

# ASPHALT ROAD LAYER DETECTION FOR CONSTRUCTION PROGRESS MONITORING

Steven Vick<sup>1</sup> and Ioannis Brilakis<sup>2</sup>

**Abstract:** Transportation construction projects consistently underperform, with an estimated \$82.6 billion globally in annual cost overruns. Current progress monitoring practices contribute to this poor performance thanks to their manual, subjective, inaccurate, and time-consuming nature. Automating this task could address these shortfalls and improve project performance. One way to accomplish this automation compares 3D Civil Infrastructure Model design surfaces to 3D point cloud reconstructions of the as-built scene. This requires automated detection of the design surfaces in the as-built data. Research in this area has focused on all-or-nothing detection of structural building components using methods that are a poor fit for large, complex, and closely-layered road design surfaces. These approaches ignore the kind of incremental progress detection needed on transportation projects. This paper proposes a method for detecting large road design surfaces in discrete regions (i.e. increments) of as-built point cloud data, contributing a novel model-guided and sparse hierarchical data structure ('layerTree') that addresses the limitations of existing state-of-the-art methods. The authors collected as-built and as-planned data during construction of a small residential road in Cambridge, UK. A total of 640 experiments on this data examined different combinations of layerTree parameters and classification rules, producing a peak accuracy of 86.62%, peak precision of 80.65%, and peak recall of 92.50%. The most balanced combination produced an accuracy of 86.50%, precision of 68.17%, and recall of 60.99%.

**Keywords:** Construction Progress Monitoring, Transportation, Drones.

## 1 INTRODUCTION

Transportation construction projects consistently underperform, with an estimated \$82.6 billion globally in annual cost overruns (Vick and Brilakis 2016). The nature of these projects contributes to this inefficiency, with sites often in rural areas that can stretch across many kilometres. This presents a logistical challenge for inspectors and managers charged with overseeing project execution, as it can be very costly and time-consuming to conduct the regular in-person site assessments needed in current practice to identify schedule deviations. In addition to these logistical progress monitoring challenges, the current methods employed for accomplishing the task lack the accuracy, consistency, reliability, and timeliness needed to enable effective project control actions. They rely on checklists and daily reports that offer room for subjective interpretation and provide no physical measurement of the work done on site. Progress monitoring remains largely a 2D process despite the increasing proliferation of 3D and 4D Civil Infrastructure Models (CIM), the use of which is mostly limited to marketing and communication of design intent

---

<sup>1</sup> PhD Candidate, Department of Engineering, University of Cambridge, Cambridge, UK, [sv364@cam.ac.uk](mailto:sv364@cam.ac.uk)

<sup>2</sup> Laing O'Rourke Lecturer of Construction Engineering, Department of Engineering, Director of the Construction Information Technology Laboratory, University of Cambridge, Cambridge, UK, [ib340@cam.ac.uk](mailto:ib340@cam.ac.uk)

and/or construction sequencing (Guo et al. 2014). A method that leverages the rich 3D information available in CIMS and accurate 3D reality capture technologies, like laser scanning or photogrammetry, could address both the logistical and performance shortfalls of the current practice in transportation construction progress monitoring.

An essential task in implementing such a progress monitoring approach is the detection of relatively thin design surface layers in 3D as-built data. In this context, detection refers to both the classification (i.e. which element is present) and localization (i.e. exactly where it is present in the scene) of design surface layers in the as-built point cloud data (PCD). This paper proposes a method for accomplishing this detection and presents experimental results on as-built data collected during the construction of a small residential road in Cambridge, UK. The following section provides background and context to this effort by reviewing recent relevant research. The proposed method is then described, and experimental results are discussed in subsequent sections. Finally, we draw conclusions and make recommendations for further research in this area.

## 2 BACKGROUND

The scope of this paper is limited to detection of layered as-planned design surfaces in as-built PCD for asphalt road construction progress monitoring. Research in this area has been very limited, so recent approaches for detection of non-layered design surfaces in as-built PCD for building construction will also be examined in this literature review.

