TRANSPORTATION POOLED FUND PROGRAM
QUARTERLY PROGRESS REPORT

Lead Agency (FHWA or State DOT): Kansas DOT

INSTRUCTIONS:

Lead Agency contacts should complete a quarterly progress report for each calendar quarter during which the projects are active. Please provide a project schedule status of the research activities tied to each task that is defined in the proposal; a percentage completion of each task; a concise discussion (2 or 3 sentences) of the current status, including accomplishments and problems encountered, if any. List all tasks, even if no work was done during this period.

Transportation Pooled Fund Program Project # *(i.e, SPR-2(XXX), SPR-3(XXX) or TPF-5(XXX)* **TPF-5(535)**	**Transportation Pooled Fund Program - Report Period:** ☒Quarter 1 (January 1 – March 31) ☐Quarter 2 (April 1 – June 30) ☐Quarter 3 (July 1 – September 30) ☐Quarter 4 (October 1 – December 31)
TPF Study Number and Title: **TPF-5(535): Human-centered Steel Bridge Inspection enabled by Augmented Reality and Artificial Intelligence**	

Lead Agency Contact: **David Behzadpour**	**Lead Agency Phone Number:** **785-291-3847**	**Lead Agency E-Mail** David.Behzadpour@ks.gov
Lead Agency Project ID: Click or tap here to enter text.	**Other Project ID (i.e., contract #):** Click or tap here to enter text.	**Project Start Date:** 10/1/2024
Original Project Start Date: **10/1/2024**	**Original Project End Date:** **9/30/2027**	**If Extension has been requested, updated project End Date:** Click or tap to enter a date.

Project schedule status:

☐On schedule	☐On revised schedule	☐Ahead of schedule	☒ Behind schedule

Overall Project Statistics:

Total Project Budget	Total Funds Expended This Quarter	Percentage of Work Completed to Date
$600,000	$100,526	36%

Project Description:

The main objective of this proposed research is to provide state DOTs practical tools for supporting human-centered steel bridge inspection with real-time defect (e.g., fatigue cracks and corrosion) detection, documentation, tracking, and decision making. The proposed research will not only bridge the gaps identified in the IDEA project, but also expand the existing capability by developing AI algorithms for crack and corrosion detection. In addition to AR headsets, the project will also develop AR-based inspection capability using tablet devices. The tablet device can be used to perform AR-based inspection directly in a similar way to the AR headset. It can also leverage Unmanned Aerial Vehicles (UAV) for remote image and video acquisition during inspections, enabling bridge inspections from a distance in a human-centered manner.

Progress this Quarter
(includes meetings, work plan status, contract status, significant progress, etc.):

1. **Task 1: CV and AI algorithms for crack and corrosion inspection**
   Task 1.1: Video-based crack detection algorithm
   This quarter, the team investigated improvements to the video-based crack detection approach using video stabilization and motion magnification techniques to detect fatigue crack movement. However, results showed that crack opening displacements in field conditions are too small to be reliably captured, and even after stabilization and magnification, crack motion was not discernible. As a result, the research shifted toward a frame-by-frame processing approach using ensemble learning, which successfully extracted and tracked fatigue crack features from video data.
   Task 1.2: Deep learning algorithms for crack and corrosion detection
   Progress was made in enhancing the MetaCNN framework through integration of CBAM attention mechanisms and residual connections to improve segmentation of fine cracks and corrosion boundaries. The enhanced model was applied to both image-based and video frame-based inference, enabling improved detection accuracy and robustness.
2. **Task 2: AR-based software for human-centered bridge inspection**
   Development focused on evaluating AR projection methods and advancing inspection data management and 3D reconstruction capabilities. Unity URP decal projection was investigated but found unsuitable due to projection artifacts, lack of perspective projection, and incompatibility with Magic Leap 2, leading to continuation with the existing system. Progress was also made on the temporal analysis feature using a SQLite database, enabling inspection results to be organized and visualized by date. In parallel, RGB-D/SLAM-based point cloud generation, mesh reconstruction, and real-time depth visualization were advanced for tablet and UAV-based inspection workflows.

