Iterative Tensor Voting for Pavement Crack Extraction Using Mobile Laser Scanning Data

Article *in* IEEE Transactions on Geoscience and Remote Sensing · March 2015

DOI: 10.1109/TGRS.2014.2344714

7 authors, including:

Haiyan Guan
Nanjing University of Information Science & …
52 PUBLICATIONS **428** CITATIONS
SEE PROFILE

Yongtao Yu
Huaiyin Institute of Technology
41 PUBLICATIONS **370** CITATIONS
SEE PROFILE

Michael Chapman
Ryerson University
21 PUBLICATIONS **250** CITATIONS
SEE PROFILE

Cheng Wang
Xiamen University
157 PUBLICATIONS **894** CITATIONS
SEE PROFILE

Iterative Tensor Voting for Pavement Crack Extraction Using Mobile Laser Scanning Data

Haiyan Guan, Jonathan Li, *Senior Member, IEEE*, Yongtao Yu, Michael Chapman, Hanyun Wang, *Member, IEEE*, Cheng Wang, *Member, IEEE*, and Ruifang Zhai

***Abstract*—The assessment of pavement cracks is one of the essential tasks for road maintenance. This paper presents a novel framework, called ITVCrack, for automated crack extraction based on iterative tensor voting (ITV), from high-density point clouds collected by a mobile laser scanning system. The proposed ITVCrack comprises the following: 1) the preprocessing involving the separation of road points from nonroad points using vehicle trajectory data; 2) the generation of the georeferenced feature (GRF) image from the road points; and 3) the ITV-based crack extraction from the noisy GRF image, followed by an accurate delineation of the curvilinear cracks. Qualitatively, the method is applicable for pavement cracks with low contrast, low signal-to-noise ratio, and bad continuity. Besides the application to GRF images, the proposed framework demonstrates much better crack extraction performance when quantitatively compared to existing methods on synthetic data and pavement images.**

***Index Terms*—Georeferenced, intensity, iterative tensor voting (ITV), ITVCrack, mobile laser scanning (MLS), pavement crack extraction.**

I. Introduction

PAVEMENT cracks, as the most common type of asphalt concrete-surfaced pavement distress, can be caused by fractures due to excessive loading, fatigue, thermal changes, moisture damage, slippage, or contraction. Usually, in regard to shape and position, cracking is grouped into one of the following types: fatigue, longitudinal, alligator, edge, reflection, block, and transverse [1], [2]. In the past, crack inspection and evaluation involved high degrees of subjectivity and hazardous exposure, as well as low production rates. Until now, visual inspection techniques have been explored for evaluating pavements. These techniques involved the capture of, mostly on video and cameras, images collected using specially equipped vehicles.

Dynamic-optimization-based methods effectively handle blurry and discontinuous pavement images [3]–[5]. However, most of them are computationally intensive. The effectiveness of the thresholding-based segmentation methods, based on either grayscale discontinuity or similarity [6]–[9], mostly depends on the pavement environment and material, leading to unreliable crack extraction results. Although wavelet-based transforms, such as beamlet, contourlet, and their variants, are another common type of technique for crack extraction [10], [11], due to the anisotropic properties of wavelets, they often fail to process cracks with high curvatures or poor continuity. Mathematical-morphology-based methods have been used to detect cracks in pavement images [12]. However, these algorithms are limited to three structural elements (i.e., disk, line, and square) and by the choice of parameters. A number of efforts on crack extraction have been made in the fields of artificial intelligence, data mining, machine learning, and neural networks [13]–[15]. However, the selection of parameters depends on crack variations and image quality. Additionally, image-/video-based crack extraction algorithms suffer from the influence of several environmental factors, such as the following: 1) shadows cast by trees and moving vehicles; 2) weather conditions; and 3) imaging time of day, which has the greatest impact on the visibility of road surfaces.

In recent years, mobile laser scanning (MLS) has become a rapidly developing technology, particularly for accurate corridor mapping (e.g., railroads, highways, and roads) because this technology enables the collection of millimeter-level survey-grade data in unprecedented detail at highway speeds and at less than traditional survey costs [16]. Along a corridor, MLS systems capture (and represent by 3-D point clouds) the following: visible trees, bridges, streetlight poles, buildings, power lines, road markings, cracks, etc. Therefore, data collected from a single mission can be used for multiple tasks without further field visits, thus increasing data usability and efficiency. More importantly, MLS systems enable zero-traffic-impact data acquisition about road corridors because less congestion occurs at night. Thus, in terms of abundant and detailed data, safety,

Manuscript received January 9, 2014; revised April 22, 2014; accepted July 21, 2014. This work was supported by the National Natural Science Foundation of China under Grant 41471379. *(Corresponding author: Jonathan Li.)*

H. Guan is with the Department of Computer Science, Xiamen University, Xiamen 361005, China, and also with the Department of Geography and Environmental Management, University of Waterloo, Waterloo, ON N2L 3G1, Canada.

J. Li is with the Department of Computer Science, Xiamen University, Xiamen 361005, China, and also with the Department of Geography and Environmental Management,University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: junli@xmu.edu.com.cn; junli@uwaterloo.ca).

Y. Yu and C. Wang are with the Department of Computer Science, Xiamen University, Xiamen 361005, China.

M. Chapman is with the Department of Civil Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada.

H. Wang is with the School of Electronic Science and Engineering, National University of Defense Technology, Changsha 410073, China.

R. Zhai is with the Department of Computer Science, School of Informatics, Huazhong Agricultural University, Wuhan 430070, China.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2014.2344714

and efficiency, MLS systems are recognized for their ability to enrich the available 3-D databases of geographical information systems for transportation-related applications [17].

However, the ability of MLS systems to capture highly dense point clouds presents a great challenge in postprocessing of a very large volume of MLS data in order to obtain readable and comprehensive information about cracks. For example, a Trimble MX-8 system that integrates two Riegl VQ-250 laser scanners can produce up to 35 GB of data in 20 min. In particular, road points account for a large proportion of the scanned data. Thus, an interpolation method that converts unorganized 3-D point clouds into a 2-D grayscale image is considered for feature extraction using established image processing algorithms.

MLS intensity data that physically reflect the power of the received echoes [18] have been widely used to automatically extract different types of street-scene objects [19], such as highly reflective road markings [20] and even illuminated structures (e.g., tunnels and culverts) [21]. Based on intensity, MLS data are interpolated into a georeferenced feature (GRF) image, in which a crack is typically represented as a curvilinear structure. However, compared to a real pavement image, a GRF image contains a large amount of noise. In addition, owing to the particle materials of asphalt concrete-surfaced roads, curvilinear cracks in the GRF image are represented with nonuniform intensity, low contrast with their surroundings, and low signal-to-noise ratio (SNR). Therefore, the aforementioned methods fail to extract cracks from the GRF images.

Tensor voting, a perceptual grouping method, as proposed in [22], is more powerful and efficient than the other methods for inferring curvilinear structures from noisy and corrupted data. Developed on the foundation of Gestalt psychology, the tensor voting method is based on tensor representations of image features and nonlinear voting. In the 2-D case of tensor voting, input data are first encoded as structure-aware tensors, where the structures are either points or curves in the feature space. The support information of proximity and continuation constraints propagates from tensor to tensor in a neighborhood through a voting process, by which the saliencies of the perceptual structures can be estimated from noisy and corrupted data in the form of votes. The more votes received by a given tensor, the higher the probability of a salient feature being present at the corresponding location [23]. Although noniterative tensor voting (ITV) is claimed to provide good results in many cases, an iterative version of the tensor voting framework demonstrated that revoting more effectively deals with complex data configurations as well as improves the orientation estimation at the input primitives and the overall curve inference results [24], [25]. The efficacy of the ITV framework, which combines tensor voting and iterative voting, was discussed and proven in [26] for the ill-defined curvilinear structures of medical cell membranes. The algorithm performed in [27] starts by encoding every pixel in the image as an unoriented ball tensor. Through ball tensor voting, all of the tensors obtain their preferred orientations, which indicate the potential curvilinear structures. A set of iterative stick tensor voting procedures is then imposed. Each iteration aims at refining the previous result at gradually reduced scales. Therefore, the iteration operation enhances the concentration of the votes over promising curvilinear structures.

Given that the cracks in the GRF images are represented as diffused and heterogeneous curvilinear structures, an ITV scheme is adopted to improve crack grouping and inference. Compared to traditional tensor voting algorithms, the modified ITV algorithm has the following two distinctions: 1) Prior to sparse voting, crack candidates are segmented and encoded as unit ball tensors. Due to the use of crack candidates, rather than all of the pixels in the image, the processing complexity is dramatically reduced. 2) In each iteration (in the form of dense voting), we consider both ball tensor voting and stick tensor voting, rather than stick tensor voting alone, for refining salient curvilinear structures by gradually reducing the aperture of the stick voting field. Dense ball-and-stick tensor voting can preserve many subtle curvilinear crack details.

