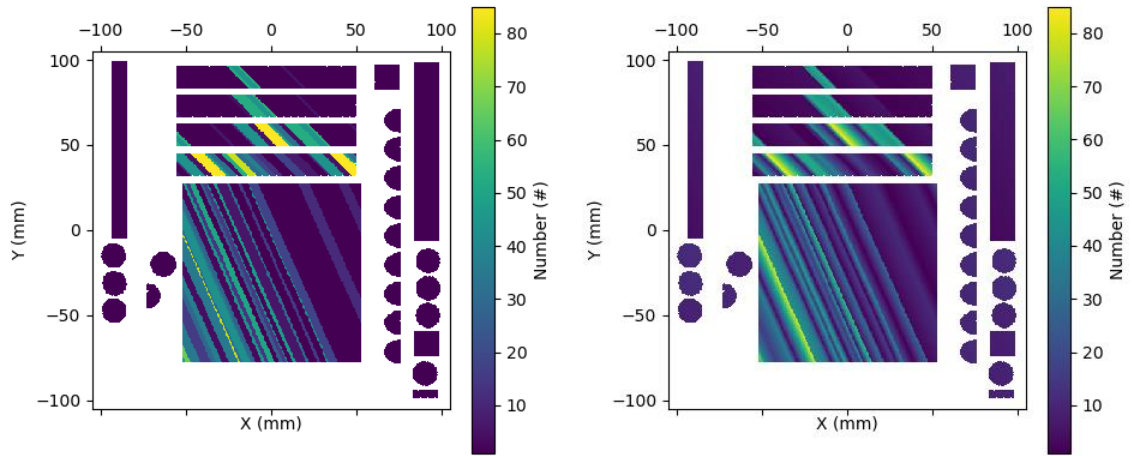


Figure 30. Spatially mapped smoke detection counts for an Arcam Q10 machine with (left) no interpolation and (right) linear interpolation at a pixel resolution of 500 μ m.

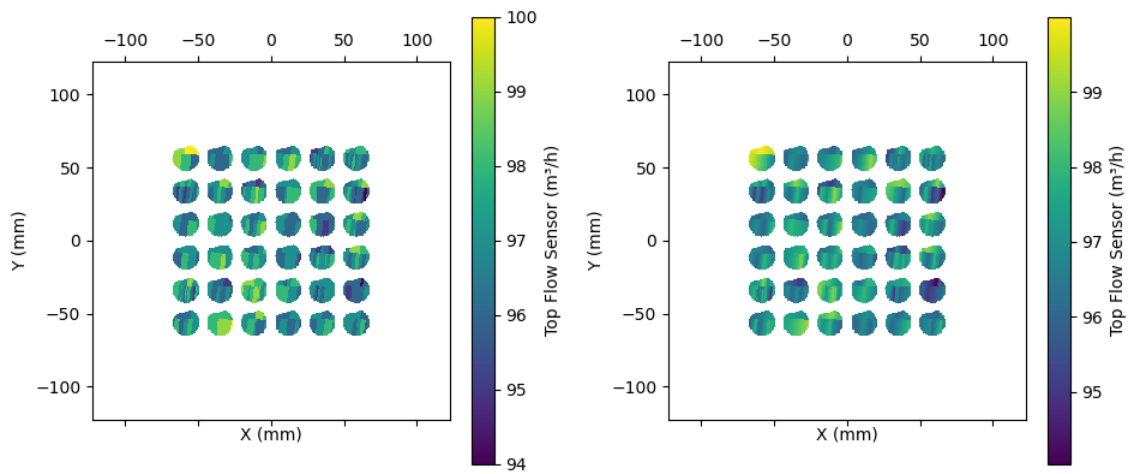


Figure 31. Spatially mapped top flow rate for a Concept Laser M2 machine with (left) no interpolation and (right) linear interpolation at a pixel resolution of 500 μ m.

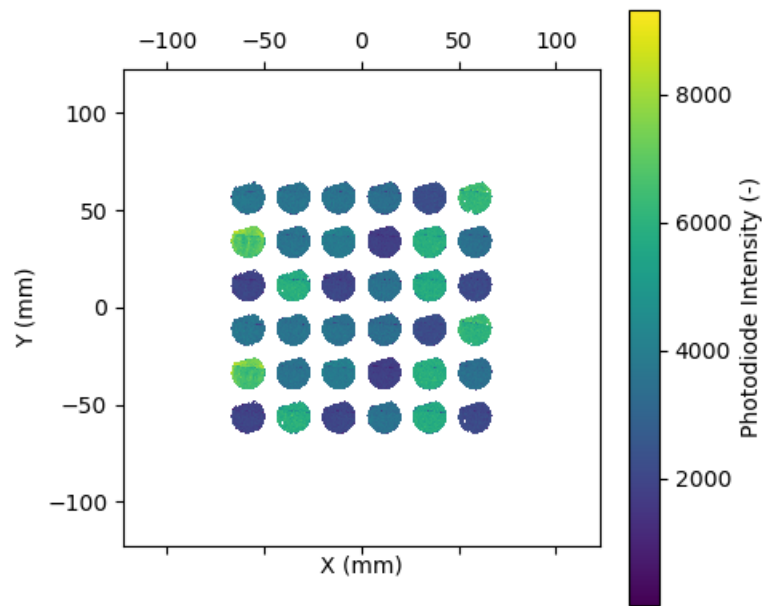


Figure 32. Spatially mapped photodiode data for a Concept Laser M2 machine at a pixel resolution of 100 μ m.

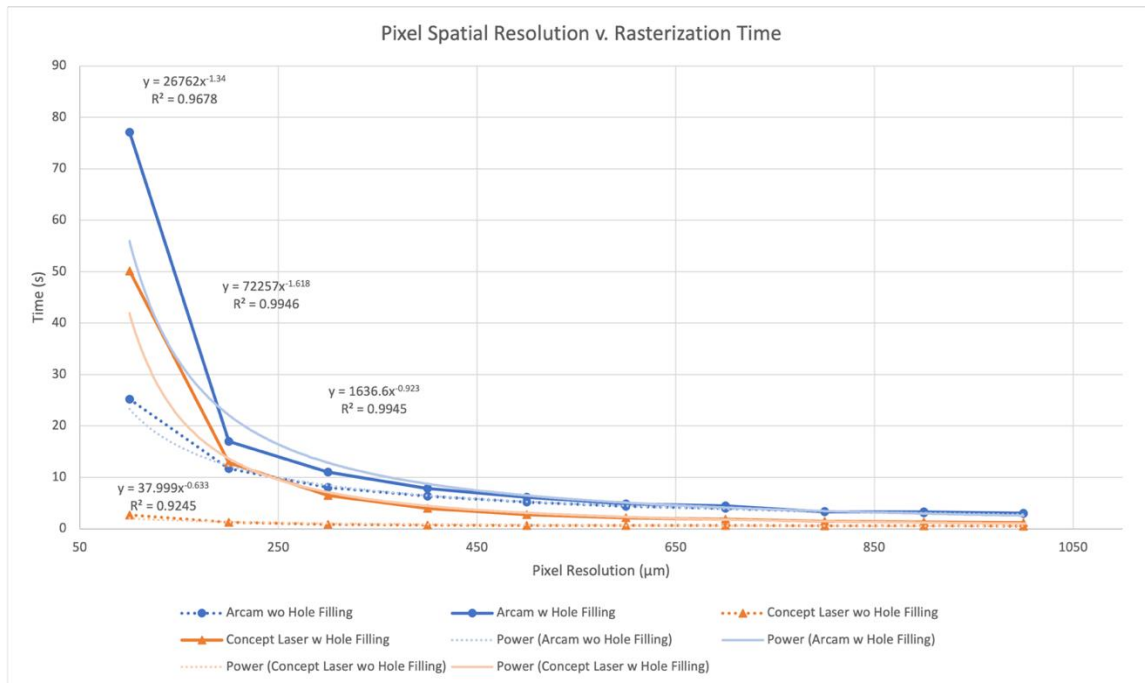


Figure 33. Change in rasterization time as a function of target pixel resolution.

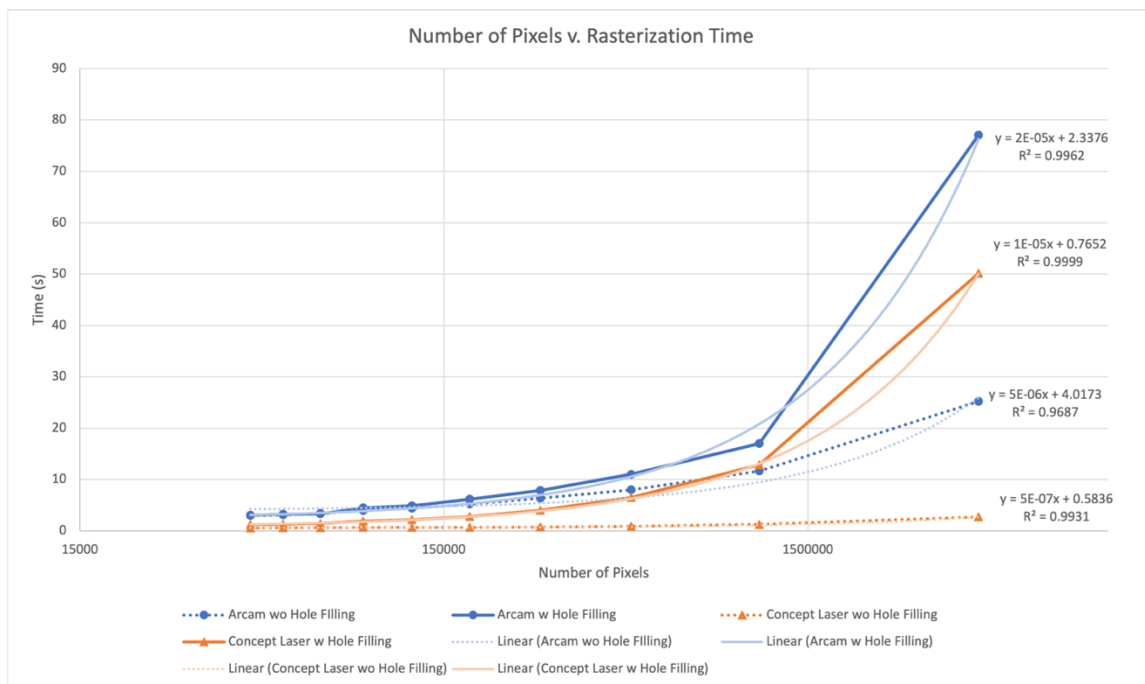


Figure 34. Change in rasterization time as a function of number of pixels in target image space.

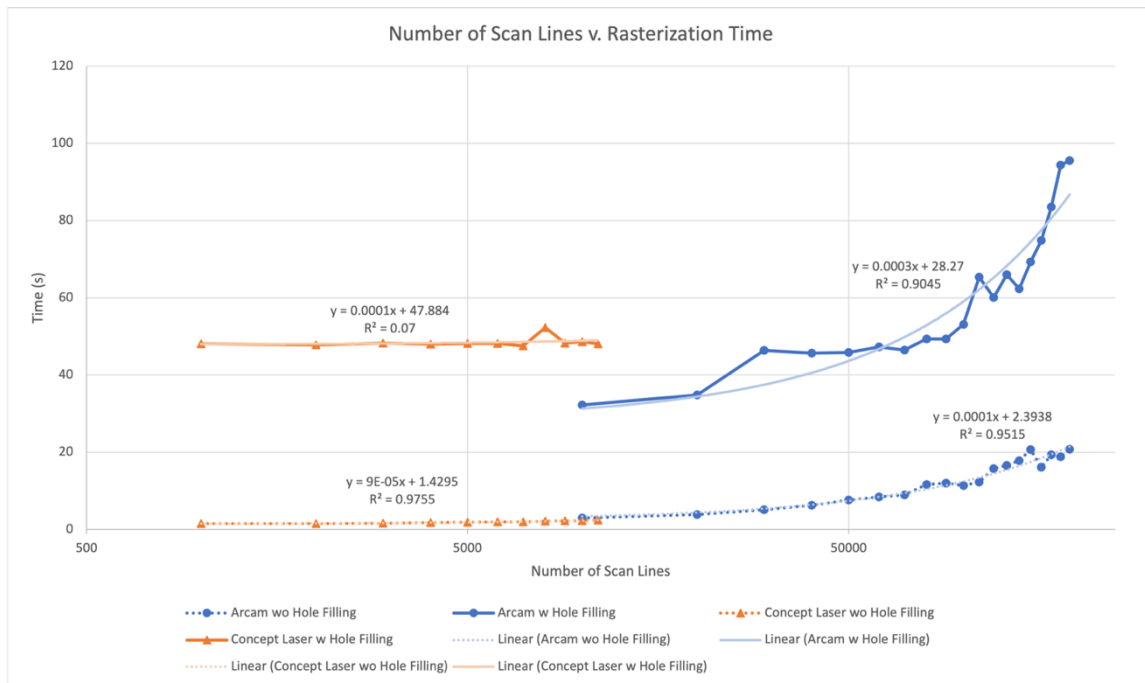


Figure 35. Change in rasterization time as a function of number of scan lines. Times are for a target pixel resolution of 100µm.