Anticipated work next quarter:

1. **Task 1: CV and AI algorithms for crack and corrosion inspection**
   - Conduct systematic ablation studies to quantify the contributions of key MetaCNN components, including the Convolutional Block Attention Module (CBAM), residual connections, and base learners.
   - Investigate advanced edge detection methods to improve localization accuracy and boundary delineation of subtle structural defects.
2. **Task 2: AR-based software for human-centered bridge inspection**
   - Continue development, testing, and refinement of the temporal analysis feature to ensure reliable functionality. completion, conduct comprehensive system-level testing to further optimize and finalize the AR inspection platfo followed by preparation of the final project report.

- Quantify the accuracy of generated point clouds and mesh models using reference objects with consistent tracking, and evaluate the processing pipeline on the Jetson AGX Orin to assess latency and feasibility for autonomous UAV-based inspection.

Significant Results:

1. **Task 1: CV and AI algorithms for crack and corrosion inspection**

Task 1.1: Video-based crack detection algorithm

In this quarter, we investigated the feasibility of improving video-based crack detection using video stabilization and motion magnification techniques. This effort was motivated by observations from field data, where traffic-induced crack opening and closing in steel bridges are extremely small and difficult to detect directly from raw video. To enhance the visibility of such subtle structural responses, motion magnification was applied prior to feature-based crack detection.

Video Stabilization and Motion Magnification: Videos captured during bridge inspections contain significant global camera motion, including 3D rotation and perspective changes, which must be removed before applying motion magnification. To address this, a homography-based stabilization approach was implemented. KLT optical flow was used to track feature points across frames, and corresponding 3×3 homography matrices were estimated to model inter-frame transformations. These transformations were sequentially applied to align all frames to a common reference perspective, followed by edge padding and cropping to isolate regions of interest such as fatigue cracks.

The stabilized video was then processed using a deep learning-based motion magnification method to amplify sub-pixel movements. However, evaluation results showed that crack opening displacements under field conditions are too small to be reliably detected. Figure 1 shows a frame of the stabilized and motion magnified video that contain a fatigue crack. Even after applying stabilization and magnification, crack motion remained indiscernible, indicating that motion-based approaches are not effective for crack detection in real-world bridge inspection scenarios (see stabilized and magnified video results).

Figure 1. A video frame of fatigue crack after video stabilization and motion magnification

Frame-by-Frame Video Processing and Ensemble Learning: Because the motion magnification approach proved insufficient under field conditions, the methodology was shifted to a frame-by-frame video processing pipeline leveraging ensemble learning to track crack features. In this approach, input video is processed using OpenCV to extract individual frames, which are resized to 512×512 resolution. Each frame is then processed in parallel by multiple YOLO-based segmentation models to generate pixel-wise probability maps. These outputs are stacked and passed to a trained MetaCNN, which refines the predictions and produces the final inference for each frame.

As shown in Figure 2, this ensemble-based approach enables crack detection based on visual features rather than motion cues and was shown to successfully extract and track fatigue crack patterns from the video data, where the motion magnification approach had failed (see MetaCNN inference results).

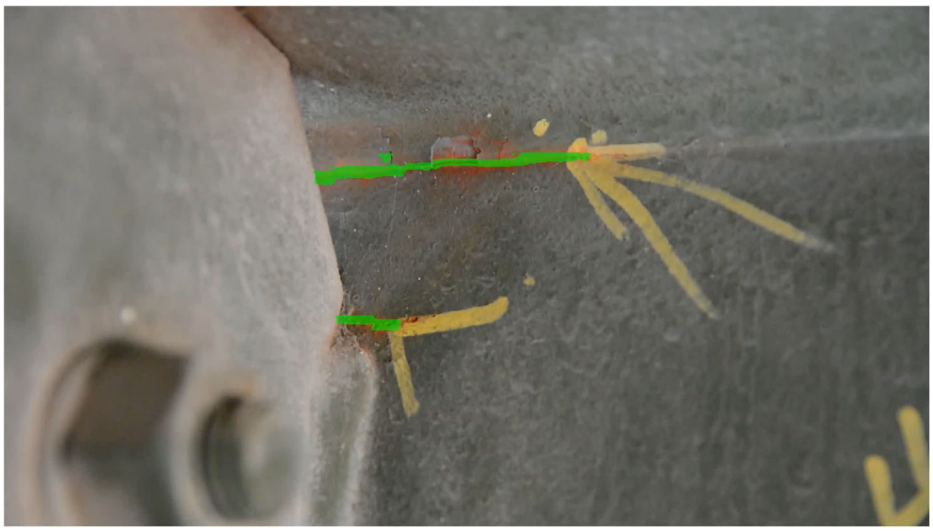

Figure 2. Fatigue crack identification using the same video based on ensemble learning

