DEVELOPING IN-SITU PROCESS MONITORING CAPABILITIES FOR MATERIAL EXTRUSION ADDITIVE MANUFACTURING

Zachary Renda, Jan Petrich, Callie Zawaski, and Joseph Bartolai
Applied Research Laboratory
Penn State University
University Park, PA

Abstract

A machine agnostic framework for in-situ data collection during Material Extrusion (MEX) Additive Manufacturing (AM) builds with user-defined anomaly tagging is presented. To enable the use of Machine Learning (ML) algorithms for detection and identification of MEXAM build anomalies, a large set of training and test data is required. The tagging framework is integrated into a data collection system that includes infrared imaging, visible light imaging, accelerometer data, homography-based telemetry data, temperature, and environmental data. This data is registered both in time and 3D space, allowing the build anomaly data to be traced to a specific location on the as-built part. The presented framework allows users to create a database by identifying anomalies during a MEXAM build and automatically marks data around the anomaly time step across all collected sensor modalities. This tagged data can then be used as ground truth for ML training and validation.

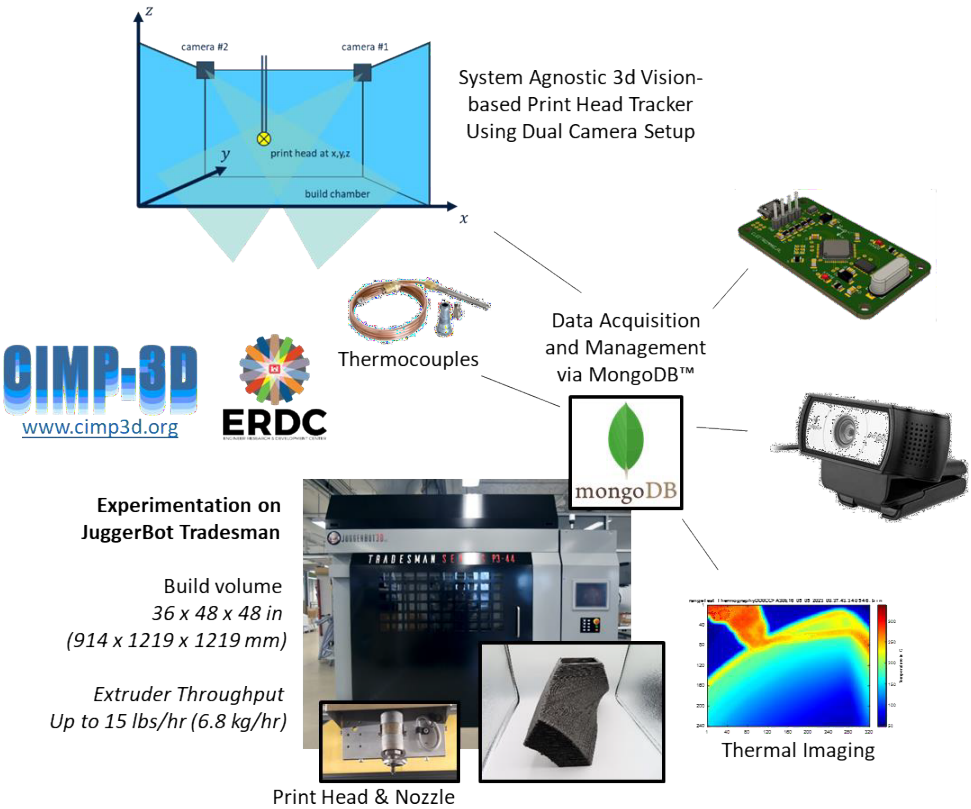

FIGURE 1. A GRAPHICAL ABSTRACT OF THE DEVELOPED MEXAM MONITORING FRAMEWORK.

1. - Background

Parts manufactured through polymer MEXAM are often limited by low reproducibility and inconsistencies due to print anomalies and failures that are caused by the low stability of the manufacturing process [1]. Instabilities are often caused by changes in processing conditions, such as nozzle temperature and pressure [2]. Therefore, monitoring these factors during the print process is critical in order to detect, qualify, prevent, and/or mitigate the defects and failures in situ [1]. Nozzle temperature inconsistency can be caused by asymmetrical cartridge heaters used to melt the polymer during the extrusion process [2] as well as degradation of thermocouples used in the system's temperature regulation. Inconsistency in this temperature affects the polymer melting process and the initial weld contact temperature at the layer-to-layer interface which in turn causes inconsistent weld strengths throughout the part [6]. Nozzle pressure inconsistency can be caused by filament buckling and annular backflow during the deposition process [2,3] as well as material degradation and blockages in the nozzle. During annular backflow, molten filament flows back up between the solid filament and the heated nozzle wall and subsequently cools back into a solid [3], which in turn affects the flow rate of the polymer melt [2]. Environmental factors are also responsible for the inconsistency of polymer MEXAM processing. The environmental temperature of the manufacturing chamber affects the cooling rate of the polymer, impacting the final part properties such as inter-layer weld strength [4]. Humidity inside the chamber affects the filament diameter as the filament absorbs moisture from the air and swells [5] and the absorbed water can also then boil and bubble within the extruding material. The swelling and boiling changes the diameter of the extruding material, the nozzle pressure and ultimately the polymer flow rate [3]. Nozzle and print bed positional control also often exhibit some variability and error during the printing process which will cause the part geometry, tool pathing, and final mechanical properties to skew from desired values. Understanding these factors and the underlying physics and capturing them during process monitoring facilitates a real-time assessment of the process and provides indicators for the quality and properties of the final part. Currently, part properties are estimated through destructive testing of test specimens developed in the same orientation as the loaded area of the final part, such as through dog-bone tensile testing of polymer MEXAM printed specimens [5].