Approaches to design object detection in building construction typically search for as-built points within distance tolerances of design surfaces after PCD-to-model registration. Bosché (2010) used a point-to-point method labelling as-built points with the nearest model element during registration. Others (e.g. Tuttas et al. 2014; Golparvar-Fard et al. 2015; Braun et al. 2015) partitioned the registered scene into equilateral axes-aligned volume elements (voxels) and/or raster cells before conducting the point-to-surface/plane search. The object detection decision has been accomplished using a decision tree based on voxel and raster cell labels (Tuttas et al. 2014), a Support Vector Machine classifier (Golparvar-Fard et al. 2015), and a threshold on recognized points (Zhang and Arditi 2013). Braun et al. (2015) used a precedence relationship graph to account for activity sequence and technical dependencies, which improved the method's robustness to occlusions. All of these methods focus on detecting progress at the object level; i.e. the design object is either there or not, and incremental progress is not considered. The spatial partitioning methods show promise, but either require analysing one surface triangle at a time (e.g. Tuttas et al. 2014) which can be costly for large and complex road surfaces, or use recursive axes-aligned subdivisions that cannot guarantee a uniform voxel 'fit' to non-planar surfaces (e.g. Golparvar-Fard et al. 2015). This poor voxel fit could result in detection conflicts, e.g. where two closely-layered surfaces intersect the same voxel, and/or inconsistent results, e.g. when a surface intersects the middle of some voxels and only a small corner of others. These drawbacks limit the effectiveness of existing methods on road construction projects, where design surfaces are large, layered and complex, and require incremental progress determination.

One study dealing directly with asphalt road design layers (Kivimaki and Heikkila 2015) used CIM surfaces for cloud-based as-built comparisons on infrastructure projects. Survey data from various project sources (e.g. GNSS surveys, Total Stations, etc.) was uploaded to a cloud-based server. The researchers then assigned points to the applicable design surface, and elevation differences between the surveyed points and design surface were calculated to determine compliance with specified tolerances. This study, however,

stopped short of classification and/or detection, as the matching of surveyed points to the relevant design surface was accomplished using a specific code assigned to each point. Such point codes must be manually set on data collection equipment prior to operation, or manually applied to various sections of the point cloud during post-processing which requires a-priori knowledge of the surveyed scene. The authors concede that manual point-to-surface matching is required; a drawback that limits the usefulness of this approach for automated progress monitoring.

Considering the shortfalls in current progress monitoring practice for road construction and limitations in the body of research, there exists a need for a method capable of incrementally detecting large, complex, and relatively-thin design surface layers in unlabelled as-built data. The next sections propose and evaluate such a method, with the stated goal of answering the following questions: How can we partition the 3D as-planned space to consistently detect road design surfaces in discrete increments? What combination of design and/or input parameters will produce the most desirable results?

### 3 PROPOSED SOLUTION

For the reasons mentioned in the previous section, we propose a novel model-guided and sparse hierarchical data structure, termed the 'layerTree', for assigning as-built points to voxels corresponding to discrete regions of the layered design surfaces. A threshold on the number of points attributed to each voxel is then used to determine if that cell is a candidate for positive detection. Finally, we resolve detection conflicts between layered cells occupying the same x-y space using a branch classification rule. Figure 1 illustrates the key concepts and parameters used in the implementation of this solution. Figure 2 on the following page provides an overview of the process.