Task 1.2: Deep learning algorithms for crack and corrosion detection

The CBAM is a lightweight attention mechanism designed to refine intermediate feature maps by adaptively focusing on the most informative elements within the input data. The MetaCNN receives a stacked 9-channel probability map (from three base YOLO models across three classes). The CBAM acts as a filter, helping the network decide "what" is important and "where" to look. It operates sequentially across two dimensions:

- Channel Attention (What to pay attention to): The stacked input contains probability maps from different base learners (e.g., YOLOv8s, YOLOv8m). Channel attention evaluates these input channels and assigns higher weights to the most reliable and informative feature maps. For example, if YOLOv8x generates a highly confident probability map for a specific crack, the channel attention module will emphasize that channel over a noisier channel produced by a smaller model.
- Spatial Attention (Where to pay attention): Following channel refinement, spatial attention analyzes the structural layout of the image. It generates a spatial map that highlights regions containing critical structural information (such as the boundaries of severe corrosion or the localized path of a hairline crack). It effectively tells the network to focus computational effort on these specific regions while suppressing irrelevant background noise (e.g., shadows, intact paint).

By applying both channel and spatial attention, the CBAM ensures that the MetaCNN does not simply average the YOLO base predictions. Instead, it intelligently prioritizes the most accurate base-model inputs and aggressively focuses on subtle defect morphologies, significantly improving the detection of fine features like cracks and reducing false positives in noisy backgrounds.

Also, traditional deep convolutional networks often suffer from the "vanishing gradient" problem (as the network gets deeper, the gradients used for training can become infinitesimally small, making the model harder to optimize and sometimes decreasing accuracy). Residual connections, the core innovation of the ResNet architecture, solve this by introducing "skip connections" that bypass one or more layers. In the context of the MetaCNN:

- Instead of forcing the meta-learner's deep layers to reconstruct a completely new segmentation map from scratch (learning a direct mapping F(x)), the residual connection adds the original input x (the stacked YOLO probability maps) directly to the output of the convolutional block (F(x) + x).
- This means the deep convolutional layers and the CBAM are only responsible for learning the *residual*—the necessary corrections or refinements to the base YOLO predictions.

How it can improve results:

1. Stable Training: Skip connections preserve the flow of gradients during backpropagation, allowing us to build a deeper, more expressive meta-learner without stalling during training.

2. Preservation of Confidence: By adding the original stacked probability maps back into the refined output, the network explicitly preserves the strong baseline confidence generated by the YOLO models. It allows the MetaCNN to act as a precision-correction tool—fixing localized errors via the CBAM pathway while maintaining the fundamentally correct predictions already established by the base learners.

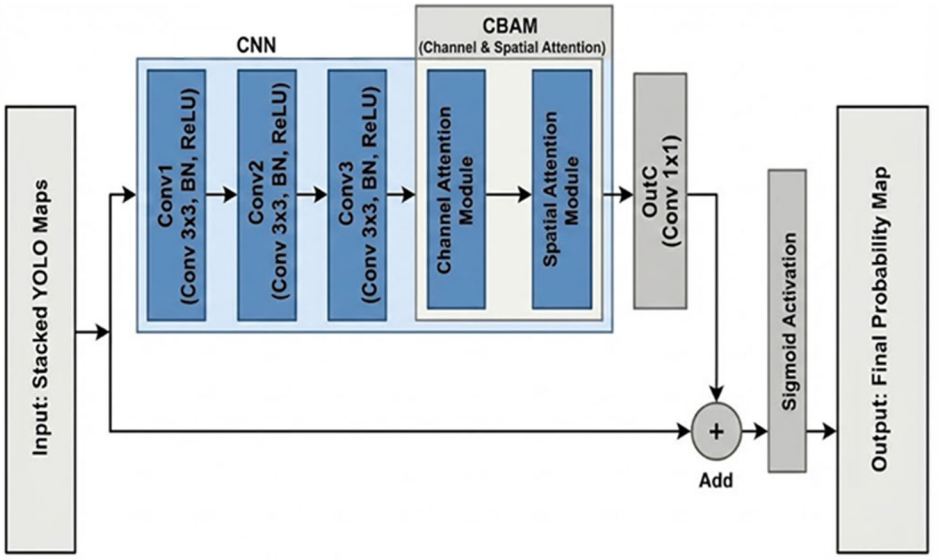

Figure 3. MetaCNN architecture with CBAM and Residual network.