1.1 - Thermal Gradients

Throughout the polymer MEXAM process, different areas of the manufactured part experience varying thermal gradients which in turn have discernable and measurable effects on the interlayer weld-strength at these locations [6]. The influx and distribution of thermal energy is primarily dictated by the part geometry as well as the chosen tool path for printing. These two qualities may be modified to minimize the impact of thermal gradients onto part quality once an understanding of their impacts has been developed. Traditional manufacturing methods, such as injection molding, are near isothermal which causes the part strength to be near isotropic [6]. However, polymer MEXAM is highly non-isothermal [6] and varies throughout part locations based on environmental temperature [4], nozzle temperature, deposition rate [2], and tool-pathing [7] which causes variations in nozzle speed and location over time. This variation in thermal gradients throughout the part's layer interfaces affects the polymer chain reptation across deposition boundaries [6]. The polymer welding process begins with surface wetting, then polymer chains from the two layers begin reptating across the two surface interfaces, and then

finally the polymer chains begin to become entangled forming the final polymer-polymer weld [6]. This process has a strong, measurable time-temperature relationship governed by the temperature profile and cooling rate of the interface location [6]. By measuring and understanding the thermal history of a polymer MEXAM part during the manufacturing process, some researchers are studying the associated weld strengths throughout the part [6] in order to predict part properties without requiring testing to be performed after manufacturing. This is just another step on the way to the "born-certified" part: parts produced and certified based on known mechanical properties.

1.2 - Machine Learning

Machine learning (ML) enables systems that are monitoring complex datasets, like those found while monitoring an additive manufacturing (AM) process, to create better and more useful correlations of the information collected. In general, ML model training can be broken down into three major training categories [8]. The three major training categories are supervised, unsupervised, and semi-supervised learning. Supervised learning trains an ML model with pre-labeled data of known classifications or results. In the context of MEXAM, data may be labeled as nominal or anomalous by a human operator. For anomalous data, a classification may be added such as over/under extrusion. Unsupervised learning trains an ML model with data that has not been analyzed or labeled, allowing the system itself to discover any hidden patterns or anomalies. In the context of MEXAM, unsupervised learning could occur for time-series data such as nozzle temperature, and deviations from nominal could be flagged without explicit labels provided. Semi-supervised learning is a combination of supervised and unsupervised training techniques that can be broken down into three major methods itself: active, passive, and self-training. All semi-supervised training starts with some pre-labeled and some unlabeled data that allows the ML model to help train itself. Active learning models are able to query human experts for help labeling some key data points to help them formulate their classifications. Passive learning models just use the data given and do not actively seek out new information for additional training. Finally, self-learning models retrain themselves iteratively over the unlabeled data to form better classification methods over time. These semi-supervised training methods are useful for large and high dimensional data sets that require costly experiments and for scenarios in which labelling is not feasible or practical.

ML models are designed to complete one of 4 major tasks that allow researchers to derive improved predictions and/or understandings about training data and subsequent experimentation data. These tasks are regression, classification, clustering, or dimensionality reduction [8]. A regression model is used to predict quantitative outcomes. In process monitoring for AM, this includes predicting numerical mechanical properties like weld strength. A classification model is used to assign data into predefined classes based on known patterns. In process monitoring for AM, this relates to detecting defects, anomalies, and part quality. A clustering model is used to categorize data into groups based on detected patterns. In process monitoring for AM, this includes defect detection as well as modes of part failure. Finally, a dimensionality reduction model decreases its data dimensionality while maintaining a similar level of output variation and control by identifying essential input features in order to increase its efficiency and interpretability. In process monitoring for AM, this reflects to investigating parameter-property

correlation and determining manufacturing formulas. Decision making of an ML model can be categorized as over-fit, under-fit, or well-fit [13]. An over-fit model is defined as a model that fits its input training data well based on the value of the cost function used for training. However, the model underperforms when exposed to new test and validation data. An over-fit model is called “optimistically-biased” because its training set accuracy is much higher than its validation accuracy. Generally, an assumption is made that the training data is representative of the validation and test data. An over-fit model is indicative that this assumption is violated. An under-fit model poorly fits its training data and also underperforms at predicting new scenarios in the validation and test data. The goal of an ML model is to develop well-fit data that uses enough training data and constraints to make accurate predictions for previously unseen data [13]. Obtaining representative and sufficient training data is often the key to achieve good generalizability of the ML model. ML may provide the last step to “born-certified” parts: an automated system of analyzing manufacturing data to determining part quality and characteristics.

