Title:
A Digital Twin-based Vision System for Layer-Level Defect Detection in Fused Deposition Modeling

Authors:
Kiet Q. Huynh, huynhk4@myumanitoba.ca, University of Manitoba
Qingjin Peng, Qingjin.Peng@umanitoba.ca, University of Manitoba

Keywords:
Digital Twin (DT), Additive Manufacturing (AM), Defect Detection, Computer Vision, Layer Monitoring, Fused Deposition Modeling (FDM)

DOI: 10.14733/cadconfP.2026.124-129

Introduction:
Fused Deposition Modeling (FDM) in Additive Manufacturing (AM) requires precise control of the thermal dynamics and deposition rates across layers during the 3D printing process [1, 11]. Conventional quality control techniques typically rely on post-production inspection, which results in material waste when defects are discovered late [2, 8, 9]. In-situ monitoring can achieve “zero defects,” such as using vision-based monitoring for real-time assessment of layer contours and surface conditions [2, 6]. Digital Twin (DT) technology enables this quality assurance method using a virtual model that synchronizes with its physical part in real time, enabling comprehensive monitoring and proactive analysis [1, 7]. Recent progress in this field is summarized in Tab. 1.

Despite these advances, significant limitations remain. Most DT systems focus on inactive monitoring or offline simulation, failing to provide meaningful feedback for immediate correction or real-time decision-making. For example, a pause after completion of each layer was introduced for inspection during the layer-by-layer scanning process, which hinders real-time monitoring and may affect the overall quality of the finished part [11]. The DT system proposed by Henson et al. requires simulating layer images for each camera before printing, and its detection classification remains simple and lacks details on defect type or severity level [3]. In addition, the developed DT-based system is computationally intensive, resulting in a lack of prompt feedback for closed-loop control [6]. Although detailed computational models can improve accuracy, their latency reduces effectiveness for preventing cumulative defects across layers.

These research gaps motivate the development of a real-time DT-based vision system that enables fully in-process layer-wise inspection by capturing and comparing layer images against expected ones. The system is designed to detect, categorize, and quantify different types of defects, such as over-extrusion, under-extrusion, and geometric distortion, during the 3D printing process without the need for pre-simulated images from multiple cameras. Furthermore, the image acquisition and processing time must be short enough to enable prompt autonomous interventions, ensuring high material integrity and geometric fidelity, and reducing waste. The main contributions of this work include: (1) a real-time synchronized Unity–Ender 3 Pro digital twin, (2) a layer-wise vision pipeline with object-centric deviation metrics for reliable multi-defect detection without the need for pre-generated reference layer images, and (3) experimental validation proving the system’s effectiveness in terms of latency and detection performance.

Main Idea:

Proposed method: A digital twin–based vision system is proposed to detect layer-level defects, including missing material, over/under extrusion, global geometry distortion, boundary shift, and major shape errors, by comparing layer images acquired from twin models in real time. The system is designed for real-time and modular synchronization, ensuring extensibility for full closed-loop control. It consists of hardware and software components with a data exchange and analysis pipeline, as shown in Fig. 1. The system includes a Creality Ender 3 Pro, 1080p Webcam, a host computer running Unity, and Python software. The digital printer model is built and connected to the physical printer. The printing operation is based on Marlin firmware and standard G-code commands. The camera is mounted on the top frame of the printer to monitor the print bed and capture layer images after each layer is completed. A single-host computer runs the Unity DT model and executes Python scripts for data communication, image processing, and network communication. The software components include the Unity engine, Python middleware, and a communication layer. The Unity engine provides a platform for building a virtual and synchronized printer model with efficient functions for real-time visualization of nozzle movement and extrusion. Python middleware serves as a data exchange and processing center. Multiple communication protocols are used in the communication layer to link different elements, including serial, WebSocket, and HTTP.