The implemented MetaCNN model is illustrated in Figure 3. The integration of CBAM and Residual Connections transforms the MetaCNN from a simple aggregator into an intelligent, adaptive refinement system. The Residual Connections provide a stable foundation, preserving the baseline predictions of the YOLO ensemble and easing network optimization. Meanwhile, the CBAM acts as a targeted spotlight, dynamically emphasizing the most reliable channels and focusing aggressively on critical spatial boundaries to identify difficult, fine-scale defects.

Table 1 compares the performance of the proposed MetaCNN framework with benchmark results reported by (Ameli et al. 2023) for YOLOv8 and Mask R-CNN evaluated on the same test dataset. MetaCNN achieves the best overall performance, with an mAP50 of 0.75 and an F1 score of 0.79, outperforming both benchmark models in detection reliability and balanced segmentation accuracy. Qualitative results for corrosion segmentation is shown in Figure 4.

Table 1. Corrosion segmentation performance comparison with benchmark models by Ameli et al. 2023

Metric	Ameli et al.: YOLOv8 (Test)	Ameli et al.: Mask R-CNN (Test)	MetaCNN (Test)
mAP50	0.726	0.674	**0.750**
Precision	0.853	0.625	**0.860**
Recall	0.647	**0.922**	0.734
F1 Score	0.730	0.745	**0.790**

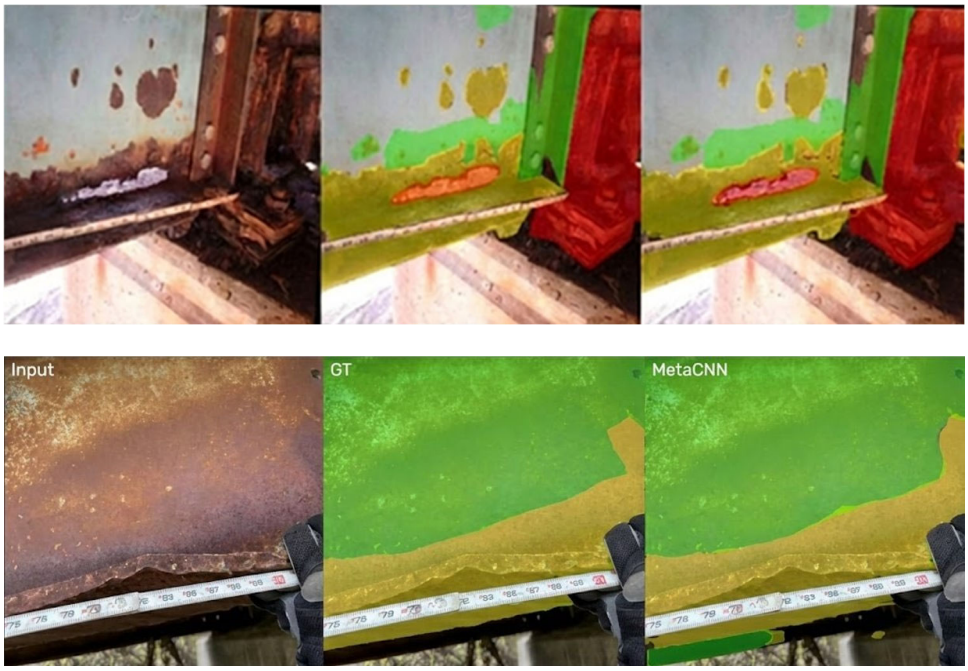

Figure 4. Qualitative corrosion segmentation results using the proposed MetaCNN-based ensemble framework: (left) input bridge inspection image, (middle) ground-truth annotation, and (right) MetaCNN prediction. Corrosion severity classes are visualized as Fair (green), Poor (yellow), and Severe (red).

