CAIT-UTC-REG65

Development of a Geometric Extraction Framework as Part of a Pilot Digital Twin Framework for Open-Deck Rail Bridges

FINAL REPORT
April 2022

Submitted by:

Amirali Najafi
Research Associate
Center for Advanced Infrastructure and Transportation
Rutgers, The State University of New Jersey
Rm 224, 100 Brett Rd. Piscataway, NJ, 08854

External Project Manager
Richard Schaefer
Chief Engineer, Design and Environmental
New Jersey Transit Corporation

In cooperation with

Rutgers, The State University of New Jersey
And
New Jersey Transit Corporation
And
U.S. Department of Transportation
Federal Highway Administration

Disclaimer Statement

The Center for Advanced Infrastructure and Transportation (CAIT) is a Regional UTC Consortium led by Rutgers, The State University. Members of the consortium are Atlantic Cape Community College, Columbia University, Cornell University, New Jersey Institute of Technology, Polytechnic University of Puerto Rico, Princeton University, Rowan University, SUNY - Farmingdale State College, and SUNY - University at Buffalo. The Center is funded by the U.S. Department of Transportation.

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
CAIT-UTC-REG65		

4. Title and Subtitle	5. Report Date
Development of a Geometric Extraction Framework as Part of a Pilot Digital Twin Framework for Open-Deck Rail Bridges	April 2022
	6. Performing Organization Code CAIT/Rutgers
7. Author(s) Amirali Najafi https://orcid.org/0000-0002-7845-0859	8. Performing Organization Report No. CAIT-UTC-REG65
9. Performing Organization Name and Address Center for Advanced Infrastructure and Transportation Rutgers, the State University of New Jersey 100 Brett Road, Piscataway, NJ 08854	10. Work Unit No.
	11. Contract or Grant No. 69A3551847102
12. Sponsoring Agency Name and Address Center for Advanced Infrastructure and Transportation Rutgers, The State University of New Jersey 100 Brett Road, Piscataway, NJ 08854	13. Type of Report and Period Covered Final Report 09/15/2021-03/15/2022
	14. Sponsoring Agency Code

15. Supplementary Notes
U.S. Department of Transportation/OST-R
1200 New Jersey Avenue, SE
Washington, DC 20590-0001

16. Abstract
Open-deck railway bridges require expensive and customized timber sleepers. When these sleepers are due for replacement, a manual process for geometry measurement is necessary which can be time consuming, inaccurate, and expensive. In addition, significant downtime is required for the safety of the inspectors that measure and assess the conditions of open-deck bridges. In this report, an alternative approach is proposed for geometry extraction of timber sleepers using unmanned aerial vehicle (UAV) inspections and use of artificial intelligence. First, a photogrammetric procedure for development of three-dimensional (3D) bridge models from UAV-based images is provided. Next, a deep learning-based algorithm for segmentation of 3D bridge model into recognizable components is described. Finally, a geometric primitive fitting algorithm is outlined for identifying the geometry of individual components. The aim for this development with 3D scans and automation is to reduce the maintenance and sleeper replacement procedure costs and challenges for open-deck bridges.

17. Key Words	18. Distribution Statement
Geometry extraction, track and sleeper alignment, artificial intelligence, unmanned aerial vehicles	

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	Total 14	

Form DOT F 1700.7 (8-69)