Discussion

Common Data Representations and Complex Data Processing Pipelines

By successfully performing the spatial mapping operation with data from two different machines, we have demonstrated the utility of having a common data representation. Such representations allow for building useful and complex data processing pipelines that are immediately utilizable for new technologies upon developing the corresponding data parsers. Specifically, the outlined common data representations for temporal data and scan path data that are produced by AM processes are robust and have the potential to accommodate different data sources. In the case of temporal data, this includes low and high frequency data as well as irregularly sampled data. The common scan path data representation is certainly amenable to raster scan lines. However, we also posit that it should be a suitable representation for contours for any PBF technology as well accommodate the point melt scan strategies that are possible on EB-PBF machines. For contours, the measured location is already represented by a set of sampled locations. In the section *Example of converting sampled trajectory data to the common data representation*, we have already discussed how such data may be represented in the proposed Common Data Representation. For point melt strategies, each point that is printed will have the same start and end point and a corresponding duration. As such the representation can account for spatial and temporal discontinuities.

Time Complexity Analysis

The experiments have shown, unsurprisingly, that the spatial mapping operation is sensitive to both the target resolution of the rasterized representation as well as the number of scan lines — which in turn depends on the cross-sectional area of the geometry and the scanning parameters. The time complexity of the operation appears consistent for scan paths from the two different technologies. Namely, the execution time increases as a power of the target resolution; subsequently, time increases linearly with the number of pixels. Execution time also increases linearly with the number of scan lines. Further, we note the execution penalty incurred when "hole filling" is required — discussed further in the following section. A more efficient "hole filling" algorithm would mitigate some of this penalty.

Key Considerations for Spatial Mapping Operation

The pixel spatial resolution is not only an important factor with respect to execution time but also regarding the quality and utility of the final spatially mapped data. Indeed, two situations may be present in the final mapped data depending on the pixel spatial resolution and hatch spacing: (1) some pixels that are in the printed region are never intersected by an infinitesimally thin scan line producing "holes" in the mapped data and (2) pixels may be intersected by multiple scan lines at multiple, distinct points in time which requires special handling. Both situations are germane for other spatial mapping algorithms as well.

In case (1) some sort of hole filling method is required. Our approach was to perform an image-like *hole filling* operation on the rasterized scan path data. This type of operation is commonly done on binary images and consists of a *dilation* operation that extrapolates relatively high pixel values to neighboring pixels with lower relative values. The *hole filling* operation is completed by then performing an *erosion* operation - which is the logical inverse of the *dilation* - extrapolating relatively low pixel values to neighboring pixels with higher relative values. For internal data "holes", the dilation operation serves to fill those holes, and the erosion operation pares back the boundaries without reopening any holes. This method requires a parameter regarding the neighborhood size that requires some tuning. The presence of holes in some of the results indicates that the neighborhood size was not adequate to completely fill the holes. Additionally, other interpolation methods may be used for hole filling that may be either more efficacious, faster, or both.

For case (2), several scan lines may intersect with a given pixel at different points in time as the mapped pixel spatial resolution decreases. Our approach to handle this situation is to assign the pixel with the time corresponding to the last intersection of the heat source with that pixel, because we posit that the most salient conditions that determine localized melt conditions and thermal properties occur at the last time a point is visited by the heat source. However, this need not be the case, and other methods for handling this case — such as the average of the mapped values — may also be viable.

In addition to spatial interpolation considerations, the best methods for temporal interpolation of the sensor data to be mapped should also be considered. For data that is captured at a sufficient frequency relative to the velocity of the heat source, this is not an issue; however, for lower frequency data, such as several of the machine sensor data presented herein, the interpolation scheme affects the final mapping. Here, we demonstrate two different temporal interpolation schemes for the low-frequency, irregularly sampled temporal data — a constant interpolation and a linear interpolation.

A final important consideration — and one that we have not yet implemented — for the spatial mapping operation is to factor in any temporal latency that may occur between the time a sensor registers a measurement and the time that such a change may affect the build chamber. Several important sensors in the machine and build chamber are sufficiently far away such that a significant temporal delay may exist — changes in speed of the inert gas that flows over the powder bed for L-PBF that is sensed at an inlet or outlet for example. As such, a "latency function" may be required for each sensor that maps a measured change from a sensor to when that change would be experienced or present in the build chamber or at the area that is being melted. However, we posit that some of the currently available spatially mapped data may be useable and useful even without implementing any such latency function.

Extensions and Future Work

We see several paths forward to apply this work. The most obvious is to begin to leverage the spatially mapped data as inputs to machine learning models to assist in flaw detection. This data, mapped at an appropriate resolution, may be easily fused with in-situ, layer-wise images of the powder bed for such a purpose. Ideally, however, such a system would be able to operate in real-time during a build. Currently the data parsers are written to operate on data as a post-build analysis. Thus, creating parsers for ingesting and transforming streaming, layer-wise scan path and temporal data will be necessary. However, the time complexity analyses of the spatial mapping algorithm show that it could readily support such a real-time application. Future work will also include writing the necessary scan path and temporal data parsers for other machines and data streams, such as acoustic data. Other future work will include adding other common data representations - spatial data, including images and modeling and simulation data, for example. Furthermore, with the necessary parsers and common data representations, we plan to continue to expand the data visualization, exploration, and analysis capabilities.

Conclusions

We have introduced a set of algorithms for analyzing data from additive manufacturing operations - the main goal of which is to craft reusable data processing pipelines to facilitate the creation of component digital twins and assist in component qualification. The concept of common data representations has been introduced and discussed.

Specifically, implementations of such representations for temporal and scan path data that are ubiquitous in AM have been presented.

Through a case study we have demonstrated how these concepts enable complex data processing pipelines. First, data parsers may be written to transform raw data into the conforming common data representations. Further, a spatial mapping algorithm was described and applied to two distinct powder bed fusion AM technologies. This allows for the creation of a more fully featured, voxelized, component digital twin.

Looking forward, such spatially mapped data may be integrated into anomaly detection models. For example, spatially mapped data could be included as additional input channels for CNN-based flaw detection models. The time complexity studies demonstrate the possibility of including such data in real time, layer-wise flaw detection workflows. Further, the concepts described unlock the possibility for creating other complex data processing pipelines with the opportunities for extending to additional technologies and sensor streams.

APPENDICES

APPENDIX A — Glossary of Scan Path Terms

Scan strategy	A set of policies and associated parameters that dictate how scan paths will be algorithmically generated to melt the geometrical cross section for each layer.
Scan path or trajectory	The specific path that the heat source follows during melting for a layer.
Infill	The policy and parameters of the scan strategy that dictate how the bulk regions of each layer will be melted.
Contours	The policy and parameters of the scan strategy that dictate how edges — interfaces between melted regions and powders — of each layer will be melted.
Raster	A specific infill policy where bulk regions are melted as a series of sequential, parallel lines that stretch across the cross section of a part. This policy is further subdivided into unidirectional and bidirectional. [6]
Point melt	A specific infill and/or contour policy where regions are melted as a series of discrete points.
Scan line	A single line traversed by the heat source in a scan path generated by a raster policy. A scan line is characterized by start and end coordinates as well as a duration.
Hatch spacing	A key parameter for the raster infill policy that dictates the spacing between parallel scan lines.
Sub scan [line]	One portion of a longer, logical scan line.

APPENDIX B — Example Rasterized Spatio-Temporal Maps

Table 10: Example rasterized spatio-temporal map for an Arcam Q10 trajectory.

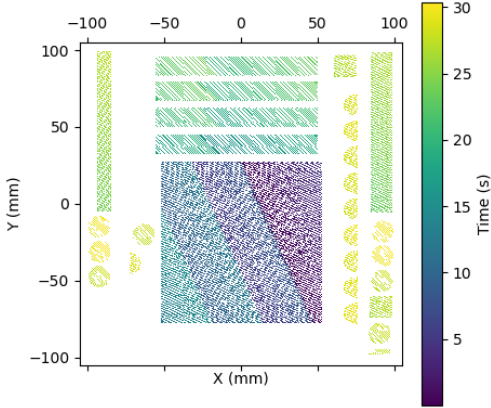
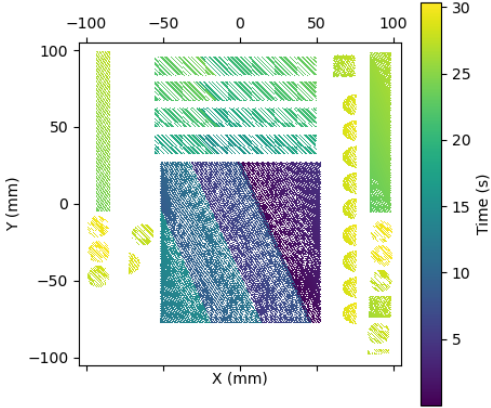
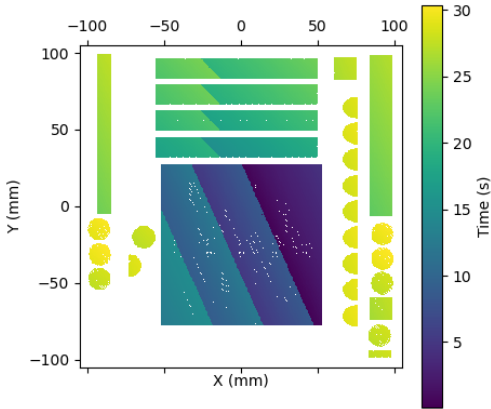
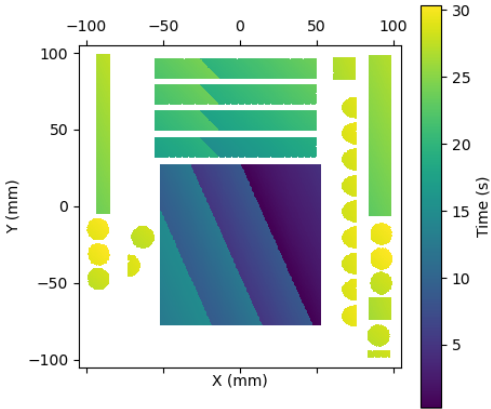
Pixel Spatial Resolution (μm)	Without Hole Filling	With Hole Filling
100		
500		

Table 10 Continued

Pixel Spatial Resolution (μm)	Without Hole Filling	With Hole Filling
1000		

Table 11: Example rasterized spatio-temporal map for a Concept Laser M2 trajectory.

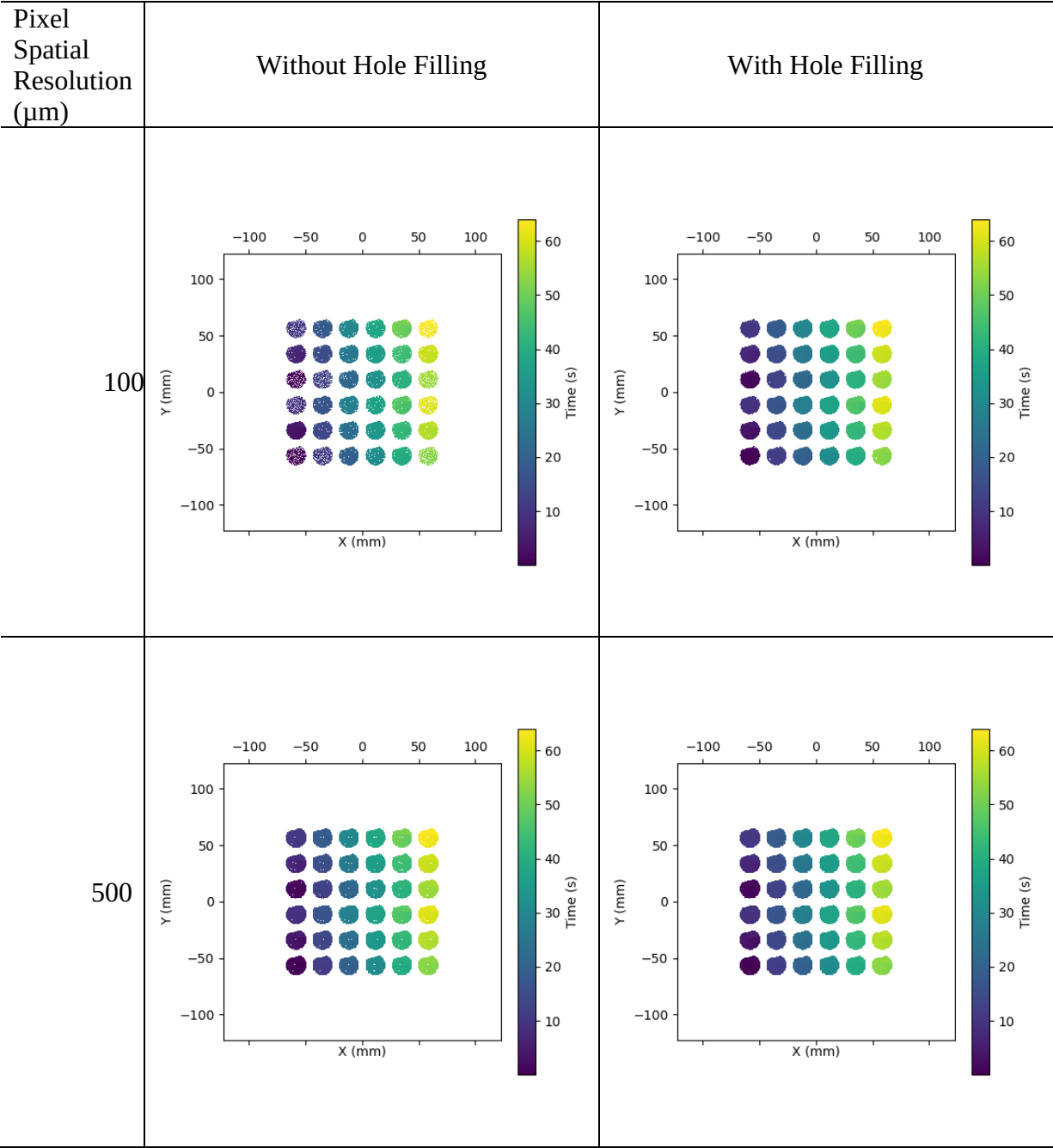
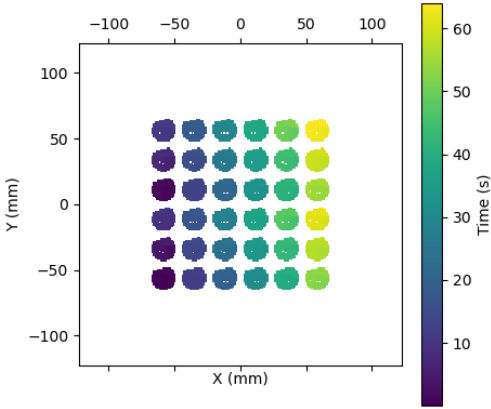
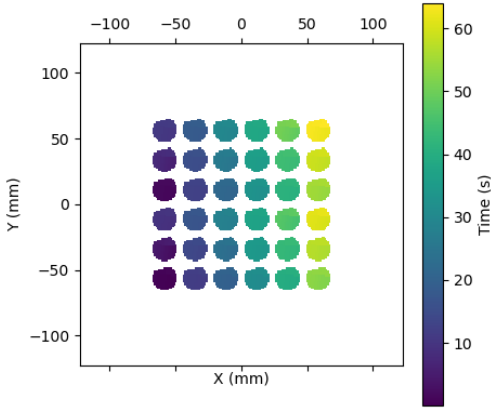