2. Task 2: AR-based software for human-centered bridge inspection

Subtask 2.3: AR software environment for AR headset

In this quarter, development efforts focused on evaluating alternative projection methods and advancing temporal analysis capabilities within the AR inspection system. A major effort was devoted to investigating Unity's Universal Render Pipeline (URP) decal projectors as a potential replacement for the deprecated standard projector component currently used in the system. Figure 5 illustrates one example application of the decal projector. This transition was motivated by the need to address known projection artifacts—such as unintended projection onto occluded or back-facing surfaces—and to modernize the implementation by avoiding legacy rendering features.

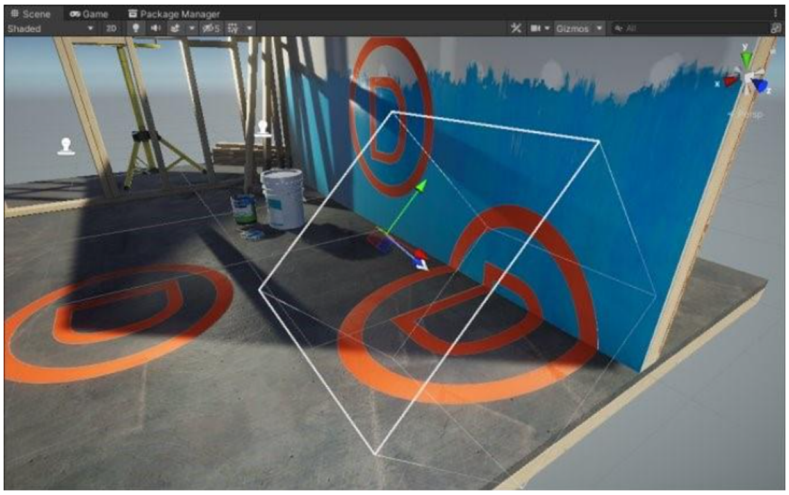

Figure 5. Unity URP Decal Renderer Example

To support this investigation, the Unity project was migrated from the built-in render pipeline to URP. Because such migrations can introduce visual inconsistencies, the application was thoroughly tested to ensure rendering fidelity was preserved. After validation, the URP decal renderer was enabled, and decal projectors were implemented and evaluated within the desktop development environment.

Testing revealed that several limitations persisted. As shown in Figure 6, projections continued to appear on back-facing surfaces and did not properly account for object occlusion. In addition, decal projectors only support orthographic projection, whereas the inspection system requires perspective projection for accurate alignment of reprojected camera imagery. As illustrated in Figure 7, orthographic projection does not exhibit depth-dependent scaling, unlike perspective projection where the projection frustum expands with distance. This limitation prevents accurate spatial alignment and reduces applicability for AR-based inspection.