Table of Contents

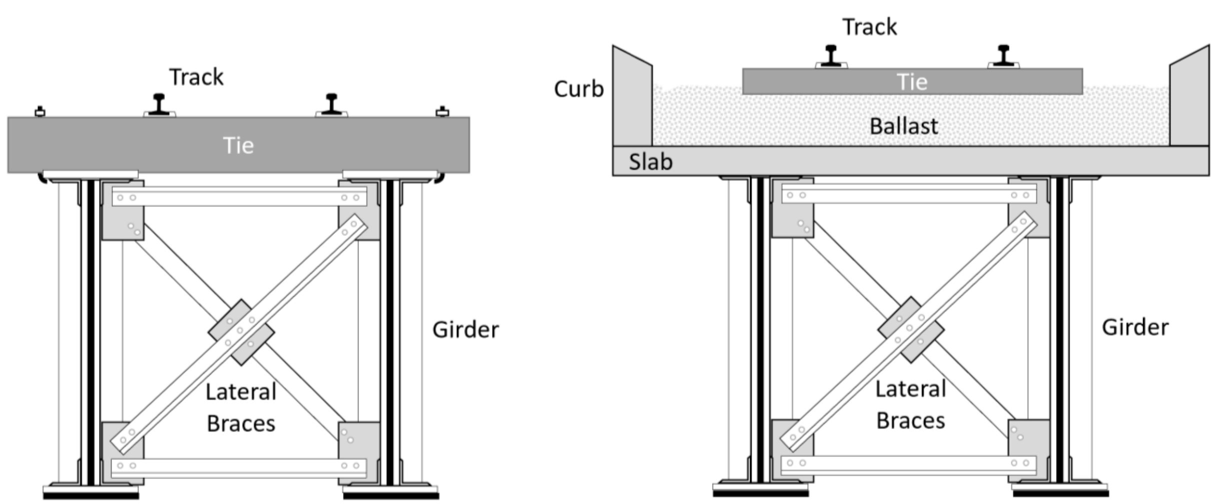
List of Figures

List of Tables

DESCRIPTION OF THE PROBLEM

The US rail transport network is among the largest in the globe. The rail network carries 16% of the nation's freight by weight [1]. The day-to-day dependence and cost-effectiveness are reasons why the rail transport network is a critical infrastructure asset. Railway bridges provide the critical role of connecting the rail network over vast bodies of water, roads, and ravines. Increases in demand in recent years have translated to additional pressure on the railway bridge infrastructure through increases in speed of the traffic and the axle loads. As a result, a greater emphasis has been placed on condition monitoring of railway bridges.

Railway bridges are typically constructed in ballasted or open-deck configurations. Initial construction costs are higher for ballasted bridges; however, maintenance costs are lower as special sleepers are not required, superstructure is better protected due to reinforced concrete cover, and maintaining a level track (by adding more ballast) is easier. Open-deck bridges have simpler sleeper-on-girder designs and are more cost-effective to construct due to lower material use. Operational costs are higher in open-deck bridges, due to the need for specialized dimensions and customized fitting of the sleepers. Maintenance cost of the superstructure is higher too, as exposure to the elements result in more rapid degradation of the girders and other supporting elements. From a track elevation perspective, the open-deck configuration leads to a fixed track grade. Maintaining a flat track surface becomes a difficult task, as the bridge approaches are more flexible than the rigid deck. The approach grades must therefore be frequently fixed, adding to extensive maintenance costs. The typical railway bridge configurations discussed are illustrated in Fig. 1.



(a) Open-deck (b) Ballasted

Figure 1 - Typical railway bridge configurations.

New Jersey Transit (NJT) is one of the largest transit operators in the nation, having a large fleet of bridges. Like many legacy railroads, NJT has inherited a large inventory of open-deck bridges. For NJT, sleeper replacement is costly as custom fits are necessary for every sleeper. The traditional procedure for deck sleeper replacement begins with a survey of the structure, including general line, grade, dimensions, and other features. This data is provided to a structural engineer for calculating sleeper configurations and generation of sleeper tables, which are sent to a fabrication shop for manufacturing. Each sleeper is unique and requires specific dapping (e.g., holes and notches). Sleepers are then shipped to the open-deck bridge and installed with minimal adjustments. NJT is interested in expediting the sleeper assessment, outage times, and replacement procedures, through the use of modern technologies in order to potentially reduce maintenance costs. For these reasons, NJT has elected to investigate the use of 3D scanning technologies for condition monitoring of their open-deck bridges.