Table 11 Continued

Pixel Spatial Resolution (μm)	Without Hole Filling	With Hole Filling
1000		

CHAPTER III
LOCALIZED DEFECT DETECTION FROM SPATIALLY MAPPED, IN-SITU
PROCESS DATA WITH MACHINE LEARNING

A version of this chapter was originally published by William Halsey, Derek Rose, Luke Scime, Ryan Dehoff, and Vincent Paquit:

William Halsey, Derek Rose, Luke Scime, Ryan Dehoff, and Vincent Paquit. “Localized Defect Detection from Spatially Mapped, In-Situ Process Data With Machine Learning,” *Front. Mech. Eng.* 7 (2021): 1–14.

The content of this chapter is the same as the original manuscript. The student was responsible for ideation, implementation, data collection, analysis, and manuscript writing. Dr. Derek Rose assisted in ideation — specifically best practices for preprocessing the data after spatial mapping and before model training. Dr. Luke Scime developed Peregrine that was employed to generate the ground truth labels and served as subject matter expert for data generated by the Concept Laser M2 machine. Dr. Ryan Dehoff served as the PBF subject matter expert, and Dr. Vincent Paquit assisted in funding this work. All coauthor reviewed the manuscript.

Abstract

In powder bed fusion additive manufacturing, machines are often equipped with in-situ sensors to monitor the build environment as well as machine actuators and subsystems. The data from these sensors offer rich information about the consistency of the fabrication process within a build and across builds. This information may be used for process monitoring and defect detection; however, little has been done to leverage this data from the machines for more than just coarse-grained process monitoring. In this work we demonstrate how these inherently temporal data may be mapped spatially by leveraging scan path information. We then train a XGBoost machine learning model to predict localized defects – specifically soot –using only the mapped process data of

builds from a laser powder bed fusion process as input features. The XGBoost model offers a feature importance metric that will help to elucidate possible relationships between the process data and observed defects. Finally, we analyze the model performance spatially and rationalize areas of greater and lesser performance.

Introduction

Additive manufacturing (AM) is a set of exciting and promising processing methods that are expected to continue to revolutionize manufacturing. One exciting subset is AM for metals, which is being investigated for applications in several industries such as aerospace, automotive, biomedical, and nuclear[3], [5] The excitement behind AM is attributed to its flexibility in component design – AM may be used to fabricate geometries not possible by other conventional means.

At the same time, however, the technologies have not achieved widespread adoption and are typically relegated to non-critical applications. The inability to gain traction comes from the poor repeatability of the process from one build to the next stemming from the prevalence of seemingly stochastic defects[12] and anisotropic microstructure[15] yielding inconsistent and heterogeneous properties throughout a component and across nominal duplicates of a component. This is true for all metal AM technologies including the more mature subdomain of powder bed fusion (PBF).

The key to mitigating these shortcomings is to characterize more accurately the complex relationships between the PBF processes, the components' structural characteristics (both microstructural as well as macro-scale defects), and the resulting mechanical properties that ultimately dictate the quality of the component. These sets of intricately coupled relationships have collectively been deemed process-structure-property relationships[16].

Some researchers have proposed the use of machine learning (ML) as a way to establish relationships between these different regimes. Such relationships would aid in more robust qualification of AM builds. The ultimate goal would be to predict component characteristics and mechanical properties from in-situ process data collected during fabrication. This would allow for circumvention of costly ex-situ characterization and destructive testing. Additionally, discovering and leveraging relationships between in-situ process data and component properties would allow for the determination of part properties in real-time during fabrication and possibly even the triggering of reparative measures.

Currently, research predominantly focuses on some form of in-situ imaging data for localized defect detection and property prediction as well as closed-loop control[83][11]. However, other, less utilized in-situ data – specifically temporal sensor data – is also being investigated for process quality monitoring. Several commercial machines yield such data in the form of a build log file. Much of the data in build log files is generated by sensors that directly monitor machine subsystems and the build environment, the

status of which can directly affect build quality[6]. Analyzing these streams may help to directly relate the process to corresponding properties.

Recent work has shown the value of this temporal sensor data for build qualification — especially for electron beam PBF (EB-PBF). Grasso et al.[84] showed how a statistical model could leverage data from Arcam log files to predict swelling. Additionally, Chandrasekar et al.[85] showed how rake data from these log files, in addition to powder reuse information, informs build quality. Steed et al.[86] shows how a visual analysis tool called Falcon may be used for EB-PBF builds to analyze the time series data from log files to determine process consistency and relate it to build quality. Currently, however, it has been only possible to use these data on a layer-wise or build-wise basis; no published technique exists to leverage this data for more localized, intra-layer analyses.

Some machines produce sensor data records at a sufficient sampling rate to detect changes in these important subsystems during a single layer and even during a single melt-phase. However, leveraging this data for localized analysis is unintuitive due to its different modality — temporal instead of spatial. As such, some technique must be used to convert the temporal data to space in order to have a one-to-one relationship with the spatial data. Vandone et al.[77] have proposed leveraging tool paths for directed energy deposition (DED) to fuse spatial (melt track morphology data) and temporal (melt pool imaging video data and laser power) datasets together for modeling. However, their work centers on experimental data from single melt tracks, the temporal input feature space is

small and targeted, and the overall findings are likely not to scale to larger and more complex geometries and will be confounded by other process intricacies. Our approach, rather than being bottom up, begins to chip away at the process monitoring and defect detection problem in a top-down way. We begin with data from multiple full-scale builds and fuse together as many in-situ process signals as are available and leverage ML to mine for relationships between the process and structure.

For this study, we focus on a ConceptLaser M2 Laser Powder Bed Fusion (L-PBF) machine instead of Arcam EBM or DED sensor data. It records data from multiple sensors during a build and compiles it into a log file. Several of these data streams contain enough records to allow for investigating how perturbations in these process signals relate to localized defects that are detectable in in-situ layer-wise images. This work demonstrates an initial attempt of leveraging these temporal process data to detect localized, layer-wise defects within four entire builds using XGBoost. Additionally, we analyze the XGBoost prediction results to better understand key relationships between process data and the resulting defects.

This work specifically investigates the relationship between the process data and the occurrence of soot within a layer. Soot has been shown to cause significant degradation within the L-PBF process. While studying the effects of powder reused in L-PBF processes, Heiden et al.[87] found that soot is potentially detrimental to both mechanical properties of components and to the reuse of powders for subsequent builds. They

recommend process parameter tuning to reduce the occurrence of soot and spatter. Huang and Li[88] found that the prevalence of soot that has resettled is likely related to decreased ductility of components due to reduced effective laser power in those regions. Finally, in a recent white paper GE additive discusses the poor materials processing that occurs as a result of soot and outlines ways that they are improving the gas flow systems on the ConceptLaser M2 machines to remove soot from the build area more effectively[89].

Materials and Methods

Data Sources and Descriptions

This work was performed on process data produced from four builds fabricated at the Oak Ridge National Laboratory's (ORNL) Manufacturing Demonstration Facility. Here we outline relevant information about the builds and the data collection procedure.

Build setups and descriptions

The builds were fabricated on a ConceptLaser M2 L-PBF system that has two 400 W laser modules and a build chamber volume of 245 mm x 245 mm x 450 mm. Four builds were chosen that had approximately the same geometry to reduce confounding effects that different geometries induce in the in-situ process data – especially the trajectory laser data. Figure 36 shows the two geometries included in this analysis.

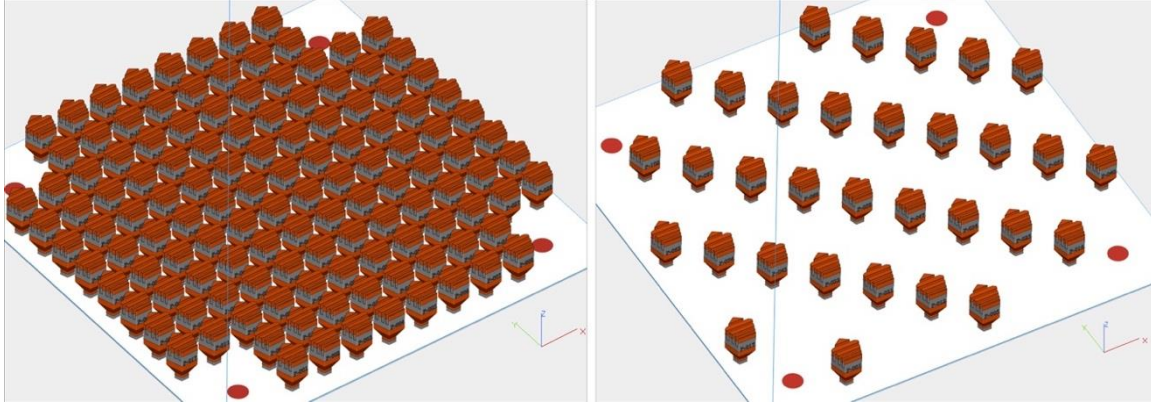


Figure 36. Left: geometry for build 1. Right: geometry for builds 2, 3, and 4. For builds 1 and 2 the gray and red regions represent where laser modules 1 and 2 are used respectively.

The builds were fabricated out of 316L stainless steel. Power was set to 370 W, speed to 1350 mm/s, spot size to 130 μm , and hatch spacing to 0.09 mm. The gas flow rate was set at 52 mm/hr, and the powder dosing factor was set to 350%. Builds 1 and 2 utilize both laser modules for different sections of the build. Build 3 uses only laser module 1; build 4 utilizes only laser module 2. Note that these builds were not fabricated specifically for this analysis. Instead, this work demonstrates the training, prediction, and analysis of an ML model on real-world data.

Data collection and description

Data were collected during the fabrication of these builds from several different distinct sources. These data sources are in-situ process data from various machine sensors that are recorded in log files, trajectory data that record relevant information about the laser scan path for each layer, and finally defect detection labels that were derived from in-situ visible-light layer-wise images.

In-situ process data

During builds, the ConceptLaser M2 machine records data from several different in-situ sensors that monitor important subsystem statuses. These sensors monitor the build environment during fabrication and perturbations in these sensor signals may correlate with the quality of the fabricated component. In L-PBF the build chamber is filled with an inert gas, typically argon, to prevent oxidation of the material during fabrication. The laminar gas flow serves to mitigate the occlusion of the melt surface caused by the laser

scattering off of the vapor plume and to remove spatter, soot, and other ejecta from the build area. A number of embedded sensors monitor this gas flow and the system that controls it.

In addition to gas flow rate, the machine monitors the oxygen concentration within the machine. The L-PBF process must be conducted within an inert environment because oxygen can react with the material during fabrication and lead to variable consistency and properties. Several sensors detect the level of residual oxygen in the system.

Finally, thermocouples are also placed in several location in the build chamber to monitor the global temperature. Figure 37 shows an example plot of temporal, in-situ process data from a gas flow sensor for one layer.