2. - Technical Approach

Our technical approach towards in-situ monitoring of MEXAM processes combines sensors and instrumentation, data acquisition and multiprocessing, data management, as well as human-machine interaction. We outline our sensor suite and instrumentation strategy in Section 3.1, while Section 3.2 discusses data acquisition and multiprocessing of sensor data. The data management approach is summarized in Section 3.3, and details of the human-machine interface that can be used to generate labeled data is provided in Section 3.4.

2.1 - Sensors and Instrumentation

This project is investigating the use of a number of common sensors used in AM process monitoring concurrently: thermal cameras, a 3-axis accelerometer, optical cameras, thermocouples, and a chamber environment sensor measuring temperature and humidity. The graphical abstract in Figure 1 shows an overview of the sensors and methods being used.

The sensors are all mounted and tested in the Juggerbot Tradesman Series P3-44. The Juggerbot Tradesman machine is a medium-scale pellet-fed polymer MEXAM system. This machine’s built plate is 1219 x 914 mm, and its extrusion nozzle diameter is 3 mm.

The thermal cameras are two SEEK Thermal S314SPX cameras from the Mosaic Core series. The cameras output a 320x240 floating point array of temperature values in °C and are accurate between -40°C to 330°C with a frame rate of up to 27 Hz. One thermal camera is positioned on the deposition head pointed down at the nozzle to view the nozzle and the material at the point of material deposition. This thermal data is useful for tracking the thermal history of weld contacts to determine temperature fluctuations, nozzle temperature variations, and weld strengths. The other thermal camera is located on the build chamber wall pointed at the entirety of the build plate to track the thermal history and cooling rate of the part as a whole. Figure 2 shows the thermal camera captures from an active polymer MEXAM process.

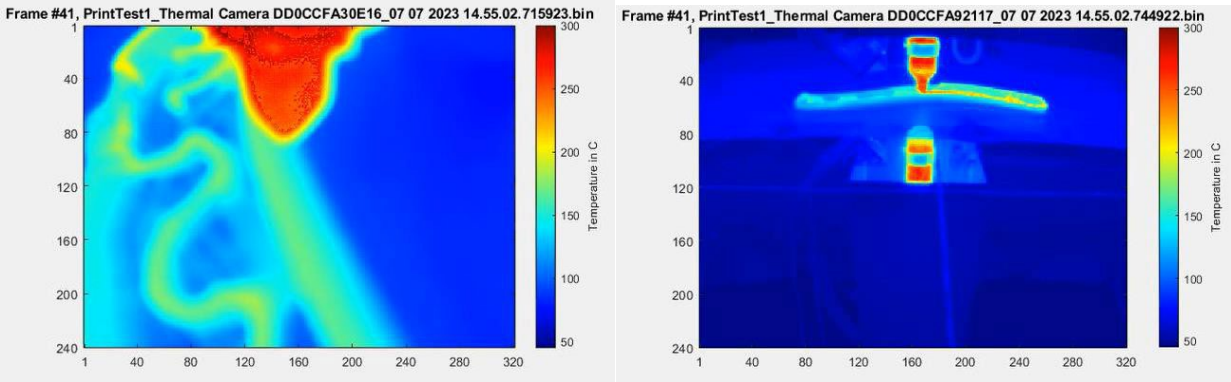

FIGURE 2. SEEK THERMAL CAMERA IMAGES TAKEN DURING A LIVE POLYMER MEXAM PRINT. (LEFT) NOZZLE MOUNTED CAMERA. (RIGHT) PROFILE POSITIONED CAMERA.

The 3-axis accelerometer is the ADXL345 sensor with a data-capture rate of 3200 Hz. The acceleration range is ±2 g with a 10 bit resolution. The accelerometer is mounted on the deposition head and monitors vibrations and movements, such as direction changes, decelerations and accelerations primarily in horizontal XY plane.

Two optical cameras are located at the top-front left and right corners of the build camber and are pointed towards the build area. The current project is utilizing two Logitech C930e USB web cameras with a 90° field of view and a 1920 x 1080 pixel high definition resolution at 30 fps. The optical cameras are used for visual inspection of the manufacturing process as well as for tracking the 3D position of the nozzle and build plates.

Thermocouples are used for thermal camera accuracy confirmation as well as interlayer material temperature tracking. For our setup, all thermocouples communicate through an external DI-2008 DATAQ data acquisition (DAQ) system with a data input framerate of about 2000 Hz for one thermocouple and 200 Hz for more than one with a 16 bit resolution.

Finally the chamber humidity and temperature is being monitored through a DATAQ EL-USB-2+ environmental sensor. This sensor has a temperature range of -35 to +80° and a humidity range of 0 to 100% RH.