APPROACH

This project uses non-contact 3D scanning technology and AI for developing a geometry extraction framework which assists in the inspection and maintenance procedures for deck sleepers and rails on open-deck bridges. The geometry extraction approach outlined for this project is as follows:

i. Laser scan or unmanned aerial vehicle (UAV) photography of a benchmark bridge.
ii. Photogrammetry and construction of 3D point cloud of the bridge.
iii. Segmentation of the various point cloud components into recognizable elements (e.g., sleepers, rail, and girders).
iv. Identification and quantification of geometric volumes and deviations in various recognized elements.

Due to the pilot nature of this investigation, a real-life bridge is not used in this development. A database of graphically generated open-deck bridges is developed instead to assist with tasks i and iii.

METHODOLOGY

The methodologies described in this section are organized in relation to the four tasks listed in the previous section. The steps for photogrammetry and generation of 3D point cloud are first described. Several open-deck bridges are created in a computer graphics imagery (CGI) environment. A deep learning approach is proposed next for 3D component segmentation. And lastly, individual components are fitted with geometric shapes.

3D Scan and Photogrammetry

A point cloud is the simplest form of a 3D model and is comprised of individual coordinates and RGB color values. Generating a 3D point cloud of a bridge can be attained through laser scans or photographic pictures in a process called *photogrammetry*. Laser scanners operate by emitting laser pulses which bounce off the surface of an object, and the time for reflection is measured. Photogrammetry begins by detecting specific features in a picture, which are aligned with the same features in other pictures using triangulation. The various angles for each feature point on the objects are used to generate the position of the feature point in the 3D point cloud. The geometric extraction described in this report requires two separate procedures: (i) photogrammetry and (ii) direct point cloud generation. The first procedure pertains to the *benchmark* bridge under investigation. The second procedure is used for rapid generation of training data for the component segmentation step. Variations are created in the features of the open-deck bridges in both procedures. Some of the features include the bridge length, curvature, approach level, superelevation, and horizontal alignment.

Several commercially available software are used in study. These include: (i) *SketchUp,* for construction of a CGI environment, (ii) *RealityCapture,* for point cloud generation through photogrammetry, and (iii) *CloudCompare,* for direct conversion of solid models to point clouds. For the photogrammetry procedure, an open-deck bridge with specific features is constructed in Sketchup. A hypothetical UAV path is planned around the bridge. Images collected by the UAV are converted to a 3D point cloud using RealityCapture. The direct procedure is designed for rapid production of 3D bridge models; hence the UAV-based photogrammetry step is eliminated. Instead, the 3D bridge models are converted directly to point clouds via the CloudCompare software. An illustration of both procedures is provided in Fig. 2.

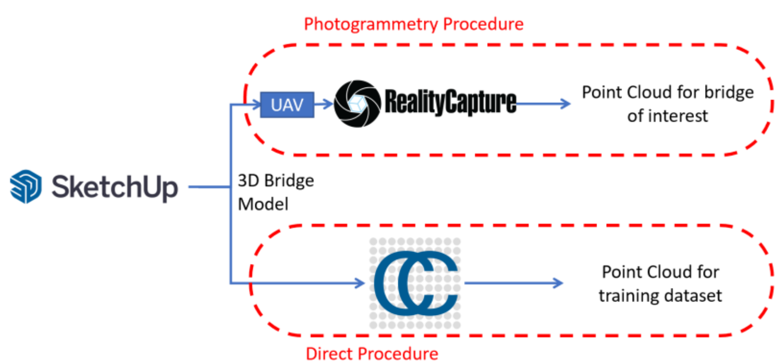


Figure 2 - Procedures for point cloud generation.

The point clouds are next post-processed for scale and data structure size. Result from the photogrammetry procedure is scaled back to the original dimension of the benchmark bridge. Results from the direct procedure need not be scaled. All point clouds are downsampled due to the computational challenge posed by the semantic segmentation procedure discussed next.