These data are collected and recorded in text-based log files that are available for analysis at the end of a build. Importantly, the data in these build logs are recorded asynchronously — values are only written out upon a significant change as dictated by the machine's operating system. For this work, a specific value from one of the sensors is assumed to persist until a new value is recorded. A set of the sensor streams do not generate enough records in the log file to be included in this analysis. Table 12 shows the subset of sensors that were used in this analysis. Also, note that the log files contain timestamps of when the melt phase begins for each layer.

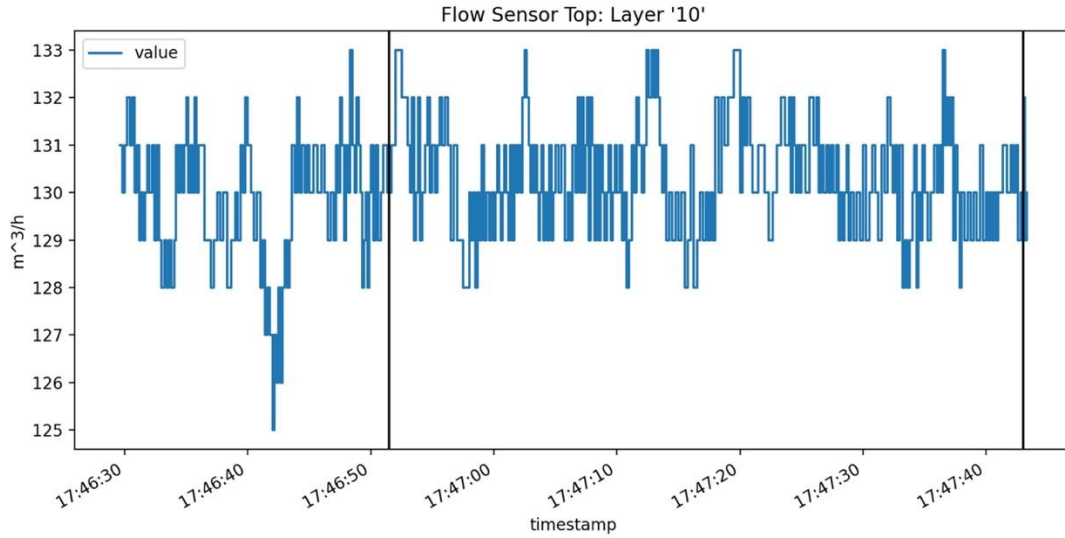


Figure 37. A plot of a single layer of log file data for 'FlowSensorTop' from build 1. The spikes show where the machine logs several records quickly in succession, while the timing between other records is greater. The black vertical lines indicate where the melt phase begins and ends for this layer.

Table 12: Description of in-situ process data used to detect defects for this analysis.

<i>Variable Name</i>	Description
<i>FlowSensorTop</i>	Gas flow at the top of the build chamber in m ³ /h
<i>FlowSensorBottom</i>	Gas flow at the bottom of the build chamber near the powder bed in m ³ /h
<i>Ventilator control Actual</i>	The speed of the motor of the ventilation system as a percentage of the maximum command possible
<i>Ventilator rotation speed</i>	Another measure of the motor of the ventilation system as a percentage of the maximum command possible
<i>Residual oxygen value in module</i>	The oxygen concentration in the build chamber as a percentage
<i>Residual oxygen value in gas loop</i>	The oxygen concentration in closed inert gas circuit as a percentage

Trajectory data

In addition to the previously described temporal process data, spatial information from the laser modules is captured during the melt phase. This data is captured during a build by a National Instruments PXIe- data acquisition device at an approximate sampling rate of 40 kHz and stored in a TDMS file. The data that are included for each sample are scaled x and y location on the build plate, an indicator of which laser module is melting, and intensities from an on-axis photodiode. From the sequence of coordinates and sampling rate the absolute order and timing of the melt is known.

For this work, these data are transformed from the format described and interpreted as line segments. The start and end coordinates of these line segments are the coordinates of the current sample and the coordinates of the subsequent sample. Thus, we can derive a distance for each scan line, and since the sampling rate is constant, velocity can also be derived. Finally, the relative start time for each scan line since the beginning of the melt can also be found. This can be extended further to find the time elapsed since the beginning of the melt for any arbitrary location in the trajectory.

Also, for this work the photodiode data was *only* used to infer part boundaries and not as an input feature. Figure 38 shows a close up of a plot of trajectory where the color is from the photodiode data and shows the component boundaries. Determining the component boundaries allows for the analysis of only voxels that are melted and where process perturbations may affect the local material properties. This will be elaborated upon later

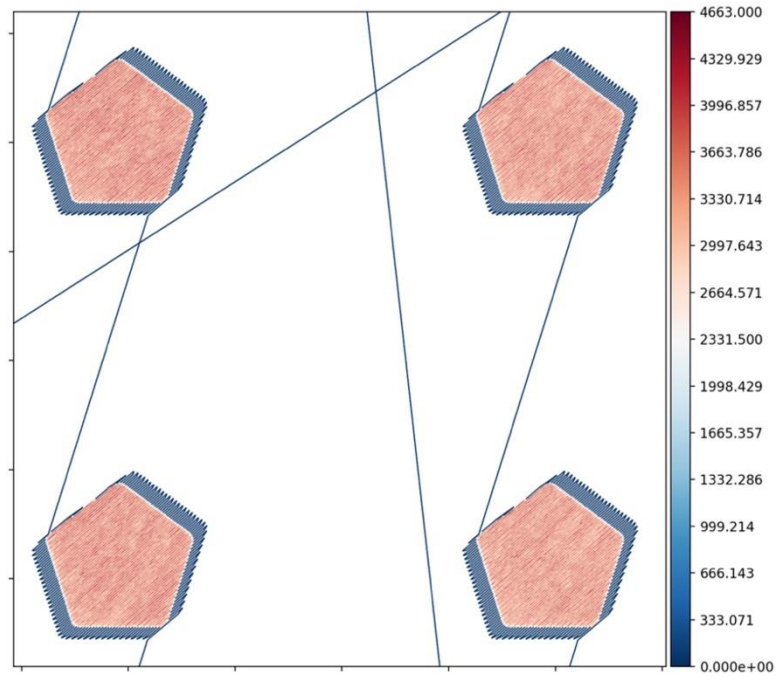


Figure 38. Example of trajectory data from build 1 zoomed in on four components. The image shows the melt areas, skywriting, and jump lines, and the visualized data is are the raw digital intensity levels from the on-axis photodiode that was used to determine the component boundaries for spatial mapping.

in this paper. With the part boundaries identified, consecutive sub-scans could be consolidated into a single raster line. Table 13 shows the variable names and descriptions of the trajectory data.

Defect data

Finally, the defect data are generated from in-situ images of the powder bed that are collected by an off-axis, five-megapixel, visible light camera focused on the powder bed. The camera captures the full 245 mm x 245 mm build plate, and the build area is illuminated from above by an LED array[90]. Two images are taken for each layer – one after powder spreading and another after the melt is complete.

The defect predictions are generated by Peregrine — a tool developed at ORNL[90]. This tool uses a deep convolutional neural network (CNN) — what the authors specifically refer to as a dynamic segmentation convolutional neural network (DSCNN) — to perform semantic segmentation on the in-situ, layer-wise images. Semantic segmentation is a computer vision task for assigning one or multiple class labels to each pixel that is a constituent of an object within an image. As stated previously, we focus specifically on the ‘*Soot*’ defect because this defect is generated during the melt and is influenced by build chamber gas flow and possibly by other process factors such as the laser trajectory and possibly amount of oxygen in the chamber — all features that are captured in the process log or trajectory data.

Table 13: Description of in-situ process data used to detect defects for this analysis.

<i>Variable Name</i>	<i>Description</i>
<i>Phase</i>	An indicator variable denoting the laser module utilized to melt
<i>Distance</i>	Length of the consolidated scan line
<i>Duration</i>	Elapsed time for the laser to traverse the scan line
<i>Speed</i>	Velocity of the laser
<i>Time</i>	Elapsed time since the start of the melt for any arbitrary location along the trajectory

Peregrine outputs a categorical label for each voxel in a layer. Since only one label is investigated for this work the defect data are transformed to a Boolean value indicating the presence of soot or not. Figure 39 shows the Boolean defect detections for a single layer as a binary image. Note that the authors state that the off-axis setup of the imaging system leads to differing effective resolutions across the build chamber [90].

XGBoost Background

XGBoost is a fast and effective ML model implementation [91]. It has been used effectively for many different, real-world applications – often achieving top performance for structured data tasks [91][92].

Model description

XGBoost is an implementation of Gradient Boosted Machine (GBM). GBM, in turn, is a robust ML model that aims to train many *weak learners* that, when leveraged in conjunction, act as one strong model[93]. A weak learner is a model whose predictions are only slightly better than random guessing; they are simple and typically have few learnable parameters. As such, GBM is an example of ensemble learning, and specifically as the name denotes, it implements a paradigm of ensemble learning called *boosting*. Boosted models are still trained using gradient descent; however, it differs from typical gradient descent training used in other, monolithic models such as logistic regression or neural networks. A brief discussion of similarities and differences between training GBM's and other ML models that the reader may be familiar with is now presented. The

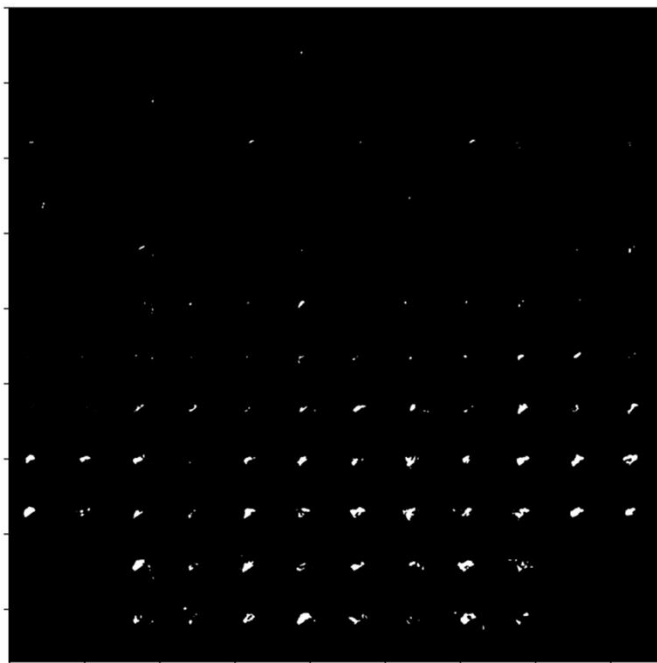


Figure 39. Binary class labels for 'Soot' for each voxel of the build chamber in layer 10 of build 1 as predicted by Peregrine.

equations are taken from the original paper on GBM's by Friedman[93] with minor simplifications for brevity and clarity.

The goal of supervised ML is to find an optimal function, $F^*(x)$, minimizes a loss function, Ψ , with respect to provided labels or response values, y , as shown in Equation 23.

Equation 23.

$$F^*(x) = \arg \min_{F(x)} E_{y,x} \Psi(y, F(x))$$

Many different types of ML models have been developed to this end, and many of these models, such as logistic regression and neural networks, are parameterized. This means that finding an optimal function for a given model occurs when optimal model parameters, P^* , are found as shown in Equation 24.

Equation 24.

$$P^* = \arg \min_P E_{y,x} \Psi(y, F(x, P)); F^*(x) = F(x, P^*)$$

The optimization of the parameters occurs by updating the parameters over the course of M iterations from an initial guess, p_0 , as shown in Equation 25.

Equation 25.

$$P^* = \sum_{m=0}^M p_m$$

Each update is a small increment — scaled by a learning rate, α — along the negative gradient of the loss function with respect to the current parameter values, as shown in Equation 26 and Equation 27.

Equation 26.

$$p_m = -\alpha g_m$$

Equation 27.

$$g_m = \left[\frac{\partial \Psi(y, F(x, P))}{\partial P} \right]_{P=P_{m-1}}$$

In fact, GBM's leverage a similar algorithm for model training. As an ensemble method, the optimal function, $F^*(x)$, is a summation of constituent weak learner models, and like conventional models, the optimization of this ensemble can be performed by iterative updates as shown in Equation 28.

Equation 28.

$$F^*(x) = \sum_{m=0}^M f_m(x)$$

However, instead of updating one set of model parameters, the ensemble incrementally adds additional learners along the negative gradient of the ensemble loss with respect to all the current learners in the ensemble, as show in Equation 29 and Equation 30.

Equation 29.

$$f_m(x) = -\rho g_m(x)$$

Equation 30.

$$g_m(x) = E_y \left[\frac{\partial \Psi(y, F(x))}{\partial F(x)} | x \right]_{F(x)=F_{m-1}(x)}$$

So, gradient boosting performs gradient descent by iteratively training additive models that each attempt to lower the global loss of the ensemble. For a complete technical description of the model and underlying algorithm, the reader is referred to original paper on GBM's by Friedman[93] as well as a technical overview provided in the XGBoost documentation [94].

Typically, XGBoost employs decision trees as the base, weak learners; however, other base models may be used instead. Also, it may be used for either classification or regression. For this task we used the classification version with the decision tree base learner. The model has several user-defined parameters that affect training and performance of the algorithm. A subset of the parameters that are considered the most

important are shown in Table 14. The values for these parameters were empirically optimized. This process is covered in Section *Model parameter optimization and tuning*.

Model interpretability

Additionally, XGBoost offers several metrics for evaluating the importance of input features on the performance of the final model. This enables a user to gain a better understanding of the key relationships between the input features and the output. This is an important advantage over other, deep neural network ML models. Specifically, XGBoost offers three distinct metrics for feature importance: gain, weight, and cover[95]. The gain metric calculates the average improvement to the model's loss contributed by each feature. Cover and frequency refer to how many samples are associated with a feature and how often a feature was utilized for splits within the ensemble respectively. Based on these definitions, gain is a more relevant evaluation metric when compared to the other two. Therefore, we focus on it to analyze the input features. This will allow for the investigation and initial determination of which process data may be most closely associated with the chosen defect class — soot.

Methodology

Before the analysis is possible on these data, several transformations and preprocessing steps must be performed. These steps are outlined here, but for the sake of brevity and scope some of the more technical details are omitted.

Table 14: Description of a subset of XGBoost model parameters.

<i>Parameter Name</i>	<i>Value</i>
<i>n_learners</i>	The maximum number of decision trees for the ensemble
<i>scale_pos_weight</i>	A weighting factor to balance the positive and negative class weights
<i>colsample_bytree</i>	Fraction of features to include when building each tree
<i>subsample</i>	Fraction of training samples to include when building each tree
<i>max_depth</i>	Maximum depth of each tree in the ensemble
<i>min_child_weight</i>	Number of samples that must exist in a leaf node
<i>learning_rate</i>	Parameter that controls shrinkage of feature weights between boosting rounds

Temporal and trajectory data transformation

The temporal data from the in-situ sensors as well as some of the trajectory data, and spatial data from the in-situ images and defect detections, are not immediately useful in conjunction. Indeed, typical data analysis is restrained to using each of these modalities only in isolation. As such, the need to transform one of these modalities to the other is necessary for them to be leveraged together. To this end and for this work, we propose a “spatial mapping” transformation that takes the inherently temporal in-situ process data and turns it into a spatial dataset.

Several items are needed to complete this transformation: an absolute timestamp for the start of the melt phase for each layer and the location and timing information of the heat source. For the ConceptLaser M2 these data come from the build log and the trajectory data respectively. The ultimate goal is to transform the temporal data into an image-like array of the same shape as the in-situ image data and Peregrine defect predictions.

The basic process is to use the location information from the trajectory data to determine which voxels each scan line travels through for a given layer. The timing information of the trajectory data gives the relative time from the start of the melt that the pixels are intersected by the laser. The melt start time given in the build log gives the absolute time for the start of the melt, and together yield the absolute time that the laser passed through each pixel. With this information, one can trivially index into the in-situ process data and pull any sensor value from a given time that corresponds to a physical location where the

laser was melting material. Figure 40 illustrates this general concept. This spatial mapping and voxelization of the data can be performed for any arbitrary voxel resolution; however, intuitively, the data are mapped to match the resolution — approximately 100 μm — of the Peregrine defect prediction data.

Once this data transformation is complete, the in-situ process data as well as the trajectory data are in the same domain as the defect data. Each voxel of each component contains localized in-situ process data, trajectory data from the laser scan path, as well as local defect detections.

Data preprocessing

After the data are transformed to the same voxelized, spatial domain, as shown in Figure 41, several more preprocessing steps are performed before the analysis. These steps and rationale are outlined in this section.

Defect data preprocessing

First, the defect data undergo a dilation operation. Through qualitative analysis it was determined that many of the ‘Soot’ defect detections occurred over the powder bed and proximal to the printed components with vanishingly few detections over the components themselves. This operation expands the defect predictions from the original set of voxels to adjacent voxels, and it serves to reasonably increase the number of ‘Soot’ voxels that intersect with voxels where mapped process data exist. The likely reality is that the soot

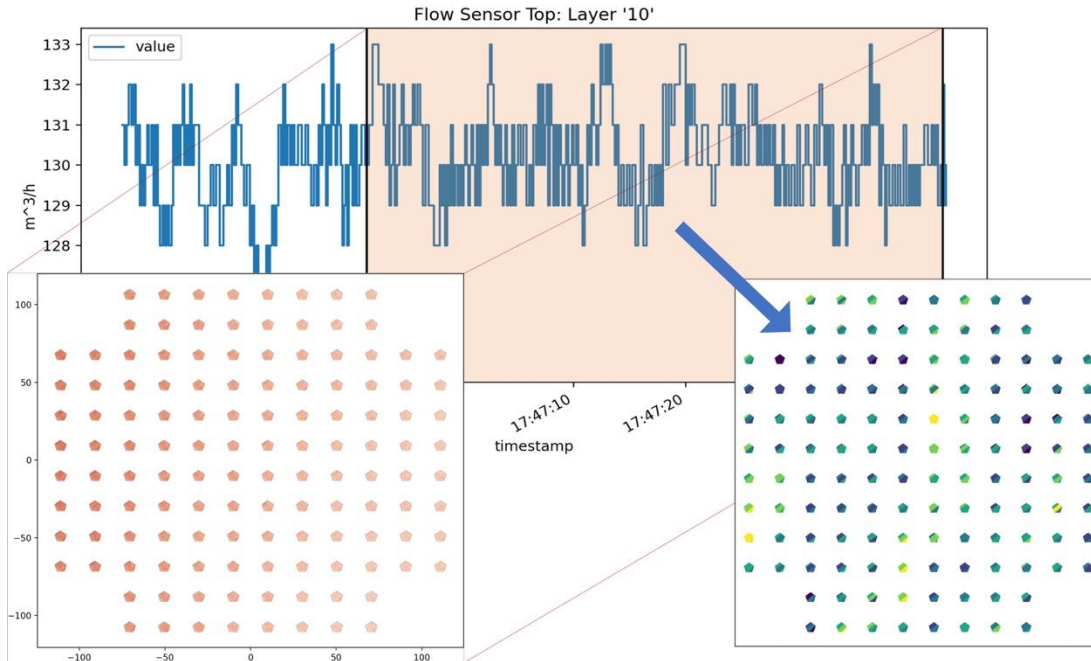


Figure 40. A sample layer from build 1 includes trajectory data, log data for 'FlowSensorTop', and the subsequent mapped data. The orange highlighted portion of the log data corresponds to the melt phase for the layer as indicated by the 'Exposure started for' and 'Exposure ended for' log file entries.

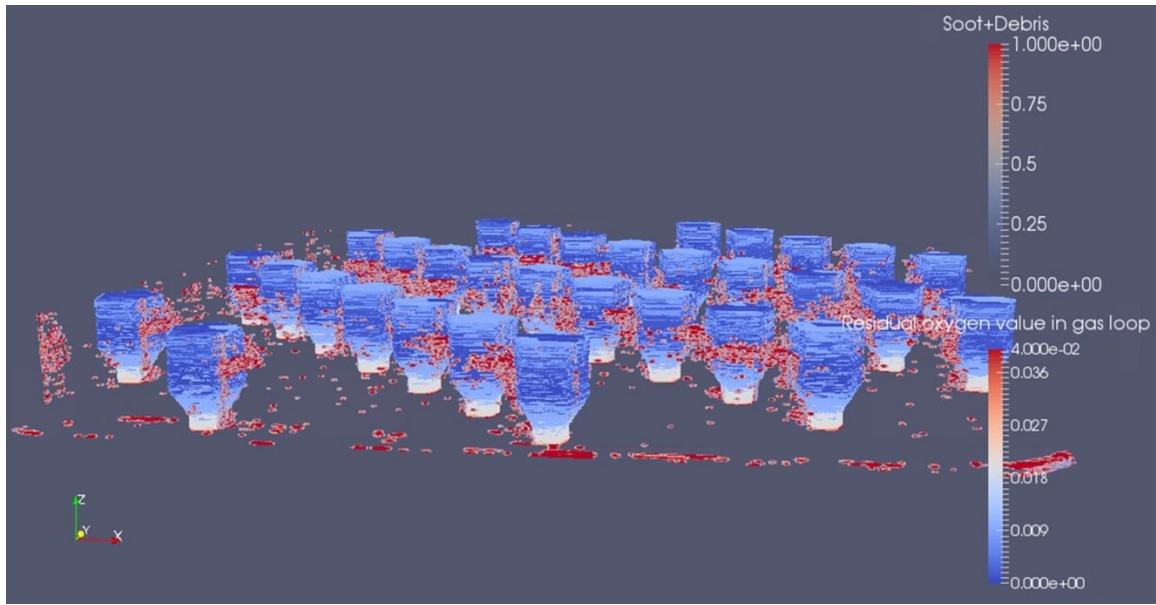


Figure 41. 3D plot of the build volume of spatially mapped process data for "Residual oxygen value in gas loop" with defects overlaid for build 2. Note that this image shows combined soot and debris defects [16], but for the actual analysis only soot was considered.

particles are also present on the printed surfaces in quantities proportional to their presence near the part; however, the imaging conditions make their on-part in-situ detection challenging.

Full data preprocessing

The next operation is a Gaussian blur of defects and in-situ process data — both the log data and the trajectory data. Within the AM process, many complex interactions occur within local regions, so this Gaussian blurring of the data serves to blend the data together and eliminate hard boundaries between defects and nominal voxels as well as voxels with one sensor value proximal to another with a different value from the same sensor.

Afterwards, the data were resized spatially to reduce the data burden. This decreased the voxel spatial resolution of the data from $100\text{ }\mu\text{m} \times 100\text{ }\mu\text{m}$ to $300\text{ }\mu\text{m} \times 300\text{ }\mu\text{m}$. This was determined to be a reasonable processing step since soot may be detected from in-situ layer images at lower resolutions.

The blurring and resizing of the data create a target set, ‘Soot’, that is no longer simply 0’s and 1’s, so we threshold the defect data of the lower resolution dataset at 0.5 to recreate a Boolean target for classification.

Now the data are in a form where many voxels in the build chamber contain data from all three data sources (any voxel outside the geometry where no in-situ process data exist is removed). This dataset is now appropriate for training a supervised ML model. Each voxel represents a sample in the dataset with process and trajectory data acting as the input features and the defect labels serving as the training targets.

Several final preprocessing steps are taken. Namely, the removal of some bad data points. Specifically, these were voxels where the trajectory ‘Speed’ value had been corrupted at some point along the way and had a value of infinity. This represented a very small number of overall voxels.

At this point a subset of the data was generated by randomly choosing one million voxels with spatially mapped data from each of the four builds. For the first build this represented approximately 1.5 % of the voxels, and for the remaining three builds it represented approximately 6 %. This subset of the data was used for parameter optimization, as outlined later in the paper, and for final model training.

Finally, the input features were duplicated, standardized in two different ways, and concatenated to create the final input feature set. The first copy of input features was each standardized globally across all four builds. For each feature a global mean and standard deviation were calculated. Then for each sample, each respective feature was zero centered and scaled to have a unit standard deviation. The second copy of input features

were standardized per build — meaning that the mean and standard deviation were calculated for each feature and build separately and then each sample was standardized based upon which build it belonged to. The rationale for these two, simultaneous standardization methods is that the global standardization will allow a ML algorithm to learn important trends and relationships in the absolute ranges for each sensor. The build-wise standardization, on the other hand, will help account for and mitigate any possible sensor drift or changes in calibration between builds and still allow for the ML algorithm to learn relationships between changes in relative sensor values and defects.

Model parameter optimization and tuning

ML models have distinct parameters that affect the training characteristics and performance of the model. For this work we approximately followed the parameter optimization procedure as outlined in [96], and we only optimized the parameters discussed in Section *Model description*. Parameters of interest were set to default values as outlined in the XGBoost documentation until optimized with one exception. The *'scale_pos_weight'* value was initialized using a heuristic recommended by the creators of XGBoost — a ratio of the number of negative samples to the number of positive samples [97].

Additionally, receiver operating characteristic (ROC) area under the curve (AUC) is used as the evaluation metric during parameter optimization. It was chosen because it is well suited for binary classification tasks when there is a significant class imbalance. Using

this evaluation metric mandates that the model use logistic regression as its objective to perform the classification [97].

To determine the model performance more accurately for a given parameter set, the dataset is typically divided into multiple *folds*, or training and validation set pairs, that are each used to evaluate the model performance and across which an average performance is calculated in a process called *cross validation*. For this work, the voxels were divided into four folds each consisting of approximately four million voxels. The validation set of each fold contained voxels from only one of the four builds — one million voxels; the training set contained the remaining voxels from the other three builds — three million voxels. This was done to emulate a likely scenario where an ML model is trained on data from existing builds and then used to predict properties of new builds. Later, for model training this same principle was applied — models were trained on data from three builds to predict defects for the fourth build.

The first optimization was to determine an initial number of learners, $n_learners$. An XGBoost model was trained with an initial goal of 1000 learners; however, early stopping was enabled that would end training after 50 consecutive trees failed to reduce the loss and the number of learners with the best average loss across the four folds was used.

Next, with the new number of learners established, *'scale_pos_weight'* was optimized from its original, heuristic-based number. This optimization was done by using random search as implemented in the python library scikit-learn [98]. Twenty randomly selected values between the square root of *'scale_pos_weight'* and the original, heuristic-base *'scale_pos_weight'* were tested while holding the remaining model parameters constant. A second random search of a narrower range was performed if it was deemed necessary.

Then, *'max_depth'* and *'min_child_weight'* were jointly optimized followed by another round of updates to *'n_learners'*. Afterwards, *'colsample_bytree'* and *'subsample'* were jointly optimized. Finally, *'learning_rate'* was manually set to 0.01 and the final value for *'n_learners'* was determined. The optimized model parameter values are shown in Table 15. An asterisk indicates that the parameter was chosen manually and did not undergo any optimization.

Input feature subsets

Several different subsets of the features were used to train models (each using the same model parameters previously outlined). This ablation study was performed to establish the effect on model performance that different in-situ data had for identifying defects. These subsets were as follows: the full feature set – both in-situ sensor data and trajectory data, the full feature set excluding the ordering information from the trajectory data, and finally a subset that consisted only of in-situ sensor data and laser module data. Table 16 shows these three different input feature sets. Recall that each feature listed, except

Table 15: Description of a subset of XGBoost model parameters.

<i>Parameter Name</i>	<i>Value</i>
<i>n_learners</i>	1249
<i>scale_pos_weight</i>	24.7
<i>max_depth</i>	4
<i>min_child_weight</i>	7
<i>colsample_bytree</i>	0.612
<i>subsample</i>	0.617
<i>learning_rate*</i>	0.01
<i>seed*</i>	27

Table 16: Description of a subset of XGBoost model parameters.

Feature Name	Full	WO Time	Log
FlowSensorTop	X	X	X
FlowSensorBottom	X	X	X
Ventilator control Actual	X	X	X
Ventilator rotation speed	X	X	X
Residual oxygen value in module	X	X	X
Residual oxygen value in gas loop	X	X	X
Phase	X	X	X
Distance	X	X	
Duration	X	X	
Speed	X	X	
Time	X		

‘Phase’, actually represents two input features — one that is globally standardized and another that is standardized per build.

These subsets were chosen with the following respective rationales: to train a model on a comprehensive feature set that did not include any explicit location information, to train a model on a feature set that did not include *implicit* location information (in the form of scan path order as given in the ‘*Time*’ feature), and to train a model strictly on in-situ process data and establish initial relationships between these signals and detected defects. Note that ‘*Phase*’ – information about which laser module was melting – was included in all three feature sets because the log files also indicate when each laser is being utilized. In summary, a total of twelve models were trained – four models trained on data from three builds to predict defects in the fourth and replicated for all three subsets of the feature set.

Results

Models were trained with the optimized XGBoost model parameters. One model was trained to make voxel-level soot predictions for each of the four builds with sampled data from the other three builds — approximately three million voxels worth of data. This was repeated for each of the three distinct input feature sets. This process produced several distinct results that will each be presented. The overall performance of each model, the feature importance rankings for each model, and the location-based performance for each model will be presented.

Model Performance

Confusion matrices are a canonical way of presenting classification results; however, the imbalanced nature of this problem makes their use impractical because the overwhelming majority of predictions are true negatives. Instead, we will discuss several metrics in conjunction that will allow for analyzing the model performances. These are *sensitivity* and *specificity*, *balanced accuracy*, as well as *false omission rate* and *false discovery rate*.

Sensitivity is also known as the true positive rate and is the fraction of ground-truth positives that were correctly predicted as a percentage of all ground-truth positive examples. Similarly, specificity is also known as the true negative rate and is the fraction of ground-truth negatives that were correctly predicted as a percentage of all ground-truth negative samples. Balanced accuracy is the average of sensitivity and specificity and is a good metric for analyzing performance in light of class imbalance. False omission rate is the fraction of data that were ground-truth positive that were incorrectly predicted as a percentage of all model-predicted negatives; false discovery rate is the fraction of data that were ground-truth negative that were incorrectly predicted as a percentage of all model-predicted positives. Table 17 shows the performance for all twelve models as well as aggregated metrics across the four build-wise models for each input feature set.

Table 17: Model performances. The top lines for each input feature set represents the aggregated performance across all four models and respective builds.

Model	Sensitivity	Specificity	Bal. Acc.	FOR	FDR
<i>Full</i>	<u>39.69%</u>	<u>94.11%</u>	<u>66.90%</u>	<u>0.63%</u>	<u>93.78%</u>
Build 1	41.68%	92.94%	67.31%	0.37%	96.59%
Build 2	34.66%	96.55%	65.60%	1.00%	86.93%
Build 3	45.36%	93.68%	69.52%	1.01%	88.85%
Build 4	34.14%	96.79%	65.47%	0.86%	88.06%
<i>WO Time</i>	<u>30.52%</u>	<u>94.37%</u>	<u>62.45%</u>	<u>0.72%</u>	<u>94.93%</u>
Build 1	41.86%	92.38%	67.12%	0.38%	96.82%
Build 2	17.28%	98.03%	57.65%	1.25%	88.37%
Build 3	38.14%	94.63%	66.38%	1.13%	88.96%
Build 4	14.33%	98.48%	56.41%	1.10%	89.27%
<i>Log</i>	<u>22.34%</u>	<u>94.28%</u>	<u>58.31%</u>	<u>0.80%</u>	<u>96.29%</u>
Build 1	35.50%	92.21%	63.86%	0.42%	97.34%
Build 2	4.01%	98.89%	51.45%	1.43%	94.88%
Build 3	26.60%	94.57%	60.59%	1.34%	92.11%
Build 4	13.11%	97.75%	55.43%	1.12%	93.07%

Feature Importance

As previously discussed, XGBoost offers metrics for ranking the importance of each feature for producing predictions. Figure 42 shows the top ten features for each of the twelve models divided up by input feature sets.

Location-Based Model Performance

The performance results can also be calculated per component and visualized. This may lead to some interesting observations about the model performance. Figure 43 show the component-wise balanced accuracy for each build and input feature set. Note that the coloring scale is not consistent across the images. This was done to ensure that differences among the components and trends across the build chamber could be observed easily. Also, changes in balanced accuracy are dominantly influenced by changes in the sensitivity metric across the build chamber. For context, the spatial distribution of the soot prevalence per build is shown in Figure 44.

Discussion

Analysis

The results from this analysis show fair performance but with ample room for improvement with respect to the various metrics. The model achieved an aggregated best sensitivity of approximately 40 % and balanced accuracy of approximately 67 %. However, the model still performed significantly better than a baseline of random

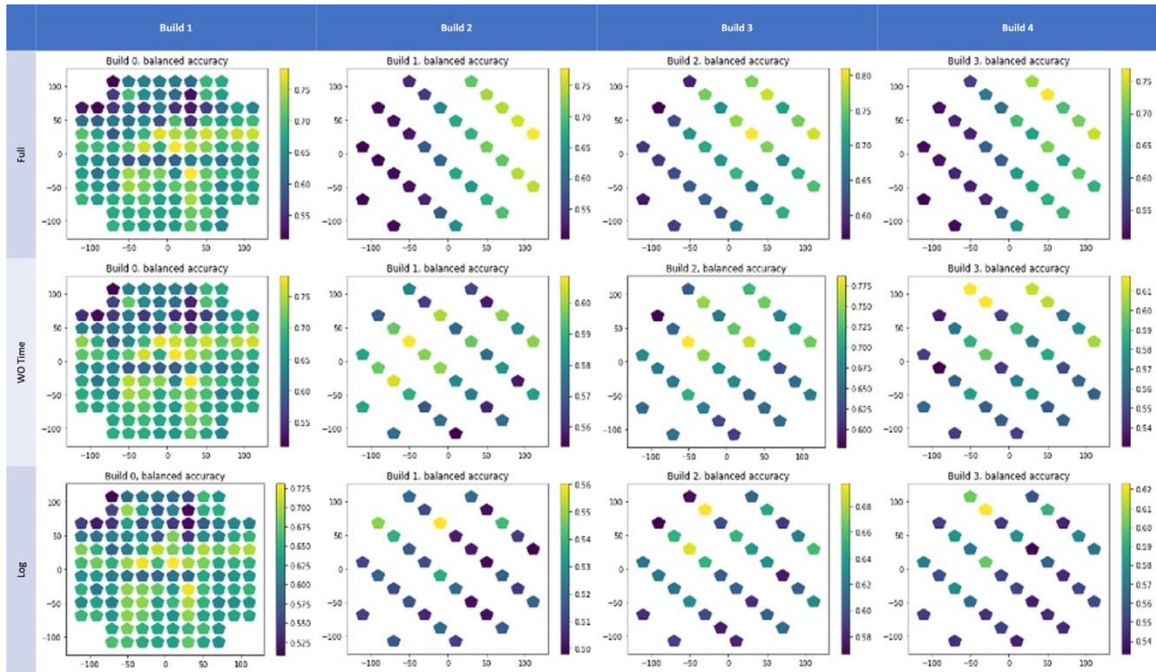


Figure 43. Visualization of balanced accuracy per component of each build for the 'Log' input feature set.

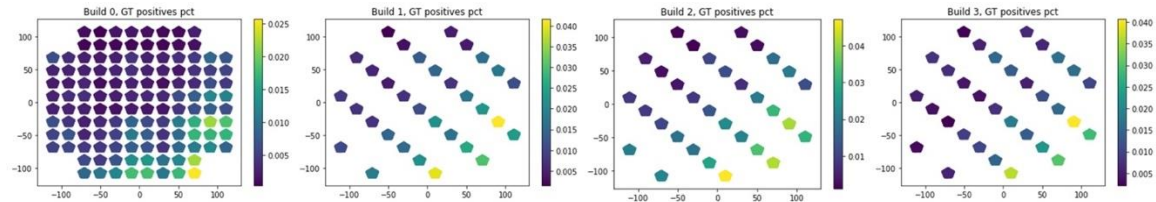


Figure 44. Spatial distribution of soot prevalence for the four builds.

guessing. In general, models for builds 1 and 3 had better performance. Additionally, the performance consistently decreased with more restricted input feature sets.

Across all twelve XGBoost models, two input features were consistently measured to have the greatest importance for model predictions across all builds and input feature sets. These were '*Residual oxygen value in module_full_norm*' and '*Residual oxygen value in gas loop_full_norm*' constantly occupying the first and second positions respectively. This makes strong intuitive sense as increasing levels of oxygen lead to greater levels of oxidation on the exposed surface that manifest as darker regions that are easy for Peregrine to detect as soot. Additionally, these two features were the globally standardized versions (indicated by the '*_full_norm*' suffix), so it is the absolute ranges for residual oxygen that is most informative (as opposed to the relative changes within a relative range in the sensor readings within a build).

Finally, several interesting spatial trends are observed when visualizing model performance in different regions of the build area – especially as observed in builds 2-4. Figure 44 shows how the prevalence of soot increases toward the front of the build chamber. The prevalence is also skewed toward the right side of the build chamber. When '*Time*' is included as an input feature, the XGBoost models likely consistently use it to approximately determine location. This occurs because the ConceptLaser M2 machine printed these components at the extreme left side of the machine first proceeding from front to back. The '*Time*' feature indirectly gives the model approximate spatial

information. As such, the models are able to correctly predict more samples on the right side of the machine where the prevalence is greater as shown in Figure 43 — labelled “Full”. However, when the ‘*Time*’ feature is removed, performance across the build plate changes. Row two of Figure 43 — labelled “WO Time” — shows that the balanced accuracy tends to be greater toward the back of the build chamber albeit only slightly. This may stem from the data lacking information that relates to the higher prevalence of soot at the front of the build chamber leading to decreased model sensitivity. Using only in-situ process log data shows a similar pattern. Across all three input feature sets, the component-wise balanced accuracy is very similar for build 1. This is likely related to the models that predicted defects for build 1 having been trained on builds 2 through 4 that have a different geometry. Since different geometries yield different trajectory data, the model was likely unable to leverage those features effectively for defect detection and thus the little apparent spatial performance change from one input feature set to the next.

Future Work

This work demonstrates a first take on leveraging already-available in-situ process and trajectory data modalities, but it only begins to scratch the surface of testing the possible utility of these data. When training the ML model this work completely neglected spatial information and makes predictions for each voxel based only on the sensor values for that voxel. Instead of XGBoost these data could be used to train a deep CNN which would be able to learn from the spatial relationships between the mapped voxels in addition to the mapped data. Further, they could be employed as additional input channels in existing

deep learning NN models, like the DSCNN that generated the initial defect data in this study, for more accurate defect detection. Alternatively, with spatial and trajectory data omitted, as was the case for the third input feature set, the existing models could be utilized directly with sampled, streaming in-situ process data to make defect predictions in real-time as a coarse AI-based process monitoring system.

Finally, this type of analysis should be readily employable for any AM technology that has the requisite data — in-situ temporal process data, trajectory data, and localized property data. One can even imagine leveraging this method to mine for relationships between other localized properties such as microstructure and mechanical properties by utilizing the feature importance scores produced by XGBoost to discover relationships between the inputs and targets to investigate further with targeted experiments.

Conclusion

This analysis represents a first attempt at leveraging in-situ process data for predicting localized build defects. It elicits interesting possibilities for leveraging temporal data modalities for predicting localized spatial properties. Specifically, that spatially mapped, in-situ process and trajectory data may be fused into a common representation and related to image-based defect data. The overall performance of the model is likely not adequate to be deployed as the sole means for process monitoring and soot detection. However, the use of XGBoost as the chosen classifier yields desirable auxiliary capabilities — namely the ability to narrow down which in-situ data are most relevant for detecting certain in-

situ defects. Additionally, the spatially transformed in-situ process dataset may be used as additional inputs to other ML models for more accurate prediction of defects.

CONCLUSION

At the outset of this work several key themes for metal PBF AM were outlined.

1. Metal PBF AM is a promising set of technologies that still have room for improvement to reduce mechanical anisotropy and inconsistent quality in fabricated components.
2. The inconsistent quality is due, in large part, to localized heterogeneities in microstructure and flaws throughout fabricated components.
3. Investigating process-structure relationships is necessary to understand how the complex AM process affects the evolution of these structures.
4. The scan path is an important key to understanding this inherently spatio-temporal process as well as the corresponding dynamic interactions that introduce significant process complexity.

The work herein, has focused on the scan path as a key locus to understand the spatio-temporal nature of the metal PBF AM process and to explore localized heterogeneities that are still pervasive. This work represents just a small foray into understanding the inextricable and intricate links between the scan path and resulting structures. The two tasks have sought to leverage scan path data to explore heterogeneities of both microstructure and larger process flaws.

Task one proposed methods for optimizing the scan path to induce and control different microstructure morphologies in arbitrary, local regions. Such a capability will add an

entirely new dimension of design for additively manufactured components — the ability to tailor mechanical properties locally that are best suited for the forces that the component will experience during use. Importantly, the optimizations made minimal assumptions and produce solutions that are uniquely tailored for the specific microstructural regions. Additionally, with few a priori assumptions, it was shown how the resulting scan paths may be probed towards creating new scan strategies.

The first portion of task two leveraged scan path data for spatially mapping temporal, in situ process data. These data streams may include records from sensors that monitor the build chamber and give insight to fluctuations that occur in the build environment.

Spatially mapping such data allows for investigating how they correlate with other inherently spatial data such as in situ images or ex situ x-ray computed tomography or micrographs. Additionally, abstract common data representations for analogous and ubiquitous data streams from metal AM were introduced and discussed. The addition of these abstractions will facilitate creating complex data processing pipelines that are portable and reusable for different metal AM machines and processes.

The work for the second portion of task two represents the most conservative and constrained possible implementation of an ML model trained on spatially mapped, in-situ process data, and it has resulted in a baseline performance for using such data for predicting spatial flaws that are related to the state of the build environment. The groundwork has been laid to further explore possibilities of leveraging multiple different

data modalities for process monitoring and instance qualification and may lead to better flaw detections and process understanding. The work offers many opportunities to extend the work and further explore process-structure relationships.

Together these tasks serve to give a glimpse of what is possible when applying the principles of data science to metal PBF AM as well as uncover a plethora of future opportunities for the same. The potential and possibilities at the intersection of data science and AM are not only exciting but crucial in order to continue to improve the process and push the bounds of what is possible for metal AM.

REFERENCES

- [1] C. R. Deckard, "Part generation by layer-wise selective laser sintering," University of Texas, Austin, TX, 1986.
- [2] ISO/ASTM, "Additive Manufacturing - General Principles Terminology (ASTM52900)," *Rapid Manufacturing Association*. pp. 10–12, 2013.
- [3] E. Herderick, "Additive manufacturing of metals: A review," *Mater. Sci. Technol. Conf. Exhib. 2011, MS T'11*, vol. 2, no. 176252, pp. 1413–1425, 2011.
- [4] H. Kim, Y. Lin, and T. L. B. Tseng, "A review on quality control in additive manufacturing," *Rapid Prototyp. J.*, vol. 24, no. 3, pp. 645–669, 2018.
- [5] X. Lou and D. Gandy, "Advanced Manufacturing for Nuclear Energy," *Jom*, vol. 71, no. 8, pp. 2834–2836, 2019.
- [6] W. J. Sames, F. A. List, S. Pannala, R. R. Dehoff, and S. S. Babu, "The metallurgy and processing science of metal additive manufacturing," *Int. Mater. Rev.*, vol. 61, no. 5, pp. 315–360, 2016.
- [7] B. M. Sharratt, "Non-Destructive Techniques and Technologies for Qualification of Additive Manufactured Parts and Processes: A Literature Review," *Dep. Natl. Def. Canada*, vol. 55, no. March, pp. 95–100, 2015.
- [8] M. F. Zäh and S. Lutzmann, "Modelling and simulation of electron beam melting," *Prod. Eng.*, vol. 4, no. 1, pp. 15–23, Feb. 2010.
- [9] S. Dryepontdt, M. Kirka, B. Pint, and D. Ryan, "Comparison of Electron Beam and Laser Beam Powder Bed Fusion Additive Manufacturing Process for High Temperature Turbine Component Materials, Phase II," *ORNL Rep.*, vol. ORNL/TM-20, 2019.
- [10] M. Ashraf, I. Gibson, M. G. Rashed, and M. G. Rashed, "CHALLENGES AND PROSPECTS OF 3D PRINTING IN STRUCTURAL ENGINEERING," no. November 2017, 2018.
- [11] S. K. Everton, M. Hirsch, P. I. Stavroulakis, R. K. Leach, and A. T. Clare, "Review of in-situ process monitoring and in-situ metrology for metal additive manufacturing," *Mater. Des.*, vol. 95, pp. 431–445, 2016.
- [12] M. Grasso and B. M. Colosimo, "Process defects and in situ monitoring methods in metal powder bed fusion: A review," *Meas. Sci. Technol.*, vol. 28, no. 4, Apr. 2017.
- [13] "Inside Electron Beam Melting."
- [14] C. Körner, H. Helmer, A. Bauereiß, and R. F. Singer, "Tailoring the grain structure of IN718 during selective electron beam melting," *MATEC Web Conf.*, vol. 14, p. 08001, 2014.
- [15] Y. Kok *et al.*, "Anisotropy and heterogeneity of microstructure and mechanical properties in metal additive manufacturing: A critical review," *Mater. Des.*, vol. 139, pp. 565–586, 2018.
- [16] T. DebRoy *et al.*, "Additive manufacturing of metallic components – Process, structure and properties," *Prog. Mater. Sci.*, vol. 92, pp. 112–224, 2018.
- [17] M. A. Groeber, E. Schwalbach, S. Donegan, K. Chaput, T. Butler, and J. Miller, "Application of characterization, modelling, and analytics towards understanding process-structure linkages in metallic 3D printing," *IOP Conf. Ser. Mater. Sci.*

- Eng.*, vol. 219, no. 1, 2017.
- [18] J. M. Waller, B. H. Parker, K. L. Hodges, E. R. Burke, and J. L. Walker, "Nondestructive Evaluation of Additive Manufacturing State-of-the-Discipline Report," *Nasa/Tm-2014-218560*, no. November, pp. 1–36, 2014.
 - [19] M. Seifi *et al.*, "Progress Towards Metal Additive Manufacturing Standardization to Support Qualification and Certification," *Jom*, vol. 69, no. 3, pp. 439–455, 2017.
 - [20] S. S. Babu, L. Love, R. R. Dehoff, W. Peter, T. R. Watkins, and S. Pannala, "Additive manufacturing of materials: Opportunities and challenges," *MRS Bull.*, vol. 40, no. 12, pp. 1154–1161, 2015.
 - [21] J. P. Kruth, G. Levy, F. Klocke, and T. H. C. Childs, "Consolidation phenomena in laser and powder-bed based layered manufacturing," *CIRP Ann.*, vol. 56, no. 2, pp. 730–759, Jan. 2007.
 - [22] T. Ishimoto, K. Hagihara, K. Hisamoto, S. H. Sun, and T. Nakano, "Crystallographic texture control of beta-type Ti–15Mo–5Zr–3Al alloy by selective laser melting for the development of novel implants with a biocompatible low Young's modulus," *Scr. Mater.*, vol. 132, pp. 34–38, Apr. 2017.
 - [23] K. Hagihara, T. Nakano, M. Suzuki, T. Ishimoto, Suyalatu, and S. H. Sun, "Successful additive manufacturing of MoSi₂ including crystallographic texture and shape control," *J. Alloys Compd.*, vol. 696, pp. 67–72, Mar. 2017.
 - [24] T. Amine, J. W. Newkirk, and F. Liou, "Methodology for Studying Effect of Cooling Rate During Laser Deposition on Microstructure," *J. Mater. Eng. Perform.*, vol. 24, no. 8, pp. 3129–3136, Aug. 2015.
 - [25] W. Xu *et al.*, "Additive manufacturing of strong and ductile Ti–6Al–4V by selective laser melting via in situ martensite decomposition," *Acta Mater.*, vol. 85, pp. 74–84, Feb. 2015.
 - [26] H. Helmer, A. Bauereiß, R. F. Singer, and C. Körner, "Grain structure evolution in Inconel 718 during selective electron beam melting," *Mater. Sci. Eng. A*, vol. 668, pp. 180–187, Jun. 2016.
 - [27] X. Tan *et al.*, "Graded microstructure and mechanical properties of additive manufactured Ti–6Al–4V via electron beam melting," *Acta Mater.*, vol. 97, pp. 1–16, Sep. 2015.
 - [28] A. A. Antonysamy, J. Meyer, and P. B. Prangnell, "Effect of build geometry on the β -grain structure and texture in additive manufacture of Ti6Al4V by selective electron beam melting," *Mater. Charact.*, vol. 84, pp. 153–168, 2013.
 - [29] T. Machry, D. Eatock, J. Meyer, A. Antonysamy, A. Ho, and P. Prangnell, "Effect of microstructure on the tensile strength of Ti6Al4V specimens manufactured using additive manufacturing electron beam process," *Powder Metall.*, vol. 59, no. 1, pp. 41–50, Jan. 2016.
 - [30] S. Yoder *et al.*, "Characterization of topology optimized Ti-6Al-4V components using electron beam powder bed fusion," *Addit. Manuf.*, vol. 19, pp. 184–196, 2018.
 - [31] X. Tan *et al.*, "Revealing martensitic transformation and α/β interface evolution in electron beam melting three-dimensional-printed Ti-6Al-4V," *Sci. Reports 2016*

- 61, vol. 6, no. 1, pp. 1–10, May 2016.
- [32] X. Tan *et al.*, “An experimental and simulation study on build thickness dependent microstructure for electron beam melted Ti–6Al–4V,” *J. Alloys Compd.*, vol. 646, pp. 303–309, Oct. 2015.
 - [33] S. H. Sun, Y. Koizumi, S. Kurosu, Y. P. Li, H. Matsumoto, and A. Chiba, “Build direction dependence of microstructure and high-temperature tensile property of Co–Cr–Mo alloy fabricated by electron beam melting,” *Acta Mater.*, vol. 64, pp. 154–168, Feb. 2014.
 - [34] J. L. Bartlett, F. M. Heim, Y. V. Murty, and X. Li, “In situ defect detection in selective laser melting via full-field infrared thermography,” *Addit. Manuf.*, vol. 24, no. October, pp. 595–605, 2018.
 - [35] and R. C. R. Carter, Luke N., Moataz M. Attallah, “Laser powder bed fabrication of nickel-base superalloys: influence of parameters; characterisation, quantification and mitigation of cracking,” *Superalloys*, vol. 6, pp. 2826–2834, 2012.
 - [36] R. R. Dehoff *et al.*, “Site specific control of crystallographic grain orientation through electron beam additive manufacturing,” *Mater. Sci. Technol.*, vol. 31, no. 8, pp. 931–938, 2015.
 - [37] H. Gu, H. Gong, D. Pal, K. Rafi, T. Starr, and B. Stucker, “Influences of energy density on porosity and microstructure of selective laser melted 17-4PH stainless steel,” *24th Int. SFF Symp. - An Addit. Manuf. Conf. SFF 2013*, no. August, 2013.
 - [38] J. Beuth *et al.*, “Process mapping for qualification across multiple direct metal additive manufacturing processes,” *24th Int. SFF Symp. - An Addit. Manuf. Conf. SFF 2013*, pp. 655–665, 2013.
 - [39] A. Simchi and H. Pohl, “Effects of laser sintering processing parameters on the microstructure and densification of iron powder,” *Mater. Sci. Eng. A*, vol. 359, no. 1–2, pp. 119–128, 2003.
 - [40] Y. M. Arisoy, L. E. Ciales, T. Özel, B. Lane, S. Moylan, and A. Donmez, “Influence of scan strategy and process parameters on microstructure and its optimization in additively manufactured nickel alloy 625 via laser powder bed fusion,” *Int. J. Adv. Manuf. Technol.*, vol. 90, no. 5–8, pp. 1393–1417, 2017.
 - [41] S. P. Narra, L. Scime, and J. Beuth, “Integrated Control of Melt Pool Geometry and Microstructure in Laser Powder Bed Fusion of AlSi10Mg,” *Metall. Mater. Trans. A Phys. Metall. Mater. Sci.*, vol. 49, no. 10, pp. 5097–5106, 2018.
 - [42] C. Guo, W. Ge, and F. Lin, “Effects of scanning parameters on material deposition during Electron Beam Selective Melting of Ti-6Al-4V powder,” *J. Mater. Process. Technol.*, vol. 217, pp. 148–157, 2015.
 - [43] T. Scharowsky, V. Juechter, R. F. Singer, and C. Körner, “Influence of the Scanning Strategy on the Microstructure and Mechanical Properties in Selective Electron Beam Melting of Ti-6Al-4V,” *Adv. Eng. Mater.*, vol. 17, no. 11, pp. 1573–1578, 2015.
 - [44] P. Liu, Z. Wang, Y. Xiao, M. F. Horstemeyer, X. Cui, and L. Chen, “Insight into the mechanisms of columnar to equiaxed grain transition during metallic additive manufacturing,” *Addit. Manuf.*, vol. 26, pp. 22–29, Mar. 2019.
 - [45] J. P. Choi *et al.*, “Densification and microstructural investigation of Inconel 718

- parts fabricated by selective laser melting,” *Powder Technol.*, vol. 310, pp. 60–66, 2017.
- [46] Y. Lu *et al.*, “Study on the microstructure, mechanical property and residual stress of SLM Inconel-718 alloy manufactured by differing island scanning strategy,” *Opt. Laser Technol.*, vol. 75, pp. 197–206, 2015.
 - [47] H. E. Helmer, C. Körner, and R. F. Singer, “Additive manufacturing of nickel-based superalloy Inconel 718 by selective electron beam melting: Processing window and microstructure,” *J. Mater. Res.*, vol. 29, no. 17, pp. 1987–1996, 2014.
 - [48] A. Plotkowski, M. M. Kirka, and S. S. Babu, “Verification and validation of a rapid heat transfer calculation methodology for transient melt pool solidification conditions in powder bed metal additive manufacturing,” *Addit. Manuf.*, vol. 18, pp. 256–268, 2017.
 - [49] R. R. Dehoff, M. M. Kirka, F. A. List, K. A. Unocic, and W. J. Sames, “Crystallographic texture engineering through novel melt strategies via electron beam melting: Inconel 718,” *Mater. Sci. Technol.*, vol. 31, no. 8, pp. 939–944, 2015.
 - [50] N. T. Nguyen, A. Ohta, K. Matsuoka, N. Suzuki, and Y. Maeda, “Analytical Solutions for Transient Temperature of Semi-Infinite Body Subjected to 3-D Moving Heat Sources,” *Weld. J. (Miami, Fla)*, vol. 78, no. 8, 1999.
 - [51] B. Stump and A. Plotkowski, “An adaptive integration scheme for heat conduction in additive manufacturing,” *Appl. Math. Model.*, vol. 75, pp. 787–805, 2019.
 - [52] M. Gäumann, C. Bezençon, P. Canalis, and W. Kurz, “Single-crystal laser deposition of superalloys: Processing-microstructure maps,” *Acta Mater.*, vol. 49, no. 6, pp. 1051–1062, 2001.
 - [53] J. D. Hunt, “Steady state columnar and equiaxed growth of dendrites and eutectic,” *Mater. Sci. Eng.*, vol. 65, no. 1, pp. 75–83, 1984.
 - [54] M. Haines, A. Plotkowski, C. L. Frederick, E. J. Schwalbach, and S. S. Babu, “A sensitivity analysis of the columnar-to-equiaxed transition for Ni-based superalloys in electron beam additive manufacturing,” *Comput. Mater. Sci.*, vol. 155, no. August, pp. 340–349, 2018.
 - [55] N. Raghavan *et al.*, “Numerical modeling of heat-transfer and the influence of process parameters on tailoring the grain morphology of IN718 in electron beam additive manufacturing,” *Acta Mater.*, vol. 112, pp. 303–314, 2016.
 - [56] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Mass.: Addison-Wesley Pub. Co., 1989.
 - [57] B. Ozcelik, H. Oktem, and H. Kurtaran, “Optimum surface roughness in end milling Inconel 718 by coupling neural network model and genetic algorithm,” *Int. J. Adv. Manuf. Technol.*, vol. 27, no. 3–4, pp. 234–241, 2005.
 - [58] K. Saravanakumar, M. Kumar, and A. Dawood, “Optimization of CNC turning process parameters on Inconel 718 using genetic algorithm,” *Optimization*, vol. 2, no. 4, pp. 532–537, 2012.
 - [59] J. S. Senthilkumaar, P. Selvarani, and R. M. Arunachalam, “Intelligent optimization and selection of machining parameters in finish turning and facing of Inconel 718,” *Int. J. Adv. Manuf. Technol.*, vol. 58, no. 9–12, pp. 885–894, 2012.

- [60] P. K. Shrivastava and A. K. Pandey, "Parametric optimization of multiple quality characteristics in laser cutting of Inconel-718 by using hybrid approach of multiple regression analysis and genetic algorithm," *Infrared Phys. Technol.*, vol. 91, pp. 220–232, 2018.
- [61] K. S. Prasad, C. Srinivasa Rao, and D. N. Rao, "Optimization of fusion zone grain size, hardness, and ultimate tensile strength of pulsed current micro plasma arc welded Inconel 625 sheets using genetic algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 85, no. 9–12, pp. 2287–2295, 2016.
- [62] E. Falkenauer and S. Bouffouix, "A genetic algorithm for job shop," *ICRA*, vol. April, pp. 824–829, 1991.
- [63] G. Syswerda, "No TitleSchedule optimization using genetic algorithms," in *Handbook of genetic algorithms*, 1991.
- [64] S. S. Razvi, S. Feng, A. Narayanan, Y. T. T. Lee, and P. Witherell, "A review of machine learning applications in additive manufacturing," *Proc. ASME Des. Eng. Tech. Conf.*, vol. 1, pp. 1–10, 2019.
- [65] S. S. Babu *et al.*, "Additive Manufacturing of Nickel Superalloys: Opportunities for Innovation and Challenges Related to Qualification," *Metall. Mater. Trans. A Phys. Metall. Mater. Sci.*, vol. 49, no. 9, pp. 3764–3780, 2018.
- [66] E. Popova, T. M. Rodgers, X. Gong, A. Cecen, J. D. Madison, and S. R. Kalidindi, "Process-Structure Linkages Using a Data Science Approach: Application to Simulated Additive Manufacturing Data," *Integr. Mater. Manuf. Innov.*, vol. 6, no. 1, pp. 54–68, 2017.
- [67] L. Scime, A. Singh, and V. Paquit, "A scalable digital platform for the use of digital twins in additive manufacturing," *Manuf. Lett.*, 2021.
- [68] Michael Grieves, "Digital Twin: Manufacturing Excellence through Virtual Factory Replication," *White Pap.*, pp. 1–7, 2014.
- [69] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
- [70] T. DebRoy, W. Zhang, J. Turner, and S. S. Babu, "Building digital twins of 3D printing machines," *Scr. Mater.*, vol. 135, pp. 119–124, 2017.
- [71] G. L. Knapp *et al.*, "Building blocks for a digital twin of additive manufacturing," *Acta Mater.*, vol. 135, pp. 390–399, 2017.
- [72] T. Mukherjee and T. DebRoy, "A digital twin for rapid qualification of 3D printed metallic components," *Appl. Mater. Today*, vol. 14, pp. 59–65, 2019.
- [73] O. Fergani and K. Eissing, "A machine learning-based digital twin of the manufacturing process: metal powder-bed fusion case," no. February, pp. 1–8, 2020.
- [74] M. A. Groeber and M. A. Jackson, "DREAM.3D: A Digital Representation Environment for the Analysis of Microstructure in 3D," vol. 3, p. 5, 2014.
- [75] Z. C. Cordero, H. M. Meyer, P. Nandwana, and R. R. Dehoff, "Powder bed charging during electron-beam additive manufacturing," *Acta Mater.*, vol. 124, pp. 437–445, 2017.
- [76] S. Clijsters, T. Craeghs, S. Buls, K. Kempen, and J. P. Kruth, "In situ quality

- control of the selective laser melting process using a high-speed, real-time melt pool monitoring system,” *Int. J. Adv. Manuf. Technol.*, vol. 75, no. 5–8, pp. 1089–1101, Oct. 2014.
- [77] A. Vandone, S. Baraldo, and A. Valente, “Multisensor data fusion for additive manufacturing process control,” *IEEE Robot. Autom. Lett.*, 2018.
 - [78] J. Petrich, Z. Snow, D. Corbin, and E. W. Reutzel, “Multi-modal sensor fusion with machine learning for data-driven process monitoring for additive manufacturing,” *Addit. Manuf.*, vol. 48, no. PB, p. 102364, 2021.
 - [79] D. Kozjek *et al.*, “Iterative closest point-based data fusion of non-synchronized in-situ and ex-situ data in laser powder bed fusion,” *J. Manuf. Syst.*, vol. 66, no. August 2022, pp. 179–199, 2023.
 - [80] “OpenVectorFormat.” Digital Production Aachen, 2023.
 - [81] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965.
 - [82] V. Dhiman, “BRESENHAM’S LINE ALGORITHM IN N-DIMENSIONS (PYTHON RECIPE),” *ActiveState*, 2012. [Online]. Available: <https://code.activestate.com/recipes/578112-bresenhams-line-algorithm-in-n-dimensions/>. [Accessed: 10-Nov-2021].
 - [83] A. H. Moltumyr, M. H. Arbo, and J. T. Gravdahl, “Towards Vision-based Closed-loop Additive Manufacturing : A Review,” 2020.
 - [84] M. Grasso, F. Gallina, and B. M. Colosimo, “Data fusion methods for statistical process monitoring and quality characterization in metal additive manufacturing,” *Procedia CIRP*, vol. 75, pp. 103–107, 2018.
 - [85] S. Chandrasekar *et al.*, “Investigating the effect of metal powder recycling in Electron beam Powder Bed Fusion using process log data,” *Addit. Manuf.*, vol. 32, no. July 2019, 2020.
 - [86] C. A. Steed, W. Halsey, R. R. Dehoff, S. L. Yoder, V. Paquit, and S. Powers, “Falcon: Visual analysis of large, irregularly sampled, and multivariate time series data in additive manufacturing,” *Comput. Graph.*, vol. 63, pp. 50–64, Apr. 2017.
 - [87] M. J. Heiden *et al.*, “Evolution of 316L stainless steel feedstock due to laser powder bed fusion process,” *Addit. Manuf.*, vol. 25, pp. 84–103, 2019.
 - [88] D. J. Huang and H. Li, “A machine learning guided investigation of quality repeatability in metal laser powder bed fusion additive manufacturing,” *Mater. Des.*, vol. 203, p. 109606, 2021.
 - [89] B. Roidl and T. Fauner, “Continuous Improvement in Gas Flow Design.”
 - [90] L. Scime, D. Siddel, S. Baird, and V. Paquit, “Layer-wise anomaly detection and classification for powder bed additive manufacturing processes: A machine-agnostic algorithm for real-time pixel-wise semantic segmentation,” *Addit. Manuf.*, vol. 36, no. March, p. 101453, 2020.
 - [91] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
 - [92] “Awesome XGBoost.” [Online]. Available: <https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge->

- winning-solutions. [Accessed: 10-Aug-2021].
- [93] J. H. Friedman, “Greedy Function Approximation: a Gradient Boosting Machine,” *Ann. Stat.*, no. 2, p. 34, 2001.
- [94] “Introduction to Boosted Trees,” *XGBoost*. [Online]. Available: <https://xgboost.readthedocs.io/en/latest/tutorials/model.html>. [Accessed: 05-Oct-2021].
- [95] “Python API Reference.” [Online]. Available: https://xgboost.readthedocs.io/en/latest/python/python_api.html#module-xgboost.plotting. [Accessed: 11-Aug-2021].
- [96] J. Aarshay, “Complete Guide to Parameter Tuning in XGBoost,” *Analytics Vidya*, 2016. [Online]. Available: <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>.
- [97] “XGBoost Parameters.” [Online]. Available: <https://xgboost.readthedocs.io/en/latest/parameter.html>. [Accessed: 10-Aug-2021].
- [98] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.

VITA

William Halsey was born in and grew up in Knoxville, TN. Upon graduation from Karns High School he pursued undergraduate studies at Clemson University in Clemson, SC where he earned a Bachelor of Science degree in computer engineering. Upon completion, he returned to Knoxville to attend the University of Tennessee for graduate studies. He completed a dual degree program which yielded a Master of Science degree in computer engineering and a Master of Business Administration. During these graduate studies he undertook an internship with Oak Ridge Associated Universities to work at Oak Ridge National Laboratories assisting in the development of Falcon, a data exploration and visual analytics tools for time series data from Arcam EB-PBF AM machines. This work was continued at ORNL during a post-master's program through UT. After the conclusion of the post-master's appointment, William chose to pursue a Ph.D. in Data Science and Engineering through the Bredesen Center for Interdisciplinary Graduate Education at UT. He has subsequently been hired as an R&D Assistant Staff member at ORNL where he works on a diverse portfolio of data science projects for advanced manufacturing with a special emphasis on metal PBF AM.