2.2 - Vision-based Print Head Tracking

The 3D position of the nozzle head and build plate are determined using a dual-camera computer vision tracking system. This allows for the real-time estimation of nozzle and build plate position. The approach and sensor configuration are system agnostic, and the software is written in Python using the OpenCV library [17]. Real-time imagery is acquired through USB connected high-definition webcams. The system's original tracking method was based on bright-light, circular, colored LED array identification. This system would have an HSV color range that it filtered images for. This range would be determined experimentally based on the color LED array being used. Once the image was filtered the remaining white pixels on a black background would be eroded and then dilated to help eliminate noise. Then the image was broken down into contours using a built-in OpenCV function, and only the largest contour would be kept. Finally, a minimum enclosing ellipse would be determined for the contour using another built-in function, and its center point would be saved as the position point for the nozzle tracker

within the camera frame. This center point ideally represented the real-world center of the circular LED array. Later forms of image filtering involved using a Gaussian Mixture Model (GMM) to determine the brightest input region of the image matching that of the LED array's light and iteratively improve the threshold for this filtering by using continuously more frames as the experiment goes on. The tracking method is based on detecting ArUco markers within the image frames captured. An ArUco marker is a computer-generated visually identifiable image comprised on a thick white border and an inner black-and-white binary matrix [16], similar to a QR code. Figure 3 shows examples of ArUco markers.

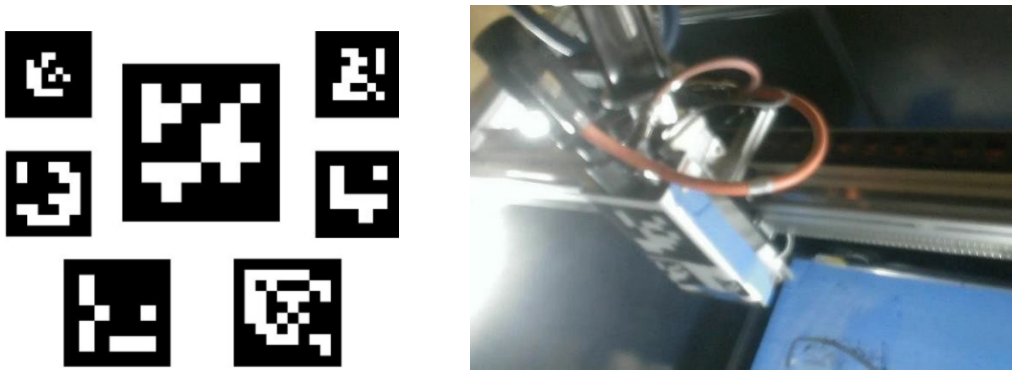

FIGURE 3. (LEFT) EXAMPLES OF ARUCO MARKERS USED FOR COMPUTER VISION IMAGE INSPECTION [16]. (RIGHT) DIFFERENT SIZED ARUCO MARKERS BEING TESTED INSIDE THE JUGGERBOT TRADESMAN.

Using built-in OpenCV functions [17], the system recognizes and tracks the ArUco marker in the image frame. In addition to location, the system also estimates relative orientation using the four corners of the marker. The system then records the lower left corner as the position of the marker in the image frame. During camera calibration, homography arrays can be calculated from a set of known calibration points that allow for the position of the marker within each camera's frame to be mapped to its real-world 3D position. Figure 4 is a visualization of the projection of a 2D position to the two different camera frames and vice versa. By using and distinguishing different markers, one for the nozzle and one for the build plate, for a machine that has both a moving nozzle (mainly horizontally in *xy*) and a moving build plate (mainly vertically in *z*), this 3D position can be determined for both moving systems simultaneously in each frame. For the dual-camera telemetry, the homography arrays calculated during the calibration procedure for each camera is stored in the dual-camera telemetry collection to be used to process collected marker data into xyz positions. The tracking system was calibrated and tested using the thirteen 2D position point pattern shown in Figure 5 on the Juggerbot Tradesman Series P3-44. The cameras were located about 800 mm above the nozzle and in the front-top-left and front-top-right corners of the machine. The cameras were angled to view the largest amount of the nozzle movement range as possible. The calibration points tested were a 3 x 3 grid spaced 250 mm apart. The 4 diagonal points between the center point and corners served as validation points.