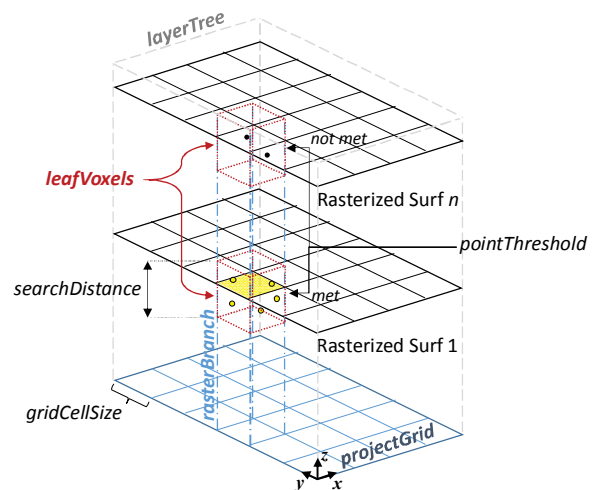


Figure 1: The voxelTree Data Structure

The first step in the process establishes a 2D grid of cells, the 'projectGrid', defining the x and y boundaries of each 'rasterBranch' in the tree structure. The lowest design surface is transformed from the CIM project coordinate system to the global origin and aligned using Principal Component Analysis so the transverse and longitudinal axes of the road are parallel to the global x- and y-axes respectively. This analytical coordinate system is used to provide stability for calculations performed using floating point numbers, and to ensure proper alignment of grid cells for the following analysis. We define an outer bounding box and subsequently divide it into square cells with side length 'gridCellSize'. This parameter defines the granularity of the ultimate progress determination; i.e. incremental progress on a design surface layer will be defined at the per-cell level. If all else is perfect in the detection algorithm, this gridCellSize parameter defines the acceptable level of error in the final determination, as projectGrid cells located in areas of as-built transition between layers will be partially incorrect in their classification.

In the following step, we generate the layerTree by projecting projectGrid cells onto the layered design surfaces and building 'leafVoxels' at each projected cell location. This construction topology explains the model-guided and sparse nature of the solution, as voxels are only generated where rasterBranches intersect a design surface. The 'searchDistance' parameter defines the leafVoxel height oriented normal to the design surface within the rasterBranch boundaries. The selection of the optimal searchDistance parameter is a focus of experiments in the following section, but care should be taken to ensure it is selected to limit overlap with voxels from adjacent design surface layers.

Next, we populate the leafVoxels with as-built points. The registered PCD is transformed to the analytical coordinate system, and an octree is constructed to enable rapid identification of only those points nearest to each leafVoxel. A hull cropping algorithm then assigns points to the leafVoxels. First, we traverse the as-built octree to identify only those points nearest the leafVoxel in question. This is done to limit the number of points fed into the hull cropping algorithm. The algorithm then crops these points using a 3D hull constructed from the leafVoxel such that only points falling within this volume are assigned to the voxel. We use a threshold ('pointThreshold') on the number of points assigned to label each leafVoxel as a 'candidate' or 'non-candidate' for detection. This threshold is designed to eliminate voxels containing sparse noise from further consideration. The optimal pointThreshold value is another focus of the experiments discussed in the next section.

Given the closely-layered nature of the design surfaces, random noise in the as-built data exceeding the pointThreshold could result in multiple leafVoxels being marked 'candidate' in the same rasterBranch region. To address this and correct any potential detection conflicts, we implement a further rasterBranch classification rule. This study considers three possible rules: maximum points (i.e. select the voxel with the most points), minimum distance (i.e. select the voxel with the minimum average orthogonal distance between the as-built points and the design surface), and minimum vertical variance (i.e. select the voxel whose points have the lowest variance in the z-direction). The following section tests this solution for a variety of different parameters using each of the three classification rules.