Semantic Segmentation

Semantic segmentation is the process of classifying the 3D point cloud data structure into regions belonging to the same component and having the same properties. Manual semantic segmentation is time consuming and inefficient. Other segmentation methods proposed in the literature include color-based segmentation, Random Sample Consensus (RANSAC), and Euclidean Clustering [2]–[4]. These methods however also a large degree of require manual tunning. PointNet is a supervised deep learning model which allows for fully autonomous component segmentation for point cloud data [5]. The PointNet requires a training database of pre-segmented bridge point clouds.

The PointNet architecture accepts raw point cloud data as input and produces labeled point cloud as output. Each point in the point cloud is represented as (x_i, y_i, z_i). There are n points and m segments. Multi-layer perceptron (mlp) blocks are installed throughout the model, mapping each point from 3 dimensions to 64, and from 64 dimensions to 1024, as shown in Fig. 3. The 64-dimensional features are concatenated with global features, resulting in a space of $n \times 1088$. Next, two mlp layers map to lower dimensional spaces of 128, and m for the number of segments. Concatenation of global features with point features allows the PointNet to combine global and local semantics for determining the interaction of neighboring points which carry useful information. To make the network geometric transformation invariant, networks of T-Net are trained to perform affine transformation of the input coordinates. Due to the unstructured nature of point cloud data, $n!$ many permutations of data representation are possible. The commutative property makes max pool a symmetric function. To make the feature detection permutation invariant, a max pooling operation is selected.

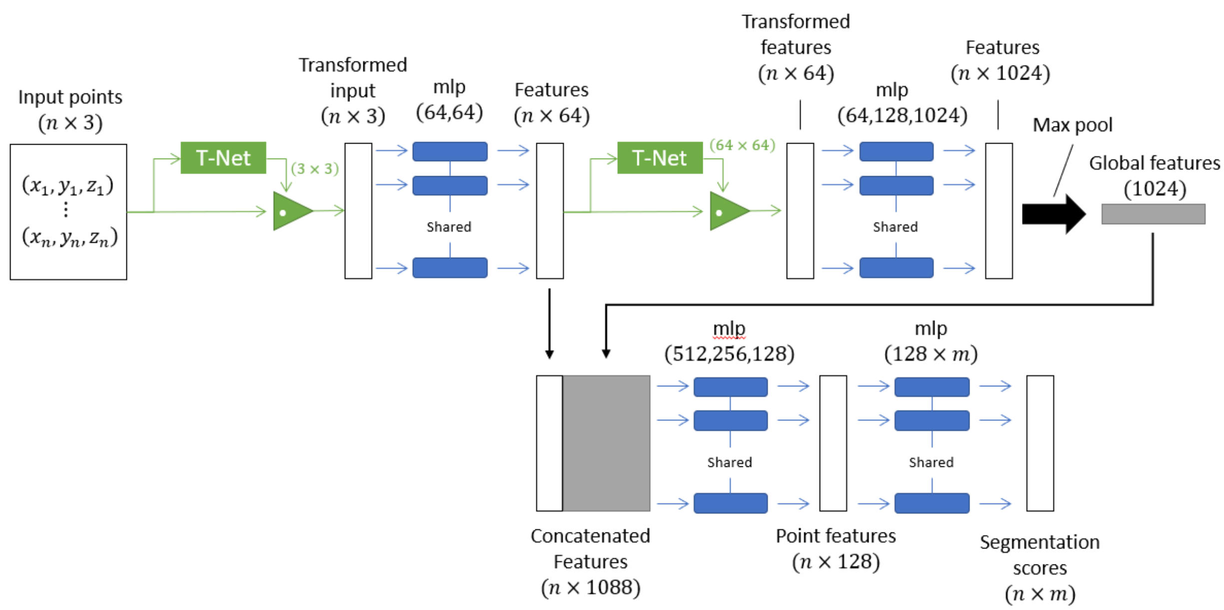


Figure 3 - PointNet architecture.

A standard Softmax loss (i.e., Softmax activation plus a cross-entropy loss) is used for training. The large number of parameters associated with the transformation matrices may lead to overfitting and instability problems during training. Hence, an L_2 regularization term is added to the Softmax loss. See Qi et al. [5] for additional information on the loss functions.