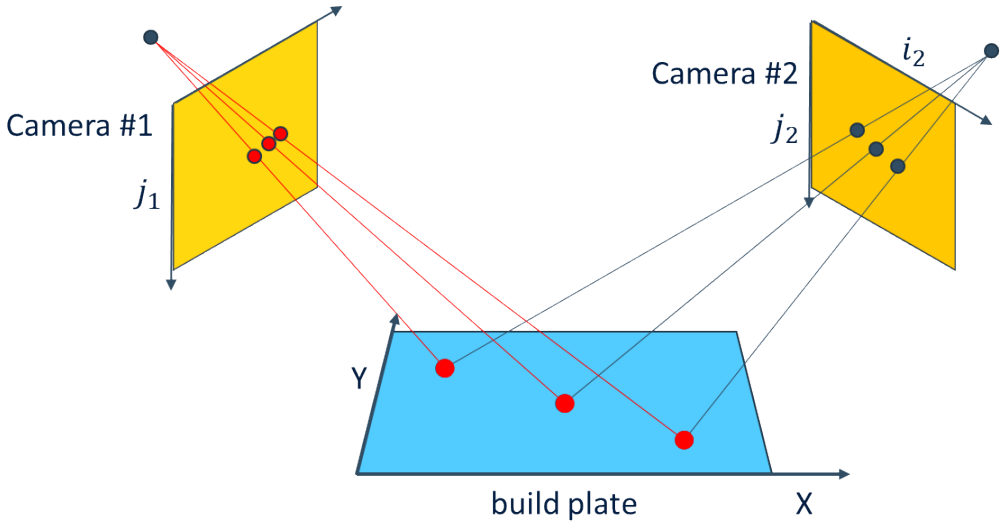

FIGURE 4. REAL-WORLD 2D POINT TO DIGITAL CAMERA 2D FRAME POSITION CALIBRATION USED FOR DEVELOPING HOMOGRAPHY ARRAYS THAT ALLOW FOR DUAL-CAMERA, COMPUTER VISION TELEMETRY.

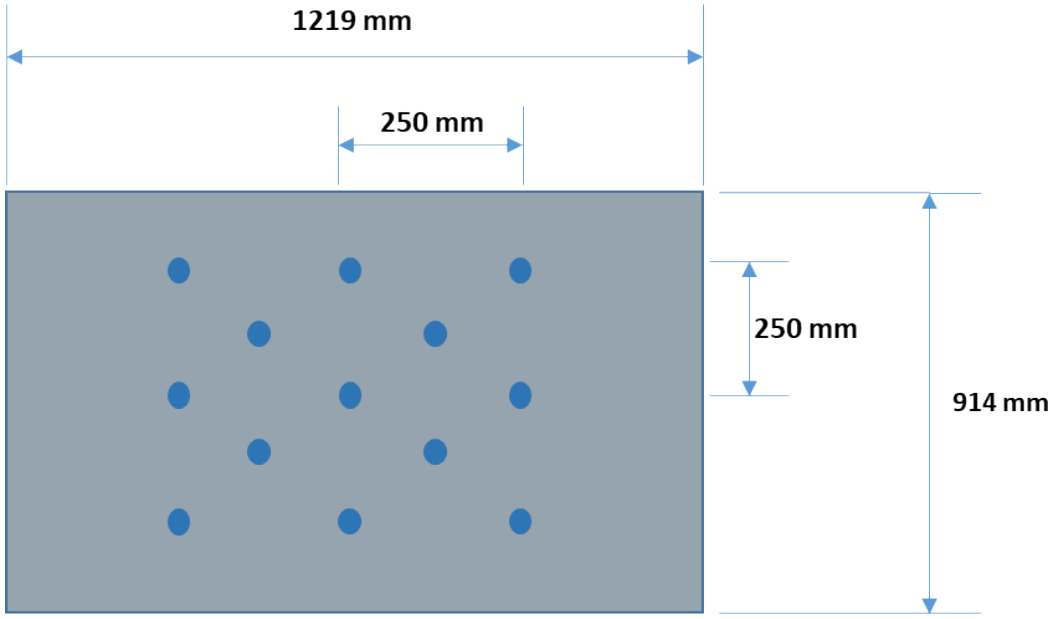

FIGURE 5. THIRTEEN POINT 2D DUAL CAMERA NOZZLE TRACKING CALIBRATION GRID.

2.3 - Multiprocessing

In order to effectively and efficiently collect data input from multiple sensors with different data acquisition rates by a single computer system, multiprocessing and multithreading have been implemented into the DAQ methodology. Multithreading and multiprocessing are often used together and get confused but their distinction is important to the design and operation of the DAQ program [9]:

- Multithreading is the concurrent running of multiple threads.
- Multiprocessing is the parallel running of multiple processes.

- A thread is the tied connection between events within a process.
- A process is the execution of a computer program [10].
- Concurrency means the threads are executing in an interweaving pattern where one thread will start running while waiting for the results of another to compute, but two threads will not run at the exact same time [9].
- Parallel means that the process tasks execute simultaneously across multiple processors whenever available.

Multiprocessing is the primary approach for this data collection approach because it allows for the process task of calling to a sensor and reading the data sent back to occur for each sensor simultaneously. Therefore, any sensors that have slower data-capture rates like the thermal cameras will not slow down the data-capture rate of the fast sensors, like the accelerometer, because these tasks are being performed on separate processors. The proposed DAQ strategy also offers maximal extensibility and flexibility so that the framework can seamlessly scale if the deployment of additional sensor is desired. Data acquisition via multithreading and multiprocessing has been implemented using the multiprocessing and threading Python libraries.