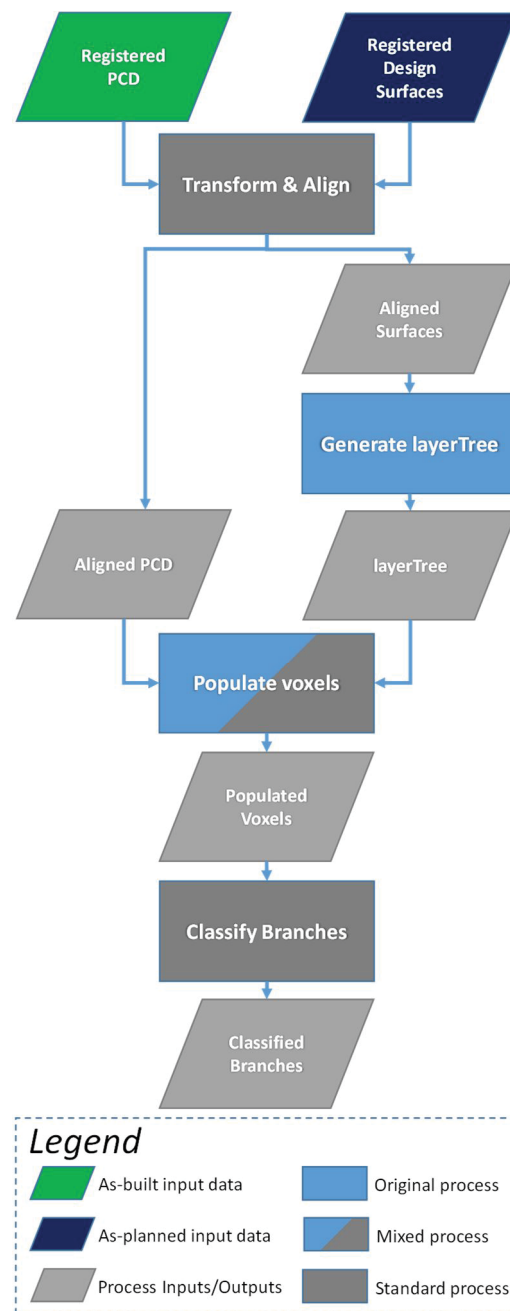


Figure 2: Solution process

## 4 METHODOLOGY, EXPERIMENTS & RESULTS

As-built and as-planned data were collected on the construction of a residential road for a new development in Cambridge, UK. The asphalt road design included five distinct surfaces: (1) the formation, or bottom of excavation, (2) a 520 mm thick sub-base layer, (3) a 125 mm thick base course, (4) a 65 mm thick asphalt binder course, and (5) a 40 mm thick asphalt wearing course. To facilitate the following experiments, we generated a 3D corridor model from the design information provided by the contractor using Autodesk's Civil 3D application.



Figure 3: Trimble UX5

We collected as-built aerial imagery using Trimble's UX5 fixed wing unmanned aerial vehicle (UAV), shown in Figure 3. This method of data collection was selected because it is the most cost- and time-effective technology capable of producing accurate survey data for large areas (Vick and Brilakis 2016). The UX5 collected a total of 224 images during the 11:13 minute flight, with a ground sample distance of 0.02 m from an average flying height of 93.5 m above ground level. We performed photogrammetric post-processing of the collected images in Trimble's Business Center application. A total of eight GNSS-surveyed ground control targets were used to georeference the photogrammetric products in the design model's coordinate system, requiring no further registration between the model and as-built data. The post-processing produced an orthorectified image and point cloud with an average density of 335 pts/m<sup>2</sup>. We selected the southern-most section of the road to run the following experiments because it contained a transition between two relatively thin layers (base course and asphalt binder) at the time of data collection. Figure 4 depicts the site, photogrammetric surveying products, and ground truth conditions.

We conducted a series of 640 experiments testing different combinations of

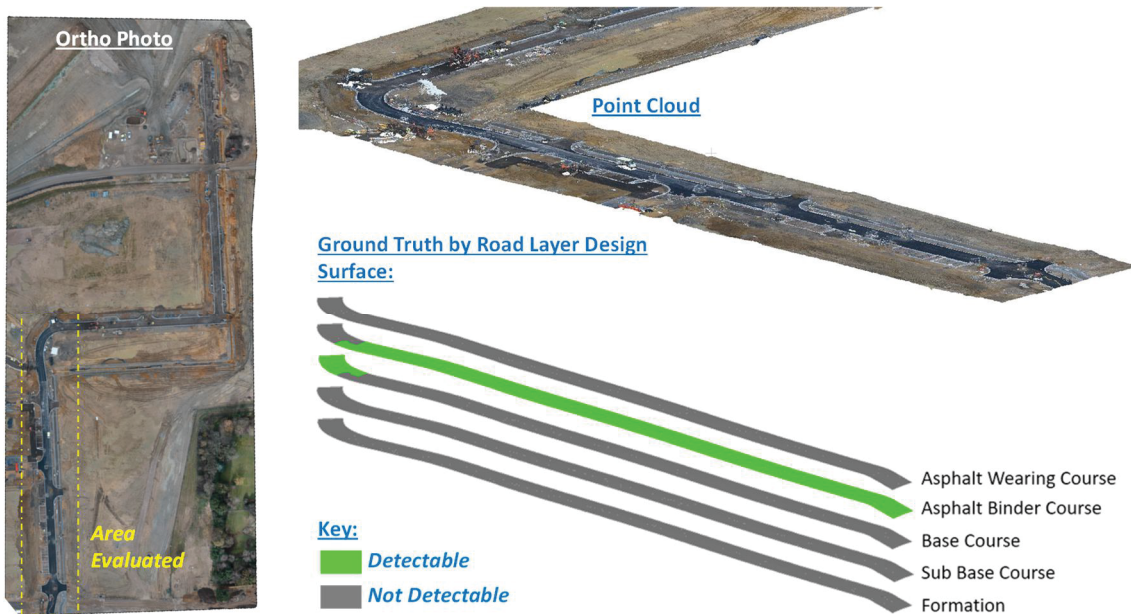


Figure 4: As-built data and ground truth conditions searchDistance and pointThreshold parameters for the three classification rules discussed

in the previous section. The experiments also considered the 'no classification' case, skipping the final step shown in Figure 2 to provide a better feel for the impact of the classification rules. The `gridCellSize` parameter was set to 0.5m and controlled for in the following experiments based on conversations with local project management personnel. For each of the four classification cases, we varied the `searchDistance` between 0.01m and 0.04m in 0.01m increments and tested `pointThreshold` values between 1 and 40. Design layer thickness drove selection of the `searchDistance` values tested, being careful to limit overlap in grid cell voxels between adjacent design surface layers. The range of `pointThreshold` values was determined experimentally, running enough iterations to ensure an observable peak in accuracy. We ran all experiments on a computer with a 4.0 GHz Intel i7 processor, 32 GB of RAM, a dedicated 1,280-core GPU with 2 GB memory, and Windows 10 64-bit operating system. Average accuracy, precision, and recall were used as the performance metrics (Equations 1-3), where  $c$  is the number of design surface classes, true positive ( $TP_i$ ) is the number of leafVoxels correctly detected, false positive ( $FP_i$ ) is the number incorrectly detected, true negative ( $TN_i$ ) is the number correctly undetected, and false negative ( $FN_i$ ) is the number incorrectly undetected for each class respectively. Table 1 lists the top results and the combination of `searchDistance` (sD) and `pointThreshold` (pT) parameters that produced each. We include a fourth metric, 'Balance' calculated as the mean of the three performance metrics, to highlight the combination of parameters producing the most balanced results. Figure 5 depicts the results for the combination of parameters that produced the most balanced results.

$$\text{Average Accuracy} = \frac{\sum_{i=1}^c \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{c} \quad (1)$$

$$\text{Average Precision} = \frac{\sum_{i=1}^c \frac{TP_i}{TP_i + FP_i}}{c} \quad (2)$$

$$\text{Average Recall} = \frac{\sum_{i=1}^c \frac{TP_i}{TP_i + FN_i}}{c} \quad (3)$$

Table 1: Maximum performance metric values

Classification Rule	Avg. Accuracy	Avg. Precision	Avg. Recall	Balance
None	86.013% sD: 0.02, pT: 15	80.654% sD: 0.01, pT: 29	92.502% sD: 0.04, pT: 1	71.757% sD: 0.03, pT: 14
Maximum Points	86.623% sD: 0.04, pT: 27	80.654% sD: 0.01, pT: 29	62.981% sD: 0.03, pT: 1	71.891% sD: 0.03, pT: 10
Minimum Distance	86.501% sD: 0.04, pT: 29	80.654% sD: 0.01, pT: 29	60.885% sD: 0.04, pT: 1	70.880% sD: 0.04, pT: 18
Minimum Variance	85.825% sD: 0.02, pT: 16	80.654% sD: 0.01, pT: 29	53.677% sD: 0.02, pT: 11	69.080% sD: 0.02, pT: 15

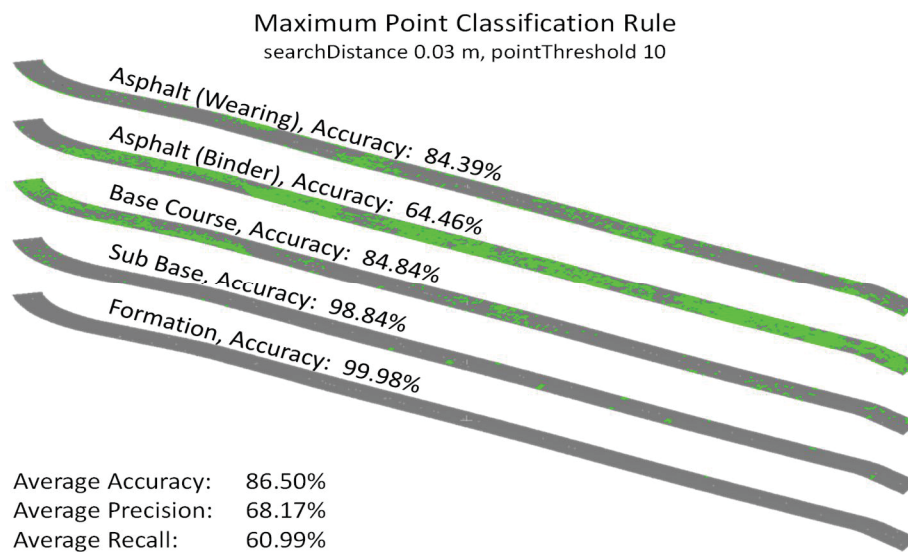


Figure 5: Results for most balanced mix of parameters

## 5 SUMMARY AND CONCLUSIONS

This study presented a novel approach for detecting incremental progress on layered asphalt road design surfaces in PCD, contributing a model-guided and sparse hierarchical data structure known as the layerTree to accomplish the task. One advantage of the layerTree is its ability to uniformly fit voxels to non-planar surfaces, allowing for a more consistent point-to-surface assignment compared to standard space partitioning approaches like octrees. Additionally, limiting the detection decision to discrete regions enables identification of incremental progress on large design objects, unlike existing methods focused on all-or-nothing detection of building structural elements. We conducted 640 experiments testing the proposed method's performance across a wide range of input and design parameter combinations for four separate classification cases. The results show promise, but also highlight a need for further research. The poor reliability results (as measured by precision and recall) appear to be caused by a combination of misclassification, surface occlusion, and potential construction error (see Figure 6). Our future research will examine the automated classification and handling of these different error cases to improve overall performance. Interestingly, the maximum recall achieved prior to classification (92.5%) indicates that the required progress information is available in the data. However, the drastic reduction in recall following classification indicates that a more sophisticated classification rule (or set of rules) may be required to improve performance. Further research in this area will examine additional classification approaches along with inclusion of a priori construction sequencing and technique information to improve results.

## 6 ACKNOWLEDGMENTS

This research is made possible through funding from the Cambridge Trust and the United States Air Force. The views expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of the Air Force, the Department of Defense or the U.S. Government.