In this paper, we develop a novel framework, called ITVCrack, for automated crack detection from MLS data. We first propose a curb-based road edge extraction algorithm that separates road points from nonroad points using MLS data. After classifying the MLS data, we interpolate the road points into GRF images using a modified inverse distance weighted (IDW) algorithm. We then extract cracks using modified ITV and morphological thinning.

The rest of this paper is organized as follows. Section II defines the tensor voting framework. Section III details the proposed ITVCrack. Section IV states three data sets for validating ITVCrack and discusses the experimental results. Section V concludes our work.

II. Tensor Voting Framework

Tensor voting consists of two components: tensor calculus for representation and nonlinear voting for data communication [26]–[32]. In 2-D, a second-order, symmetric, and nonnegative definite tensor is represented by a 2×2 matrix, decomposed as

$$T = (\lambda_1 - \lambda_2)e_1e_1^T + \lambda_2(e_1e_1^T + e_2e_2^T) \tag{1}$$

where λ_1 and λ_2 $(\lambda_1 > \lambda_2)$ are the eigenvalues; e_1 and e_2 are the corresponding eigenvectors. Geometrically, the tensor is visualized as an ellipse shaped by the tensor's eigenvectors' directions and eigenvalues' magnitudes. Specifically, the size and shape of a tensor are given by its eigenvalues, while the orientation is determined by the corresponding eigenvectors. The tensor's shape defines the structural type of interest (such as curves), and its size represents the saliency. The first term in (1) is termed stick tensor, indicating an elementary curve element with e_1 as its curve normal. The second term is called ball tensor, indicating a perceptual structure without any preferred orientations. For example, a crack pixel in the GRF image is represented by a stick tensor and visualized by a thin ellipse, whose major axis indicates the estimated preferred orientation e_1 and whose length $(\lambda_1 - \lambda_2)$ represents the saliency of the estimation.

Based on the tensor representation in 2-D, an input is first encoded as a tensor. If the input has no orientation, then it is encoded as a ball tensor with the eigenvalues of $\lambda_1 = \lambda_2 = 1$,

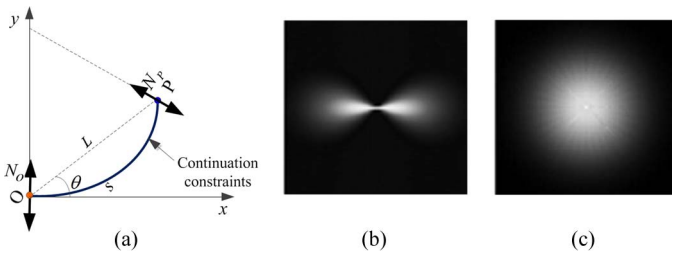


Fig. 1. (a) Vote generation. (b) Magnitude (saliency) of the 2-D stick voting field. (c) Magnitude (saliency) of the 2-D ball voting field.

in the form of the 2×2 identity matrix $T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. If the input has the orientation $\vec{n}(n_x, n_y)$, then it is encoded as a stick tensor with the eigenvalues of $\lambda_1 = 1$ and $\lambda_2 = 0$, in the form of the 2×2 matrix

$$T = \begin{bmatrix} n_x n_x & n_x n_y \\ n_x n_y & n_y n_y \end{bmatrix}.$$

After the inputs have been encoded as tensors, their information is propagated to their neighbors following the Gestalt principles of smoothness, proximity, and good continuation. This propagation is termed tensor voting. As presented in Fig. 1(a), two tensors, positioned at O and P with normal $\vec{N}$ in the $o{-}xy$ coordinate system, are named the voter and the receiver, respectively. The arc from O to P is the simplest way to model their smoothness and proximity. Let L denote the distance between the voter and the receiver; let θ denote the angle between the tangent to the osculating circle at the voter and the line going through the voter and the receiver. The arc length s and the curvature κ are given by $s = \theta L / \sin(\theta)$ and $\kappa = 2\sin(\theta)/L$, respectively. Thus, the path from O to P, defined by an osculating circle, is the most likely smooth path, given the assumption of the voter and the receiver belonging to the same perceptual structure. This is a consequence of the path maintaining a constant curvature. Thus, the vote at P is defined as

$$\begin{aligned} &V(P) = DF(s, \kappa, \sigma) N_P N_P^T \\ &\begin{cases} N_P = N_O \left[-\sin(2\theta), \cos(2\theta)\right]^T \\ DF(s, \kappa, \sigma) = e^{-\left(\frac{s^2 + c\kappa^2}{\sigma^2}\right)} \end{cases} \end{aligned} \tag{2}$$

where N_P and N_O are the normals to the tangents to the same osculating circle at P and O, respectively. $DF(s, \kappa, \sigma)$ is the saliency decay function. The scale parameter σ is viewed as a measure of smoothness and is the only free parameter for users. The coefficient c, which controls the degree of decay, is defined as

$$c = -16 \log(0.1) \times (\sigma - 1)/\pi^2. \tag{3}$$

The voting field serves as a look-up table that stores the precomputed votes cast by both stick and ball voters at their neighboring receivers at various distances and angles. Similar to linear convolution, this precomputed look-up table can accelerate the voting process. The tensor voting framework provides two forms of voting fields: stick voting field [see Fig. 1(b)] and ball voting field [see Fig. 1(c)]. The extent of the voting field is controlled by the scale parameter σ. A small σ corresponds to a small voting neighborhood; it makes the voting process local, leading to higher susceptibility to outlier corruption but better preservation of details. On the other hand, a large σ corresponds to a large voting neighborhood; a large σ enforces a higher degree of smoothness, thus assisting in noise removal. The stick field is limited to exist only for $|\theta| \leq 45^\circ$, and θ is called the field aperture. Beyond this scope, the smoothest path from O to P cannot be represented by the osculating circle formed by the tensors at O and P. To compute a vote cast by a tensor, the voting field is aligned to the tensor. Then, the magnitude and the orientation of the receiver can be looked up from the voting field.

Each input collects all of the votes cast by the tensors in its neighborhood and integrates them into a new tensor, eventually revealing behavioral coherence among the image primitives. Vote accumulation is performed by tensor addition: more specifically, the summation of 2×2 matrices. For example, the resulting tensor at P can be represented by

$$T_P = \sum T_O V(P) \tag{4}$$

where T_P is the summation tensor obtained by accumulating all of the votes $V(P)$ from its neighbors T_O at location O. Thus, after the votes are cast from tensor to tensor and accumulated by tensor addition, a new tensor at P is generated for structure extraction. The expected structures can be interpreted from the tensors. For example, in 2-D, if the new tensor at P has $\lambda_1 - \lambda_2 > \lambda_2$, then this indicates a curve with the estimated normal e_1 at that location. However, because outliers receive only inconsistent contradictory votes, the difference between the eigenvalues is small, leading to low saliency. Low saliency, in turn, indicates that the point in question is noise to be removed.

There are two types of voting in the voting process: sparse tensor voting and dense tensor voting. Sparse voting restricts tensors to cast votes to only other tensors, while dense voting involves tensors casting votes to all locations within their neighborhood, regardless of the presence of tensors or the lack thereof.

III. Description of ITVCrack

The proposed ITVCrack includes three steps: preprocessing, GRF image generation, and ITV-based crack extraction. The input to the ITVCrack algorithm is unorganized MLS point clouds, and the output is a group of curvilinear cracks. In this paper, we explore the application of MLS data to crack extraction. Cracks extracted from MLS data might require repair due to point resolution; as a result, we focus on extracting the type and location of pavement cracks and not the crack width.

A. Preprocessing

In most urban cities, curbs, which are nearly vertical surfaces, separate roads from sidewalks or green spaces. Based on this phenomenon, we propose a curb-based road extraction method. In this method, sharp jumps in height (representing curbs), which are used to identify road edges, are detected from the MLS data.

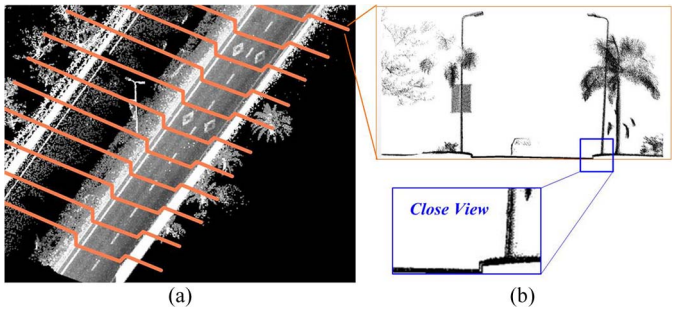


Fig. 2. Road surface extraction. (a) Data profiling. (b) Sample of a profile.