Geometry Fitting

The final step in developing a geometric extraction framework is to fit solid geometric volumes around the segmented point cloud components. By fitting standard geometric primitives (e.g., planes, lines, cubes, cylinders, and spheres) around the segmented point clouds, the geometry of various components can be quantified. For instance, fitting a cuboid around a sleeper point cloud allows for extraction of the component's rotation along the longitudinal axis of the bridge (i.e., superelevation). Geometry fitting is a mathematical optimization process that computes the best fitting geometric primitive around some point cloud. Optimization often involves iterative solving of the best fit until some objective function is minimized. A RANSAC-based algorithm is employed for geometry fitting.

High resolution point clouds are necessary for geometry fitting. The segmented point clouds from the previous step are first upsampled via a nearest neighbor approach. This algorithm begins with a segmented parent point, selection of the nearest K points, and segmenting of those points with the same label as the parent point.

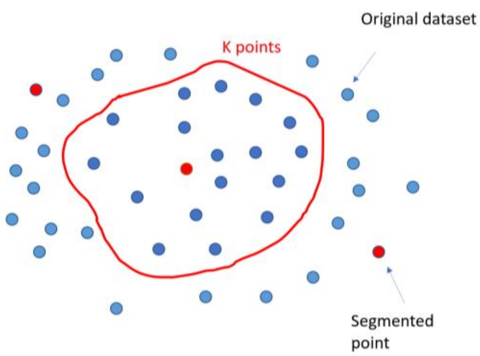


Figure 4 - Upsampling using the nearest neighbor approach.

The RANSAC algorithm for fitting geometric primitives works by selecting a random set of points from a point cloud and minimizing the Euclidean distance between the random set and a geometric primitive. Some of the geometric primitives suitable for geometry extraction in open-deck bridges are cuboids for sleepers, lines for tracks, and combinations of planes for girder flanges and webs. This process is iterated until a suitable fit is acquired between point clouds and the designated geometric primitive. The RANSAC algorithm for cuboid fitting is outlined below.

Algorithm 1 - RANSAC algorithm for cuboid fitting

Determine:
N – the maximum number of iterations

```
    D – the distance threshold for well-fitting points
for i = 1:6
    while j < N
        Draw random points from segmented point cloud
        Fit a plane to the drawn points
        Calculate the Euclidean distance between the plane and points
        if Euclidean distance < d
            Select all points close to the plane and move them from point
        cloud
            end
        Use the best fit plane from the collection and remove all point
        close to the plane.
    end
    Find intersection between planes
end
Present cuboid coordinates and parameters
```

FINDINGS

This section presents results and findings from the geometry extraction study. First, the CGI-generated open-deck bridges are constructed. UAV paths are planned around the benchmark bridge and photogrammetry software is used for point cloud construction. A database of pre-segmented bridge point clouds is constructed. The training process and predictions are visualized. Lastly, the geometry fitting results are presented.

Photogrammetry and Point Clouds

A database of 41 (1 benchmark and 40 training bridges) CGI-based open-deck bridges is constructed using SketchUp. Each bridge is generated with different features. The range of possible variations for the bridge features are summarized in Table 1. Presence of variations in the training database is intended for a more thorough training of the PointNet model.

Table 1 - Variations in bridge features

Feature	Variation range
Length	80-115 meters
Curvature	1/1000 ft - straight
Girder count	2-4
Track count	2-4
Sleepers	Superelevation: 0-0.5% Dapping: with/without notches
Misc.	Track signs, panels, and fences

For the benchmark structure, the surrounding environment are designed and textured in order to assist the photogrammetry software (RealityCapture) with feature detection. The training database does not need a surrounding environment as the solid-to-point cloud conversion feature of the software (CloudCompare) can directly and rapidly convert models. An illustration of a CGI-generated bridge is available in Fig. 5.