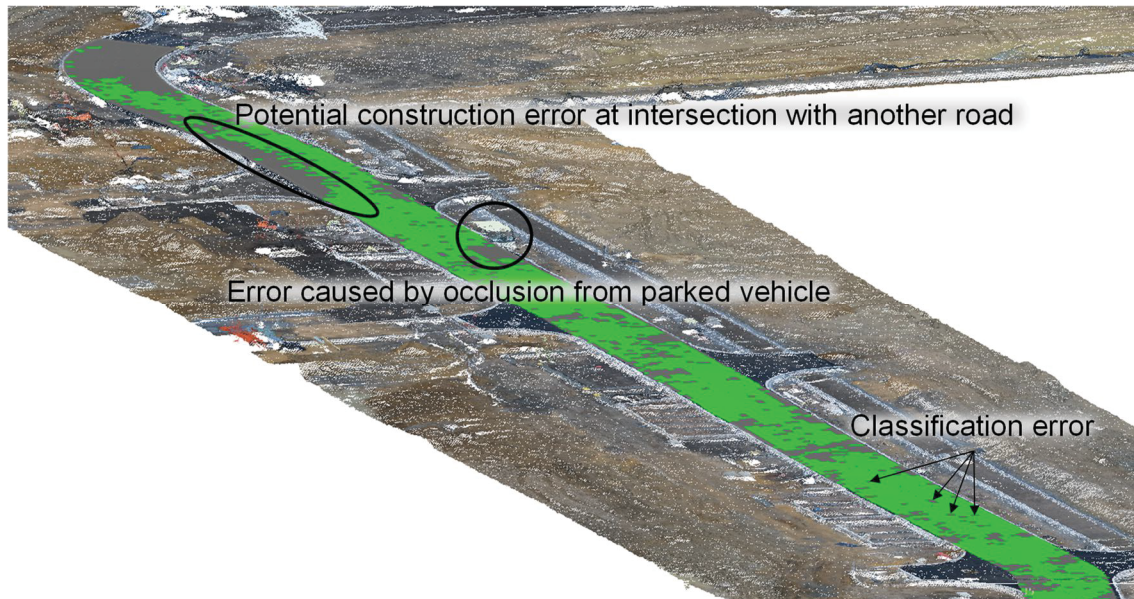


Figure 6: Binder layer results overlaid with PCD

## 7 REFERENCES

- Bosché, F. (2010). Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction. *Advanced Engineering Informatics*. 24 (1). p.pp. 107–118.
- Braun, A., Tuttas, S., Borrmann, A. and Stilla, U. (2015). A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds. *ITcon*. 20. pp. 68–79.
- Golparvar-Fard, M., Peña-Mora, F. and Savarese, S. (2015). Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models. *Journal of Computing in Civil Engineering*. 29 (1). p.p. 4014025.
- Guo, F., Turkan, Y., Jahren, C.T. and Jeong, H.D. (2014). Civil Information Modeling Adoption by Iowa and Missouri DOT. *Computing in Civil and Building Engineering*. p.pp. 463–471.
- Kivimaki, T. and Heikkila, R. (2015). Infra BIM based Real-time Quality Control of Infrastructure Construction Projects. In: *Proceedings of the 32nd International Symposium on Automation and Robotics in Construction and Mining*. 2015, Oulu, Finland, pp. 877–882.
- Tuttas, S., Braun, A., Borrmann, A. and Stilla, U. (2014). Comparison of Photogrammetric Point Clouds with BIM Building Elements for Construction Progress Monitoring. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. XL-3. pp. 341–345.
- Vick, S.M. and Brilakis, I. (2016). A Review of Linear Transportation Construction Progress Monitoring Techniques. In: *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering*. 2016, Osaka, Japan, pp. 1106–1113.
- Zhang, C. and Arditi, D. (2013). Automated progress control using laser scanning technology. *Automation in Construction*. 36. p.pp. 108–116.