Moreover, vehicle trajectory data, which provide real-time position information about a vehicle, facilitate curb-based road extraction. Perpendicular to the trajectory data, we cross section, at intervals, the MLS data into a number of profiles as thin as a few centimeters, as shown in Fig. 2(a). Empirical analysis shows that, to keep enough points for curb detection, a suitable profile size is 30 cm, and a suitable interval length is 3 m for nearly straight roads and 1 m for sharp-turning roads. Fig. 2(b) shows one of the profiles, on which curbs are clearly demonstrated as vertical lines connecting road surfaces and sidewalks. Thus, two simple thresholds, namely, slope and elevation difference, are used to detect curbs—profile by profile.

Specifically, we first resample points for a profile to form a pseudoscan line with a given sampling size. The sampling size is dependent on the point density. In this paper, given the average point space of 2–6 cm, we set the sampling size to 5 cm. Then, we calculate the slope and elevation difference of two consecutive sampling points on the pseudoscan line. Most countries' street design and construction manuals specify that a curb is a nearly vertical surface, with a height generally ranging from 10 to 30 cm. Accordingly, for curb detection, we define 25 and 8 cm as the maximum and minimum elevation difference thresholds, respectively, and 75° as the slope threshold. For a point on the pseudoscan line, if 1) the slope is larger than the given slope threshold and 2) the elevation difference in the vicinity of the point is in the range between the maximum and minimum elevation difference thresholds, then the point is labeled as a curb candidate; otherwise, it is labeled as a noncurb point.

All curbs detected from the profiles are quite sparse because we section the MLS data along the vehicle trajectory data at certain intervals. Therefore, constrained by the trajectory data, we use a cubic spline interpolation method to generate two smooth road edges that separate road points from nonroad points.

B. GRF Image Generation

Then, we interpolate the extracted road surface point clouds into a GRF image by a modified IDW method. The traditional IDW interpolation method estimates the value of a given cell by averaging all of its point values. The closer a point is to the center of the cell, the higher its weight in the averaging process. This weight is termed distance weight. As we investigate the applicability of MLS intensity data to crack extraction, the

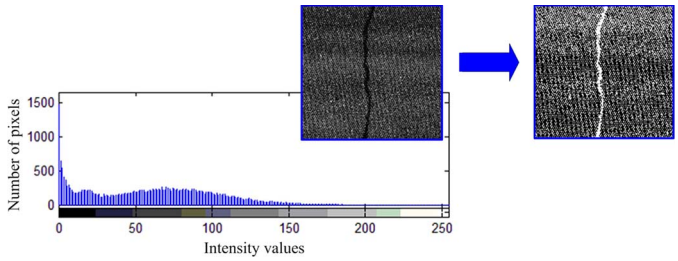


Fig. 3. Histogram analysis for a GRF image.

IDW method is modified by introducing intensity weight into the calculation of cell values. A point's intensity weight is proportional to its intensity value, i.e., a point with higher intensity is given a greater intensity weight. Image spatial resolution is determined by point density. A detailed description and discussion of GRF image generation and relevant parameters is found in [20].

Asphalt pavements mainly contain the following: 1) rocky components that probably vary with geological regions and 2) asphalt mixtures that are made up of a variety of chemical components. Thus, shadows and increasing surface roughness cause reflectance differences of up to 7%–8% in the near infrared range between the actual pavement and high severity cracks [33]. In addition, concave-shaped cracks in the visible/near-infrared range make noncracked areas brighter. Furthermore, compared to road surfaces, deeper layers, exposed by cracks, contain higher contents of the original asphalt mix, thereby increasing hydrocarbon absorption, which highlights the cracks' contrary spectral signals. Thus, the visual appearance of cracks in the near-infrared range is usually darker than that of the normal road surface. Based on this observation, an optimal threshold can be found to segment potential crack pixels from noncrack pixels by making use of image histograms and an objective function derived from information theory. Without noise, it would be successful to segment cracks from the background by means of a bimodal histogram structure. However, a huge amount of noise is scattered in a GRF image. Therefore, the image histogram in Fig. 3 displays no obvious peaks and valleys, resulting in the issue of finding the optimal separation value (T_E). As a result, we adopt the maximum entropy sum method to detect possible crack pixels by maximizing the information measures between crack and noncrack pixels [34].

C. ITV-Based Crack Extraction

Fig. 4 shows the flowchart of the ITV-based crack extraction algorithm. After thresholding, we assume that $P = \{p_1, p_2, p_3, \ldots, p_i, \ldots, p_n\}$ is the crack candidate data set, where n is the number of crack candidates; p_i is the ith crack candidate. First, as crack candidate p_i has no orientation preference, it is initially encoded, in the form of a 2×2 identity matrix, by a ball tensor with unit saliency. After construction of the tensor space, a first round of sparse voting is performed using the ball voting field with σ_{ball}.

After large-scale sparse ball voting, all tensors corresponding to crack candidates obtain rough orientations (e_1 and e_2) and

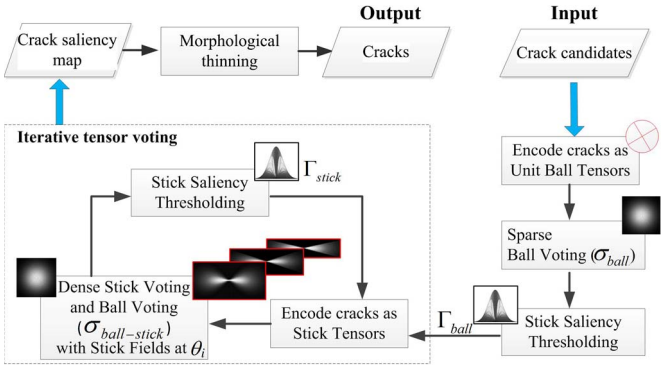


Fig. 4. Proposed ITV-based crack extraction algorithm.

magnitudes (λ_1 and λ_2). However, mapped cracks are inaccurate and lack saliency; therefore, a round of stick voting is required to refine the orientations and to obtain a saliency map of cracks. By nature, curvilinear structures in tensor representation should have high $\lambda_1 - \lambda_2$ values, i.e., crack candidates with $\lambda_1 - \lambda_2$ values that are smaller than the ball-saliency threshold Γ_{ball} are ruled out in this step. Eliminating the tensors with low stick saliencies increases computational efficiency because fewer crack candidates participate in ball-and-stick voting.

Each oriented crack candidate is further encoded as a stick tensor. A round of dense voting is then executed using the stick field. According to eigendecomposition, although ball tensors have no orientation preferences, they can still cast meaningful information to other tensors, contributing to the saliency concentration. For example, a potential curve could be influenced by two nearby ball tensors. Thus, we adopt both ball voting and stick voting using the stick voting field with $\sigma_{\text{ball-stick}}$ for the saliency map.

Usually, after the dense ball-and-stick voting process, curvilinear structures are enhanced on the resulting saliency map. However, the cracks of interest are presented with much noise and a low contrast with their surroundings. Only one round of dense ball-and-stick voting (namely, a combination of ball-and-stick voting) could not achieve a good saliency map for the cracks. An iterative scheme is thus proposed to gradually refine the previous results of the dense ball-and-stick voting.

For each iteration, dense ball-and-stick voting is employed using the stick voting field with $\sigma_{\text{ball-stick}}$. A stick saliency thresholding similar to the aforementioned ball-saliency thresholding is subsequently used to remove the resultant tensors with low $\lambda_1 - \lambda_2$ values, i.e., only the tensors with $\lambda_1 - \lambda_2$ values larger than the stick threshold Γ_{stick} will go to the next iteration. As such, each iteration refines the previous one. With the iterative scheme, the tensors with high $\lambda_1 - \lambda_2$ values appear to be concentrated and accurate with little disturbance and interference from the tensors with low $\lambda_1 - \lambda_2$ values; thus, we call this ITV.

Using dense ball-and-stick voting, the curvilinear structure becomes gradually more concentrated and accurate as the number of iterations increases, which means that the field aperture θ for the stick field can be correspondingly reduced to focus on the promising votes for enhanced results. For the stick field, let $\theta_{\max}$ and $\theta_{\min}$ denote the maximum and minimum field apertures, respectively. Also, let $\Delta\theta$ denote the voting aperture step. The number of iterations is calculated as

$$N = (\theta_{\max} - \theta_{\min})/\Delta\theta + 1. \tag{5}$$

For example, for the ith iteration, we employ the stick field with the field aperture of $\theta_i (\theta_i = \theta_{\max} - (i-1)\Delta\theta)$ for dense voting. Apart from assigning the voting aperture step for calculating the number of iterations, we can also empirically predefine N to stop the iterative processing. Finally, with ITV, a refined crack probability map is generated to enhance the crack pixels, simultaneously suppressing the background and the noise.

To further remove noise and obtain cracks in the crack probability map, a four-pass-per-iteration morphological thinning algorithm [26] is applied. This algorithm serves to thin the cracks to their median axes, by peeling off their boundary pixels. After implementation, the algorithm proposed in [35] produces a converged 8-connected one-pixel-thick skeleton.