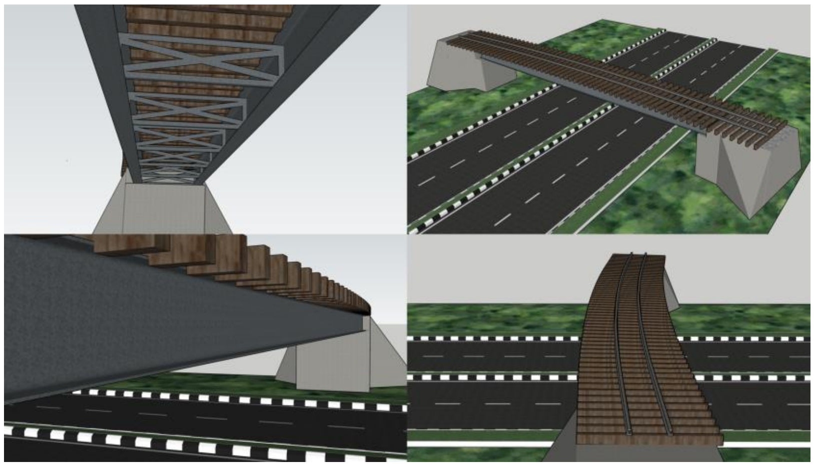
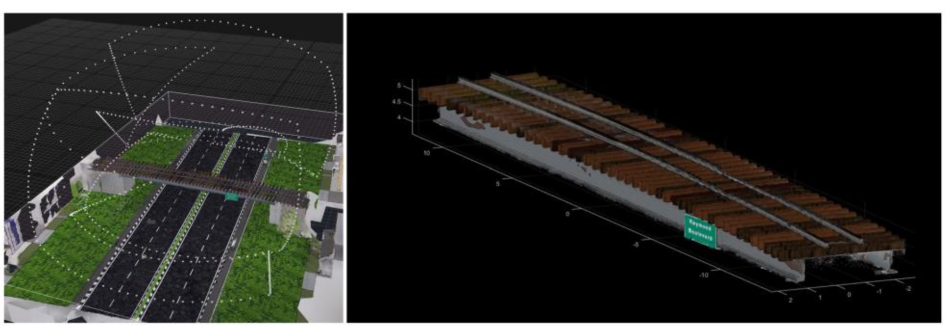


Figure 5 - CGI-based bridge model.

The photogrammetry procedure outlined in Fig. 1 is used to generate point cloud of the benchmark bridge. First, the UAV flight path, shown in Fig. 6(a), is planned. The aerial camera feature of SketchUp allows simulation of a UAV flight and photographing of the bridge from various angles. Photos of the bridge are next imported into RealityCapture for point cloud construction. The generated point cloud does not have the same scale as the original structure. Therefore, a known dimension from the bridge (e.g., sleeper length) is used to scale the point cloud. Finally, the surrounding environment are cropped out to create the 3D reconstruction shown in Fig. 6(b).



(a) UAV path (b) Cropped point cloud

Figure 6 - Point cloud created through the first procedure.

The second procedure is used to generate training point clouds. The points in the training data are automatically categorized with the correct segmentation labels. A total of 40 bridges are generated as shown in Fig. 7. The color coding in this figure is as follows: yellow for tracks, green for sleepers, red for girders and cross-braces, and blue for other miscellaneous components.