2.4 - Human Machine Interface - UIET

It is imperative to allow human operators to tag the data in real-time in order to support data labeling and to manually indicate process anomalies. To accommodate this need, our team has augmented the data acquisition and management system with a program we named UIET-User-Input Error Tracking. UIET is not just the collection of the sensor control and data management software, but it also represents the overall goal of the data collection protocol: flagging of and identifying process anomalies. The UIET program runs alongside the data recording process and allows the user to monitor and flag data for any errors that are observed by or purposefully inserted by the human operator, such as over or under extrusion. It takes in two key fields and automatically marks the timespan for 5 seconds before and after the error is marked, and this timespan can be adjusted during post-process analysis. The main goal is to enable to retrieval of anomalous data snippets from all sensors after build completion. The first input field is the error tag. This drop-down menu represents commonly and repetitively occurring failure modes that occur during the polymer MEXAM process such as over and under extrusion, and it allows for rapid grouping of similar data during the post-process analysis step. In other words, the failure tag indicates the classification of the process anomalies that can be utilized for supervised ML. The other input field is for notes, and it allows the user to add any important information that was observed during the error occurrence that might be useful during the analysis step. Both fields are able to be adjusted after the recording process has finished, and all of the error flags are collected as documents within the experiment database inside a collection titled “UIET.” Figure 6 shows the UIET user interface. All data visualization and error flagging is performed parallel to the sensor data collection processes so as to prevent any user interference with the data collection process. The UIET collection is the key takeaway from the data analysis and processing steps and can be used by researchers not only for building understandings of failure modes that occur during the polymer MEXAM process but also for developing training data libraries for developing and testing ML models.

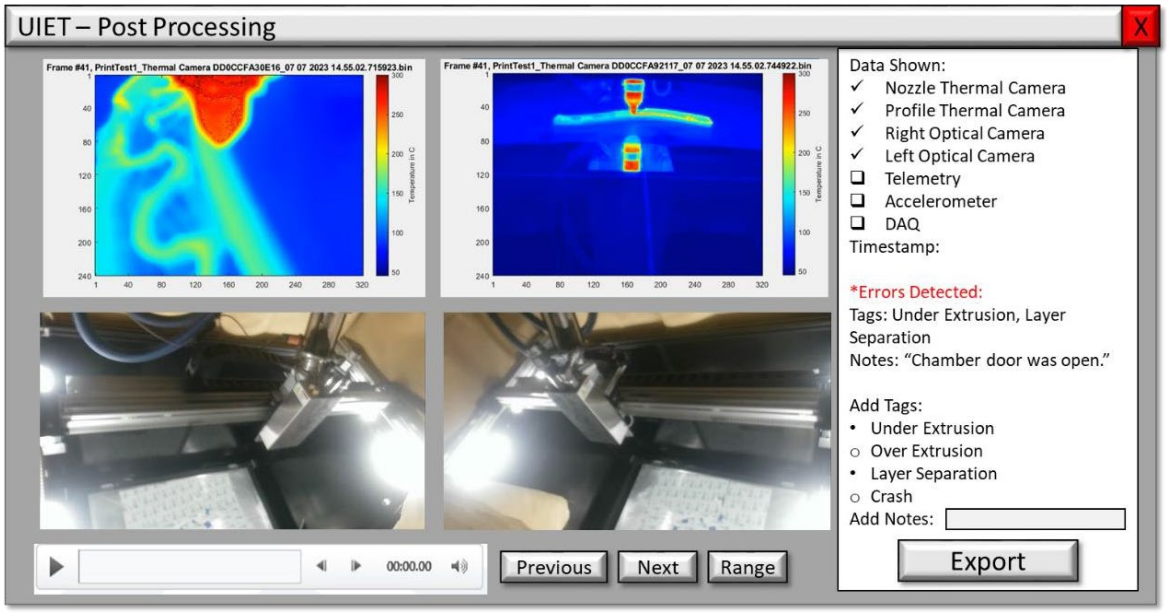

FIGURE 6. UIET USER INTERFACE USED FOR IN-SITU ERROR FLAGGING AND POST-PROCESS DATA ANALYSIS.

2.5 - Data Management

All data captured and recorded directly from the connected sensors are processed then packaged into a BSON (binary JSON) object document in a MongoDB hosted database [15]. For the given application, MongoDB offers many benefits such as object-based data storage, built-in compression capabilities through WiredTiger, and an open-source Python Library (PyMongo). Documents contain field-value pairs to organize the data stored within themselves. These documents are stored in collections which are grouped together in databases. The overall data structure of the sensor data management can be broken down into experiments, sensors, and timestamped data frames. Each experiment (or 3D print) is stored in its own database on a given computer or server. Each collection within the given experiment database corresponds to an individual sensor and/or processed data type as is in the case of the xyz telemetry data calculated from the dual-camera homography techniques discussed previously. Each document within a collection corresponds to a timestamped data frame where the "Data" field pairs to the data frame and the "Timestamp" field corresponds to a Python generated timestamp. Exceptions to these general rules are documents and collections stored for data processing and analysis, such as error flagging and sensor calibration. By timestamping all of the data that is captured with the same system, events can be tracked by time of occurrence or by position by determining the time in which the nozzle was depositing at this location. Figure 7 visualizes the database setup being used. A large amount of data is collected however, so a way for managing and tagging events of interest, i.e. labeling the data, is necessary.