IV. Results and Discussion

The stability and capability of ITVCrack were evaluated using synthetic data, pavement images, and GRF images. To objectively evaluate the performance, we used the manual interpretations of the crack curves in these images as the ground truth.

A. Data Sets

The following three data sets were used in this study.

1) **Two groups of synthetic data created with two different noise models**. The first group was generated with the standard additive white Gaussian noise model, while the second group was created with the multiplicative gamma noise model. In many cases, noise in pavement images is found to be additive in nature with uniform power in the whole bandwidth following the Gaussian probability distribution. In addition, multiplicative gamma noise, in the form of speckles, normally appears in laser-based images, thus degrading the quality of the images and affecting the performance of the image processing techniques [36]. All synthetic images are 200 × 200 pixels.
2) **A group of 1.5-mm ground sample distance (GSD) pavement images taken by a Canon IXUS 125HS camera with a megapixel count of 16.1**. This group contains three images with the size of 300 × 255 pixels.
3) **A group of GRF images interpolated from MLS point clouds acquired on April 23, 2012, by a RIEGL VMX-450 system in a tropical urban environment, Xiamen, a port city in southeast China**. The 25-km two-way four-lane road surveyed contained an increased number of cracks in its surface due to the hot and wet weather and the increased load caused by an ever-increasing traffic flow. This complete survey was conducted once in a forward direction and once in the reverse direction at an average speed of 50 km/h. A 105-m section of the road that contained 8.4 million points was selected. Using the vehicle trajectory, the road section

was first segmented into road and nonroad points in the preprocessing stage. Then, given that the point density on the road surface was as high as 4000–7000 points/m^2, a 2-cm GSD was used for generating a GRF image from the segmented road points. From the GRF image, we selected five areas that contained a variety of cracks, ranging from small cracks that were a few centimeters in width to large alligator cracks that were up to 10 cm wide.

B. Quantitative Assessment Measures

To quantitatively evaluate the crack extraction results, we used a buffered Hausdorff distance metric $(H(A,B))$ by comparing the detected cracks with the human-labeled cracks [3], [37]. $A=\{a_1,a_2,\ldots,a_p\}$ and $B=\{b_1,b_2,\ldots,b_q\}$ are the finite pixel sets corresponding to identical locations within the extracted crack image and the human-labeled image, respectively. The Hausdorff distance metric is given by

$$H(A,B)=\max\left(h(A,B),h(B,A)\right) \tag{6}$$

where

$$h(A,B)=\max_{a\in A}\min_{b\in B}\|a-b\| \tag{7}$$

and $\|\bullet\|$ is the Euclidean norm of the pixel sets A and B. The function $h(A,B)$ is called the directed Hausdorff distance from A to B, describing the degree of difference between two shapes. $h(A,B)$ identifies the point $a\in A$ that is the farthest from any point in B and measures the distance from a to its nearest neighbor in B. Essentially, $h(A,B)$ ranks each point in A based on its distance from the nearest point in B and then uses the distance corresponding to the highest ranking point. We used a buffer of size L to create a searching region, within which the Hausdorff distance metric was adopted to evaluate the crack extraction performance based on the ground truth. In the evaluation, the scoring measure (SM) is calculated by

$$\text{SM}=100-\frac{H(A,B)}{L}\times 100. \tag{8}$$

The value of SM ranges from 0 to 100. The higher the value of SM, the better the crack extraction performance. Considering that the cracks in the three data sets were not wider than 3 pixels, we assigned $L=5$ pixels for computing the values of SM.

C. Synthetic Data Tests

The synthetic data set is used to investigate the applicability of ITVCrack, in which the following five parameters are used: σ_{ball}, $\sigma_{\text{ball-stick}}$, Γ_{ball}, Γ_{stick}, and $\Delta\theta$. Among these five parameters, the thresholds Γ_{ball} and Γ_{stick} are both used to delete tensors with low stick saliencies and preserve tensors with high stick saliencies. Two scales of voting, namely, σ_{ball} and $\sigma_{\text{ball-stick}}$, control the neighborhood sizes for the sparse ball voting and the dense ball-and-stick voting in the iterations, respectively. In addition, the voting aperture step $\Delta\theta$ is used to control the number of iterations.

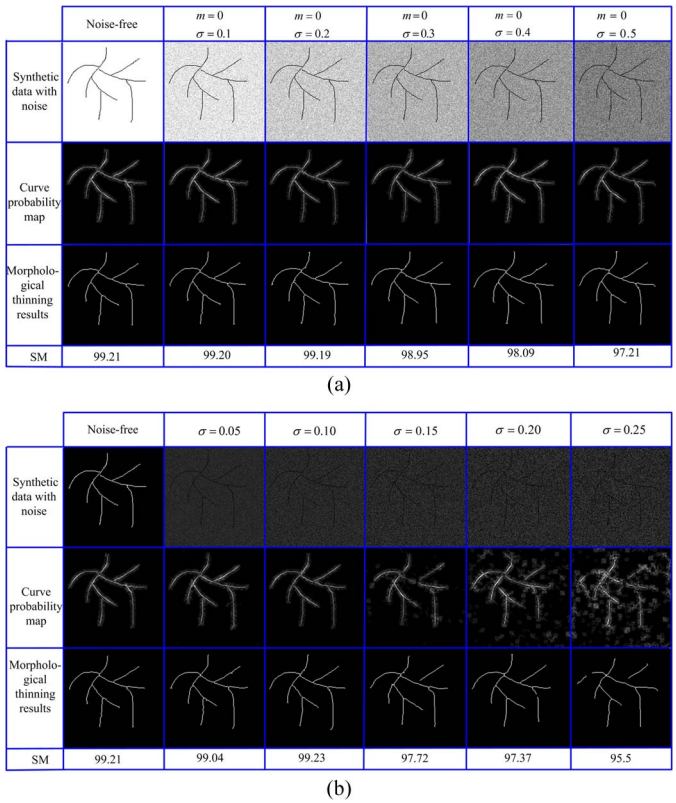


Fig. 5. Two groups of the synthetic data set. (a) Synthetic data with additive Gaussian white noise. (b) Synthetic data with multiplicative gamma noise.

Fig. 5(a) and (b), respectively, illustrates the synthetic images corrupted with additive and multiplicative noises with different variances. The processing parameters are $\sigma_{\text{ball}}=10.0$, $\sigma_{\text{ball-stick}}=3.0$, $\Gamma_{\text{ball}}=0.4$, $\Gamma_{\text{stick}}=0.05$, and $\Delta\theta=20^\circ$. Through visual inspection of the sample results, all parameters were defined and used for two groups of the synthetic image data sets, as well as throughout subsequent comparative experiments. As shown in Fig. 5(a), the synthetic data are corrupted by a set of additive noises following a zero-mean normal distribution $(m=0)$ with five different variances $(\sigma=0.1,0.2,0.3,0.4,\text{ and }0.5)$. In this group, the SM values of the detected curves exceed 97 for all five values of σ, demonstrating that the ITVCrack algorithm is noise robust and capable of preserving the details of curvilinear structures. In Fig. 5(b), the synthetic data are corrupted by a set of multiplicative noises with five different variances $(\sigma=0.05,0.10,0.15,0.20,\text{ and }0.25)$. Again, the SM values of all five detected curves exceed 95, indicating that ITVCrack can handle cracks, which have low contrasts with their surroundings (such as asphalt concrete-surfaced pavements) in the GRF image.

D. Comparative Tests With Pavement Images

In order to further evaluate the performance and feasibility of our ITVCrack method, we compare it with two newly proposed methods—FoSA (F$*$ seed growing) [4] and CrackTree [38]—for extracting cracks in the real pavement images. The dynamic-optimization-based method, suggested in [3], outperforms the other five methods for segmenting low-SNR images. We also selected it for comparison [39]. In working toward

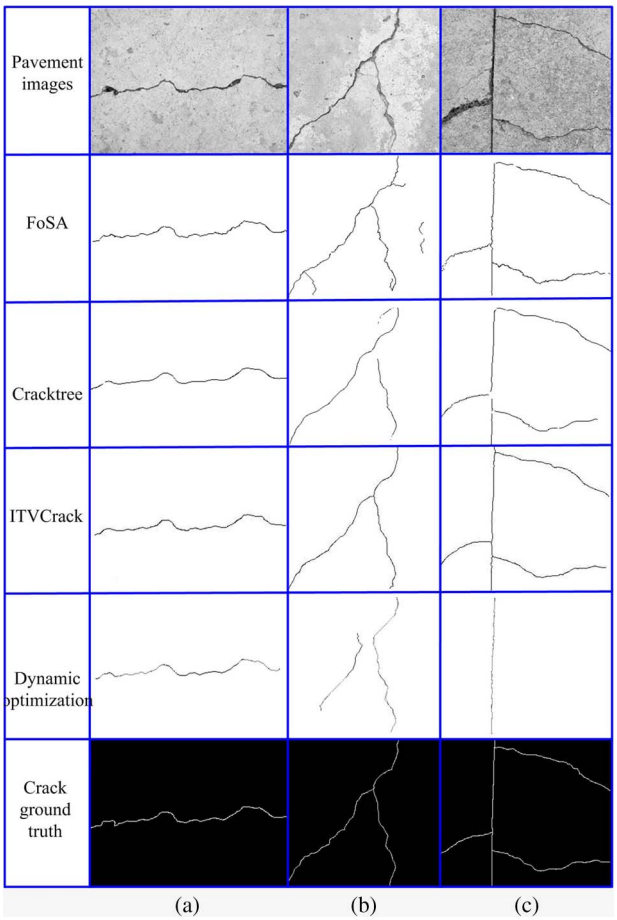


Fig. 6. Comparison of ITVCrack with the other approaches using the pavement images. (a) Image 1. (b) Image 2. (c) Image 3.

