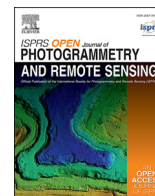


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

## ISPRS Open Journal of Photogrammetry and Remote Sensing

journal homepage: [www.editorialmanager.com/OPHOTO](http://www.editorialmanager.com/OPHOTO)

# Pavement distress detection using terrestrial laser scanning point clouds – Accuracy evaluation and algorithm comparison



Ziyi Feng<sup>a,\*</sup>, Aimad El Issaoui<sup>a,1</sup>, Matti Lehtomäki<sup>a</sup>, Matias Ingman<sup>b</sup>, Harri Kaartinen<sup>a,c</sup>, Antero Kukko<sup>a,b</sup>, Joonas Savela<sup>a</sup>, Hannu Hyyppä<sup>b</sup>, Juha Hyyppä<sup>a,b</sup>

<sup>a</sup> Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute (FGI), the National Land Survey of Finland, Geodeetinrinne 2, 02430, Masala, Finland

<sup>b</sup> Department of Built Environment, Aalto University, 02150, Espoo, Finland

<sup>c</sup> Department of Geography and Geology, University of Turku, 20500, Turku, Finland

## ARTICLE INFO

## Keywords:

Terrestrial laser scanning  
Pavement  
Road  
Crack  
Distress  
Point cloud

## ABSTRACT

In this paper, we compared five crack detection algorithms using terrestrial laser scanner (TLS) point clouds. The methods are developed based on common point cloud processing knowledge in along- and across-track profiles, surface fitting or local pointwise features, with or without machine learning. The crack area and volume were calculated from the crack points detected by the algorithms. The completeness, correctness, and  $F_1$  score of each algorithm were computed against manually collected references. Ten 1-m-by-3.5-m plots containing 75 distresses of six distress types (depression, disintegration, pothole, longitudinal, transverse, and alligator cracks) were selected to explain variability of distresses from a 3-km-long-road. For crack detection at plot level, the best algorithm achieved a completeness of up to 0.844, a correctness of up to 0.853, and an  $F_1$  score of up to 0.849. The best algorithm's overall (ten plots combined) completeness, correctness, and  $F_1$  score were 0.642, 0.735, and 0.685 respectively. For the crack area estimation, the overall mean absolute percentage errors (MAPE) of the two best algorithms were 19.8% and 20.3%. In the crack volume estimation, the two best algorithms resulted in 19.3% and 14.5% MAPE. When the plots were grouped based on crack detection complexity, in the 'easy' category, the best algorithm reached a crack area estimation MAPE of 8.9%, while for crack volume estimation, the MAPE obtained from the best algorithm was 0.7%.

## 1. Introduction

Pavement deterioration is usually the result of inappropriate road design and maintenance, improper construction material, overloading, poor road surface drainage, seepage, and challenging climate factors such as frost. Road distresses slow traffic flow and affect road safety, resulting in increased fuel costs, extended travel time, etc. for road users. It is crucial to identify road distress at an early stage, because preventive road maintenance and effective remedies can be carried out before the distress worsens, or the pavement becomes completely unqualified. Proper, timely, and selective road maintenance extends pavement lifetime and decreases maintenance cost.

The required accurate information of the road conditions for maintenance is insufficient.

Currently, when road condition is inspected manually, the inspector

travels along the road to find possible distress elements. However, the process is slow, costly, and laborious, and the traffic hazards entail potential risks for the inspection. An automated distress detection system is therefore needed to quantify the quality of road surfaces, which provides assistance in determining and planning road network maintenance.

Lots of research has been conducted in the last two decades that has aimed to develop pavement distress recognition and detection algorithms based on 2D image intensity. The crack depth information is not offered by these methods. Moreover, digital image quality and resolution limit the image analysis, which makes it challenging to perform fully automated crack detection in various lighting and poor intensity contrast conditions (Tsai and Li, 2012). In other words, 2D image-based data acquisition methods are sensitive to lighting effects. Natural features such as shadows, illumination changes, uneven crack widths, and low intensity contrast between cracks and surrounding pavement surfaces

\* Corresponding author.

E-mail address: [ziyi.feng@maanmittauslaitos.fi](mailto:ziyi.feng@maanmittauslaitos.fi) (Z. Feng).

<sup>1</sup> Co-first author.

<https://doi.org/10.1016/j.ophoto.2021.100010>

Received 28 June 2021; Received in revised form 20 October 2021; Accepted 29 November 2021

Available online 3 December 2021

2667-3932/© 2021 The Authors. Published by Elsevier B.V. on behalf of International Society of Photogrammetry and Remote Sensing (isprs). This is an open access

article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

therefore have serious effects on the function of image-based crack detection methods. Furthermore, non-crack features, e.g. joints, sealed cracks, and white painting marks, can also be erroneously identified as crack features. The performance of 2D intensity images has been improved by using new computational methods, mathematical morphology, structured learning, and artificial intelligence such as machine learning, including neural networks and deep learning (Hu and Zhao, 2010; Zou et al., 2012; Salman et al., 2013; Shi et al., 2016; Guan et al., 2014; Oliveira and Correia, 2008, 2012, 2014; Varadharajan et al., 2014; Zhang et al., 2016; Fan et al., 2019; Zhong et al., 2020). However, the selection of the parameter values of these methods is challenging, because it strongly depends on crack variations and image quality.

According to Zhong et al. (2020), excellent performances have been achieved in recent years using neural networks and deep learning technology, but crack recognition is still challenging when analysing based on 2D images, which are frequently obscured by shadows, illumination, stains, rust, and noise.

Currently, the most promising results have been achieved using 3D measurements. The Lidar (light detection and ranging) technique is not sensitive to lighting effects. Cracks can be identified with a segmentation algorithm as long as the crack depth is large enough to be recognised. Recently, several studies or reviews have been conducted with Lidar-related crack detection (Kunyan et al., 2015; Mathavan et al., 2015; Medina et al., 2014; Guan et al., 2014, 2015; Laurent et al., 2008; Li et al., 2009; Zhang et al., 2015; Yu et al., 2014; De Blasiis et al., 2020; Ravi et al., 2020; Tsai and Chatterjee, 2018; Tsai and Yang, 2020; Yang et al., 2021). De Blasiis et al. (2020) used MLS data from a 100-m-long urban road stretch to evaluate surface distress focusing on potholes, swells, and shoves wider than 100 mm using several local planes. Tsai and Chatterjee (2018) developed a watershed method to detect potholes by using 179 images based on 3D pavement data. They classified images containing potholes, and reported an accuracy of 95.0%. Ravi et al. (2020) highlighted the application of pothole detection and patching quantity estimation in their study which focused on mapping potholes from a 10-km-long MLS dataset with a relative accuracy of  $\pm 1$ –2 cm. Tsai and Yang (2020) reported a 6-years-long change detection study using 3D laser technology continued by Yang et al. (2021) analysing crack length changes.

According to the best studies, it can be stated that with mobile laser scanning (MLS), road cracks that are wider than 2 mm can be detected with 95% precision even in the case of low intensity contrast and lighting conditions (Zhang et al., 2015). Additionally, Guan et al. (2015) claimed that road cracks with widths greater than 2 cm could be detected, achieving a completeness of 96% and correctness of 85%. Zhong et al. (2020) reported that transverse, longitudinal, and oblique crack detection achieved  $F_1$  scores of 96.6%, 87.1%, and 81.5% respectively, using an approximately 220-m stretch of highway containing numerous cracks. Choi et al. (2016) reported a crack detection rate of 86.4%. Ragnoli et al. (2018) reviewed pavement distress detection methods and confirmed terrestrial laser scanning (TLS) feasibility to survey medium and high severity cracks, but considered the state of TLS automation for distress mapping poorly developed. Barbarella et al. (2018) suggested a field process for terrestrial laser scanner data collection and a data processing workflow to measure the size of the fault at each joint of apron slabs for rigid airport pavement management. In all the laser-based studies cited above, either the amount and type of studied cracks or the quantitative accuracy evaluation was limited or even missing. In addition, the accuracy of the crack area and volume obtainable from TLS have not been reported in previous studies.

Accordingly, we can summarise the shortcomings of the state-of-the-art in using 3D Lidar for pavement crack detection. To the best of our knowledge, there is

- a need for studies reporting accuracies in real-life conditions and experimental set-ups in detail;

- a need for comparison studies of the performance of various algorithms for crack detection, crack area, and volume estimation;
- a need for studies reporting the crack detection accuracies versus parameters such as crack width.

Therefore, this paper reports the performance of static terrestrial laser scanning (TLS) in crack detection, using five different crack detection algorithms developed based on point cloud processing expertise. Crack depth, crack areal extent, and volume were the major parameters to be extracted from TLS data. The main purpose of this publication is to provide a comparative analysis of these algorithms for mapping pavement distresses using highly accurate point cloud data. The study results provide an indication of which methods are best suited for pavement quality analysis and which methods could be further developed for the operational use of distress mapping with mobile laser scanning. The algorithm comparison was performed for ten test plots, extracted from a 3-km-long road, and key pavement distress features, such as depth, area and volume, were extracted fully automatically.

## 2. Experimental plan, materials, and methods

### 2.1. Data description

In this study, TLS point cloud data collected statically from the roof of a car (Fig. 1) were used for algorithm comparison. The data were collected from a 3-km road section in the municipality of Kirkkonummi, Southern Finland, where 64 1-m-by-3.5-m samples of road surface (called plots hereafter) were measured. Of these 64 plots, 10 plots containing significant pavement distresses were selected for this study. Scans were performed with FARO Focus S 350 (FARO technologies, Inc., Lake Mary, FL, USA) phase-shift laser scanner, which has a measurement speed up to 976,000 points/second, and it is using 1550 nm wavelength. The mean point spacing of the TLS data on the road was 2.3 mm, and the point density in test plots was approximately 150,000 points/m<sup>2</sup>. El Issaoui et al. (2021) used the same dataset as a reference for mobile laser scanning (MLS) data in road rut depth measurements, and the accuracy of the TLS data was verified by photogrammetric measurements to be better than 0.5 mm. For more detail, the reader is referred to El Issaoui et al. (2021).

In the pre-processing, the TLS data from the road surface were rotated by calculating the angle between the x-axis and a line created across the road to align the road surface in the xy direction, allowing the data to be easily divided into different sections for a profile based and surface fitting methods. The TLS data consisted of a tilted scan pattern from the across-direction on the road. For profile-based methods, such data had to be pre-processed in across-track profiles. Thus, the profiles were created by resampling the data into 3-mm-wide slices (since point spacing was averaged at 2.3 mm) along the across-track direction (x direction), and each slice was classified as an individual profile. Similarly, along-track profiles (y direction) were formed.

### 2.2. Ground truth

In Finland, a pavement crack should be repaired when wider than 5 cm (Finnish Transport Infrastructure Agency, 2013). As early detection of road damage is needed for road maintenance planning, we decided to segment all surface distresses more than 1 cm wide. The segmentation was done manually using CloudCompare (version 2.10.2, <http://www.cloudcompare.org/>). To make classification easier and more accurate, a local elevation difference was used as follows: as pavements consist of planar features, test plots were first cut into smaller areas in the xy plane, after which the selected point cloud was levelled, making it parallel with the xy plane. Variations in height were used to extract the pavement distress points from the surrounding pavement points. The result was further improved using the CloudCompare Segment tool during the visual inspection. Fig. 2 shows the total area and volume of each manually

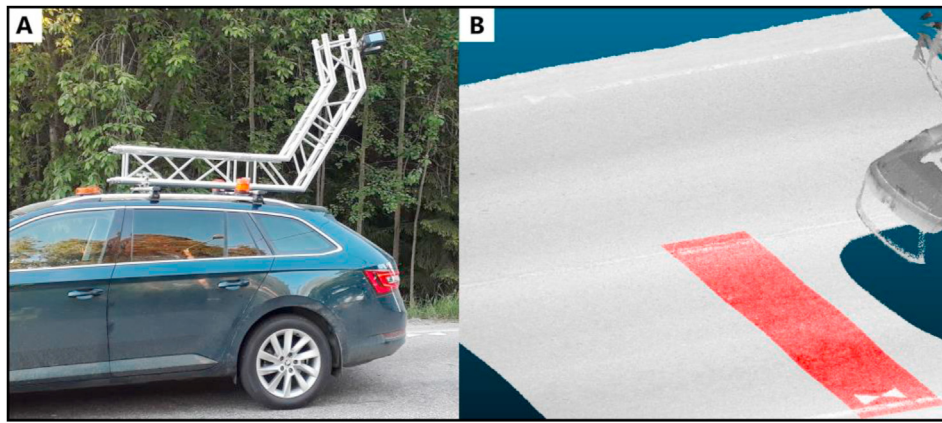


Fig. 1. Principle and measuring system for static TLS measurements in this study. Figure A shows the TLS scanner mounted on the roof of the car. Figure B shows the point cloud provided by TLS. The plot area is marked in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

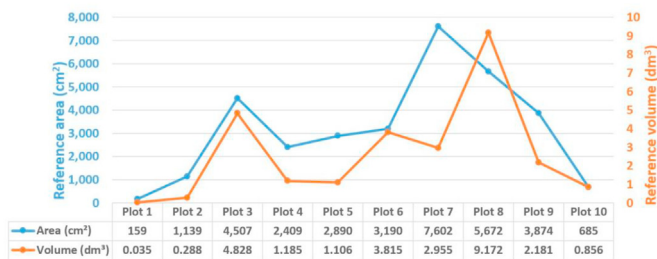


Fig. 2. Areal extent and volume of manually segmented cracks by plot.

segmented plot, and it can be seen that the total crack volume and areal extent varied significantly across plots.

During the segmentation, the plots were classified into three groups according to their level of difficulty: easy, medium, and difficult. The classification was based on the complexity of the manual classification. If the plot contained clear cracks that were easily identifiable, the plot was easy. Difficult test plots were those where determining the edge of the distress was difficult as the distress was no longer clear. Examples of such are potholes that have been patched multiple times and in which the patches have partially erupted, which has resulted in several types of distresses. In this case, the determination of such distress can have subjective differences depending on the individual.

### 2.3. Crack area and volume calculation

Due to the irregular shapes of the cracks, the crack area was analysed pixel-wise at a plot level. Considering the adjacent points' average distance of the studied TLS data is 2.3 mm, the pixel size was set to be 5 mm to ensure the crack area calculation accuracy and avoid gaps between crack pixels. The plot was divided into 5-mm-by-5-mm pixels, and crack points were allocated to corresponding pixels according to their coordinates. A pixel containing at least one classified crack point was treated as a crack pixel. Finally, the crack area was estimated corresponding to the number of crack pixels. Fig. 3a presents an example plot, and Fig. 3b is the detail of the partial plot.

The crack volume was calculated using an automated tool developed for this study. The tool consisted of two parts: partitioning and volume calculation. In the partitioning phase, the aim was to divide the point cloud of segmented distresses into 40-cm-by-40-cm sections. The operating principle of the volume calculation phase was based on the assumption that the points surrounding the detected distress were pavement points, in which case fitting the plane to these points would create an ideal pavement surface over the distress. The distance of each distress point to the pavement plane was calculated, and the sum of the volumes corresponded to the volume of the whole distress. Sometimes, in the partitioning phase, the edge of the 40-cm window was placed on top of the distress so that the window did not contain both edges of the distress. In these cases, the neighbouring point must be interpolated with

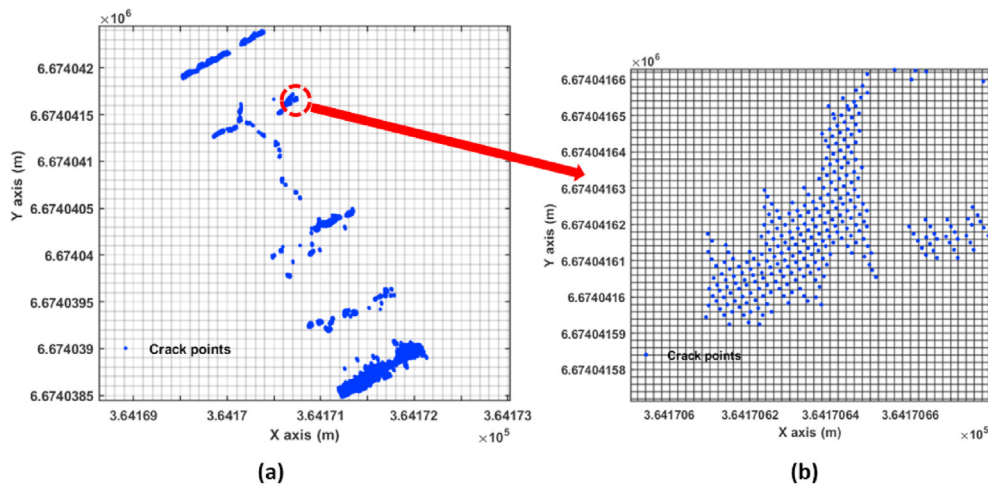


Fig. 3. (a) Crack points of one plot in the pixel map; (b) part of the crack pixel map.

the nearest pavement points to enable plane fitting. Fig. 4 illustrates the effect of selecting neighbouring points from different distances. To improve the tool's accuracy, neighbouring points were selected from a distance of 1–1.5 cm. Underestimation of volume can thus be avoided.

#### 2.4. Accuracy evaluation methods

Crack detection accuracy was evaluated pixel-wise, using the same rasterization as in the crack area analysis (see Section 2.3).

In our study, a confusion matrix (Table 1) is used to evaluate the crack detection accuracy of each algorithm, with the reference data as the actual class. The crack pixel is assumed to be 'positive'; otherwise the pixel is 'negative'. For every algorithm, the crack detection results of each plot will be compared with reference data, resulting in a corresponding confusion matrix.

In a confusion matrix, there are four possible comparison results, which are named true positive (TP), true negative (TN), false negative (FN), and false positive (FP) respectively. TP and TN indicate the number of actual positive and negative objects predicted accurately, while FN and FP show different types of prediction error.

Completeness, correctness, and  $F_1$  score are three commonly used statistical measures to evaluate the detection accuracy of the test results. They are calculated for each algorithm. Completeness measures the percentage at which the actual positive objects are identified, and it is calculated using equation (1). Correctness indicates the accurate positive rate among the objects identified by the algorithm, and it is calculated using equation (2).

$$\text{Completeness} = TP / (TP + FN), \quad (1)$$

$$\text{Correctness} = TP / (TP + FP). \quad (2)$$

Completeness and correctness measure different aspects of the algorithm, while the  $F_1$  score considers both completeness and correctness, and offers an overall accuracy estimation of the algorithm. More specifically, the  $F_1$  score is the harmonic mean of completeness and correctness, and it is calculated by equation (3):

$$F_1 = \frac{2}{\text{Completeness}^{-1} + \text{Correctness}^{-1}}. \quad (3)$$

Concerning the crack area and volume, in this study, one plot accounted for 35% of the total distress of all plots combined. The mean absolute percentage error (MAPE) was well suited for this study, as a single high-volume plot does not significantly dominate the outcome of the results. Therefore, to compare the performance of the developed algorithms, MAPE was adopted. Assuming the estimated values are denoted by  $\hat{y}_i$ , and the manually measured reference values are denoted by  $y_i$ , where  $i$  is the plot number, MAPE over the  $N$  plots was calculated as follows:

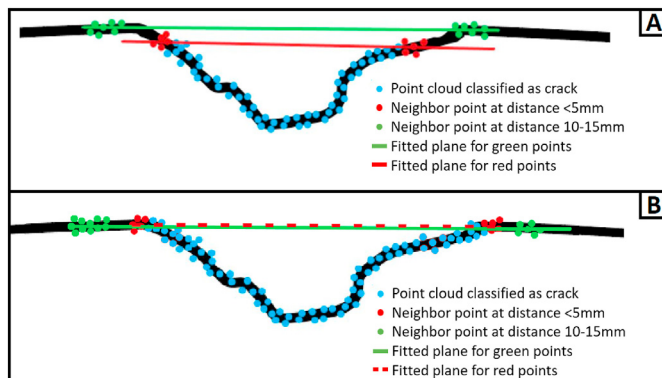


Fig. 4. Effect of selecting neighbourhood points on the plane fitting, when distress is underestimated (A) or perfectly/overestimated (B).

Table 1  
Confusion matrix.

		Predicted class	
		Positive	Negative
Actual class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

$$\text{MAPE} = \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times \frac{100\%}{N}. \quad (4)$$

The root mean square error (RMSE) is also used in the results analysis, and it is defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{N}}. \quad (5)$$

In addition, to determine the correlations of the algorithms, a coefficient of determination ( $R^2$ ) is also calculated in the results section.

### 3. Crack detection algorithms

#### 3.1. Baseline algorithm: the first derivative of height

A crack causes a sharp elevation dip. An intuitive and common method to detect cracks is therefore to examine the elevation dips in point cloud neighbourhoods. Once the elevation dip exceeds a pre-set threshold value, the corresponding point will be treated as a crack point. This method was applied to the cross-track profiles. Fig. 5 demonstrates how the baseline method finds the crack points in one profile.

The only parameter of the method, that is, the height change threshold, was equal to three times the standard deviation of all the elevation changes between neighbouring points in one profile.

In some large cracks with smooth bottom valleys, the height variation at the crack bottom was similar to the normal road surface. In this case, the crack points located at the bottom could not be detected using the height difference. Therefore, during the detection process, three flags (slope\_down, slope\_up, and on\_bottom) were set to indicate whether the detected points are located outside or inside the crack, so as to guarantee the crack points at the crack bottom can be found.

Another potential issue for this method is the profile gradient. The road surface is designed to have a certain level of gradient for water drainage, and the slope may become larger due to pavement distress. The

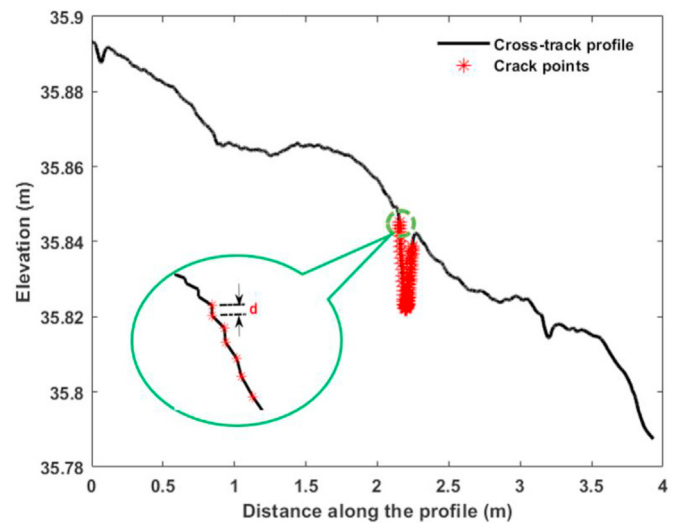


Fig. 5. Crack detection results with neighbouring points elevation inspection (d is an example of the elevation change of the neighbouring point).

elevation change of neighbouring points may therefore exceed the threshold value because of the distress, resulting in an incorrect crack point detection. Fig. 6 presents the brief algorithm process.

### 3.2. Profile-based filtering algorithm

In El Issaoui et al. (2021), a digital filter was used to reduce the data noise for a road rut study. The same digital filter with different parameter values could fit the ideal road profile and reduce the effect of cracks to the greatest extent possible. In other words, an ideal road profile without cracks was generated by applying the filter on the original profile. Then, the original profile was compared with the ideal profile without cracks, and the height differences between the points in the original profile and the corresponding points in the ideal profile were acquired. The points with a height difference exceeding a pre-set threshold value were selected as the crack points. Fig. 7 presents how the method works in one profile, and  $d$  in the figure is the height difference of the first crack point in the original profile and the corresponding point in the ideal profile. To optimize the crack detection accuracy, the method was applied to both across- and along-track profiles.

In this method, the filter parameters, namely cut-off frequency and order number, are important for the filter design. Obviously, crack detection accuracy will be affected by how much the designed filter can minimise the effect of cracks. The cut-off frequency can be figured out by analysing the signal frequency components. In our implementation, to generate an ideal across-track profile, the cut-off frequency was 200 Hz and the order was 130, while for the profiles in an along-track direction, the cut-off frequency was 50 Hz with an order of 80. In both cases, the crack threshold value was 1.5 mm.

Through the crack points demonstration in Fig. 7, there were always some crack points near the start and end part of the crack missing because of the imperfect fitting profile. This could be improved by adjusting the fitting filter parameter values, but could not be eliminated. In addition, the pre-set threshold value was an experience value based on the overall road conditions in all the plots, which did not work perfectly in certain plots with special crack types. Fig. 8 presents the general process of the algorithm.

### 3.3. Surface fitting algorithm

In this study, a surface fitting-based crack detection algorithm (SF) was developed. SF fits a polynomial surface to the road surface and separates pavement points from those points that remain below the surface, classifying them as road damage points, following polynomial curve fitting research such as (Arlinghaus, 1994; Johnson and Williams, 1976; Su et al., 2015). Our method is based on the assumption that a small piece (0.5 m × 0.5 m in this study) of asphalt would form a plane in the ideal situation. However, in reality, the road surface is not a perfect plane, as the rut depths and other changes in the road surface caused by the use of the road already cause significant differences in the flatness of the road. We therefore used a quadratic polynomial surface to approximate the shape of the road surface in the small pieces. Fig. 9A presents a surface fitted to a small test patch (50 cm × 50 cm). Fig. 9B shows a side view of the same surface, where the road pavement damage points can be clearly seen below the surface.

The workflow in the SF algorithm consists of two parts: data pre-processing and surface fitting. The surface fitting algorithm divides input data into 40-cm-by-40-cm rectangles, which in this case acts as an

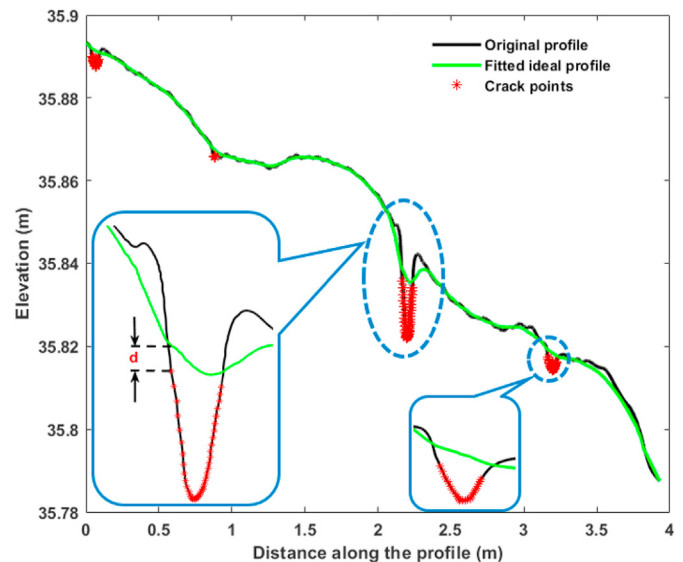


Fig. 7. Crack detection using profile-based filtering.

inspection window. In addition, a 50-cm-by-50-cm area is formed on top of the inspection window, which acts as a fitting window. The centres of both windows are the same. The fitting window, as its name implies, is the part of the data to which the polynomial surface is fitted. This method ensures that the fitting windows have a side coverage of 20% to their neighbouring windows, allowing a better continuous extraction of road damages at the edges of inspection windows. The fitted surface is compared to the data inside the inspection window, and distances between the dataset and fitted surface are calculated, after which all distances exceeding the threshold value are classified as damage points. The algorithm then starts a new iteration by removing the crack points from the fitting window dataset, allowing a better surface fitting. The algorithm implements only two iterations, as during the study it was found that most of the damage points could already be extracted in the first iteration phase, in which case the second iteration has only been used to slightly improve the result. Fig. 10 shows the workflow of the algorithm.

Fitting windows on the edge of the plot lacked points outside the plot, but we found that this had no effect on crack detection. Furthermore, in operational road quality assessment, this kind of phenomenon does not exist, because MLS data extend outside the area of interest.

### 3.4. Local surface roughness algorithm

Cracks cause discontinuities to the smooth road surface, and they can be detected by investigating how closely the surface resembles a plane – or how rough the surface is – locally. Therefore, local surface roughness was estimated for each point in the point cloud. The surface roughness measure adopted in this study was based on the variability of the surface in the surface’s normal direction. The principal component corresponding to the smallest eigenvalue of the local covariance matrix was used as an estimate of the local surface normal (Lehtomäki et al., 2016). The local covariance matrix for a particular point was estimated using the points in the local neighbourhood of the point; the radius of the local neighbourhood equalled 3 cm. The eigenvalues of the covariance matrix equalled the variances of the projections on the principal components. Therefore,

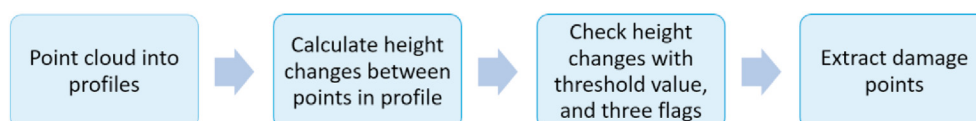


Fig. 6. Workflow of baseline method.

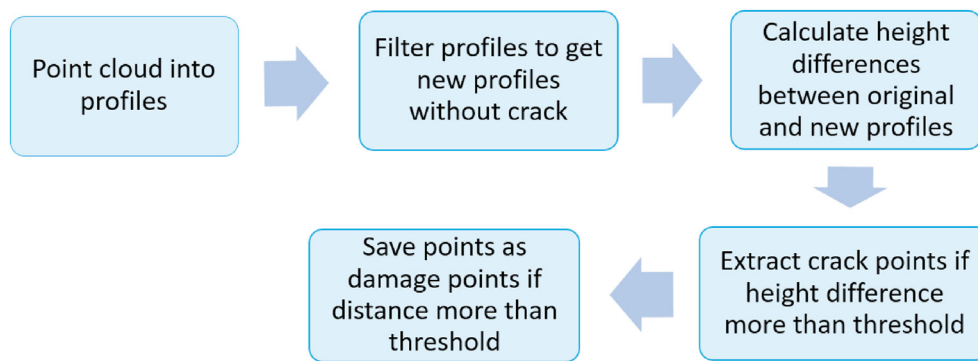


Fig. 8. Workflow of profile-based filtering algorithm.

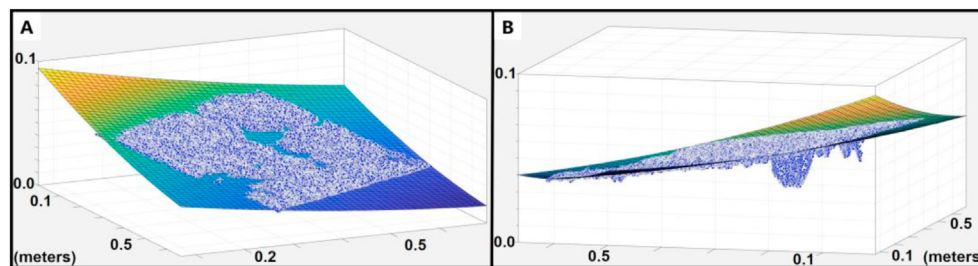


Fig. 9. Surface fitted to a 50 cm by 50 cm point cloud of a road surface.

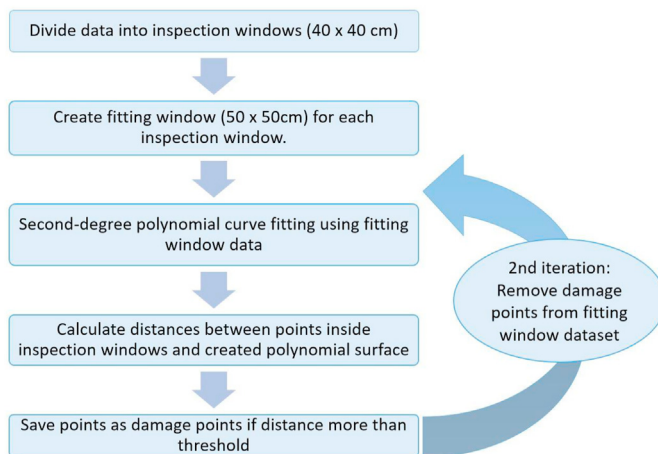


Fig. 10. Workflow of surface fitting algorithm.

the square root of the smallest eigenvalue was the standard deviation in the surface normal direction; hence it measured the goodness of the local planar fit, and was used as a surface roughness measure to detect cracks. If the roughness measure exceeded a value of 1 mm, the corresponding point was classified as ‘rough’.

Simple thresholding overestimated crack widths, because the roughness measure exceeded the threshold if a crack existed anywhere inside the local neighbourhood. Furthermore, large cracks sometimes contained smooth surfaces at the central ‘valley bottom’, which were not classified as rough, resulting in holes inside the cracks. These problems were solved using morphological operations. The point cloud was first projected onto a horizontal plane and transformed into a binary image, using a pixel size of 5 mm. Pixels containing at least one rough point were labelled as foreground, and the remaining pixels as background. Second, the holes were filled, using a flood-fill operation. Third, the image was eroded to decrease crack widths, using a disk-shaped structuring element, whose radius equalled 3 pixels. However, erosion sometimes

eradicating narrow cracks. To retain these, the skeleton of the image – retrieved using the medial axis transform (Sonka et al., 2008) – was added to the eroded image. Before the medial axis transform, the image was first dilated, using the same structuring element as in the erosion, to improve the skeleton’s connectedness. In addition, the skeleton was dilated, using a disk-shaped structuring element, whose radius equalled 1 pixel, to increase line widths. Pixels not belonging to the original foreground were removed from the dilated skeleton. The points inside the foreground pixels of the union of the eroded image and dilated skeleton were classified as cracks. Fig. 11 shows the workflow of the algorithm.

### 3.5. Random forest-based algorithm

Instead of using parametric surfaces, profile filtering, or defining rules and thresholds using expert knowledge, adaptive methods such as random forests (Breiman, 1999) can learn the classification rules from the data.

For each point, two geometric features were extracted from the point’s local neighbourhood (0.02 m radius); namely, the verticality and surface variation of the area. The geometric features were similar to those in Thomas et al. (2018) and obtained using CloudCompare 2.10.2. The surface variation and verticality stem from the eigenvalues of the covariance matrix of the point neighbourhood. The surface variation is calculated as the third eigenvalue divided by the sum of eigenvalues, and verticality as the difference of 1 and the third eigenvector of the covariance matrix (CloudCompare, 2019; Hackel et al., 2016). The surface variation was sensitive to areas deviating from the roughly planar road surface, and verticality noted areas with clear drops in surface level. Features such as height below a surface and roughness were considered, but they were sensitive to uneven parts of the road surface, where no cracks were present. Additionally, the intensity of the returned beam is expected to be lower in crack areas, which did visibly occur in our data, but using the intensity as input did not improve classification accuracy.

There were significantly fewer crack points than road points in the point clouds. Random oversampling of the crack class was therefore necessary, as using the original imbalanced classes resulted in the random forest classifier classifying all points for the road class. Adequate



Fig. 11. Workflow of local surface roughness algorithm.

results were obtained by oversampling the minority crack class to contain an equal number of points as the majority road class.

Fig. 12a) shows that the results contain salt-and-pepper noise. To remove this noise and to improve the distinction between the classes, crack points that contained fewer than five other crack points among their 15 nearest neighbours were ignored. Fig. 12b) shows that the cracks are more clearly defined after the process, and the noise is significantly reduced.

The small neighbourhood size retains small and narrow cracks even in the presence of noise. However, false negatives are found in the results, particularly in larger potholes, as the surface variation is low in flat areas, regardless of whether they are located within a crack, and the verticality seeks out the edges of the cracks. Smaller discontinuity areas are also present, with correctly classified crack areas containing small areas classified as roads. As these take the form of holes with respect to the crack area, a morphological hole filling operation – similar to that applied in the surface roughness method – was noted as a viable method for improving the results. The results show visible improvement from the noise-reduced crack classification (Fig. 12c): large holes have been filled, and most of the smaller discontinuity areas have been reclassified. Fig. 13 shows the workflow of the algorithm.

## 4. Results and discussion

### 4.1. Crack detection

The crack points of ten plots were detected by the five algorithms. With the manually detected road cracks as reference, the completeness, correctness, and  $F_1$  score of the algorithms in each plot were calculated, using pixel-wise analysis (see Section 2.4). Finally, the three overall

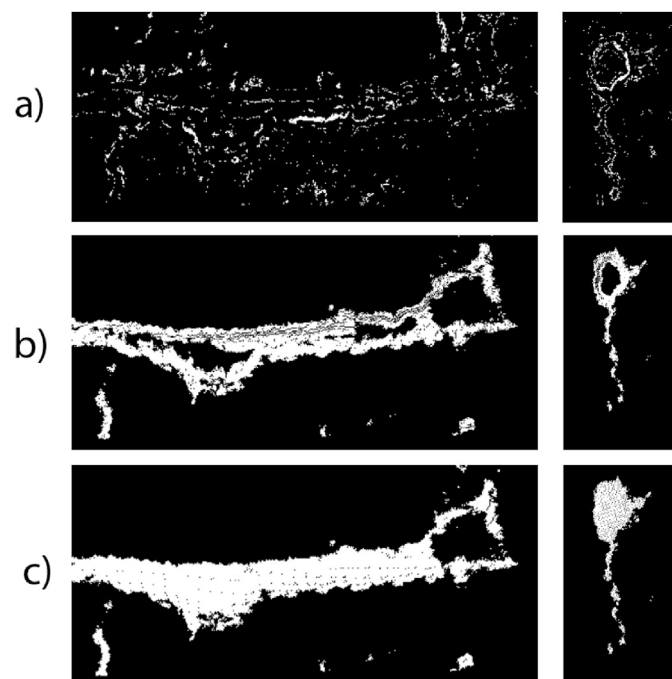


Fig. 12. Random forest classification of two crack areas: a) raw classification results; b) after the noise reduction process; c) after application of flood-fill operation.

evaluation indicators of each algorithm were calculated based on all the crack points of ten plots. The results are presented in Figs. 14–16.

Through the  $F_1$  scores in the ten plots (Fig. 16), the highest  $F_1$  score was obtained by the surface fitting algorithm in plot 3 as 0.849, and the corresponding completeness and correctness were 0.844 and 0.853 respectively. Apart from the baseline method, which offered an  $F_1$  score of 0.517, the other three algorithms achieved  $F_1$  scores greater than 0.600 in the same plot. However, the surface fitting algorithm had the lowest  $F_1$  score of 0.424 in plot 7, and the other four algorithms also acquired low  $F_1$  scores in this plot.

For the overall algorithm evaluation based on all the plots, the baseline method had the worst performance of all three indicators, and the overall  $F_1$  score was 0.409, while all the other algorithms had merits in distinct aspects. More specifically, based on Fig. 14, for the crack detection completeness, the profile-based filter method and the local surface roughness method had similar best performances of 0.642 and 0.644 respectively. However, Fig. 15 shows that the surface fitting method had the best detection correctness, which was 0.794, while the profile-based filter method and random forest-based method had slightly poorer results, which were 0.735 and 0.743. Regarding the overall performance of the algorithms, as shown in Fig. 16, the  $F_1$  score of the profile-based filter method was 0.685 as the highest one, the surface fitting method obtained a slightly lower  $F_1$  score of 0.672, and the local surface roughness method ranked third with an  $F_1$  score of 0.645.

Through the previous results, the algorithms had different performances in different plots, which were not always consistent with the overall performance. This was because the plots contained distinct crack types, which led to various levels of complexity in crack detection. Considering both the real plot condition and manual crack detection complexity, the plots were divided into three groups: easy (plots 3, 6, 10); medium (plots 1, 4, 8, 9); and difficult (plots 2, 5, 7). The crack detection completeness, correctness and  $F_1$  score of the five algorithms were calculated in each group, and the results are presented in Fig. 17.

Looking through the Fig. 17, as the detection complexity increased from the easy group to the difficult group, the crack detection completeness of the five algorithms gradually decreased respectively, proving that our plots' classification was reasonable.

For the crack detection correctness, the surface fitting and the local surface roughness methods had high correctness in both easy and difficult groups, and the surface fitting method reached the highest correctness of 0.883 in the easy group. The random forest-based method provided stable performances of correctness, which were 0.751, 0.714, and 0.779 in the easy, medium, and difficult groups respectively.

The results show that the surface fitting method had the best performance in the easy group, and its  $F_1$  score reached 0.841, while the  $F_1$  scores of the other four algorithms also increased in different levels, among them, the profile-based filter method had the second best  $F_1$  score of 0.808. In the medium group, the profile-based filter method had the best  $F_1$  score of 0.707, and the surface fitting method had a slightly lower  $F_1$  score of 0.702. In the difficult group, that of the local surface roughness was 0.592 as the best.

### 4.2. Crack area estimation

Through the amount of crack pixels, the crack area in each plot measured from the five algorithms was calculated. As the crack areas of the ten plots varied widely, the absolute crack area error percentage, which is the ratio of estimated absolute crack area error to the reference area, was adopted to present the capabilities of the five algorithms. The

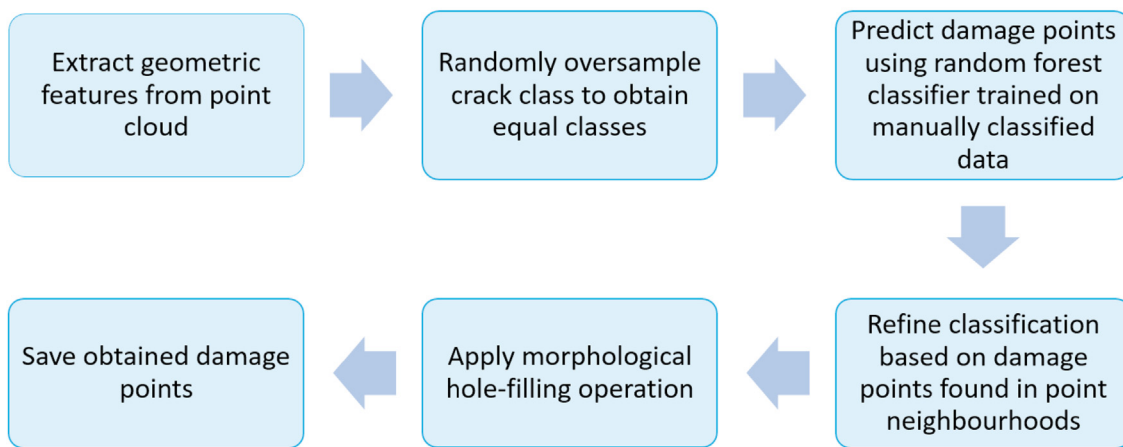


Fig. 13. Workflow of the random forest-based classification.

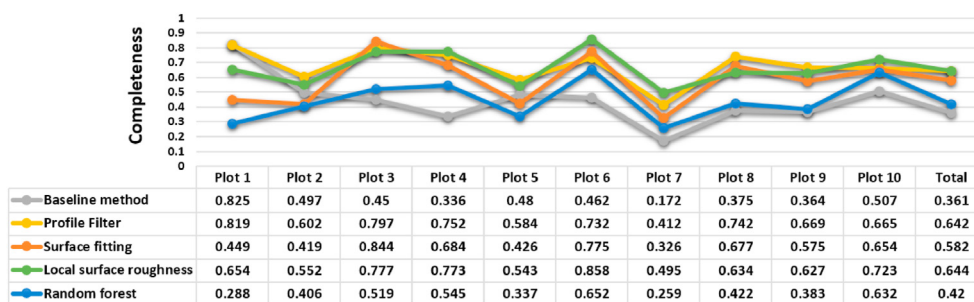


Fig. 14. Completeness for the five algorithms on the ten plots and the overall completeness.

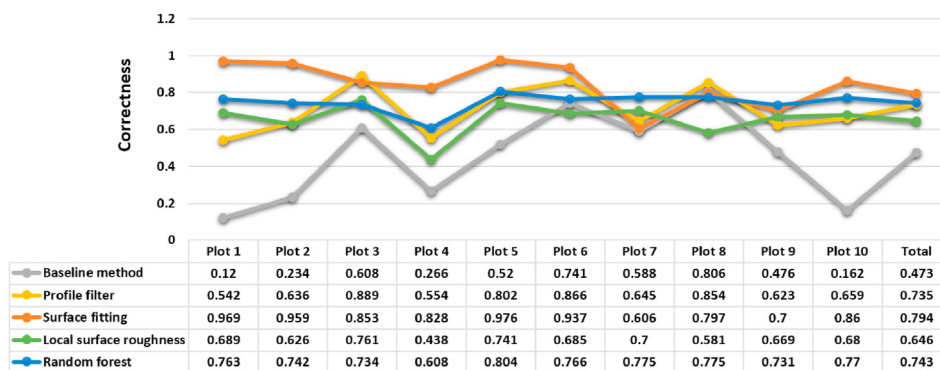


Fig. 15. Correctness for the five algorithms on the ten plots and the overall correctness.

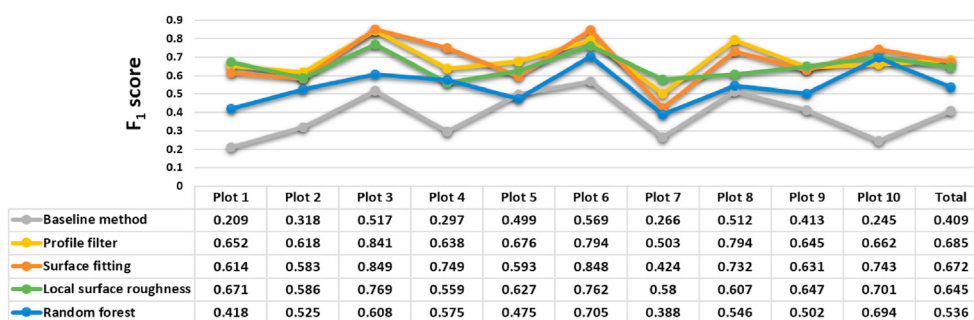


Fig. 16. F<sub>1</sub> score for the five algorithms on the ten plots and the overall F<sub>1</sub> score.

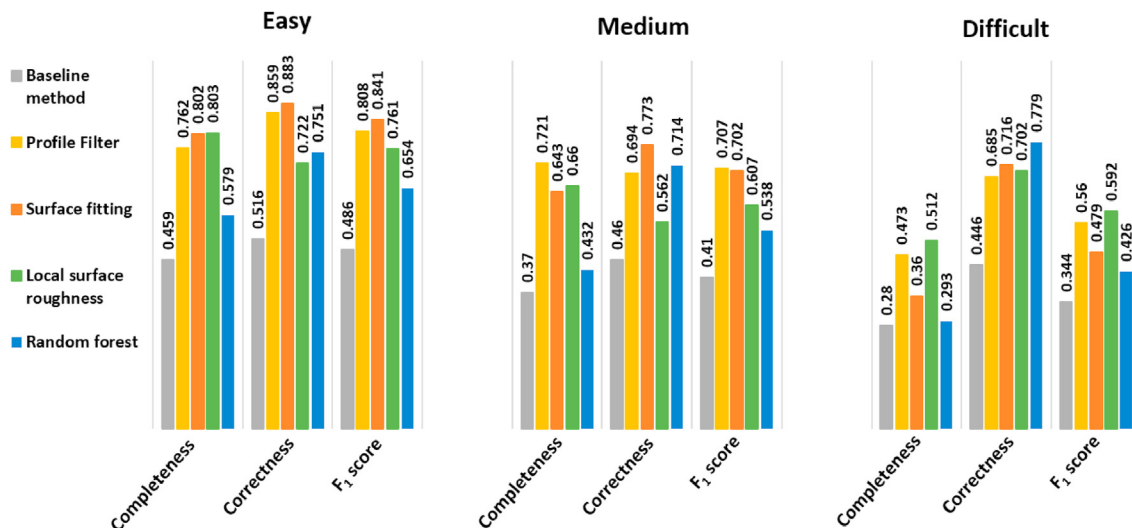


Fig. 17. The crack detection completeness, correctness, and F<sub>1</sub> score of all the algorithms for different plot categories, ranging from easy to difficult.

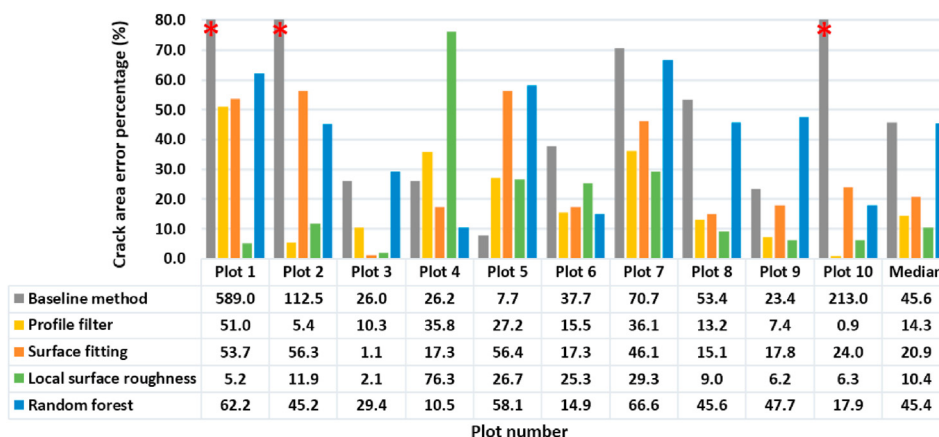


Fig. 18. The absolute crack area error percentages with five algorithms of ten plots (\*: the bar value exceeds the max limit of the y-axis and is presented in the table below).

absolute crack area error percentages in each plot are presented in Fig. 18.

Obviously, the crack area percentages from the baseline algorithm were much larger than those from other algorithms in plots 1, 2, and 10, affecting the readability of the graphic. Therefore, the y-axis value in Fig. 18 was limited to present most of the percentages, and the specific values can be checked in the table.

Considering the large performance fluctuations across the plots, the MAPEs of the five algorithms were calculated based both on the all plots and the three plot groups proposed in the previous section (Fig. 19).

Fig. 19 shows that the surface roughness algorithm achieved the lowest MAPE of 19.8% over the 10 plots, while the profile filter algorithm produced a similar MAPE of 20.3%. In the easy, medium, and difficult plot groups, the best MAPEs were 8.9% from the profile filter algorithm, and 24.2% and 22.6% from the surface roughness algorithm respectively. The baseline method performed very badly in all groups, while the surface fitting and random forest algorithms had their best performance in the easy plot group, and the performance deteriorated as the crack detection difficulty increased.

To further study the algorithm performance, the crack area estimation RMSEs of the five algorithms were calculated (Fig. 20). Fig. 20 shows that the RMSEs' distribution was quite similar to that of MAPE, while the performance of the surface fitting algorithm became a little different. Over all ten plots, the best RMSE still came from the surface roughness

algorithm of 997.3 cm<sup>2</sup>, and the profile filter algorithm achieved a similar RMSE of 1,000.8 cm<sup>2</sup>. However, the surface fitting algorithm produced the best RMSEs in the easy and medium plot groups, with 333.0 cm<sup>2</sup> and 589.6 cm<sup>2</sup>, and in the difficult plot group, it was 1,364.9 cm<sup>2</sup> from the surface roughness algorithm. The profile filter algorithm had close performances to the surface fitting algorithm in the easy and medium plots with RMSEs of 391.4 cm<sup>2</sup> and 590.3 cm<sup>2</sup>.

The coefficient of determination (R<sup>2</sup>) between the estimated crack areas and reference data was calculated for each algorithm. The profile-based filter method gave a highest R<sup>2</sup> of 0.86, while the surface fitting and local surface roughness methods gave slightly lower values of 0.82 and 0.79 respectively. The random forest-based machine-learning method attained an R<sup>2</sup> of 0.66, and the baseline method resulted in a bad R<sup>2</sup> of 0.17.

### 4.3. Crack volume estimation

#### 4.3.1. Volume calculation tool evaluation

This section presents the accuracy evaluation of the automated volume calculation tool (Sec. 2.3.). The tool was used to calculate the crack volumes for each algorithm and reference, and here we verify the tool's validity. Each distress in the reference was manually segmented, and then the total volume of the plot distresses was calculated by fitting a plane to each manually selected distress type separately. This volume was

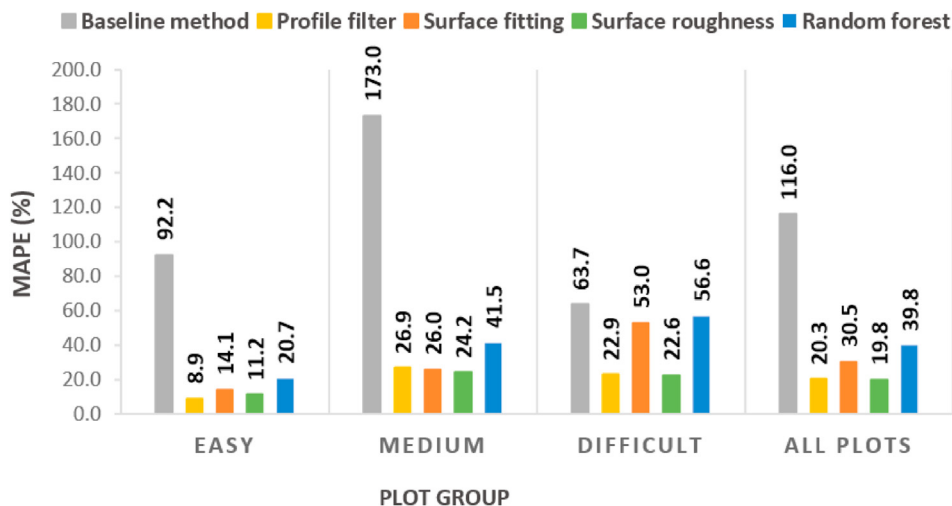


Fig. 19. Crack area MAPEs (%) of the five algorithms in different plot groups.

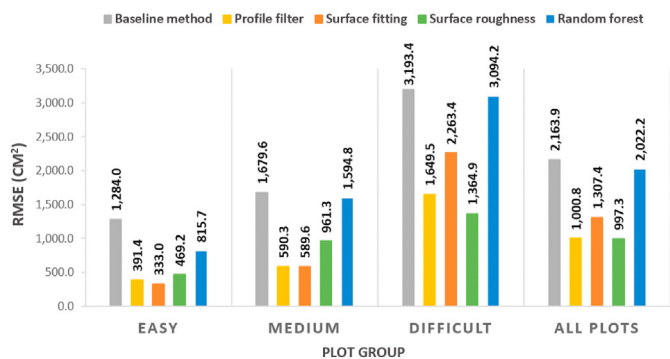


Fig. 20. Crack area RMSEs (cm<sup>2</sup>) of the five algorithms in different plot groups.

Table 2

Method evaluation for crack volume calculation tool. The tool is applied on ground truth data. The ratio in this case is the ratio of volume calculated by the tool to the manually calculated volume. The error percentage is the difference between volumes calculated automatically and manually, divided by the manually calculated value and multiplied by 100.

Evaluation for volume calculation tool				
	Manual segmentation (dm <sup>3</sup> )	Automatic segmentation (dm <sup>3</sup> )	Ratio	Error %
Plot 1	0.035	0.035	0.98	2.2%
Plot 2	0.296	0.288	0.97	2.8%
Plot 3	4.829	4.828	1.00	0.0%
Plot 4	1.276	1.185	0.93	7.2%
Plot 5	1.098	1.106	1.01	0.8%
Plot 6	3.793	3.815	1.01	0.6%
Plot 7	3.302	2.955	0.89	10.5%
Plot 8	9.522	9.172	0.96	3.7%
Plot 9	2.357	2.181	0.93	7.5%
Plot 10	0.829	0.856	1.03	3.3%
TOTAL	27.338	26.422	0.97	3.4%

used as a ground truth to verify the accuracy of the automated volume calculation tool. Table 2 shows the results for each plot. For all plots, the volume calculation tool managed to calculate 97% of the total volume obtained by manually segmented distress type measurement. The error percentage for all plots was 3.4%.

4.3.2. Plot-level volume accuracy evaluation

Crack volumes were calculated both for manually classified distresses

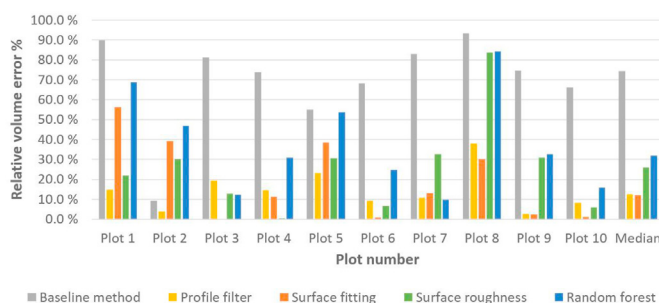


Fig. 21. The absolute volume error percentages of the algorithms.

and distresses classified by the five algorithms, using the tool presented in Section 2.3. Fig. 21 shows the absolute volume error percentage between algorithm and reference volumes for each plot, in addition to the median. The median error percentages equalled 69.5%, 12.6%, 12.1%, 26.0%, and 31.8% for the baseline method, profile-based filtering, surface fitting, local surface roughness, and random forest respectively.

Fig. 22 shows the MAPE of the crack volume estimation in relation to the plot difficulty (Sec. 4.1). The surface fitting algorithm achieved the lowest MAPE of 0.7% for the easy plots. A small correlation was observed between the error of the volume estimates and plot difficulty in the surface fitting algorithm: the more difficult the test area, the worse the MAPE. Profile-based filtering worked equally well at each difficulty level (MAPE varied between 12.3% and 17.5% across the plot difficulty). Local surface roughness also performed well in the easy plots (an 8.5% MAPE), but in the medium and difficult plots, performance was worse (MAPE between 31.1% and 34.2%). The baseline method and random forest had poorer performance than the other methods: their MAPEs were between 38.0% and 69.5% when all the plots were included in the analysis.

The RMSEs of the volume estimates were similar to the MAPEs (Fig. 23). The surface fitting algorithm performed excellently in easy plots (RMSE = 0.02 dm<sup>3</sup>), but was only slightly better than profile-based filtering when all plots were taken into account. The local surface roughness algorithm worked well in the easy and difficult plots (RMSEs of 0.39 dm<sup>3</sup> and 0.60 dm<sup>3</sup>), but in the medium plots, the RMSE was as high as 3.86 dm<sup>3</sup>, which significantly affected the total RMSE value of 2.47 dm<sup>3</sup>. The random forest algorithm performed better in the RMSE comparison than in the MAPE comparison, as its RMSE values were very consistent with the local surface roughness algorithm.

Coefficients of determination (R<sup>2</sup>) of the volume estimates of the algorithms were calculated for all plots. As a result, surface fitting and profile-based fitting algorithms gained the highest R<sup>2</sup> values among the

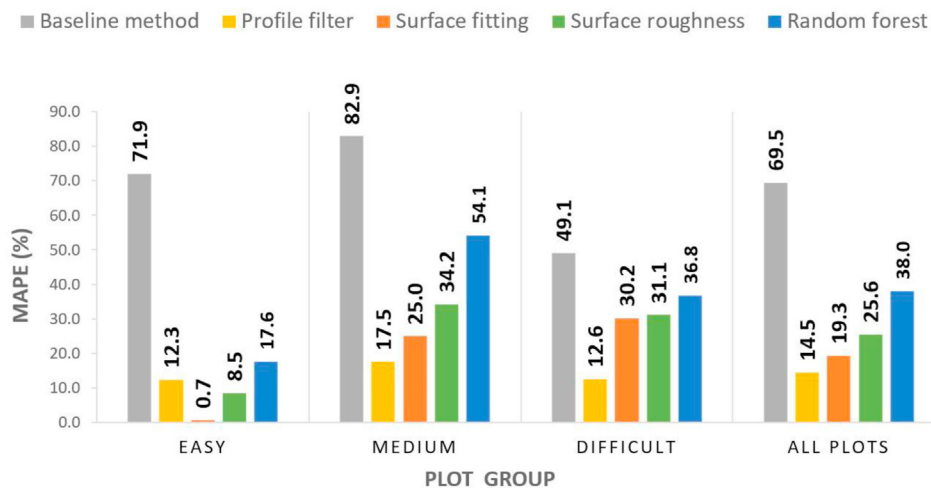


Fig. 22. Crack volume MAPE for each algorithm.

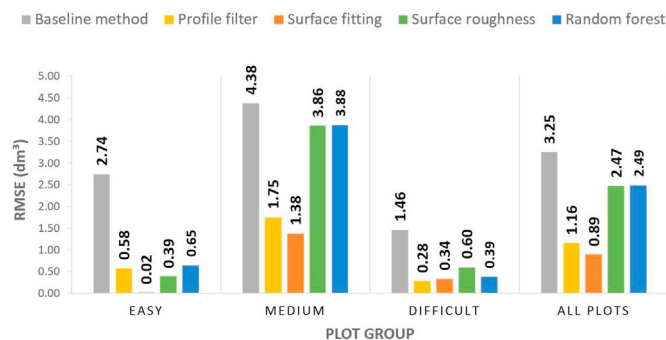


Fig. 23. Crack volume RMSE for each algorithm (dm³).

algorithms (0.97 and 0.96 respectively). Local surface roughness, random forest, and the baseline method performed equally, resulting in  $R^2$  values of 0.51, 0.50, and 0.58 respectively.

4.3.3. Object level volume accuracy evaluation

In the object level analysis, each distress was studied separately. Our ten plots contained 75 pavement distresses (hereafter referred to as objects). For each object, a true-positive volume ratio was defined as the true-positive volume of distress divided by the reference volume. Here, we evaluate the object detection rate, that is, how many distresses each algorithm found. The best-performing algorithm (the profile-based filter method) detected all 75 objects (detection rate 100%). The detection rate alone does not provide information on how comprehensively the distress was found, because some objects were only partly found. Therefore, the detection rate of the algorithms was also examined by considering only those objects found by the algorithm whose true positive volume exceeded half of the reference volume (true positive volume ratio over 50%). In this case, the best-performing algorithm detected 84% of all distresses. Fig. 24 shows the performance of each algorithm in more detail.

Object-level analysis was also used to evaluate the ability of algorithms to find different types of distress. Only the best performing algorithm, the profile-based filter, was evaluated. Objects were sorted according to maximum depths, and correlations between crack depths and different distress types were examined (Fig. 25). A trend line was fitted for each type of distress, as a result of which it was observed that certain distresses correlated better with the maximum depth of the distress than others. The results show that the algorithm found potholes well, because its true positive volume ratio was always higher than 70%. For other distresses, longitudinal and transverse cracks correlated well

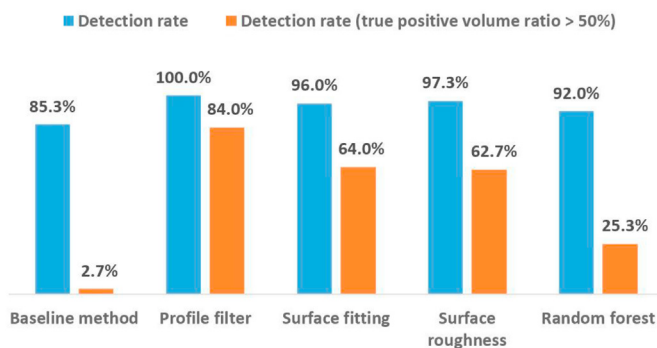


Fig. 24. Pavement distress object detection rate of each algorithm.

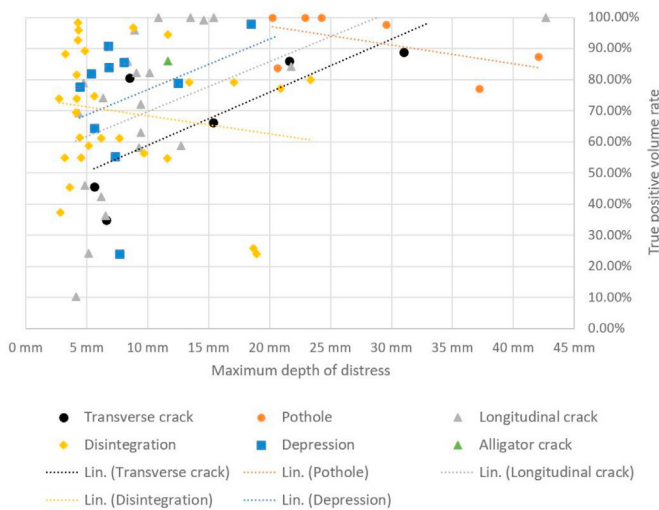


Fig. 25. Relation of maximum distress depth, true positive volume rate, and different distress types for the profile-based filter method. For each distress type, a linear trend line (Lin.) is drawn. The  $R^2$  values for transverse crack, pothole, longitudinal crack, disintegration, and depression are 0.58, 0.30, 0.26, 0.03 and 0.10 respectively.

with distress depth. The deeper the damage, the better the algorithm would find it. Depression and disintegration gave the worst  $R^2$  values. In particular, observations of disintegration were most scattered, regardless of the depth of damage. No conclusions can be drawn from the alligator

cracks, because there were too few observations.

#### 4.4. Discussion

The crack detection capabilities of the five algorithms were evaluated by analysing the accuracies of the crack point classification, crack area estimation, and crack volume estimation, in addition to an object-level analysis. The results suggest that the baseline method had the worst performance in all the aspects, which proves that only using height difference between neighbouring points (over single profile) alone does not seem to result in reliable crack detection.

The profile-based filter method and surface fitting method have similar performance in crack point detection and classification ( $F_1$  scores of 0.685 and 0.672, respectively), while the profile-based method has better performance in crack area and volume estimation. The local surface roughness method performed somewhat worse over all the plots ( $F_1$  score of 0.645), but it is better than the above two methods in complex plots ( $F_1$  score of 0.592, other two 0.56 and 0.492). Therefore, it is as important as the profile-based filter and surface fitting methods, and merits further study when developing a reliable crack detection solution for roads containing complex cracks.

The random forest-based machine-learning method has worse performance than the traditional methods except the baseline method, but it is a good start to develop effective and accurate crack detection solutions using machine-learning technology. The crack detection accuracy of the random forest method remained stable in the easy, medium, and difficult groups, which demonstrates that machine learning is not sensitive to the complexity of the road damage. It is expected that as the amount of training data increases in future, the performance of the random forest method may have the potential for significant improvement. In a future study, we will aim to develop better machine learning methods with different algorithms and larger crack datasets.

In this study, the ten 1-m-by-3.5-m plots were selected for crack study from 3 km continuous real road surface data, with totally 75 pavement distresses inside. This is comparable to other 3D point cloud based pavement distress detection studies, considering the distress amount, types or the total studied road length. The ten plots enable us to evaluate the algorithms over different kinds of cracks and various levels of crack detection complexities, which helps offer comprehensive and reliable algorithm evaluation and comparison.

As all the algorithms studied in this paper are sensitive to the road surface elevation change, the edges of road surface markings and patches may be erroneously extracted as cracks. Further studies are therefore needed to tackle these issues. One solution is to classify markings (e.g. Chen et al., 2021) and patches prior to distress detection, using reflectance of the surface or the intensity of the returning laser pulses.

As mentioned above, the algorithms are sensitive to road surface height change, therefore the performances of methods were affected by the characteristics (surface roughness, crack types, width, and depth) of the pavement, rather than the road surface material. Thus, if the point cloud geometry remains the same, there should be little difference in the distress detection between different road types (e.g., pothole in highway versus pothole in urban road). We expect the parameters we applied and the comparison results in this study are also applicable to other roads. However, this needs to be checked in a different road in our future work. It is possible that in some plots the evaluated best algorithms have poor performance if there are certain kinds of distress, or the studied plots have very complex distresses. One possibility to improve the algorithms performance is to adjust the parameters based on one sample plot in advance whenever applying the algorithms in a new road surface. Even though some additional work is needed, the whole process is still automated once the parameters have been tuned.

## 5. Conclusions

In this paper, five road distress detection algorithms developed based

on point cloud processing expertise, and using TLS point clouds were introduced and evaluated in crack detection, crack area, and crack volume estimation on ten test plots selected from a 3-km-long test road in Finland.

Compared with manual reference data, the along- and cross-track profile-based filter method performed best in crack detection. The surface fitting and surface roughness methods produced similar  $F_1$  scores. The performance of the baseline method was unsatisfactory compared with the other methods.

In the crack area estimation, the MAPEs of the profile filter, surface fitting, and surface roughness methods over the ten plots were 20.3%, 30.5%, and 19.8% respectively. The corresponding MAPEs in volume estimation were 14.5%, 19.3%, and 25.6%. In general, the deeper the damage was, the better the volume estimation accuracy. The random forest method obtained MAPEs of 39.8% and 38.0% for crack area and volume estimation respectively.