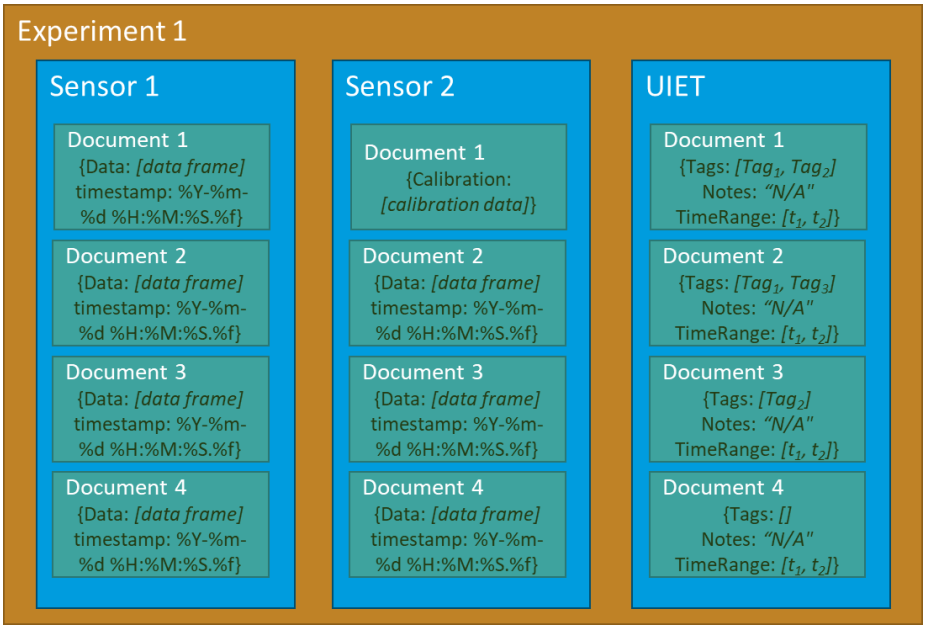

FIGURE 7. EXPERIMENTATION DATABASE SETUP INCLUDING A "UIET" COLLECTION FOR DATA PROCESSING (SECTION 3.4).

3. - Results

All framerates fall within 10% of their expected values except the thermal cameras. The thermal cameras are not programmed parallel to each other and therefor interfere with each other's capture rates. Experimentally the frame rates fall at about 24 Hz alone and 16 Hz when two cameras are used together, an 11% and 41% decrease respectively.

TABLE 1. RMS ERROR CALCULATIONS FOR COLOR-FILTERED LED ARRAY TRACKING CALIBRATION USING DUAL-CAMERA TELEMETRY.

	Full 13 points (RMS Total Error)	9-Point Grid (RMS Grid Error)	9-Point Grid (RMS Diagonal Error)	9-Point Grid (RMS Total Error)
Left Camera (pixels)	8.5	7.7	3.6	8.8
Right Camera (pixels)	7.0	6.2	2.1	6.6
Juggerbot (mm)	4.14	3.82	1.80	4.24

Table 1 shows the results calculated for the original experimentation using color-filtered LED array nozzle tracking. Although the accuracy was significant compared to the total distance traveled by the deposition nozzle, the total error was about 4 mm for both the nine point and thirteen point calibration procedures which is nearly the size of the JuggerBot's deposition nozzle, 3 mm. This means that the position recorded could be a whole tool pathing layer off. However, improvement in accuracy are expected when fusing the estimated location with G-code information. Current testing of the ArUco markers appears promising for improving the

accuracy of the telemetry system since it is not affected by the lens flare and glare from the LED indicator array.

4. - Summary and Future Work

This research outlines the development of a sensor suite and user-guided key data flagging for polymer MEXAM. The sensors data is ingested via multiprocessing in order to maintain maximum framerates for varying data acquisition speeds throughout the connected sensors. A MongoDB database was used to organize, maintain, and store collected sensor data due to its open-source programming and object document storage type. Dual-camera computer vision tracking is used to develop a system agnostic closed-loop nozzle and build plate position estimation framework. Finally, anomalous data tagging controlled by a human operator was developed to group and sort key data regions for post-process analysis and eventually ML training.