TABLE I
SM VALUES OF FOUR CRACK EXTRACTION METHODS

	Image 1	Image 2	Image 3
ITVCrack	93.13	90.04	91.64
CrackTree	93.54	89.17	89.66
FoSA	92.98	76.09	92.75
Dynamic optimization	94.22	78.38	33.77

the objectives of this study, we conducted an experimental study to compare the performance of existing crack extraction methods. It is accepted that crack detectors are neither perfect nor universally applicable. Although most of them work in most situations and with most data types, they will fail under certain environmental conditions.

Fig. 6 shows the results obtained using each of the four existing crack extraction methods. Table I lists the SM values of the extracted cracks, in comparison with the ground truth. In the CrackTree method, three parameters were selected as follows: the voting scale $\sigma = 11$, the edge-length threshold $L_e = 10$, and the path-length threshold $L_p = 60$. The FoSA algorithm maintains the searching radius at 24. As shown in Table I, given the high spectral and spatial resolutions of the pavement images, most algorithms achieve a good performance in crack extraction. However, the FoSA algorithm achieves a lower SM value (76.09) for image 2, compared to the other images. This might be spectral inconsistency around the cracks in image 2. Thus, this algorithm mistakenly identified the boundary of the

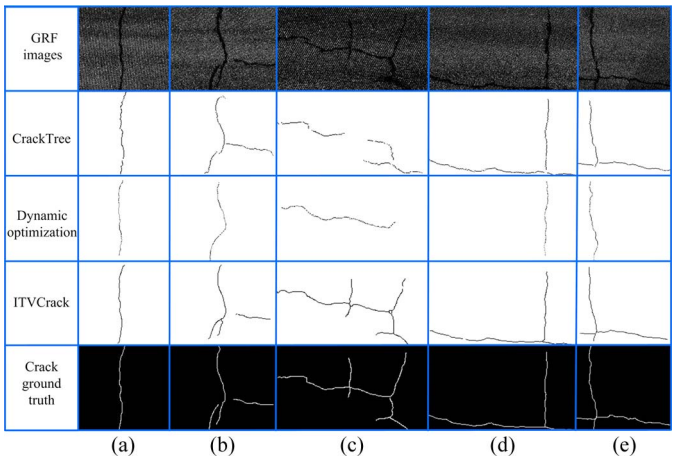


Fig. 7. Comparison of ITVCrack with the other approaches using the GRF images. (a) Crack 1. (b) Crack 2. (c) Crack 3. (d) Crack 4. (e) Crack 5.

TABLE II
SM VALUES OF THREE CRACK EXTRACTION METHODS

	Crack 1	Crack 2	Crack 3	Crack 4	Crack 5
ITVCrack	97.32	95.64	94.14	96.64	95.79
CrackTree	92.08	65.15	49.62	94.02	77.37
Dynamic optimization	84.08	51.76	46.60	31.90	43.60

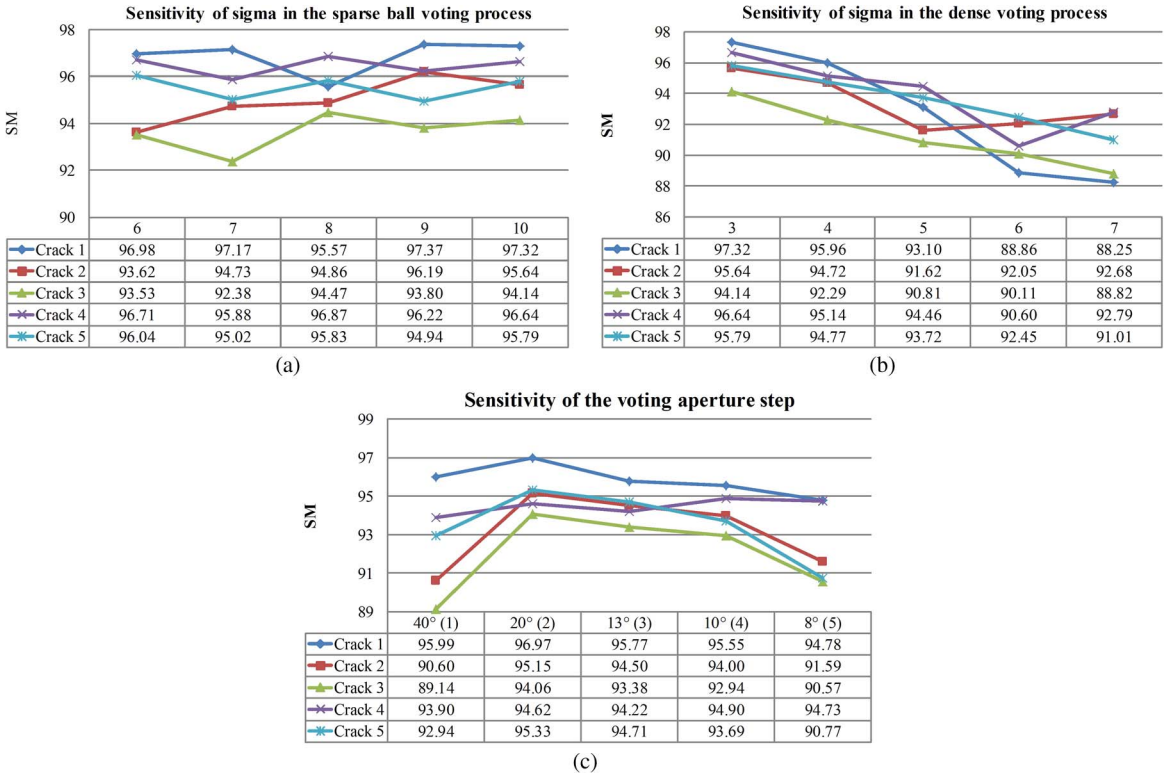
slightly dark area as a crack, thereby leading to a higher false alarm rate.

Similarly, the dynamic optimization method deals poorly with image 3, as indicated by the SM value of 33.77. This might be caused by the low contrast between the crack pixels and their surroundings. The dynamic optimization method, using connected component analysis, detects cracks from such local information as density, relative area, bounding box, and line similarity. For this reason, low contrast in a local window might cause the algorithm to inadequately extract crack information for connected component analysis. As expected, our ITVCrack attained stable performance for all three types of images. Qualitatively, all cracks were extracted completely (see Fig. 6). Quantitatively, Table I suggests that ITVCrack outperforms the other three algorithms, as indicated by the SM values being higher than 90%.

E. Comparative Tests With GRF Images

The tests on the synthetic images and pavement images indicated that ITVCrack can extract all possible sharp curvilinear structures in the presence of severe noise. In comparison with the pavement images, cracks in the GRF images show lower contrast with their surroundings and lower SNRs with a huge amount of noise. In order to evaluate the effectiveness of ITVCrack for these noisy and corrupted GRF images, we compared it with the aforementioned algorithms in this section. The first row in Fig. 7 shows five cracks sectioned from the GRF image. Cracks 1, 2, and 5 have the size of 200 $\times$ 200 pixels, and cracks 3 and 4 have the size of 250 $\times$ 150 pixels.

Given the low spectral resolution of the GRF images, the searching radius r in the FoSA algorithm must be smaller in order to obtain a consistent window for seed growing. However, the small searching radius r makes it difficult to represent the seed-growing path of cracks. Moreover, spectral inconsistency



	6	7	8	9	10
Crack 1	96.98	97.17	95.57	97.37	97.32
Crack 2	93.62	94.73	94.86	96.19	95.64
Crack 3	93.53	92.38	94.47	93.80	94.14
Crack 4	96.71	95.88	96.87	96.22	96.64
Crack 5	96.04	95.02	95.83	94.94	95.79

(a)

	3	4	5	6	7
Crack 1	97.32	95.96	93.10	88.86	88.25
Crack 2	95.64	94.72	91.62	92.05	92.68
Crack 3	94.14	92.29	90.81	90.11	88.82
Crack 4	96.64	95.14	94.46	90.60	92.79
Crack 5	95.79	94.77	93.72	92.45	91.01

(b)

	40° (1)	20° (2)	13° (3)	10° (4)	8° (5)
Crack 1	95.99	96.97	95.77	95.55	94.78
Crack 2	90.60	95.15	94.50	94.00	91.59
Crack 3	89.14	94.06	93.38	92.94	90.57
Crack 4	93.90	94.62	94.22	94.90	94.73
Crack 5	92.94	95.33	94.71	93.69	90.77

(c)

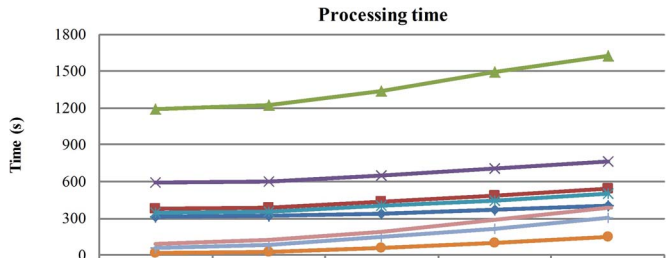
Fig. 8. Sensitivity tests with parameters. (a) σ_{ball}. (b) $\sigma_{\text{ball-stick}}$. (c) $\Delta\theta$.

caused by the point sampling pattern of the MLS data leads to the failure of the FoSA algorithm to adequately extract cracks. This algorithm might work for cracks if relevant preprocessing procedures, such as filtering, are employed.

The comparison of the results of the other three methods is shown in Fig. 7. The quantitatively compared results are listed in Table II. The proposed ITVCrack maintained a much more stable performance than CrackTree and dynamic optimization because the SM values for ITVCrack range from near 94 to 97. In addition, visual inspection shows that the extracted cracks are quite complete. These quantitative and qualitative results demonstrate that our algorithm achieves a stable performance for not only the pavement images but also the noisy GRF images. The dynamic optimization method achieves a poor performance for the complex shaped cracks in the GRF image. Due to the robustness of tensor voting under the conditions of low SNR and low spectral contrast, the tensor voting based CrackTree also outperforms dynamic optimization. However, compared to CrackTree, the proposed ITVCrack, following an iterative processing pattern, enhances promising, salient curvilinear cracks and suppresses surrounding noise by gradually reducing the voting aperture. Thus, ITVCrack can preserve many more crack details.

F. Parameter Sensitivity Analysis With GRF Images

Among the five parameters ($\sigma_{\text{ball}}, \sigma_{\text{ball-stick}}, \Gamma_{\text{ball}}, \Gamma_{\text{stick}}$, and $\Delta\theta$), histogram analysis suggests values of $\Gamma_{\text{ball}} = 0.4$ and $\Gamma_{\text{stick}} = 0.05$. In this section, we designed three groups of experiments to investigate the sensitivity of ITVCrack to the selection of the scale parameters σ_{ball} and $\sigma_{\text{ball-stick}}$, and the voting aperture step $\Delta\theta$.

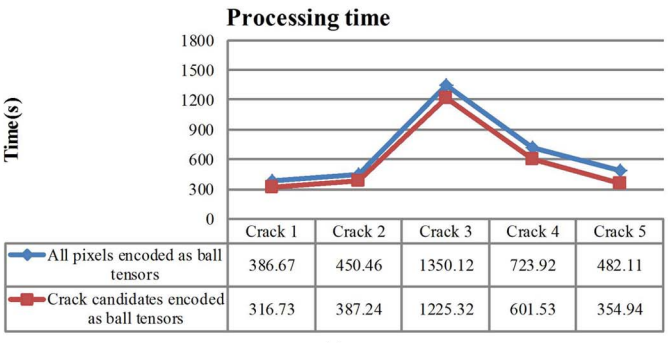


	40° (1)	20° (2)	13° (3)	10° (4)	8° (5)
Crack 1	310.45	316.73	340.67	367.90	400.66
Crack 2	378.72	387.24	435.00	484.66	539.47
Crack 3	1191.81	1225.32	1341.25	1493.20	1628.34
Crack 4	588.71	601.53	646.58	703.51	763.29
Crack 5	348.79	354.94	401.10	445.32	500.64
Image 1	17.36	28.63	62.11	103.43	152.38
Image 2	58.68	80.05	144.72	215.72	305.44
Image 3	90.01	120.21	192.46	287.06	388.81

Fig. 9. Runtime for the GRF and pavement images.

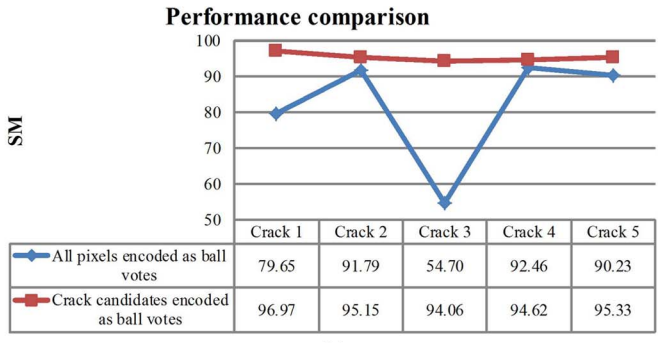
In the first group, we maintained $\sigma_{\text{ball-stick}} = 3.0$ and $\Delta\theta = 20^\circ$ and varied σ_{ball} from 10.0 to 6.0 with an interval of 1.0. Fig. 8(a) shows the experimental results for these five cracks. As shown in Fig. 8(a), the SM values of the extracted cracks dramatically vary with the parameter σ_{ball}, which increases from 6.0 to 9.0. However, the SM values tend to be stable as the parameter σ_{ball} changes from 8.0 to 10.0. The reason behind this phenomenon might be that a large σ_{ball} implies long-range interactions, leading to a higher degree of smoothness (i.e., greater noise removal), thus improving extraction performance. As such, in our study, the best crack extractions were obtained at $\sigma_{\text{ball}} = 9.0$ or 10.0.

Next, we used $\sigma_{\text{ball}} = 10.0$ and $\Delta\theta = 20^\circ$ and varied $\sigma_{\text{ball-stick}}$ from 3.0 to 6.0 with an interval of 1.0. Fig. 8(b) shows the results for these five cracks. As shown in Fig. 8(b),



	Crack 1	Crack 2	Crack 3	Crack 4	Crack 5
All pixels encoded as ball tensors	386.67	450.46	1350.12	723.92	482.11
Crack candidates encoded as ball tensors	316.73	387.24	1225.32	601.53	354.94

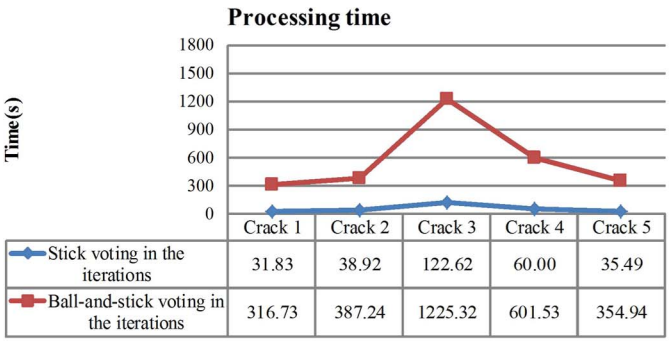
(a)



	Crack 1	Crack 2	Crack 3	Crack 4	Crack 5
All pixels encoded as ball votes	79.65	91.79	54.70	92.46	90.23
Crack candidates encoded as ball votes	96.97	95.15	94.06	94.62	95.33

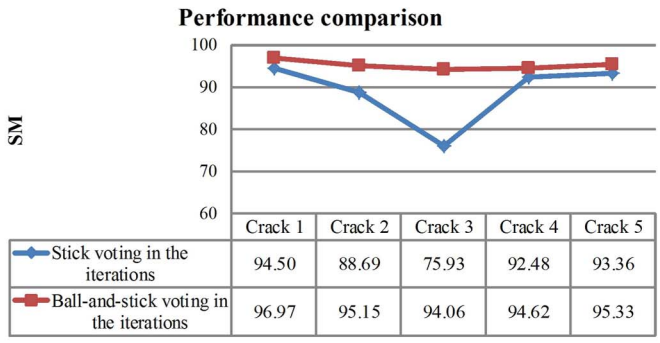
(b)

Fig. 10. Quantitative comparisons between the methods using all pixels and crack candidates encoded as ball tensors. (a) Processing time. (b) SM values.



	Crack 1	Crack 2	Crack 3	Crack 4	Crack 5
Stick voting in the iterations	31.83	38.92	122.62	60.00	35.49
Ball-and-stick voting in the iterations	316.73	387.24	1225.32	601.53	354.94

(a)



	Crack 1	Crack 2	Crack 3	Crack 4	Crack 5
Stick voting in the iterations	94.50	88.69	75.93	92.48	93.36
Ball-and-stick voting in the iterations	96.97	95.15	94.06	94.62	95.33

(b)

Fig. 11. Quantitative comparisons between the methods using stick voting alone and ball-and-stick voting in the iterations. (a) Processing time. (b) SM values.

when $\sigma_{\text{ball-stick}}$ is 3.0, the proposed ITVCrack achieved a relatively stable performance, indicated by the SM values being higher than 95%. The values of SM quickly decreased when $\sigma_{\text{ball-stick}}$ increased from 4.0 to 7.0. This is because, unlike σ_{ball} in the sparse voting process (which is given a large value to remove noise), $\sigma_{\text{ball-stick}}$ in the iteration process requires a small scale in order to preserve crack details. Due to the large amount of noise removed by σ_{ball} in the sparse voting process, iterative dense voting is able to enhance the cracks by preserving their details. In this paper, the $\sigma_{\text{ball-stick}}$ value of 3.0 obtained the best crack extraction performance.

Finally, we used $\sigma_{\text{ball}} = 10.0$ and $\sigma_{\text{ball-stick}} = 3.0$ and varied $\Delta\theta$ from 5° to 40° with five different $\Delta\theta$ settings (namely, 40°, 20°, 13°, 10°, and 8°). We used the maximum field aperture ($\theta_{\max} = 45°$) and the minimum field aperture ($\theta_{\min} = 5°$). The voting aperture step $\Delta\theta$ determines the number of iterations (N) in the dense voting process. Therefore, according to (5), the ITVCrack algorithm was performed in five different iterations (5, 4, 3, 2, and 1). As shown in Fig. 8(c), when $\Delta\theta$ is between 10° and 15° (which entails three or four iterations), the SM values of all five cracks exhibit good performance. The explanation for this phenomenon is that, in the iterative dense voting process, each iteration refines the previous one by gradually reducing the diffusion of votes and focusing the votes on only promising curves. It has been found, however, that although dense voting allows pixels to be interpolated for filling discontinuity, excessive iterative dense tensor voting (small $\Delta\theta$) produces overly smooth crack curves due to over interpolation. Consequently, some crack details could be missed, resulting in a decrease in SM values.

G. Computational Efficiency

Our analysis indicates that the proposed ITVCrack, because it gradually concentrates on the promising crack curvilinear structures by refining previous results, is capable of extracting cracks from noisy and corrupted GRF images. However, iteration results in increased required computation time. Fig. 9 shows the runtime for eight pieces of crack data, including five GRF images and three pavement images. As shown in Fig. 9, the runtime grows as the number of iterations increases. However, for all cracks, the runtime growth rates are low. The reason is that the algorithm iteratively employs a saliency thresholding scheme to delete pixels with low saliency and gradually focuses the votes on only promising curves. The first round of dense ball-and-stick voting, particularly ball voting, occupies the majority of the processing time. We also found that, compared to the five GRF images, the runtime for the three pavement images is much shorter in spite of their larger sizes. This is because the GRF images contain much more noise than the pavement images, and ITVCrack takes considerable time to concentrate the promising cracks.

Rather than all pixels, only the crack candidates binarized from the GRF images are the input to be encoded as ball tensors. As a result, the computational cost, as shown in Fig. 10(a), is reduced by 10%–25% for all five GRF images, yet as shown in Fig. 10(b), the values of SM dramatically increase by 5%–40%. With little interference from noncrack pixels, the proposed ITVCrack concentrates on crack candidates, thus improving the ITVCrack's performance and stability for crack extraction, as shown by SM values about 95. Traditional tensor voting algorithms generally use a dense stick voting process for

gradually concentrating on curvilinear structures. However, as we mentioned, a ball tensor contains implicit stick information after ball voting according to eigendecomposition. Fig. 11(a) and (b) shows the comparative results between dense stick voting and dense ball-and-stick voting in each iteration. We found that the algorithm using only stick voting dramatically reduced the computational cost by 90%. However, the accuracy of the extracted cracks is unstable as the SM value ranged from 75.93 to 94.50. On the other hand, ball-and-stick voting preserves many subtle curvilinear crack details, entailing a higher degree of accuracy; however, it leads to the disadvantage of an increase in computational costs. In order to overcome the aforementioned shortcoming, a parallel environment or a multithread scheme may be considered.

V. Conclusion

We have proposed ITVCrack, an ITV-based framework for extracting cracks in road surfaces from MLS point clouds. The presented ITVCrack combines the following: 1) curb-based road extraction; 2) GRF image generation; and 3) ITV-based crack extraction. The performance of ITVCrack was validated by the synthetic data, the pavement images, and the GRF images. Quantitatively, our algorithm demonstrated much better crack extraction performance when compared with the other crack extraction methods. Specifically, ITVCrack achieved the following: 1) the SM values of over 97 and 95 for the additive-noise-corrupted and multiplicative-noise-corrupted synthetic data, respectively; 2) the average SM value of 91.5 for the pavement images; and 3) the best SM values, ranging from near 94 to 97, for the GRF images.

One limitation is the intensive computation required due to the iterative operations involved in the tensor voting process, particularly dense ball voting. However, because the voting process of each tensor is independent, this disadvantage could be ameliorated by future research on distributed computing. Using a multithread scheme in a parallel environment, the computational burdens can be distributed to each parallel procedure, indicating that computational performance will be improved and time complexity will be greatly reduced. In addition, ITVCrack is scale independent and has been shown to perform equally well on images with high spatial resolution. As the progressing technology of laser scanning permits higher point densities, this is a feasible property of future GRF images.

Acknowledgment

The authors would like to thank Dr. Q. Zou and Dr. Y. Q. Huang of Wuhan University, Wuhan, China, for providing their valuable experimental results for our comparison, M. McAllister for his assistance in proofreading our paper, and the anonymous reviewers for their valuable comments. For details about the software tool please contact author J. Li.

References

[1] H. Lee, "Standardization of distress measurements for the network-level pavement management system, pavement management implementation," in *Proc. ASTM STP*, F. B. Holt and W. L. Gramling, Eds., 1992, vol. 1121, pp. 424–436, Amer. Soc. Testing and Materials, Philadelphia, PA, USA.

[2] K. H. McGhee, "Automated pavement distress collection techniques: A synthesis of highway practice," Transp. Res. Board, Washington, DC, USA, NCHRP Synthesis 334, 2004.

[3] Y. Tsai, V. Kaul, and R. M. Mersereau, "Critical assessment of pavement distress segmentation methods," *J. Transp. Eng.*, vol. 136, no. 1, pp. 11–19, Jan. 2010.

[4] Q. Li, Q. Zou, D. Zhang, and Q. Mao, "FoSA: F∗ seed-growing approach for crack-line detection from pavement images," *Image Vis. Comput.*, vol. 29, no. 12, pp. 861–872, Nov. 2011.

[5] T. Nguyen, S. Begot, F. Duculty, and M. Avila, "Free-form anisotropy: A new method for crack detection on pavement surface images," in *Proc. 18th IEEE Int. Conf. Image Process.*, 2011, pp. 1069–1072.

[6] H. N. Koutsopoulos, I. El Sanhouri, and A. B. Downey, "Analysis of segmentation algorithms for pavement distress images," *J. Transp. Eng.*, vol. 119, no. 6, pp. 868–888, Nov. 1993.

[7] N. Otsu, "A threshold selection method from gray-level histogram," *IEEE Trans. Syst., Manage Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[8] J. Kittler and J. Iuingworth, "Minimum error thresholding," *Pattern Recog.*, vol. 19, no. 1, pp. 41–47, Nov. 1986.

[9] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[10] A. Cuhadar, K. Shalaby, and S. Tasdoken, "Automatic segmentation of pavement condition data using wavelet transform," in *Proc. IEEE Can. Conf. Elect. Comput. Eng.*, New York, NY, USA, 2002, vol. 2, pp. 1009–1014.

[11] J. Zhou, P. S. Huang, and F. P. Chiang, "Wavelet-based pavement distress detection and evaluation," *Opt. Eng.*, vol. 45, no. 2, pp. 027007–027010, Feb. 2006.

[12] N. Tanaka, "A crack detection method in road surface image using morphology," in *Proc. IAPR Conf. MVA*, Chiba, Japan, Nov. 17–19, 1998, pp. 154–157.

[13] H. Oliveira and P. L. Correia, "Identifying and retrieving distress images from road pavement surveys," in *Proc. ICIP*, 2008, pp. 57–60.

[14] M. Gavilán, D. Balcones, O. Marcos, D. F. Llorca, and M. A. Sotelo, "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, Oct. 2011.

[15] T. Saar and O. Talvik, "Automatic asphalt pavement crack detection and classification using neural networks," in *Proc. 12th BEC*, Tallinn, Estonia, Oct. 4–6, 2010, pp. 345–348.

[16] V. Ussyshkin, "Mobile laser scanning technology for surveying application: From data collection to end-products," in *Proc. FIG Working Week*, Eilat, Israel, May 3–8, 2009, pp. 1–13.

[17] J. Li and M. A. Chapman, "Terrestrial mobile mapping towards real-time geospatial data collection," in *Geospatial Information Technology for Emergency Response*, S. Zlatanova and J. Li, Eds. New York, NY, USA: Taylor & Francis, 2007, pp. 103–119.

[18] B. Jutzi and H. Gross, "Normalization of LiDAR intensity data based on range and surface incidence angle," in *Proc. ISPRS Arch.*, 2009, vol. 38-3/W8, pp. 213–218.

[19] A. Jaakkola, J. Hyyppä, H. Hyyppä, and A. Kukko, "Retrieval algorithms for road surface modelling using laser-based mobile mapping," *Sensors*, vol. 8, no. 9, pp. 5238–5249, 2008.

[20] H. Guan *et al.*, "Using mobile laser scanning data for automated extraction of road markings," *ISPRS J. Photogramm. Remote Sens.*, vol. 87, pp. 93–107, Jan. 2014.

[21] Y. Lin and J. Hyyppä, "Geometry and intensity based culvert detection in mobile laser scanning point clouds," *J. Appl. Remote Sens.*, vol. 4, no. 1, pp. 043553-1–043553-15, Nov. 2010.

[22] G. Medioni, M. S. Lee, and C.-K. Tang, *A Computational Framework for Segmentation and Grouping*. Amsterdam, The Netherlands: Elsevier, 2000.

[23] W. Tong and C. Tang, "Robust estimation of adaptive tensors of curvature by tensor voting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 434–449, Mar. 2005.

[24] S. Fischer, P. Bayerl, H. Neumann, G. Cristobal, and R. Redondo, "Are iterations and curvature useful for tensor voting?" in *Proc. 8th Eur. Conf. Comput. Vis.*, T. Pajdla and J. Matas, Eds., 2004, pp. 158–169.

[25] B. Parvin *et al.*, "Iterative voting for inference of structural saliency and characterization of subcellular events," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 615–623, Mar. 2007.

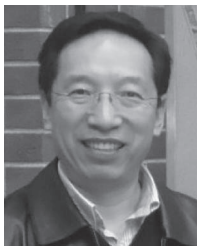
[26] L. A. Loss, G. Bebis, and B. Parvin, "Iterative tensor voting for perceptual grouping of ill-defined curvilinear structures," *IEEE Trans. Med. Imag.*, vol. 30, no. 8, pp. 1503–1513, Aug. 2011.

[27] G. Guy and G. Medioni, "Inference of surfaces, 3D curves, and junctions from sparse, noisy, 3D data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 11, pp. 1265–1277, Nov. 1997.

[28] M. K. Park, S. J. Lee, and K. H. Lee, "Multi-scale tensor voting for feature extraction from unstructured point clouds," *Graph. Models*, vol. 74, no. 4, pp. 197–208, Jul. 2012.
[29] T. N. Dinh, J. Park, and G. S. Lee, "Voting based text line segmentation in handwritten document images," in *Proc. 10th IEEE Int. Conf. CIT*, Bradford, U.K., Jun. 29–Jul. 1, 2010, pp. 529–535.
[30] G. Guy and G. Medioni, "Inferring global perceptual contours from local features," *Int. J. Comput. Vis.*, vol. 20, no. 1–2, pp. 113–133, Oct. 1996.
[31] C. K. Tang and G. Medioni, "Inference of integrated surface, curve, and junction descriptions from sparse 3D data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1206–1223, Nov. 1998.
[32] C. K. Tang and G. Medioni, "Curvature-augmented tensor voting for shape inference from noisy 3D data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 858–864, Jun. 2002.
[33] H. Martin, "Spectral characteristics of asphalt road surfaces," in *Remote Sensing of Impervious Surfaces*, Q. Weng, Ed. Boca Raton, FL, USA: CRC Press, 2008, pp. 237–247.
[34] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray level picture thresholding using the entropy of the histogram," *Comput. Vis. Graph. Image Proc.*, vol. 29, no. 3, pp. 273–285, Mar. 1985.
[35] B. K. Jang and R. T. Chin, "Analysis of thinning algorithms using mathematical morphology," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 6, pp. 541–551, Jun. 1990.
[36] Y. Hawwar and A. Reza, "Spatially adaptive multiplicative noise image denoising technique," *IEEE Trans. Image Proc.*, vol. 11, no. 12, pp. 1397–1404, Dec. 2002.
[37] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 9, pp. 850–863, Sep. 1993.
[38] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, "CrackTree: Automatic crack detection from pavement images," *Pattern Recog. Lett.*, vol. 33, no. 3, pp. 227–238, Feb. 2012.
[39] Y. Huang and J. Tsai, "Enhanced pavement distress segmentation algorithm using dynamic programming and connected component analysis," *J. Transport. Res. Board*, vol. 2225, pp. 89–98, 2011.

Haiyan Guan received the Ph.D. degree in geomatics from the University of Waterloo, Waterloo, ON, Canada, in 2014.

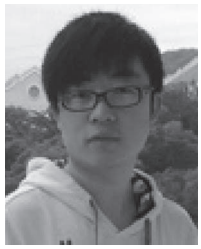
She is currently a Senior Research Fellow with the Department of Geography and Environmental Management, University of Waterloo. Her research interests are airborne, terrestrial, and mobile laser scanning data processing algorithms and 3-D spatial modeling and reconstruction of critical infrastructure and landscape. She is the coauthor of more than 30 research papers published in refereed journals, books, and proceedings.



Jonathan Li (M'00–SM'11) received the Ph.D. degree in geomatics engineering from the University of Cape Town, Cape Town, South Africa.

He is currently with the Key Laboratory of Underwater Acoustic Communication and Marine Information Technology (MOE) and the School of Information Science and Engineering, Xiamen University, Xiamen, China. He is also the Head of Waterloo Laboratory for GeoSpatial Technology and Remote Sensing, Faculty of Environment, University of Waterloo, Waterloo, ON, Canada, where he is a Professor and an elected member of the University Senate. He is the coauthor of more than 200 publications, over 60 of which were published in refereed journals. His current research interests include information extraction from earth observation images and 3-D surface reconstruction from mobile LiDAR point clouds.

Dr. Li is the Chair of the Inter-Commission Working Group I/Va on Mobile Scanning and Imaging Systems of the International Society for Photogrammetry and Remote Sensing (2012–2016), the Vice Chair of the Commission on Hydrography of the International Federation of Surveyors (2011–2014), and the Vice Chair of the Commission on Mapping from Remote Sensor Imagery of the International Cartographic Association (2011–2015).



Yongtao Yu received the B.S. degree in computer science from Xiamen University, Xiamen, China, in 2010, where he is currently working toward the Ph.D. degree in the Department of Computer Science.

His current research interests include computer vision, machine learning, mobile laser scanning, and point cloud processing.

Michael Chapman received the Ph.D. degree in photogrammetry from Laval University, Quebec, QC, Canada.

He is a Professor of geomatics with the Department of Civil Engineering, Ryerson University, Toronto, ON, Canada. Prior to joining Ryerson University, he was a Professor with the Department of Geomatics Engineering, University of Calgary, Calgary, AB, Canada, for 18 years. He is the author and coauthor of more than 160 technical articles. His current research interests include algorithms and processing methodologies for airborne sensors using GPS/INS, geometric processing of digital imagery in industrial environments, terrestrial imaging systems for transportation infrastructure mapping, and algorithms and processing strategies for biometrology applications.

Hanyun Wang (M'13) received the Master's degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2010, where he is currently working toward the Ph.D. degree in the School of Electronic Science and Engineering.

His research interests include computer vision, machine learning, pattern recognition, mobile laser scanning, and point cloud processing.

Cheng Wang (M'12) received the Ph.D. degree in information and communication engineering from the National University of Defense Technology, Changsha, China, in 2002.

He is a Professor and the Associate Dean of the School of Information Science and Technology, Xiamen University, Xiamen, China. He is the author of more than 80 papers. His current research interests include remote sensing image processing, mobile LiDAR data analysis, and multisensor fusion.

Dr. Wang is the Cochair of ISPRS WG I/3, a council member of the Chinese Society of Image and Graphics, and a member of SPIE and IEEE GRSS.

Ruifang Zhai received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2006.

She is an Associate Professor with the Department of Computer Science, School of Informatics, Huazhong Agricultural University, Wuhan. She is the coauthor of over ten research papers published in referred journals and proceedings. Her research interests include airborne and mobile laser scanning data processing algorithms, and 3-D modeling and reconstruction of objects.