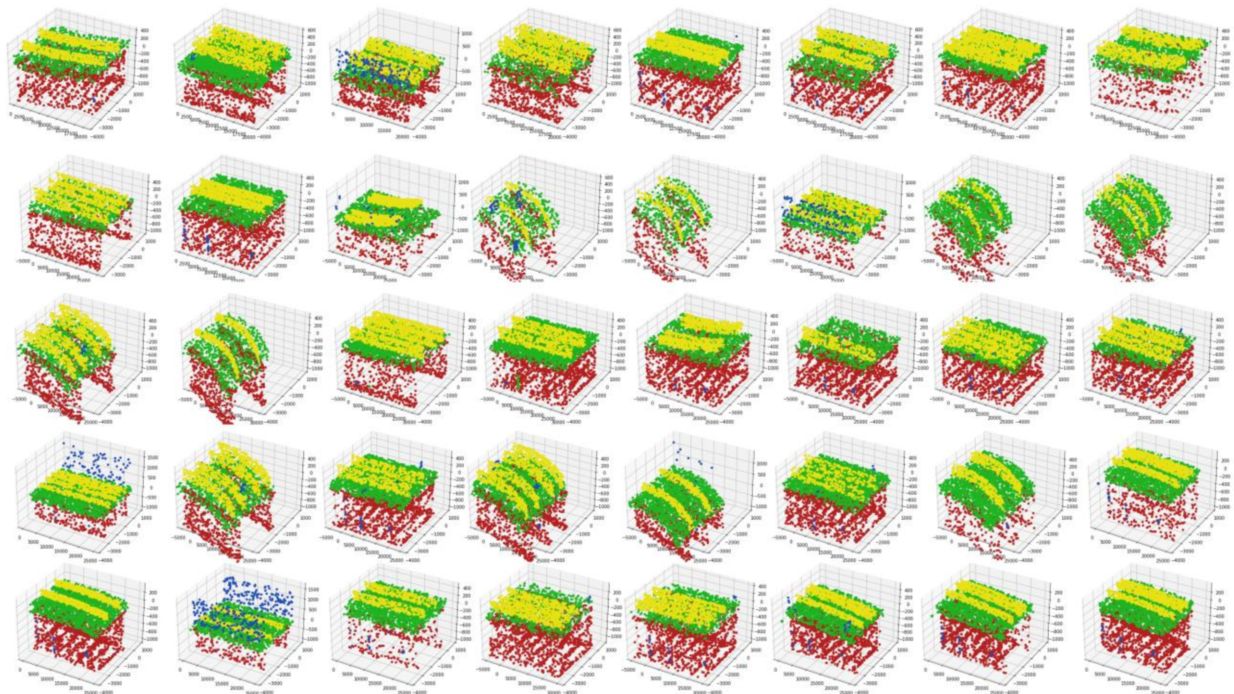
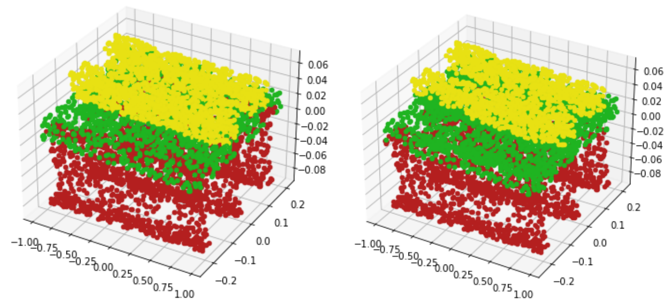


Figure 7 – Pre-segmented training database.

Segmentation Predictions

The PointNet model was trained with 1000 epochs. An L_2 multiplier 0.001 is selected. The Adam optimizing learning rate parameter of 0.001 was used. The training was conducted on an NVIDIA RTX 3080Ti. The total time to train each network was approximately 1 hour. The setup and training of the PointNet architecture were implemented in Python using Keras, NumPy, Tensorflow and other standard libraries.

The benchmark open-deck bridge was segmented following the training of the PointNet model. The predicted segmentation results are visualized in Fig. 8. Training and validation accuracies are plotted in Fig. 9. The final accuracies for the training and validation were determined as 91% and 83%, respectively.



(a) Ground Truth (b) Prediction

Figure 8 - PointNet: Prediction of benchmark.

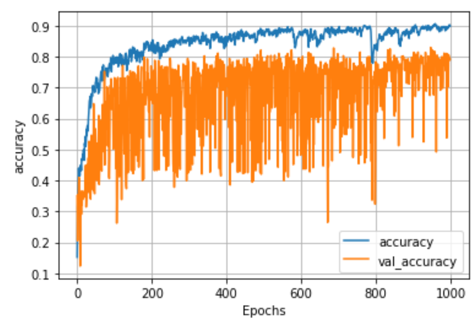


Figure 9 - Train and validation accuracy over epochs.