Extracted information from images	**Viewing angels**	**Type of defects**	**Method**	**Ref.**
Layer contours and edges, part's outer profile	Camera: top-down, ~8mm from the surface	Physical separation from the part to build the platform	Microscope + 3 DTs (plan inspection process, support edge detection, produce geometric reference)	[5]
Binary image matrices and pixels	3 cameras: left, right and front	Under/Over-extrusion	Prior printing: Simulate images layer by layer in the view of 3 cameras (DT). During printing: Capture images, convert them into pixel matrices, and compare with the corresponding simulated.	[3]
3D surface point clouds	Camera: top-down	Misaligned contour edges	Laser scanner + camera + DT (GUI)	[9]
	Camera: top-down	Major distortion detection	Point cloud + surface images	[10]
	Camera: top-down	Under/Over-extrusion	Compare the point cloud data of the inspected and normal ones	[11]
Temperature and extruding rate	Distance sensors: X, Y, Z axes Microscope: attached to the extruder head Thermal camera: top-down	Surface quality, abnormally fused beads	2D Camera (layer-wise image, overall surface quality), 2.5D laser scanner + fringe projection system (point cloud, surface parameters) System control based on comparison with DT	[5]
Image-derived features for deep learning	Distance sensors: X, Y, Z axes	Surface roughness (Ra)	Camera + CNN model (detect defects based on input images from the camera) + DT (GUI)	[2]

Tab. 1: Comparison of DT-based vision systems for monitoring and controlling FDM processes.

DT development in Unity: The 3D geometry of the Ender 3 Pro printer is represented in a STEP file created by a GitHub user [4]. The 3D model of the printer is converted to the OBJ format and imported into Unity using a built-in importer, with a scale factor of 0.001 to address the unit mismatch between meters and millimeters. There are three moving parts: the nozzle moves left and right (X-axis), the printer bed moves front and back (Y-axis), and the gantry moves top and bottom (Z-axis). However, the Y- and Z-axes are switched in Unity's default coordinate system. To replicate the physical appearance of all components, seven material types are created in Unity and assigned to the corresponding model components, such as a metal material for screws and the nozzle.

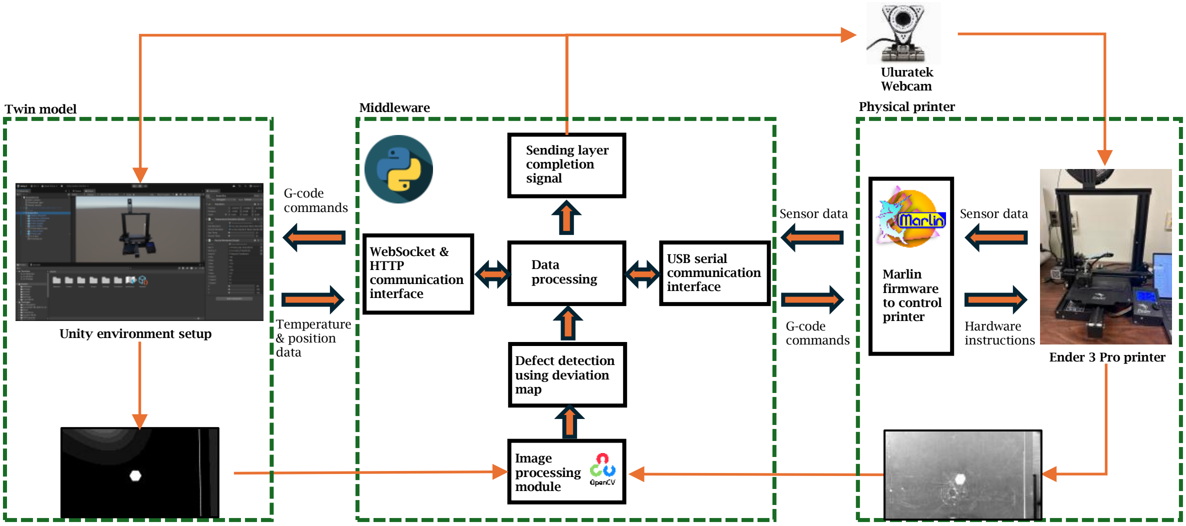

Fig. 1: Integrated Python-based control system for twin-based vision system.

To mirror the printing process, a digital twin of the Creality Ender 3 Pro is built in Unity, with critical components' motion controlled to replicate real-world printer behavior. The system converts G-code commands into equivalent virtual motions, ensuring that actual and simulated coordinate systems match. Motion is updated incrementally over time to provide smooth and consistent transitions. Furthermore, the feed rate retrieved from the G-code is used to control the movement speed in the simulation, allowing the virtual nozzle to follow trajectories at the same rates as the actual printer. This approach ensures that the digital twin accurately simulates both the kinematics and temporal behavior of the real printing process.