In the near future, we will combine the three best methods to improve the overall performance, and further develop machine-learning based methods. Due to the irregularities of pavement surface and the mixture of distresses on the pavement surface, there has been some limited success in accurately automating crack detection and classification (Oliveira and Correia, 2012, 2014), and there are continuing needs of improvement.

We expect the reported methods to be directly applicable to process data also from MLS, and high-end pavement measurement systems in an attempt to increase the automation level of road distress inventories.

## Author contribution

Z.F. was responsible for the profile-based filtering method and area-based statistical evaluation, and she contributed to writing and reviewing. A.E.I. was responsible for the surface fitting algorithm, volume calculation tool, data pre-processing, ground truth segmentation, and volume based statistical evaluation, and he contributed to the field test preparation, TLS data collection, writing, and reviewing. M.L. was responsible for the local surface roughness algorithm, and he assisted with statistical evaluation, writing, and reviewing. M.I. was responsible for the random forest-based algorithm, and he contributed to writing and reviewing. H.K. and A.K. were responsible for the study design, and they contributed to TLS data collection, writing, and reviewing. J.S. was responsible for the baseline algorithm, and he assisted with reviewing. J.H. was the PI of the project, and he assisted with data analysis, writing, and reviewing. H.H. was the co-PI of the project at Aalto University, and he acted as an advisor.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

The Academy of Finland supported this study through the 'Road Distress Mapping Combining Expertise of Sensors and Point Cloud Processing, Surveying and Road Engineering' project (323750 and 323752) and the Strategic Research Council at the Academy of Finland 'Competence-Based Growth Through Integrated Disruptive Technologies of 3D Digitalization, Robotics, Geospatial Information and Image Processing/Computing - Point Cloud Ecosystem' project (314312). Additionally, the study was supported by academy projects (337656 and 319011).

## References

- Arlinghaus, S., 1994. Practical Handbook of Curve Fitting. CRC press.
- Barbarella, M., D'Amico, F., De Blasiis, M.R., Di Benedetto, A., Fiani, M., 2018. Use of terrestrial laser scanner for rigid airport pavement management. *Sensors* 18 (1), 44.

- Breiman, L., 1999. Random Forests—Random Features. Technical Report 567. Statistics Department, University of California, Berkeley.
- Chen, S., Zhang, Z., Zhong, R., Zhang, L., Ma, H., Liu, L., Jan. 2021. A dense feature pyramid network-based deep learning model for road marking instance segmentation using MLS point clouds. *IEEE Trans. Geosci. Rem. Sens.* 59 (1), 784–800.
- Choi, J., Zhu, L., Kurosu, H., 2016. Detection of cracks in paved road surface using laser scan image data. *Int. Arch. Photogram. Rem. Sens. Spatial Inf. Sci.* 41.
- CloudCompare, 2019. CloudCompare release notes. Available: <https://www.cloudcompare.org/release/notes/20190106/>.
- De Blasiis, M.R., Di Benedetto, A., Fiani, M., 2020. Mobile laser scanning data for the evaluation of pavement surface distress. *Rem. Sens.* 12 (6), 942.
- El Issaoui, A., Feng, Z., Lehtomäki, M., Hyypä, E., Hyypä, H., Kaartinen, H., Kukko, A., Hyypä, J., 2021. Feasibility of mobile laser scanning towards operational accurate road rut depth measurements. *Sensors* 21 (4), 1180. <https://doi.org/10.3390/s21041180>.
- Fan, R., Bocus, M.J., Zhu, Y., Jiao, J., Wang, L., Ma, F., Cheng, S., Liu, M., 2019. Road crack detection using deep convolutional neural network and adaptive thresholding. In: 2019 IEEE Intelligent Vehicles Symposium (IV), pp. 474–479.
- Finnish Transport Infrastructure Agency, 2013. *Päällystevauriokartoituksen (PVK) Inventointiohje*. Liikennevirasto. Finland, Helsinki.
- Guan, H., Li, J., Yu, Y., Chapman, M., Wang, C., 2015. Automated road information extraction from mobile laser scanning data. *IEEE Trans. Intell. Transport. Syst.* 16 (1), 194–205.
- Guan, H., Li, J., Yu, Y., Chapman, M., Wang, H., Wang, C., Zhai, R., 2014. Iterative tensor voting for pavement crack extraction using mobile laser scanning data. *IEEE Trans. Geosci. Rem. Sens.* 53 (3), 1527–1537.
- Hackel, T., Wegner, J.D., Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Ann. Photogram. Rem. Sens. Spatial Inf. Sci.* 3, 177–184.
- Hu, Y., Zhao, C.X., 2010. A novel LBP based methods for pavement crack detection. *J. Pattern Recogn. Res.* 5 (1), 140–147.
- Johnson, D., Williams, R.P.D., 1976. *Methods of Experimental Physics: Spectroscopy*. Academic Press, New York.
- Kunyuuan, C., Ming, C., Menglan, Z., Xinqu, C., Yifei, C., Jonathan, L., Hongshan, N., 2015. Automated Object Extraction from MLS Data: A Survey. 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), pp. 331–334.
- Laurent, J., Lefebvre, D., Samson, E., 2008. Development of a new 3D transverse laser profiling system for the automatic measurement of road cracks. In: Symposium on Pavement Surface Characteristics, 6th, 2008. Portoroz, Slovenia.
- Lehtomäki, M., Jaakkola, A., Hyypä, J., Lampinen, J., Kaartinen, H., Kukko, A., Puttonen, E., Hyypä, H., 2016. Object classification and recognition from mobile laser scanning point clouds in a road environment. *IEEE Trans. Geosci. Rem. Sens.* 54 (2), 1226–1239.
- Li, Q., Yao, M., Yao, X., Xu, B., 2009. A real-time 3D scanning system for pavement distortion inspection. *Meas. Sci. Technol.* 21 (1), 15702.
- Mathavan, S., Kamal, K., Rahman, M., 2015. A review of three-dimensional imaging technologies for pavement distress detection and measurements. *IEEE Trans. Intell. Transport. Syst.* 16 (5), 2353–2362.
- Medina, R., Llamas, J., Zalama, E., Gómez-García-Bermejo, J., 2014. Enhanced automatic detection of road surface cracks by combining 2D/3D image processing techniques. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 778–782.
- Oliveira, H., Correia, P.L., 2008. Supervised strategies for cracks detection in images of road pavement flexible surfaces. In: 2008 16th European Signal Processing Conference, pp. 1–5.
- Oliveira, H., Correia, P.L., 2012. Automatic road crack detection and characterization. *IEEE Trans. Intell. Transport. Syst.* 14 (1), 155–168.
- Oliveira, H., Correia, P.L., 2014. CrackIT – an image processing toolbox for crack detection and characterization. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 798–802.
- Ragnoli, A., De Blasiis, M.R., Di Benedetto, A., 2018. Pavement distress detection methods: a review. *Infrastructure* 3 (4), 58.
- Ravi, R., Habib, A., Bullock, D., 2020. Pothole mapping and patching quantity estimates using LiDAR-based mobile mapping systems. *Transport. Res. Rec.* 2674 (9), 124–134.
- Salman, M., Mathavan, S., Kamal, K., Rahman, M., 2013. Pavement crack detection using the Gabor filter. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), pp. 2039–2044.
- Shi, Y., Cui, L., Qi, Z., Meng, F., Chen, Z., 2016. Automatic road crack detection using random structured forests. *IEEE Trans. Intell. Transport. Syst.* 17 (12), 3434–3445.
- Sonka, M., Hlavac, V., Boyle, R., 2008. *Image Processing, Analysis, and Machine Vision*, third ed. Thomson Learning.
- Su, W., Sun, Z., Zhong, R., Huang, J., Li, M., Zhu, J., Zhu, D., 2015. A new hierarchical moving curve-fitting algorithm for filtering lidar data for automatic DTM generation. *Int. J. Rem. Sens.* 36 (14), 3616–3635.
- Thomas, H., Goulette, F., Deschaud, J.E., Marcotegui, B., LeGall, Y., 2018. September. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In: 2018 International Conference on 3D Vision (3DV). IEEE, pp. 390–398.
- Tsai, Y.C.J., Li, F., 2012. Critical assessment of detecting asphalt pavement cracks under different lighting and low intensity contrast conditions using emerging 3D laser technology. *J. Transport. Eng.* 138 (5), 649–656.
- Tsai, Y.C., Chatterjee, A., 2018. Pothole detection and classification using 3D technology and watershed method. *J. Comput. Civ. Eng.* 32 (2), 04017078.
- Tsai, Y., Yang, Z., 2020. New pavement performance indicators using crack fundamental elements and 3D pavement surface data with multiple-timestamp registration for crack deterioration analysis and optimal treatment determination. *Transport. Res. Rec.* 2674 (7), 115–126.
- Varadharajan, S., Jose, S., Sharma, K., Wander, L., Mertz, C., 2014. Vision for road inspection. In: IEEE Winter Conference on Applications of Computer Vision, pp. 115–122.
- Yang, Z., Zhang, X., Tsai, Y., Wang, Z., 2021. Quantitative Assessments of Crack Sealing Benefits by 3D Laser Technology. *Transportation Research Record*, 03611981211027897.
- Yu, Y., Li, J., Guan, H., Wang, C., 2014. 3D crack skeleton extraction from mobile LiDAR point clouds. In: 2014 IEEE Geoscience and Remote Sensing Symposium, pp. 914–917.
- Zhang, L., Yang, F., Zhang, Y.D., Zhu, Y.J., 2016. Road crack detection using deep convolutional neural network. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 3708–3712.
- Zhang, Z., Cheng, M., Chen, X., Zhou, M., Chen, Y., Li, J., Nie, H., 2015. Turning mobile laser scanning points into 2D/3D on-road object models: current status. In: 2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), pp. 3524–3527.
- Zhong, M., Sui, L., Wang, Z., Hu, D., 2020. Pavement crack detection from mobile laser scanning point clouds using a time grid. *Sensors* 20 (15), 4198.
- Zou, Q., Cao, Y., Li, Q., Mao, Q., Wang, S., 2012. CrackTree: automatic crack detection from pavement images. *Pattern Recogn. Lett.* 33 (3), 227–238.