Finally, the nearest neighbor algorithm was used to upsample the segmented point cloud. In the example illustrated in Fig. 10, upsampling augmented the point cloud from 6000 points to 210,000 points.

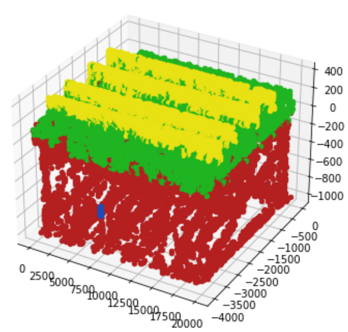
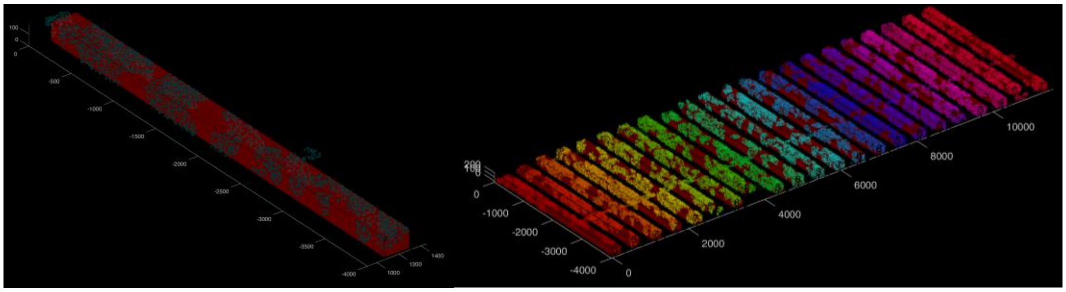


Figure 10 - Segmented and upsampled point cloud.

Geometry Fitting

For each cuboid, a 8×3 matrix of corner coordinates and a 9×1 (center X, center Y, center Z, length, width, roll, pitch, yaw) vector of best fit cuboid parameters are provided. Results from the RANSAC cuboid fitting algorithm are presented in Fig. 11.



(a) Single sleeper (b) Half of a bridge deck

Figure 11 - Results from RANSAC cuboid fitting of sleeper point cloud

CONCLUSIONS

In conclusion, a geometry extraction framework is introduced for identifying the position and alignment of sleepers and tracks on open-deck bridges. This framework utilizes UAV-based 3D scans and artificial intelligence for rapid and autonomous processing of 3D sleeper deck geometries. Due to the pilot nature of this study, open-deck bridges were synthetically generated in a computer graphics environment. Following a UAV survey, the point cloud of the structure was generated. A deep learning algorithm, called PointNet, is used to segment the point cloud into various components (e.g., sleeper, track, and girders). Finally, geometric primitives are used to convert the component level point clouds to 3D solid geometries. The potential benefits of such approach include improved bridge deck monitoring and cost savings.

REFERENCES

[1] Federal Railroad Administration, “The Freight Rail Network,” 2016. [Online]. Available: https://railroads.dot.gov/rail-network-development/freight-rail-overview.

[2] Q. Zhan, Y. Liang, and Y. Xiao, “Color-based segmentation of point clouds,” in *Laser scanning*, 2009, pp. 248–252.

[3] L. Li, F. Yang, H. Zhu, D. Li, Y. Li, and L. Tang, “An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells,” *Remote Sensing*, vol. 9, no. 5, 2017.

[4] C. Wang, X. Xiong, H. Yang, X. Liu, L. Liu, and S. Sun, “Application of Improved DBSCAN Clustering Method in Point Cloud Data Segmentation,” in *Proceedings - 2021 2nd International Conference on Big Data and Artificial Intelligence and Software Engineering, ICBASE 2021*, 2021, pp. 140–144.

[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet : Deep Learning on Point Sets for 3D Classification and Segmentation,” *In CVPR*, 2017.