Data synchronization: As shown in Fig. 1, bidirectional data exchange is implemented among the components, enabling the DT model to continuously exchange data and accurately reflect the actual printing process. Python scripts initialize and manage the serial connection between Python and the actual printer, the WebSocket/HTTP communication between Python, and Unity for smooth data synchronization between the twin model. The communication pipeline consists of data flows of input G-code files for a desired geometry, stream from Python to both physical and virtual models simultaneously, and sensor data or command sending synchronized between twin models.

In the data flow, each line of the G-code file is sent to the printer via USB serial and to Unity via the WebSocket protocol. The WebSocket communication method provides continuous and full-duplex communication, low-latency connections, and high-frequency data updates, making it suitable for streaming commands to Unity in real time. The printer and Unity receive and execute G-code commands simultaneously, allowing the virtual nozzle to closely follow the real printer motion.

Sensor data, such as the current position, temperature, and printing progress, are requested by Python via the USB serial protocol. Python sends G-code commands and reads responses via a USB serial protocol at 115200 baud. These data are formatted according to the defined JSON structure and sent to the DT model via an HTTP connection while keeping the interactive monitoring interfaces in Unity informed. The DT model controls the printer by sending a G-code command in reverse. HTTP is a request-response protocol that is simple to implement but slower than WebSocket, making it reasonable for triggering specific G-code commands or fetching non-time-critical data.

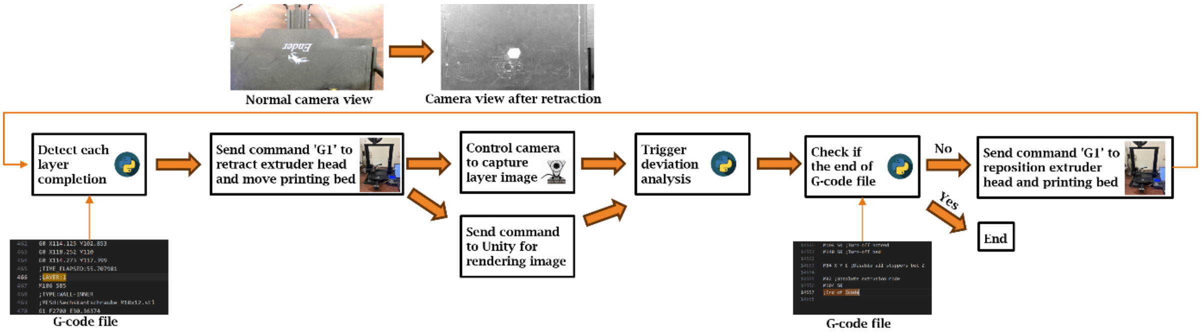

Fig. 2: Lay-by-layer image acquisition protocol integrating G-code-driven extruder control.

Image capturing and defect detection: Images are captured in the DT framework using a camera for each printed layer during the printing process, as shown in Fig. 2. These layer images are input into Python middleware through an automatic ROI extraction pipeline for processes, such as background subtraction, binary mask, object-centric cropping, and image normalizing, as illustrated in Fig. 3. To ensure consistency in pixel-wise comparisons and reduce sensitivity to camera setups and image cropping variability, the extracted object regions are resized and centered to a fixed resolution. Both objects are placed in the same coordinate frame, making the shape comparison reliable and robust. Deviation maps are computed from resized object-centric images to assess the quality of current n-layer printing for shape-based defects. The scale-dependent deviation is measured separately using the scale ratio by comparing the original bounding box dimensions. Both shape and size deviations are combined for decisions on defect types and severities, enabling process adjustments during printing.

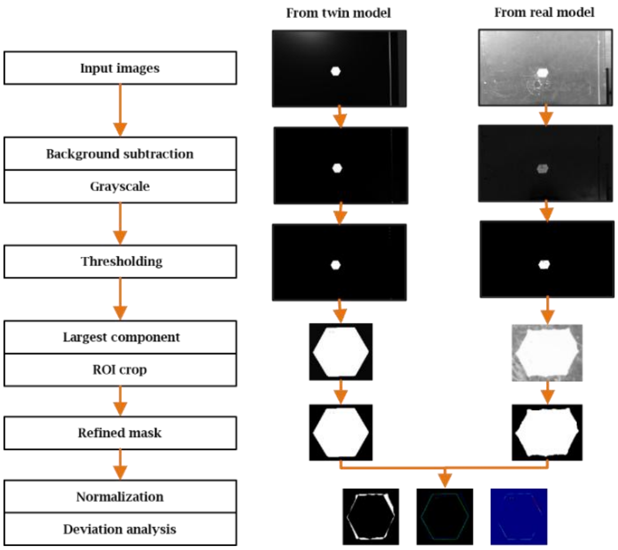

Fig. 3: Automatic region-of-interest (ROI) extraction pipeline for object-centric image comparison with the step-by-step process (left) and illustrations (right).