The next steps for this research focuses on its usefulness for training ML systems and simulating manufacturing processes. This comes from determining what data and sensors are useful and what data can be ignored to develop more accurate ML systems. Other research groups are finding that an excess amount of sensor data points such as acceleration combined with thermal profile is decreasing the accuracy of the ML models and are overfitting the calculations [14]. Some sensors are found to work for more than one data point allowing the hardware requirements to be reduced. Such dual examples include the optical cameras being used for visual inspection as well as telemetry position and the thermal cameras being useful for both visual inspection as well as thermal inspection. There are other groups already investigating the usefulness of visual inspection and show that it is a strong data point for ML assisted defect detection [11]. By creating intelligent ML models fueled with important training data that monitor live prints during a polymer MEXAM process, researchers will be able to investigate and develop automatic part property and defect assessment that will allow a machine to manufacture “born-certified” parts. As defects and failure modes are detected by trained systems, automatic ML assisted in-situ process correction like what is being investigated by Brion and Pattinson [12] will allow for these “born-certified” parts to be developed even during a print that might have failed. Finally, tool pathing has the most control over a print process and its results, so using training data for live prints to train more intelligent tool-pathing software will help to create more intentional toolpaths with failure prevention and final part properties in mind. Overall, the goal of efficient and rapid training data is to create more intelligent and purposeful AM machines and finally develop “born-certified” part manufacturing.

5. - Acknowledgments

This research was performed with the supervision and support of the US Army Corps of Engineers Engineer Research and Development Center (ERDC) contract Number is W912HZ22C0055 and the Pennsylvania State University Additive Manufacturing and Design program.

6 - References

1. Caltanissetta, F., Dreifus, G., Hart, A. J., & Colosimo, B. M. (2022). *In-situ monitoring of Material Extrusion processes via thermal videoimaging with application to Big Area Additive Manufacturing (BAAM)*. *Additive Manufacturing*, *58*, 102995.
2. Anderegg, D. A., Bryant, H. A., Ruffin, D. C., Skrip Jr, S. M., Fallon, J. J., Gilmer, E. L., & Bortner, M. J. (2019). *In-situ monitoring of polymer flow temperature and pressure in extrusion based additive manufacturing*. *Additive Manufacturing*, *26*, 76-83.
3. Gilmer, E. L., Miller, D., Chatham, C. A., Zawaski, C., Fallon, J. J., Pekkanen, A., ... & Bortner, M. J. (2018). *Model analysis of feedstock behavior in fused filament fabrication: Enabling rapid materials screening*. *Polymer*, *152*, 51-61.
4. Baechle-Clayton, M., Loos, E., Taheri, M., & Taheri, H. (2022). *Failures and flaws in fused deposition modeling (FDM) additively manufactured polymers and composites*. *Journal of Composites Science, 6(7), 202.*
5. Goh, G. D., Yap, Y. L., Tan, H. K. J., Sing, S. L., Goh, G. L., & Yeong, W. Y. (2020). Process–structure–properties in polymer additive manufacturing via material extrusion: A review. Critical Reviews in Solid State and Materials Sciences, 45(2), 113-133.
6. Seppala, J. E., Han, S. H., Hillgartner, K. E., Davis, C. S., & Migler, K. B. (2017). Weld formation ng material extrusion additive manufacturing. *Soft matter*, *13*(38), 6761-6769.
7. Bartolai, J. (2018). *Predicting and Improving Mechanical Strength of Thermoplastic Polymer Parts Produced by Material Extrusion Additive Manufacturing*. The Pennsylvania State University.
8. Nasrin, T., Pourkamali-Anaraki, F., & Peterson, A. M. (2023). *Application of machine learning in polymer additive manufacturing: A review*. *Journal of Polymer Science*.
9. Wong, K. J. (2023, August 4). *Multithreading vs. Multiprocessing Explained*. Built In. https://builtin.com/data-science/multithreading-multiprocessing
10. Brownlee, J. (2023, November 22). *Python Multiprocessing: The complete guide*. Super Fast Python. https://superfastpython.com/multiprocessing-in-python/
11. Forte, M., Eisenhour, M., Malkowski, R. M., Radhakrishnan, P., & Brown, D. C. (2022, October). *Detecting Defects in Low-Cost 3D Printing. In ASME International Mechanical Engineering Congress and Exposition* (Vol. 86632, p. V02AT02A025). American Society of Mechanical Engineers.
12. Brion, D. A., & Pattinson, S. W. (2022). *Generalisable 3D printing error detection and correction via multi-head neural networks*. Nature communications, 13(1), 4654
13. Bilmes, J. (2020). *Underfitting and overfitting in machine learning*. *UW ECE course notes*, *5*.
14. Zhang, J., Wang, P., & Gao, R. X. (2018). *Modeling of layer-wise additive manufacturing for part quality prediction. Procedia Manufacturing*, 16, 155-162.
15. MongoDB. (n.d.). *Documents - MongoDB Manual V7.0*. https://www.mongodb.com/docs/manual/core/document/
16. OpenCV. (n.d.). *OpenCV: Detection of ARUCO markers*. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html
17. OpenCV. (n.d.-a). *Detection of ARUCO markers* (4.10.0) [Software]. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html