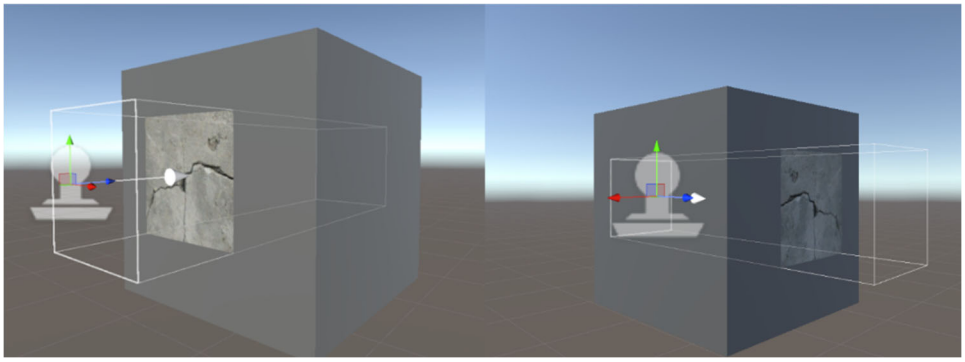

Figure 6. Back face projection and occlusion issues

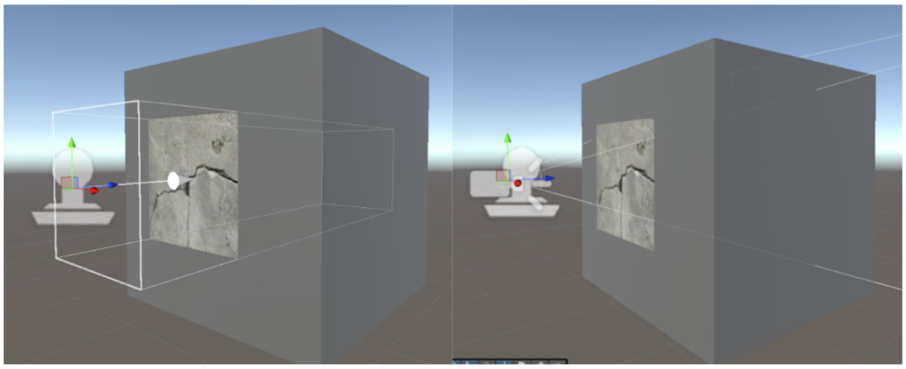

Figure 7. Orthographic vs perspective projections

Despite these limitations, decal projectors were temporarily integrated into the inspection pipeline for further evaluation. However, deployment on the Magic Leap 2 headset revealed additional issues, as decal projectors failed to render correctly in the device environment. Multiple attempts were made to resolve these issues, and alternative URP-based approaches were explored. Ultimately, due to the lack of clear visual or performance benefits and the increased implementation complexity, the decision was made to revert to the original projector-based system. Although this effort did not result in a deployable improvement, it provided valuable insight into the limitations of URP-based projection methods for AR inspection.

In parallel, development progressed on the temporal analysis feature using the previously implemented SQLite database. This functionality enables inspectors to filter and visualize inspection results by date through a graphical user interface integrated into the Magic Leap space localization menu. A dropdown component dynamically populates available inspection dates at runtime based on database queries, as shown in Figure 8.

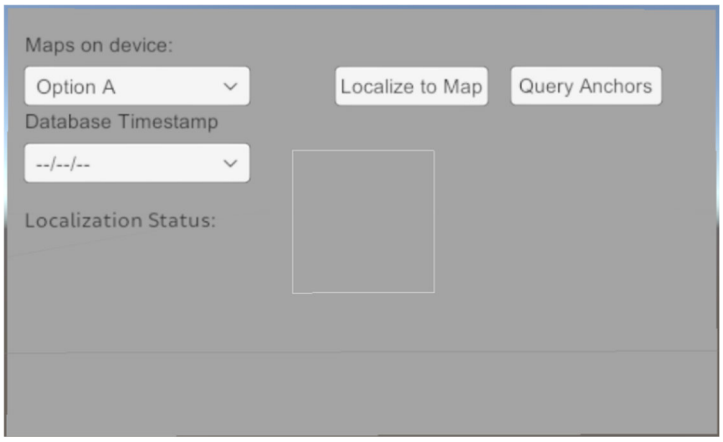

Figure 8. Temporal analysis interface

Implementation of this feature required updates to the inspection pipeline and the development of database structures to associate spatial anchors with timestamps during localization. Inspection results are grouped by date within the Unity hierarchy, allowing efficient loading and selective visualization based on user input. This design minimizes performance overhead by avoiding repeated data loading operations. The temporal analysis feature is nearing completion and is currently undergoing validation on the Magic Leap headset.

Subtask 2.4: AR software environment for tablet device and UAV

During this quarter, progress was made in integrating RGB-D imaging with SLAM-based reconstruction and automated mesh generation. The developed pipeline consists of three main stages: (1) camera pose estimation via visual SLAM, (2) dense point cloud generation, and (3) surface mesh reconstruction with post-processing. Experimental validation was conducted in an architectural materials laboratory environment containing columns, wall panels, and structural models.

Accurate camera pose estimation is critical for multi-view 3D reconstruction. A keyframe-based visual odometry approach was implemented using the Perspective-n-Point (PnP) algorithm. For each RGB frame, SIFT keypoints are detected and matched between consecutive frames using Lowe's ratio test (threshold = 0.70), followed by RANSAC-based geometric verification to remove outliers. Valid correspondences are used to estimate camera pose, with a reprojection error threshold of 1.5 pixels. Keyframes are selected only when motion exceeds predefined thresholds (2° rotation or 3 cm translation) and sufficient inliers are present, ensuring robust and efficient mapping.

To reduce drift, a lightweight sliding window optimization is applied over a window of 12 keyframes, minimizing pose errors using gradient descent in the SE(3) space. This improves global consistency while maintaining computational efficiency.