System evaluation: The implemented DT system conducted several experiments to evaluate the latency of real-time data synchronization and the position accuracy of the twin model. Compared to the physical printer, the virtual model shows lower latency on X- and Y-axis movements (180–250ms), while Z-axis movements exhibit considerable latency (2–3s) due to differences in feed rate transformation across the three axes. Despite the higher delay, the actual impact on overall system performance is limited, as Z-axis displacements per layer are relatively small and most motion occurs in the X-Y plane. Across 30 separate trials with different feed rates and moving distances on each axis, the DT system achieves high accuracy in position synchronization, with a deviation of 0 – 0.35mm. These results demonstrate real-time synchronization of motion and process data between the physical printer and its digital counterpart during continuous printing operations.

The defect-detection performance of the proposed system was evaluated in a series of controlled 3D-printing experiments, in which layer images extracted from twin models were processed using an automatic ROI extraction pipeline for object-centric comparison. A brief pause is introduced for the layer-wise image acquisition and processing procedure, approximately 2-3 seconds per layer, and the image comparison process accounts for a small portion of the total execution time, roughly 116 milliseconds per layer. This inspection process runs in near real time, layer by layer, as it is activated at layer transitions rather than continuously during extrusion. In typical FDM printing processes, where layer deposition times range from several seconds to ten seconds, depending on geometry and speed, the added delay is considered acceptable, but it remains a constraint for high-speed printing and will be improved in the future.

As shown in Tab. 2, outputs of the image comparison algorithm include shape metrics with the shape deviation ratio, IOU, and edge distance, and scale metrics including the area ratio and perimeter ratio. This combination of complementary shape- and scale-based metrics results in a more comprehensive defect taxonomy, allowing the system to distinguish between various defect types such as geometric distortion, extrusion and layer misalignment. For instance, the comparison of layer images in Fig. 3 shows over-1 area and perimeter ratios (1.2034 and 1.1299, respectively), indicating that the printed object is larger than expected due to over-extrusion or layer-shift from top-down view. Meanwhile, the IOU, deviation ratio, and edge distance are 0.8929, 0.0721, and 5.1063 pixels, indicating partial misalignment between the printed and expected shapes. This could appear as minor edge irregularities or boundary offsets, where the printed contour does not perfectly follow the intended geometry. Deviation maps and edge-distance heat maps provide an intuitive approach for defect localization, as illustrated in Fig. 3. Overall, the defect-detection performance is influenced by several factors, including the contrast between the object and bed colors, lighting conditions, and synchronization of camera positions between models.

<table>
<tr><th>Metric</th><th>Description</th><th>Assessment</th><th>Related defect types</th></tr>
<tr><td>Area ratio</td><td>Ratio of the pixel area of the object</td><td>Should be close to 1</td><td rowspan="2">Global geometry distortion, over/under extrusion, layer shift</td></tr>
<tr><td>Perimeter ratio</td><td>Ratio of the boundary length of the object</td><td>Should be close to 1</td></tr>
<tr><td>Intersection over Union (IOU)</td><td>Overlap ratio between two images</td><td>Should be close to 1</td><td rowspan="3">Layer shift, over/under extrusion</td></tr>
<tr><td>Deviation ratio</td><td>The fraction of pixels that differ between two images</td><td>Lower is better</td></tr>
<tr><td>Edge distance</td><td>Average distance between corresponding detected edge pixels</td><td>Lower is better</td></tr>
</table>

Tab. 2: Image-based comparison metrics used for defect detection.

Conclusions:

The proposed DT-based vision system represents a significant improvement in quality assurance for FDM processes. Across 30 trials, the system achieved real-time motion synchronization with 180–250 ms latency, positional accuracy within 0–0.35 mm (roughly 0.1 mm average), and efficient layer-wise inspection with ~116 ms processing time and only 2–3 s pause per layer. The proposed combination of shape-based and scale-based metrics enables robust detection of a wide range of defects, including geometric distortion, layer shift, and extrusion faults. The DT vision system acts as a live navigation assistant, constantly comparing the intended route (the expected printing simulated in Unity) with the actual position (the printed layer) and sending an alert when the process takes incorrect actions. This system enables real-time tuning of printing parameters based on feedback, preventing defects and ensuring the quality of the printed product.

