

Evaluation of FPGA-Based Track-Wise Process Quality Monitoring for Laser Powder Bed Fusion

Chen-Wei Yang¹, Gisuk Hong¹, Yang Zhuo¹, Yan Lu², Shengyen Li², Haw-Ching Yang³

¹ Research Associate of National Institute of Standards and Technology (NIST)

² Systems Integration Division, National Institute of Standards and Technology (NIST)

³ National Kaohsiung University of Science and Technology (NKUST)

Abstract

In additive manufacturing, multi-scale closed-loop control of the laser powder bed fusion process (LPBF) can potentially ensure part quality. Track-wise feedback control aims to detect and repair over-fusion or lack of fusion at the track scale, often observed by image-based melt pool monitoring and through the analysis of melting overlap between adjacent tracks. While most closed-loop control for LPBF exists at the research stage, real-time implementation remains challenging due to the need for both complex computation and high control performance, which is difficult to achieve using general-purpose PCs. In this work, we leverage the NIST AM Data Integration Testbench to evaluate the implementability of real-time track-wise monitoring using Field Programmable Gate Array (FPGA) based processing on a frame grabber of a 10 KHz melt pool image acquisition rate. We implement two scenarios of track-wise overlap ratio calculation. The results demonstrate that both computation tasks can be completed within the sampling interval on the FPGA by applying a hardware-friendly design, achieving a balance between latency and resource utilization.

1. Introduction

Laser Powder Bed Fusion (LPBF) additive manufacturing (AM) has transformed part sustainment by eliminating the need for dedicated tooling and allowing for the replacement of obsolete or low-volume parts, thus extending the service life of critical systems [1][2]. Industries such as aerospace, healthcare, and automotive increasingly rely on LPBF AM to meet the demand for complex, customized parts [3]. For scale-up production, LPBF offers the potential for distributed manufacturing and digital inventory, where parts can be produced on-demand and near the point of use, reducing lead times and reliance on the supply chain [3]. However, while LPBF is already used for end-use parts, scaling up remains challenging due to process consistency, throughput limitations, and costly qualification [4].

The current approach to process control and qualification in LPBF often involves "locking" process parameters, such as scan path, laser power, and scan speed. This rigid parameter locking is required to meet part certification, especially for those safety-critical applications regulated by the FAA (Federal Aviation Administration), NRC (Nuclear Regulatory Commission), or NASA (National Aeronautics and Space Administration) [5]. The open-loop process control restricts the

AM system's ability to handle abnormal melting or cooling caused by inherent process variability, leading to challenges in ensuring consistent part quality across builds and machines.

In-process closed-loop feedback control using real-time monitoring observations, such as melt pool temperature or emission, is used to adjust process parameters dynamically during printing and offers a promising solution to overcome these challenges [6]. By enabling closed-loop control, feedback systems can compensate for disturbances, reduce defects such as porosity and surface roughness, and improve process part quality. This approach could also support more flexible qualification strategies, allowing parameter adjustments to keep key process variables within controlled bounds. Closed-loop control can be designed at multiple scales [7]. The point scale involves real-time adjustment of parameters like laser power or scan speed based on direct measurements of the melt pool (e.g., temperature, size, or thermal emission), at high sampling rates. Controllers such as proportional (P), adaptive P, and sliding mode have been proposed and applied to regulate melt pool temperature, resulting in improved microstructure uniformity and mechanical properties [8][9]. In these studies, systems might use a pyrometer or high-speed camera to monitor the melt pool and adjust the laser within tens or hundreds of microseconds to prevent defects like lack-of-fusion, balling, or over-melting. Layer-wise closed-loop control fuses monitoring data over an entire layer and adjusts process parameters for subsequent layers. This approach can address broader process trends, such as heat accumulation or powder bed irregularities, and is useful for compensating for drift or systematic errors over the build. Layer-wise control is less real-time demanding than point control but can be limited in correcting local defects in a timely manner [10][11][12].

Track-wise control is a promising intermediate approach that balances control precision with implementation feasibility [13]. LPBF systems typically employ a raster scan pattern, where the laser moves back and forth across the powder bed scanning in both directions (left-to-right and right-to-left) within each stripe or hatch region. G-code forms a common base for LPBF command programming. Most LPBF systems use G-code for basic toolpath commands, such as laser position, velocity, and on/off signals. For example, each laser scan vector is represented as a G-code line, and parameters like laser power and scan speed are embedded in G-code commands.

```
N6 F1020 : Set scan speed to 1020 mm/min
N7 G01 X-5.925 Y-5.925 Z-0.000 : Move to (5.925, 5.925, 0.000) with laser power 200
S-200
N8 M45 : Laser ON
N9 G01 X-5.925 : Continue scan
N10 G01 Y-5.925 : Continue scan
N11 G01 X-5.925 : Continue scan
N12 G01 Y-5.925 : Continue scan
N13 M46 : Laser OFF
```

In LPBF systems, typical scan vector lengths within a stripe range from a few millimeters up to about 10 millimeters, depending on the stripe width and part geometry. Scan speeds generally fall between 200 mm/s and 3000 mm/s, meaning each scan vector is completed in just a few milliseconds. When accounting for skywriting time between 4 and 8 milliseconds, the total track-to-track control interval averages around 15 milliseconds, which is an ideal window for real-time, track-wise closed-loop control.

For track-wise control, the track-to-track fusion quality can be the control target because it serves as a critical quality indicator for detecting and preventing common defects like lack-of-fusion porosity. Multi-scale Model Predictive Control (MPC) frameworks have been developed to close the loop with the track-wise overlap ratio [13], demonstrating that track-wise control balances micro-scale and macro-scale issues, though implementation challenges remain. However, the implementability of track-wise closed-loop control based on the track-wise overlap ratio must be assessed before claiming success. Recent advances in Field Programmable Gate Array (FPGA) implementation for LPBF control have improved capabilities in real-time scan path planning, galvo motion control, and laser power control [14][15]. This paper implements the computational feasibility of track-wise fusion quality monitoring based on high-speed melt pool imaging on an FPGA provided by an image frame grabber, and measurements are conducted for computation accuracy, resource consumption, and computation time. Prior evaluations on this onboard FPGA have demonstrated its capability to implement real-time melt pool monitoring, including area, intensity, width, and length [16].

2. Track-wise Fusion Quality Monitoring based on Melt Pool Monitoring (MPM)

Despite fixed process parameters, melt pool stability and overlap ratio often fluctuate due to the stochastic nature of powder spreading and varying geometric factors. Variations in powder particle size distribution (PSD) affect thermal absorption; fine powders can overheat, while coarse powders may cause incomplete melting [17]. Complex geometries such as sharp corners or overhangs disrupt heat flow and melt pool behavior, resulting in insufficient remelting or keyholing [18]. These instabilities directly impact the overlap ratio between tracks, a key indicator of fusion quality that correlates with lack-of-fusion porosity and overheating defects [19][20]. Overlap ratio can be characterized using high-speed coaxial melt pool images, and it serves as a practical and effective state variable for real-time monitoring and control. This motivates the development of a track-wise fusion quality monitoring strategy based on Melt Pool Monitoring (MPM).

MPM fusion allows for the integration of consecutive MPM images into melting tracks. The fused result can be evaluated for critical features such as the melting width of individual tracks and the overlap ratio between two tracks. The fusion process relies on the spatial and temporal

information of the MPM images and correlated position data. Previous studies demonstrated such techniques [7][21][22].

2.1 Baseline: Image-Based Fusion of MPM Tracks

Figure 1 shows an example of fusing one original MPM image into a melting track. Grayscale thresholding is applied to the original images to remove noise. The second step is to register the processed melt pool images into the track after the MPM has been spatially and temporally synchronized to the machine coordinate system.

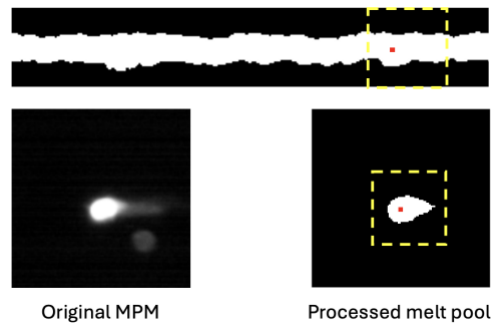


Figure 1. An example demonstrates the fusion process of an original MPM image to the machine coordinate system, forming a melting track. The red pixel represents the laser spot, while the pixels within the dashed box are those being registered [22].

Figure 2 shows an example of the fused result of two tracks constructed from 226 MPM images from the NIST AMMT Dataset. The pixel positions of the material melted twice constitute the overlap region that connects two adjacent tracks. In this figure, the zoomed-in view indicates some areas with no overlap, which may indicate a lack of fusion.

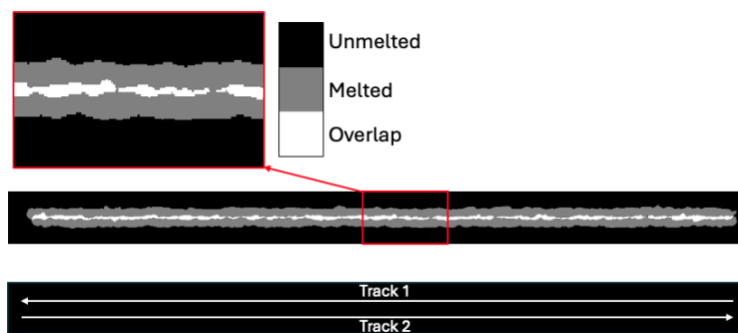


Figure 2. An example of fusing two melting tracks. The first track, located on the top, scans from right to left horizontally. The second track scans from the opposite direction and is located on the bottom. The area that melted only once is displayed in gray, and the overlapping area that melted twice is in white color.

2.2 FPGA-Efficient Overlap Estimation via Melt Pool Width

FPGA is well-suited for deterministic tasks like overlap monitoring, unlike CPUs and GPUs, which depend on operating systems. As shown in Figure 2, accurate overlap ratios can be calculated by stacking and aligning melt pool images from two tracks and counting pixels in the overlapping (white) region. However, this process cannot be fully pipelined under the current streaming architecture and would require additional dedicated logic modules, as well as significant resources. For example, in the AMMT Overhang Part X4 dataset [23], each track (scanned along the X-axis) contains 112 images at 120×120 resolution. At 800 mm/s over a 9 mm scan length with $8 \times 8 \mu\text{m}$ per pixel stacking results in an image of approximately 1125×128 total pixels per track. Computing the overlap between two such images require substantial Block Random Access Memory (BRAM), complex control logic, and compute pipelines, and must wait until both tracks are fully acquired.

Since each melt pool width strongly correlates with overlap, we can calculate the melt pool width feature directly from streaming images without going through stacking and the associated complex computations. We adopt this approach as a simpler and more resource-efficient alternative. The following sections introduce two scenarios for estimating the overlap ratio based on the melt pool width.

2.3 Spatially-Resolved Width-Based Fusion – Scenario 1 (S1)

Scenario 1's fusion approach maintains spatial granularity by analyzing individual melt pool pairs at corresponding sampling points along adjacent tracks to capture this critical phenomenon with maximum fidelity. For adjacent tracks k and $k + 1$, the overlap ratio at each corresponding point m is calculated using the equation $r_m = \frac{W_{k,m} + W_{k+1,m}}{2} - H$, where $W_{k,m}$ represents the melt pool width at the track k and point m , and H denotes the hatch distance. Taking AMMT Overhang Part X4 dataset as an example [23], process parameters (10 kHz camera rate, 800 mm/s laser speed, $80 \mu\text{m}$ interval between melt pool sampling measurements), each track (X-axis laser scanning direction) contains approximately 112 individual melt pool images, requiring 112 individual overlap calculations between adjacent tracks.

The primary advantage lies in its spatial precision and complete information preservation, enabling detection of localized defects such as a single melt pool exhibiting severe lack-of-fusion while surrounding points remain acceptable. This method can also capture transient phenomena such as powder bed irregularities affecting only specific locations. However, this method assumes that the melt pools between tracks are precisely aligned. In practice, points between adjacent tracks are often misaligned. Additional interpolation or a higher camera frame rate can help mitigate these alignment errors.

2.4 Simplified Track-Level Feature-based Fusion- Scenario 2 (S2)

Recognizing the computational constraints of real-time control systems, the simplified Track-level feature-based fusion approach was developed to achieve a practical balance between process monitoring capability and implementation feasibility, considering both timing and resource utilization in the embedded hardware. The overlap indicators are calculated as: minimum overlap estimation $\frac{\min(W_k)+\min(W_{k+1})}{2} - H$, maximum overlap estimation $\frac{\max(W_k)+\max(W_{k+1})}{2} - H$ and mean overlap estimation $\frac{\text{mean}(W_k)+\text{mean}(W_{k+1})}{2} - H$. This fundamental transformation reduces the feature dimensionality. However, this information reduction eliminates the ability to detect systematic spatial patterns that indicate process instabilities.

In summary, the point-by-point Scenario 1 (S1) approach can promptly detect localized defects through low overlap ratios at specific positions, enabling precise defect localization and potential real-time correction. In contrast, the mean melt pool width from the track-level Scenario 2 (S2) approach tends to yield an average value. Combined with a normal adjacent track, this results in a seemingly acceptable overlap, potentially masking severe local defects. Moreover, this approach cannot detect high fluctuations in melt pool width.

The following section presents the implementation and validation of both scenarios, along with an evaluation of FPGA performance and resource utilization.

3. FPGA-based Track-wise Processing

This section explains how track-wise processing is implemented on an FPGA. It starts under different scanning directions and then estimates overlap using two methods: melt pool width-based fusion and simplified feature-based fusion.

3.1 Melt-pool Width Calculation

In LPBF, scanning direction plays a critical role in determining melt-pool morphology. XY-axis scanning and tilted scanning paths are commonly used in alternating patterns to reduce residual stress and improve microstructure uniformity within layers. Therefore, we illustrate the methodology for handling and analyzing data under different laser scanning directions.

3.1.1 X-axis and Y-axis Laser Scanning Direction

Melt-pool Width Calculation Method (X-axis): Detect the top_row A and bottom_row B rows where the pixel intensity exceeds the threshold. (As shown on the left side of Figure 3.)

Algorithm:

Loop through each row in the image:

 If the first pixel > threshold and row_A is None, set row_A = current_row

 If the entire row contains only zeros after detecting row_A, set row_B = current_row

Y_axis_Interval = row_B - row_A

Return Y_axis_Interval as Width

Melt-pool Width Calculation Method (Y-axis): Detect the leftmost (min column A) and rightmost (max column B) pixel where intensity exceeds the threshold. (As shown on the right side of Figure 3.)

Algorithm:

Loop through each row in the image:

 Find the first column in this row where pixel > threshold → current_min

 Find the last column in this row where pixel > threshold → current_max

 Update:

 min_column_A = min(min_column, current_min)

 max_column_B = max(max_column, current_max)

X_axis_Interval = max_column_B - min_column_A

Return X_axis_Interval as Width

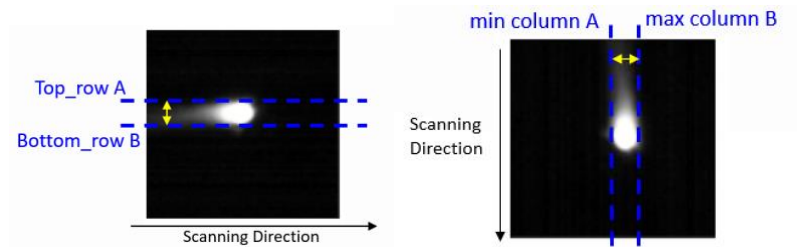


Figure 3. Melt-pool Image of X-axis and Y-axis Laser Scanning Direction

3.1.2 Tilted Laser Scanning Direction

As shown in Figure 4, the melt pool shape is rotated, and the melt-pool width (yellow arrow) is not aligned with the image axes. **a**: The red arrow represents the pixel width measured in the horizontal direction, which can be calculated using the same method described in Figure 3 for the Y-axis laser scanning direction. **θ**: The angle between the scanning direction and the X-axis. **b**: The melt-pool width in the direction vertical to the scanning direction. Based on observations, we infer that the relation $\mathbf{b} = \mathbf{a} \cdot \sin(\theta)$ can be applied to different tilted scanning directions to estimate the melt pool width. In real-time FPGA implementation: “a” needs to be measured in real time from the incoming image data. “θ” is predefined before the process starts, so $\sin(\theta)$ can be precomputed and stored in on-board memory for fast lookup during runtime.

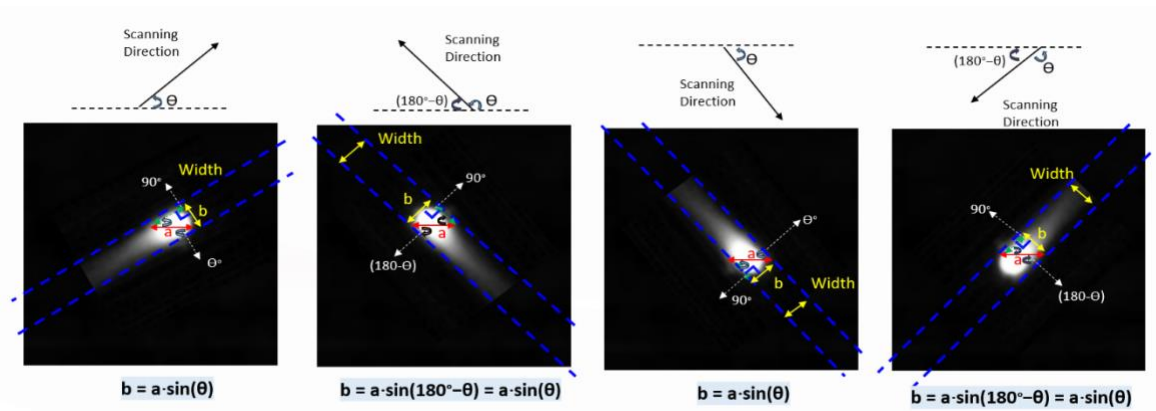


Figure 4. Width Estimation for Tilted Scanning Directions

4. Testing Dataset with Build Setting

In this paper, we utilize the Overhang Part X4 dataset provided by AMMT at NIST [23], focusing on 50 tracks from Layer 2, Part 1. Melt pool width is calculated along the X-axis (the laser scanning direction) to enable track-wise processing under two defined scenarios.

4.1 Image Sampling and Interval Configuration in Layer 2 Part 1

Table 1 summarizes the image sampling and interval settings that form the basis for subsequent analysis.

TABLE 1. Image Sampling and Interval Configuration in Layer 2 Part 1

Parameter	Value
Camera Rate	10 kHz
Size of 1 Pixel	$8 \mu\text{m} \times 8 \mu\text{m}$
Laser speed	800 mm/s
IMAGES_NUMBER (without contour)	5650
TRACKS_NUMBER	50
IMAGES_NUMBER_PER_TRACK	113
TIME_PER_TRACK	11.3 ms
TIME_BETWEEN_TRACKS	3 ms
HATCHING_DISTANCE	$0.098 \text{ mm} \approx 12\text{-pixel range}$
Interval between two points= 80um	$80 \mu\text{m}$

From Figure 5, we observe that the sampling points are not perfectly aligned due to system configuration. Additionally, laser acceleration and deceleration can cause the actual laser spot position to deviate from the intended path. As a result, point-by-point overlap may exhibit a

certain degree of spatial shift. However, this issue can be mitigated by interpolation or increasing the camera frame rate.

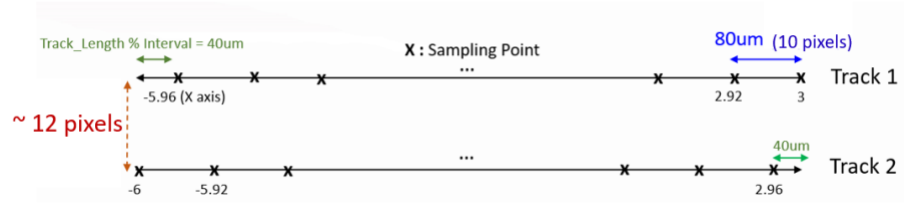


Figure 5. Sampling Point Distribution and Track Spacing in Layer 2 Part 1

4.2 Dataset Pre-Processing

The FPGA processes image data in 32-pixel packets (i.e., each packet contains 32 pixels). Therefore, the width and height of each image must be a multiple of 32 to ensure proper packet alignment and avoid issues with incomplete data transfers. The original dataset consists of grayscale images with a resolution of 120×120 pixels, which is not divisible by 32. To address this, zero-padding was applied to expand each image to 128×128 pixels, ensuring both dimensions comply with the 32-pixel boundary required by the hardware. Additionally, we observed that the first image of each track in Layer 2, Part 1 of the dataset, was entirely black. They were excluded from the evaluation to prevent these outliers from skewing track-level statistics.

5. Testing Platform – NIST AM Data Integration Testbench

To implement and evaluate both scenarios, we use an existing testbench with a high-speed camera simulator and an FPGA-based frame grabber. This setup allows us to load the dataset, simulate real-time data transmission, and evaluate FPGA processing for data acquisition and feature extraction. Figure 6 illustrates the step-by-step integration of the offline development flow with the testbench [24][25].

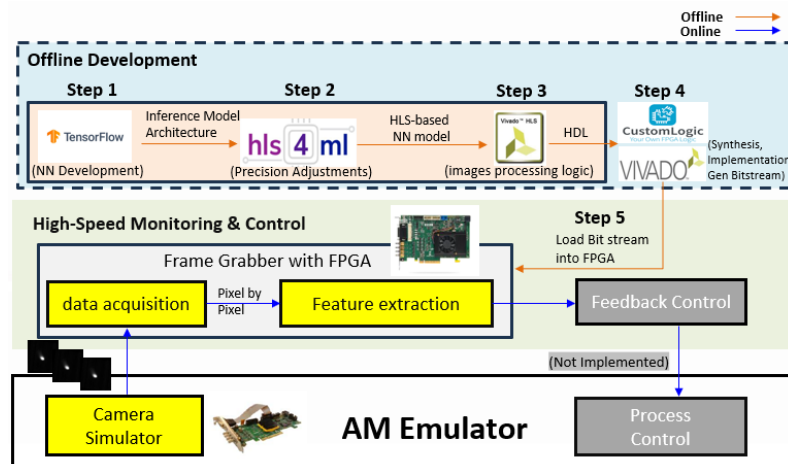


Figure 6. Development Flow with NIST AM Data Integration Testbench [16]

Top Section: Offline Development Flow (Steps 1–5)

Steps 1 and 2: Neural network development and HLS conversion using hls4ml are outside the scope of this paper. They involve training the model (e.g., in TensorFlow) and converting it to an HLS-compatible format with quantized precision for FPGA synthesis.

Step 3: Integrate image-processing and feature-extraction logic (e.g., overlap ratio, width) using Vivado HLS.

Step 4: Combine the processing logic into a complete HDL-based design and generate the bitstream using Vivado.

Step 5: Load the generated bitstream into the FPGA on the frame grabber for real-time testing.

6. Implementation and Validation (FPGA vs. PC)

To validate the values after implementation, we overwrite the last row of the image to store the calculated results. This does not affect the measurement, as the melt pool is in the middle of the image. As shown in Figure 7, we embed the melt-pool width in the last row of each MPI and embed overlapping features in the last row of the last MPI of each track.

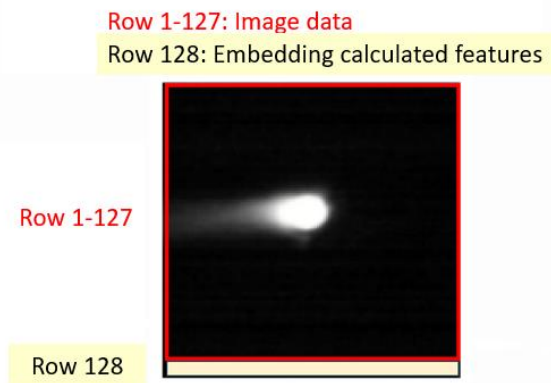


Figure 7. Melt Pool Image with Embedded Features in the Last Row

6.1 Embed Overlapping Features for S1 and S2

Figure 8 shows how overlapping features are embedded into each track's last row of the last image. In Scenario 1, this approach produces a comprehensive 116-dimensional feature vector per track pair, consisting of: the current melt pool width (position 1), statistical summaries including mean, max, and min overlap ratios (positions 2-4), and the complete overlap ratio vector for all 112 corresponding point pairs (positions 5-116). In Scenario 2, this method generates a compact 7-dimensional feature vector per track pair, consisting of: the current melt pool width, track-level melt pool width statistics including mean, max, and min values, and three aggregated overlap indicators - mean, max, and min overlap ratios derived from track-level statistics.

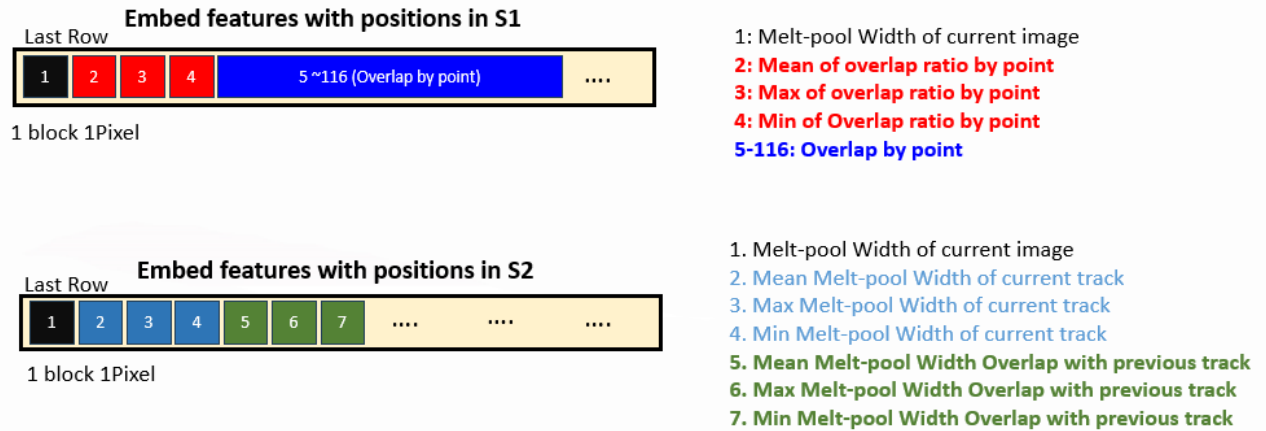


Figure 8. Feature Embedding Strategy for Track-End Image

6.2 Melt-Pool Width Validation Across 5 Tracks (FPGA vs. PC)

Figure 9 validates that the melt-pool width values calculated by the FPGA match the PC results across five consecutive tracks. In our case, the binary threshold value directly affects melt-pool width detection. If the threshold is set too high, it may lead to zero detected overlap. For this experiment, the threshold was set to 150. To optimize resource usage in the HLS implementation, integer floor division is used, whereas the PC-based implementation retains floating-point division.

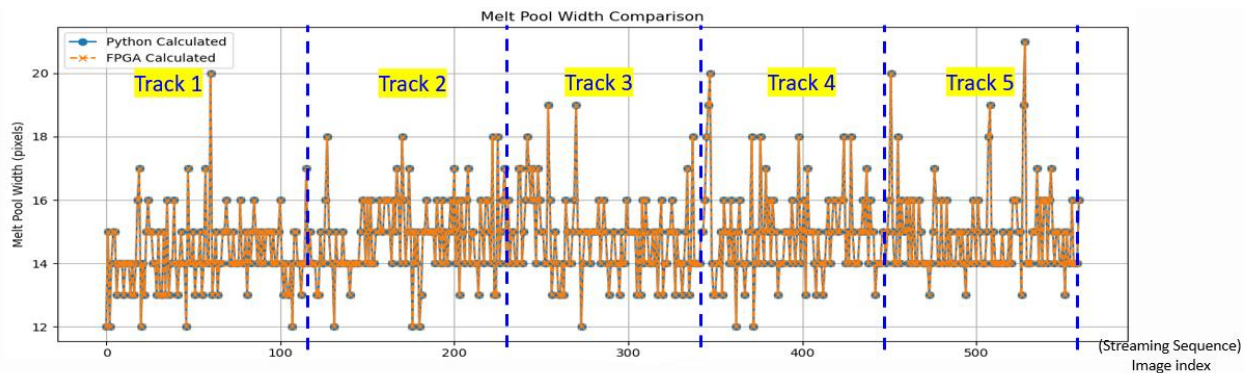


Figure 9. Melt Pool Width Validation across 5 Tracks

6.3 Validation of Spatially-Resolved Width-Based Fusion (FPGA vs. PC) – S1

Figure 10 confirms that pointwise overlap values calculated by the FPGA closely match the PC results. This validates the accuracy and reliability of FPGA-based pointwise overlap monitoring.

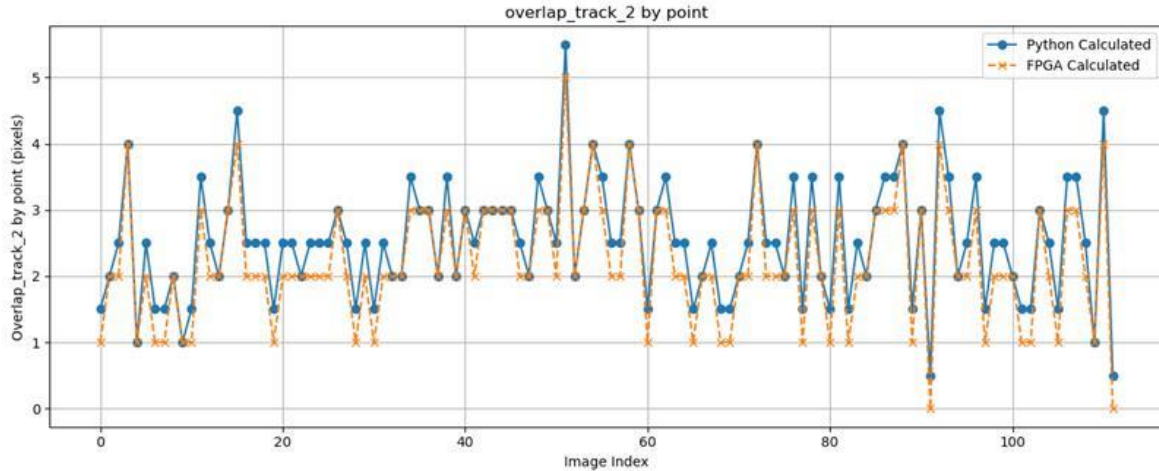


Figure 10. Overlap by Point between Track 1 and Track 2 – S1

Figure 11 shows that the mean, max, and min pointwise overlap values calculated by the FPGA closely match those from the PC across Tracks 2 to 5. At Track 2, the PC-calculated value is less than 1, which causes the FPGA result to be 0 due to integer flooring.

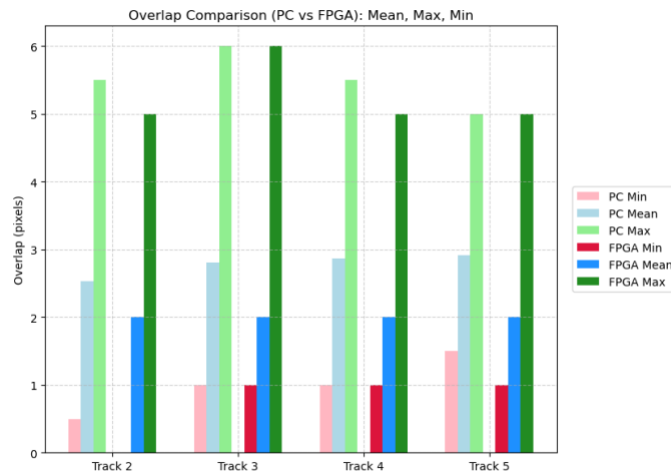


Figure 11. Pointwise Overlap Comparison across Tracks 2 to 5

6.4 Validation of Track-Level Feature-based Fusion (FPGA vs. PC) – S2

Figure 12 shows that the mean, max, and min melt pool widths (left) and overlap values (right) calculated by the FPGA are closely aligned with those from the PC across five tracks. The max and min values are based on integer pixel counts. The mean values involve division, resulting in slight differences due to using integer floor division in the FPGA and floating-point division in the PC.

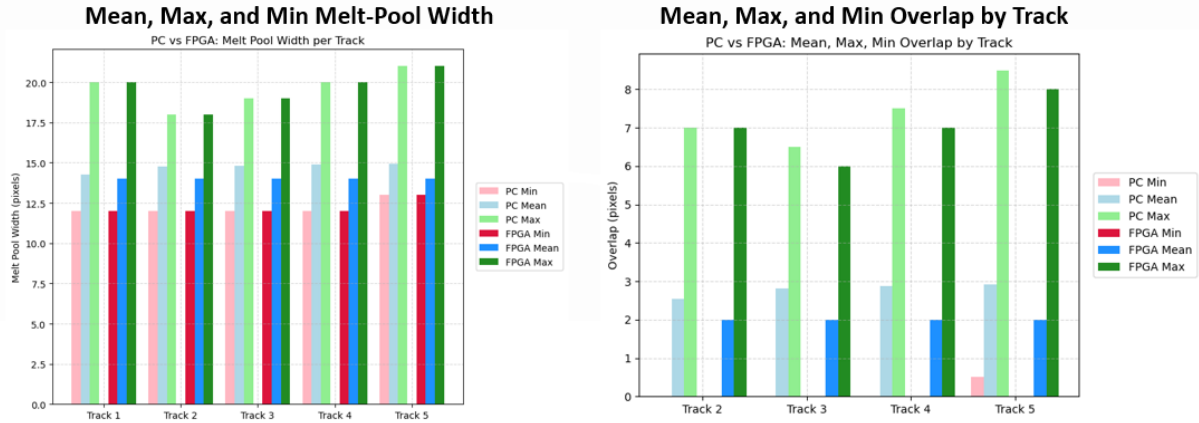


Figure 12. Track-wise Comparison of Melt-Pool Width and Overlap Statistics - S2

We have validated that the values calculated by the FPGA are correct and match those computed by the PC.

7. Performance Evaluation Results

7.1 Timing Evaluation for Design

Table 2 summarizes key timing indicators for FPGA synthesis. Setup and hold times ensure data is stable around clock edges, while Worst Negative Slack (WNS), Total Negative Slack (TNS), Worst Hold Slack (WHS), and Total Hold Slack (THS) measure slack margins. Good timing is achieved when all slack values are greater than or equal to 0 ns for totals.

Table 2. Timing Evaluation Indicators

Term	Meaning	Good Value
Setup	Data arrives before the clock edge	Slack > 0 ns
Hold	Data stays after the clock edge	Slack > 0 ns
WNS	Worst setup path slack	> 0 ns
TNS	Sum of all negative setup slacks	= 0 ns
WHS	Worst hold path slack	> 0 ns
THS	Sum of all negative hold slacks	= 0 ns

In both Scenarios, at a clock frequency of 250 MHz (corresponding to a 4 ns clock period), the design meets the required timing, meaning the slowest data path completes within the 4 ns window. As shown in Table 3. in S1, the design has a tighter margin, with 17 ps on the worst-case setup path and 30 ps on the worst-case hold path. In S2, the design achieves a timing margin of 93 ps on the worst-case setup path and 30 ps on the worst-case hold path.

Table 3. Timing Summary of S1 and S2 Implementations

Design Timing	Setup WNS	Setup TNS	Hold WNS	Hold THS
S1	0.017 ns	0 ns	0.3 ns	0 ns
S2	0.93 ns	0 ns	0.3 ns	0 ns

7.2 Resource Utilization

Table 4 lists key FPGA hardware resources and their purposes. It includes BRAM for on-chip memory storage, Digital Signal Processing Slices (DSP) for high-speed arithmetic, Flip-Flops (FFs) for sequential state retention, and Look-up Tables (LUTs) for implementing combinational logic. Efficient use of these resources is crucial for optimizing FPGA performance and area.

Table 4. Critical Resources in FPGA

Resource	Purpose
BRAM	On-chip memory blocks (36 Kb each) for buffering and data storage
DSP	Units for fast arithmetic operation
FF	A basic memory element used for storing 1-bit state for sequential logic
LUT	Implements combinational logic using precomputed truth tables

Table 5 shows the resource utilization after implementing both scenarios. The resource usage in S1, especially FFs and LUTs, is higher. However, both consume only a small percentage of the available FPGA resources. The table includes the total usage of both the original frame grabber's streaming functions and the custom processing logic.

Table 5. Resource Utilization Comparison between S1 and S2

Resource	BRAM	DSP	FF	LUT
S1	245 (Total: 45.37%) (Logic only: 10.37%)	178 (Total: 10.47%) (Logic only: 6.47%)	111,878 (Total: 27.54%) (Logic only: 11.54%)	79,375 (Total: 39.08%) (Logic only: 15.08%)
S2	244 (Total: 45.19%) (Logic only: 10.19 %)	178 (Total: 10.47%) (S2 only: 6.47%)	100,012 (Total: 24.62%) (Logic only: 8.62%)	66,763 (Total: 32.87%) (Logic only: 8.87%)

7.3 Estimate Latency Per Image

Initiation Interval (II) is the number of clock cycles between the start of loop iterations. II = 1 means a new iteration starts every clock cycle. We apply #pragma HLS pipeline II=1 to the pixel loop and confirm in the HLS report that II=1 is achieved in both S1 and S2. The FPGA efficiently handles the computations without stalling or buffering delays, as the overlap calculation is implemented without affecting the original streaming function.

Width calculation in both S1 and S2 and pointwise overlap calculation in S1 do not add additional time because they are processed in parallel with streaming and meet the 4 ns timing requirement. However, S1 and S2 require an additional 112 clock cycles for the mean overlap calculation at the final image of a track.

The latency for processing the final image of a track is around 66 μ s, as shown in Table 6, which is well below the 3 ms available between two tracks. The time required for the controller to adjust the laser power is under 50 μ s. This suggests that track-wise monitoring is feasible, and more complex computation tasks can be implemented to provide optimized laser power.

Table 6. Estimated Latency per Image of S1 and S2

Latency	Non-final image of a track	Final image of a track
S1&S2	16,384 clocks x 4 ns = 65.536 μ s	(16,384 + 112) clocks x 4 ns = 65.984 μ s

In Summary, both S1 and S2 calculations are simple for the FPGA, the designs meet timing requirements at 250 MHz, and resource utilization remains low.

8. Conclusion

Based on the correlation between melt pool width, overlap ratio, and their representation of lack of fusion or over-fusion defects affecting part quality, we have implemented and evaluated two FPGA-efficient overlap ratio calculation scenarios based on melt pool width: S1- Spatially Resolved Width-Based Fusion, S2 - Simplified Track-Level Feature-Based Fusion. We have validated that both functions can be implemented in the frame grabber. The FPGA-calculated values have been compared with PC-based calculations and show strong agreement. From a hardware perspective, both methods meet the required timing, operating at a 250 MHz clock, and maintain low resource utilization. From the track-wise process quality monitoring and control perspective, the overlap calculation is performed in line with image streaming. The required time for one image stream plus overlap calculation is less than 100 μ s. With low resource usage and latency, the system allows for implementing more complex algorithms. This includes decision-making for suggesting optimized laser power, neural network inference, or physics-based surrogate models for enhanced fidelity of real-time process monitoring.

Disclaimer and Acknowledgement

Certain commercial equipment, instruments, or materials identified in this paper are not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

The authors thank Dr. Ho Yeung for supporting and sharing information on the testbench development. We are also thankful for the support and insightful discussions Dr. Paul Witherell and Dr. Brandon Lane provided at NIST.

Reference

- [1] Totin, A., Macdonald, E., and Conner, B. Additive manufacturing for aerospace maintenance and sustainment. Defense Systems Information Analysis Center (DSIAC). Available: <https://dsiac.dtic.mil/articles/additive-manufacturing-for-aerospace-maintenance-and-sustainment/>
- [2] Godfrey, D. The future of flight: Creating an airworthy L-PBF component, SLM Solutions White Paper. Available: <https://www.slm-solutions.com/fileadmin/Content/Whitepaper/SLM-Solutions-Aerospace-future-of-flightv2.pdf>
- [3] Chowdhury, S., Yadaiah, N., Prakash, C., Ramakrishna, S., Dixit, S., Gupta, L.R., and Buddhi, D. Laser powder bed fusion: A state-of-the-art review of the technology, materials, properties & defects, and numerical modelling. *Journal of Materials Research and Technology*, 20 (2022): 2109–2172. <https://doi.org/10.1016/j.jmrt.2022.07.121>
- [4] Dejene, N.D., and Lemu, H.G. Current status and challenges of powder bed fusion-based metal additive manufacturing: Literature review. *Metals*, 13(2) (2023): 424. <https://doi.org/10.3390/met13020424>
- [5] NASA. MSFC-SPEC-3717: Standard specification for additive manufacturing. NASA Marshall Space Flight Center (2015). Available: <https://www.nasa.gov/wp-content/uploads/2015/04/msfcspec3717baseline.pdf?emrc=9ac06c>
- [6] Boulware, P. In-process monitoring techniques for laser powder bed fusion, *EWI Public Report*, Edison Welding Institute (EWI), Cincinnati (2017).
- [7] Yang, Z., et al. In-process data fusion for process monitoring and control of metal additive manufacturing, *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 85376 (2021) V002T02A072.
- [8] Shkoruta, A., Mishra, S., and Rock, S.J. Real-time image-based feedback control of laser powder bed fusion, *ASME Letters in Dynamic Systems and Control* 2(2) (2022) 021001.
- [9] Wang, R., et al. Real-time process monitoring and closed-loop control on laser power via a customized laser powder bed fusion platform, *Additive Manufacturing* 66 (2023) 103449.
- [10] Kavas, B., Balta, E.C., Tucker, M., Rupenyan, A., Lygeros, J., and Bambach, M. Layer-to-layer closed-loop feedback control application for inter-layer temperature stabilization in laser powder bed fusion, *Additive Manufacturing*, 78 (2023): 103847.
- [11] Hu, C.C., et al. A distributed AM architecture with digital twin for L-PBF cluster, *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)* (2024) 698–704.
- [12] Tran, H.C., et al. Intelligent additive manufacturing architecture for enhancing uniformity of surface roughness and mechanical properties of laser powder bed fusion components, *IEEE Transactions on Automation Science and Engineering* 20(4) (2022) 2527–2538.
- [13] Hong, G., et al. Multi-scale model predictive control for laser powder bed fusion additive manufacturing, *Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 88353 (2024) V02BT02A018.
- [14] Yeung, H., Hutchinson, K., and Lin, D. Design and implementation of laser powder bed fusion additive manufacturing testbed control software, (2021).
- [15] Yeung, H. Methodologies and implementation of laser powder-bed fusion process control, (2023).

- [16] Yang, C.W., et al. FPGA implementation for advanced real-time melt pool monitoring of laser powder bed fusion processes, *ASME IDETC-CIE* (2025) [Accepted].
- [17] Jaber, H., Kovacs, T., and Kónya, J. Investigating the impact of a selective laser melting process on Ti6Al4V alloy hybrid powders with spherical and irregular shapes, *Advances in Materials and Processing Technologies* 8(1) (2022) 715–731.
- [18] Yang, Z., et al. Analyzing remelting conditions based on in-situ melt pool data fusion for overhang building in powder bed fusion process, (2021).
- [19] Hinnebusch, S., Anderson, D., Bostan, B., and To, A.C. In-situ infrared camera monitoring for defect and anomaly detection in laser powder bed fusion: Calibration, data mapping, and feature extraction. *arXiv preprint arXiv:2407.12682* (2024).
- [20] Qu, M., Guo, Q., Escano, L.I., Nabaa, A., Hojjatzadeh, S.M.H., Young, Z.A., and Chen, L. Controlling process instability for defect-lean metal additive manufacturing. *Nature Communications* 13 (2022) 1079.
- [21] Yang, Z., et al. Enhancing part quality management using a holistic data fusion framework in metal powder bed fusion additive manufacturing, *Journal of Computing and Information Science in Engineering* 24(5) (2024).
- [22] Yang, Z., et al. Creating high-definition melting map in powder bed fusion additive manufacturing: Image-based vs. feature-based method, *ASME IDETC-CIE* (2025) [Accepted].
- [23] Lane, B., and Yeung, H. Process monitoring dataset from the additive manufacturing metrology testbed (AMMT): Overhang part X4, *Journal of Research of the National Institute of Standards and Technology* 125 (2020) 125027.
- [24] Yang, C., et al. Development of a testbench for additive manufacturing data integration, management, and analytics, *International Solid Freeform Fabrication Symposium* 34 (2023) 2166–2179.
- [25] Yang, C., et al. Integrating process data in motion for additive manufacturing industrialization, *International Solid Freeform Fabrication Symposium* 35 (2024) 585–596.