For each selected keyframe, a dense colored point cloud is generated by back-projecting valid depth pixels into world coordinates. Depth frames are preprocessed using bilateral filtering to reduce noise while preserving edges, and an erosion step is applied to remove mixed-pixel artifacts near depth discontinuities. Depth values outside the range of 0.5–3.5 m are discarded. The resulting point clouds, as shown in Figure 9, are accumulated to form a complete scene representation.

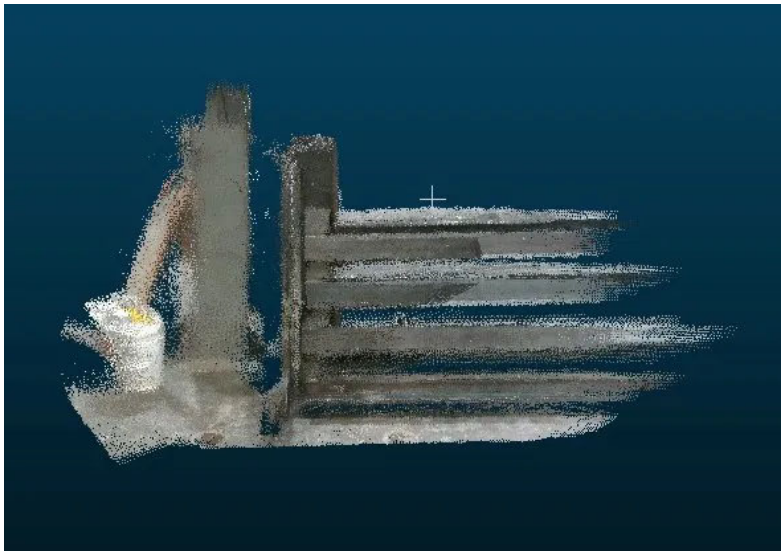

Figure 9. Generated point cloud using RGB-D image pairs

Before surface reconstruction, the accumulated point cloud undergoes several cleaning steps. Voxel downsampling at 2 cm resolution reduces redundancy and normalizes point density across the scene. Two successive rounds of statistical outlier removal, using a 20-nearest-neighbour search with thresholds of 3σ and 2σ respectively, eliminate residual noise and flying points that survive the depth filtering stage.

Surface reconstruction is performed on the cleaned and cropped point cloud using the Ball-Pivoting Algorithm (BPA). Unlike implicit methods such as Poisson surface reconstruction, BPA generates triangular faces only where input points exist, a virtual ball of a given radius rolls over the point cloud surface, and a triangle is formed each time the ball simultaneously touches three points. This property makes BPA well-suited to open scenes where large unobserved regions should remain as holes rather than be filled with geometrically plausible but fictionally interpolated surfaces. Prior to reconstruction, surface normals are estimated for each point using a 20-nearest-neighbour search with two smoothing iterations. Normal orientations are made globally consistent by propagating orientation along the tangent plane, with a viewpoint constraint applied to ensure normals face outward toward the camera. The BPA is then applied with an automatically estimated ball radius derived from the mean nearest-neighbor spacing of the point cloud.

After reconstruction, isolated fragments smaller than 300 faces are removed to eliminate spurious surface patches. Non-manifold edges are repaired to improve mesh validity. Taubin smoothing is subsequently applied for 15 iterations with parameters $\lambda=0.5$ and $\mu=-0.53$. Unlike Laplacian smoothing, which causes progressive mesh shrinkage, Taubin smoothing alternates between a positive and a negative diffusion step, effectively suppressing high-frequency surface noise while preserving the overall geometry of the reconstructed structure. Figure 10 illustrated the generated mesh model.