Looking ahead, this work lays a strong foundation for the further development of closed-loop control with proactive correction during the printing process, based on real-time evaluation of material integrity and geometric consistency. This research solution will be enhanced by using machine learning models for defect detection and parameter optimization before and during the printing process. The adaptations in this work will ensure geometric accuracy and reduce material, time, and cost, thereby increasing the overall efficiency and performance of additive manufacturing.

Kiet Q. Huynh, https://orcid.org/0009-0001-7868-1067
Qingjin Peng, https://orcid.org/0000-0002-9664-5326

References:

[1] Anand, S.; Satyarthi, M. K.; Bharti, P. S.; Kumar, P.; Kumar, A.: Digital Twin Integration for Enhanced Control in FDM 3D Printing, Industry 4.0 Driven Manufacturing Technologies, Springer Nature Switzerland, Cham, 2024, pp. 373–388. https://doi.org/10.1007/978-3-031-68271-1_17

[2] Ali, M. H.; Malik, A. W.; Jyeniskhan, N.; Mahmood, M. A.; Shehab, E.; Liou, F.: Development of Digital Twin for FDM Printer With Preventive Cyber-Attack and Control Algorithms, IEEE Access, 12, 2024, 193594–193606. https://doi.org/10.1109/ACCESS.2024.3516827

[3] Henson, C. M.; Decker, N. I.; Huang, Q.: A Digital Twin Strategy for Major Failure Detection in Fused Deposition Modeling Processes, Procedia Manufacturing, 53, 2021, 359–367. https://doi.org/10.1016/j.promfg.2021.06.039

[4] Hyotynen, T.: Ender 3 Pro 3D Printer Model, GitHub repository, GitHub, Online, 2020. https://github.com/hyotynen/Ender-3-Pro

[5] Love, A.; Behseresht, S.; Valdez Pastrana, O. A.; Sakai, J.; Park, Y. H.: Hybrid Physical-Virtual Digital Twin System for Additive Manufacturing, Journal of Advanced Manufacturing Systems, 24(01), 2025, 1–20. https://doi.org/10.1142/S0219686725500015

[6] Moretti, M.; Rossi, A.; Senin, N.: In-Process Monitoring of Part Geometry in Fused Filament Fabrication Using Computer Vision and Digital Twins, Additive Manufacturing, 37, 2021, 101609. https://doi.org/10.1016/j.addma.2020.101609

[7] Mourtzis, D.; Varsamis, A.; Zygomalas, S.; Angelopoulos, J.: Adaptive Control of 3D Printer Based on Digital Twin Concept, Innovative Intelligent Industrial Production and Logistics, Springer Nature Switzerland, Cham, 2023, pp. 3–17. https://doi.org/10.1007/978-3-031-49339-3_1

[8] Shomenov, K.; Ali, M. H.; Jyeniskhan, N.; Al-Ashaab, A.; Shehab, E.: Cost-Effective Sensor-Based Digital Twin for Fused Deposition Modeling 3D Printers, International Journal of Computer Integrated Manufacturing, 0(0), 2025, 1–20. https://doi.org/10.1080/0951192X.2025.2504085

[9] Xu, C.; et al.: Digital Twins for Defect Detection in FDM 3D Printing Process, Machines, 13(6), 2025, 448–463. https://doi.org/10.3390/machines13060448

[10] Yin, K.; Lou, S.; Qin, Y.; Xu, Y.; Scott, P.; Jiang, X.: A Digital Twin Framework of In-Line Process Optimisation for Material Extrusion-Based Additive Manufacturing, Proceedings of the UNIfied Conference of DAMAS, IncoME and TEPEN Conferences (UNIfied 2023), Springer Nature Switzerland, Cham, 2024, pp. 777–786. https://doi.org/10.1007/978-3-031-49421-5_63

[11] Ziadia, A.; Habibi, M.; Kelouwani, S.: Digital Twin-Driven Real-Time Optimization of Layer-Specific Surface Roughness in FDM 3D Printing, Progress in Additive Manufacturing, 10(11), 2025, 10053–10076. https://doi.org/10.1007/s40964-025-01227-w