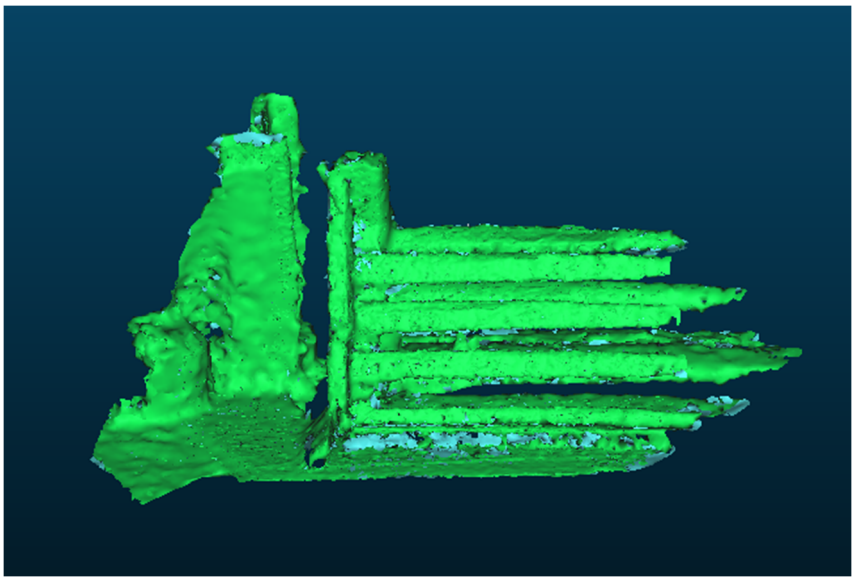

Figure 10. Generated mesh model using the generated point cloud in Figure 9

Building upon the RGB-D reconstruction and mesh generation pipeline described above, this effort further focuses on developing a user interface capable of supporting real-time data acquisition and visualization for remote AR-based bridge inspection. While the long-term objective is to stream RGB imagery and reconstructed mesh models to the inspector's device, this phase focuses on establishing reliable real-time RGB-D data capture and visualization as a foundational capability.

Integration with the Intel RealSense D455 camera enables real-time visualization of RGB-D data within the Unity environment. The system connects to the camera through a streaming pipeline, capturing Z16 depth data at a resolution of 848×480 pixels and 30 frames per second. During runtime, the raw depth data is processed frame-by-frame and converted into a colorized heatmap using a built-in colorizer. The resulting visualization is mapped onto a Texture2D and rendered in the Unity interface via a RawImage UI component.

As shown in Figure 11, the developed interface displays a real-time depth heatmap over the captured scene, where color variations represent relative distance from the sensor. This visualization provides immediate feedback on scene geometry and supports validation of depth sensing and alignment, which are critical prerequisites for accurate mesh reconstruction.

To ensure stable real-time operation, memory management mechanisms are implemented to release frame data after processing, preventing memory accumulation and potential crashes. This capability establishes a reliable front-end for RGB-D data acquisition, which will support subsequent development of mesh generation and transmission workflows. Ultimately, the system will prioritize streaming RGB video alongside reconstructed mesh representations, rather than raw RGB-D data, to improve efficiency and usability in remote inspection scenarios.

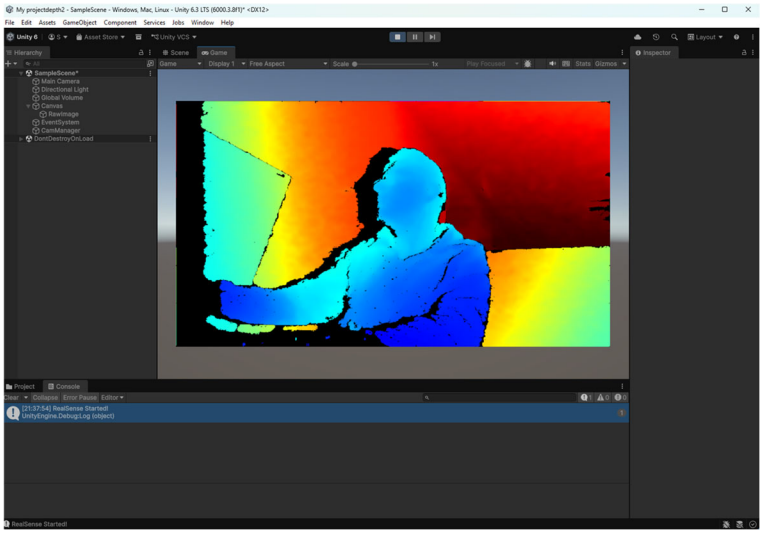

Figure 11. Realtime visualization of captured RGB-D image in the developed interface

Circumstance affecting project or budget. (Please describe any challenges encountered or anticipated that might affect the completion of the project within the time, scope and fiscal constraints set forth in the agreement, along with recommended solutions to those problems).

Due to delays in staffing, a one-year no-cost extension may be required to complete the project scope.

Potential Implementation: