A Fast and Accurate Automated Pavement Crack Detection Algorithm

Anirban Chatterjee[1] and Yi-Chang (James) Tsai[2]

Abstract— **Over the last 20 years, several crack detection algorithms have been developed to implement safe and efficient automated road condition survey (ARCS) systems. Although the current state-of-the-art algorithms can achieve a high level of accuracy, their computation time makes them infeasible to implement in real-time without massive parallelization. This paper presents a fast and accurate crack detection algorithm. The algorithm consists of the following major steps: 1) Image preprocessing; 2) Preliminary crack segmentation to minimize false negatives; 3) Crack object generation and connection to remove false positives; and 4) Refinement of the crack segmentation through a minimal path search based procedure. The proposed algorithm achieves an overall score of 80 in the Crack Detection Algorithm Performance Evaluation System (CDA-PES). With a median processing time of 0.52 seconds for 0.65 megapixel images on a single CPU thread, this algorithm makes accurate, real-time processing viable. The research presented in this paper contributes towards more widespread adoption of safer and efficient automated road condition surveys.**

I. INTRODUCTION

Regular road infrastructure condition surveys are important for optimizing the use of resources for infrastructure maintenance. Road infrastructure condition surveys can be classified as manual, semi-automated or automated. In manual surveys, engineers determine the infrastructure condition during a field survey which is unsafe, laborious and time-consuming. In semi-automated surveys, the road infrastructure is recorded using on-vehicle sensors and analyzed manually later. This makes the surveys safer but still laborious and time-consuming. Automated surveys involve the use of on-vehicle sensors and distress detection algorithms to automatically determine the pavement condition making them the preferred alternative. However, very few transportation agencies have adopted automated road infrastructure condition surveys (ARCS).

Pavement cracking is one of the most widespread pavement distresses and has the largest impact on road condition [1]. Properties such as crack type, crack length, cracked area and crack width are required to quantify the pavement condition [2] and for subsequent MR&R decision-making.

The most common approach to crack detection is based on classifying image blocks or individual pixels as cracked or non-cracked, based on the aggregate statistics of the entire image or the neighborhood of the image blocks or pixels under question. Each image block or pixel is treated as a separate classification problem. Hence, the continuity of crack features is not guaranteed. This approach will be referred to as individual classification [3], [4], [5], [6], [7]. Individual classification can also be applied to the frequency domain transform of the image. Edge detection [8] based methods [9], [10] and wavelet transform [11] based methods [12], [13], [14] fall under this category. Supervised learning has also been extensively applied for crack detection using individual classification [15], [16], [17].

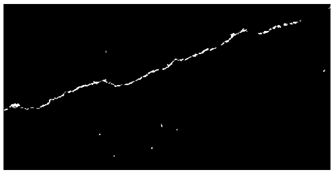


Fig. 1: Disjoint crack segmentation from individual classification

As mentioned before, individual classification does not consider that cracks are continuous, salient features. As a result, the crack segmentation generally provides a disjoint crack pattern, as shown in figure 1. Oliveira and Correia [18] label disjoint crack segments with the same identifier, but do not connect the segments. The closing operation offers a simple postprocessing step for connecting the disjoint cracks, but it connects noise close to the cracks as well. Hence, it is important to guide the direction of expansion along the crack structure. Some crack detection algorithms have developed customized morphological transforms to solve this problem [19], [20] but they only work for very small discontinuities. Tensor Voting (TV) has been very effective in joining disjoint crack segments and eliminating noise [21], [22], [23], [24]. However, TV has two shortcomings: it is computationally expensive and it results in a blurred out image of the cracks which has to be further processed to obtain the final crack pattern.

Minimal-path algorithms (MPA) attempt to detect cracks by searching for the optimal path across a potential map formed by the image, where a path along the crack is incentivized [25], [26], [27], [28]. Minimal-path based methods clearly provide the most accurate yet robust results. Unfortunately, MPAs require an enormous amount of computation. Attempts have been made to reduce the computation requirement [24] and to reduce the computation time using parallelization [29].

The objective of this paper was to develop a fast and accurate crack detection algorithm. To achieve this objective, a minimal path based approach is targeted because of the

[1] PhD Student, School of Computational Science and Engineering, Georgia Institute of Technology, North Ave, Atlanta, GA 30332 a.chatterjee@gatech.edu

[2] Professor, School of Civil and Environmental Engineering, Georgia Institute of Technology, North Ave, Atlanta, GA 30332 james.tsai@ce.gatech.edu

ISBN 978-90-827970-1-5 © EURASIP 2018

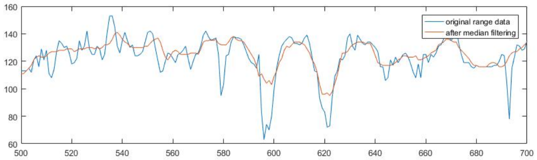


Fig. 2: Effect of preprocessing

high accuracy and robustness obtained. To address the issue of speed, a preliminary crack segmentation method using individual classification is proposed to limit the number of minimal path searches to only the crack pattern. To address the discontinuities in the crack pattern due to the individual classification approach, a novel bottom-up method for connecting the disjoint crack pattern has been developed. The connected preliminary crack pattern is then used to generate the inputs for a minimal path algorithm, which provides the final accurate crack segmentation. The Crack Detection Algorithm Performance Evaluation System (CDA-PES) has been used to evaluate and compare the proposed algorithm.

II. METHODOLOGY

The major steps of the proposed algorithm are as follows:

1) The image is preprocessed via median filtering to reduce noise.
2) A simplified individual classification algorithm is used to obtain a preliminary crack segmentation result.
3) Crack objects are generated and used to connect the disjoint crack segments and remove noise.
4) A minimal path based algorithm is used to detect the final crack pattern.

A. Preprocessing

The proposed algorithm was developed based on range images collected by a Pavemetrics LCMS laser scanner. Some pixels may contain out-of-range values because of the pavement surface being outside the measurement range of the sensor or the presence of obstructions between the sensor and the pavement surface. In either case, these out-of-range values produce a sharp gradient in the pavement image which can be mistaken for a crack. These pixels are however, almost always isolated and sparse. Therefore, a median blur is ideal for removing these outliers. A median blur with a window size of 9-by-9 pixels removes these outliers while preserving cracks. Figure 2 demonstrates the effect of the median blur.

B. Preliminary Crack Segmentation

A preliminary crack segmentation is carried out to remove the background. A simple thresholding is used to segment the crack pixels. It is difficult to determine an ideal global threshold that works for the entire image because of transverse undulations on the pavement surface caused by the camber and rutting. Hence, the image was divided into 20 × 20 subimages and an adaptive threshold was calculated for each subimage.

Let S_i be the set of pixels in subimage i. The threshold is determined using the pixel value distribution of the subimage i as follows:

1) The mean and minimum value of the subimage i is determined.

$$v_{mean}^{(i)} = \frac{\sum_{j}^{j \in S_i} v_j}{|S_i|} \tag{1}$$

$$v_{min}^{(i)} = min_{j \in S_i} v_j \tag{2}$$

Where,
v_j is the range value of pixel j,
$v_{mean}^{(i)}$ is the mean range value of subimage i and
$min_{j \in S_i} v_j$ is the minimum range value of subimage i

2) The threshold $t^{(i)}$ is then calculated using equation 3.

$$t^{(i)} = max(0, min(v_{min}^{(i)} + \alpha(v_{mean}^{(i)} - v_{min}^{(i)}) - \beta, \gamma)) \tag{3}$$

$$0 \leq \alpha \leq 1$$

$$\beta \geq 0$$

$$0 \leq \gamma \leq 255$$

Where α, β and γ are adjustable parameters. α controls the initial value of the threshold as a point between the mean and minimum range values of the subimage. For any non-zero value of α, there will be pixels which have a value below the threshold, even in subimages with no cracking. The pixel values of subimages with no cracking generally have less variance. The parameter β decreases the threshold slightly by a constant value. In case of subimages with no cracking, this slight shift pushes the threshold below the minimum value, preventing any pixels from being segmented as crack pixels. The parameter γ sets an upper limit for the threshold.

3) The preliminary crack segmentation map is generated. Pixels in S_i are classified as crack pixels if their range value i.e. depth is lower than $t^{(i)}$.

The parameters are adjusted to minimize the false negative error. This ensures that the thinner crack portions are not lost. This also results in a large number of false positives, which are removed in the next step. The optimal parameters obtained by trial-and-error are given in table I.

C. Crack Object Connection

A connected component is defined as a set of crack pixels such that every pixel in the set is connected to all other pixels through a path that passes exclusively through crack pixels through 8-neighbor connectivity. As shown in figure 3, connected components created from the true positive crack pixels (white) are generally more elongated and have an overall structure with a common local orientation. In contrast, connected components created from noise have no overall structure and no common orientation. These distinguishing properties are utilized to connect the crack features and remove the noise features.

ISBN 978-90-827970-1-5 © EURASIP 2018

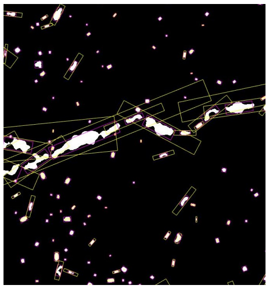


Fig. 3: Elongated crack objects overlay on preliminary image segmentation

Using the preliminary crack segmentation as the input, each connected component having at least 10 pixels is used to create a crack object. Let R_i be the set of crack pixels in the preliminary crack segmentation that belong to crack object i. Then crack object i has the following properties:

1) The centroid ($\mathbf{c} = (c_x, c_y)$) of the crack object is the centroid of the member crack pixels.
2) The length (l) is equal to the variance along the first principal component of the pixels in R_i.
3) The width (w) is equal to the variance along the second principal component of the pixels in R_i.
4) The eccentricity (e) is equal to $\sqrt{1 - w^2/l^2}$. Larger values of eccentricity correspond to a more elongated shape and smaller values denote a rounder shape.
5) The orientation (o) is given by the angle created by the principal component with the positive x-axis.

Figure 3 overlays the crack objects (magenta) over the preliminary crack segmentation (white). The crack objects are visualized as rectangles with the center, length, width and angle of the length with the positive x-axis given by c, l, w and o of the crack object respectively.

Each crack object is elongated using equations 4 and 5.

$$l' = \mu l e^{\nu} \tag{4}$$

$$w' = w e^{\nu} \tag{5}$$

Where l and l' are the initial and elongated lengths respectively, w and w' are the initial and elongated widths respectively and $\mu > 1$ and $\nu > 1$ are adjustable parameters. Equations 4 and 5 were designed to incentivize crack objects that are longer and penalize noise which generally has low eccentricity. Figure 3 overlays the crack objects before (magenta) and after (yellow) elongation on the preliminary crack segmentation (white). The false positive crack objects are mostly shrunk down while the true positive crack objects are extended towards each other. The values for the parameters were determined by trial-and-error. The finalized parameter values are given in table I. All values in table I were calibrated using the CDA-PES dataset.

For each crack object, a rectangle is drawn with the center of the rectangle coinciding with the centroid of the crack object. The length, width and the angle made by the length of the rectangle with the positive x-axis are given by l', w' and o respectively. Rectangles that overlap are considered connected. The isolated false positive connected components are much smaller than the true positive connected components which are connected together. This image of overlaid crack object rectangles is skeletonized and connected components of size smaller than the parameter δ are removed. This eliminates the false positive noise from the image, leaving only the interconnected crack pattern intact.

TABLE I: Parameter values

Parameter	Value
α	0.85
β	10
γ	65
δ	10
μ	2.7
ν	1.5

D. Minimal Path Crack Detection

From the previous subsection, a crack segmentation map is obtained with most false positives and false negatives removed and the disjoint crack segments patched together. Minimal Path detection using the fast-marching algorithm [30] was used to refine this image using the following steps.

1) The branchpoints of the image were identified. These branchpoints along with their neighboring pixels set to zero in the refined crack segmentation. As a result, the image would be left with curves with no branching.
2) The endpoints of the each curve are then identified. These endpoints serve as the input points for the fast-marching algorithm.
3) The fast-marching algorithm is used to find an optimal path between each pair of points using the original range image as the potential map. The minimal path falls along the crack pixels, which have greater depth than the surrounding pavement, hence lower weight associated with them.
4) Finally, the detected paths are filtered: paths smaller than 20mm or having a mean range value higher than the mean of the entire image are ignored. [24]
5) The removed branchpoints are added back, connecting the minimal path curves.

Figure 4 demonstrates each step of the proposed method.

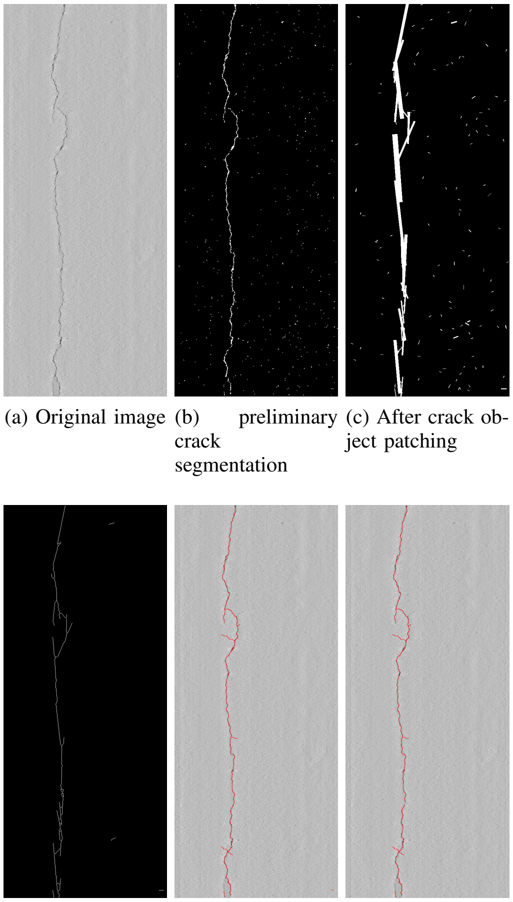
III. Results

The tests were conducted on a machine with an Intel Core i7-4770 CPU 8 cores @ 3.40GHz processor. The tests were run on a single CPU thread. The Crack Detection Algorithm - Performance Evaluation System (CDA-PES) [31] was used for evaluation.

A. Accuracy

Figure 5 shows the CDA-PES dashboard for the proposed algorithm. The overall score is 80, which is higher than previously tested algorithms [31]. The weakest category appeared to be transverse cracks, which are often difficult to detect because of their thin widths. Alligator cracking

ISBN 978-90-827970-1-5 © EURASIP 2018



(a) Original image (b) preliminary crack segmentation (c) After crack object patching

(d) Skeletonized image (e) After minimal path search (f) Final crack segmentation

Fig. 4: Demonstration of proposed algorithm on image with longitudinal crack

also pulled the overall score down. In both cases, the reason was mainly false negative error, propagated by thin cracks. These categories show the area for future improvement for the proposed algorithm. The performance is not significantly affected by crack width although it appears to clearly decline as crack complexity increases.

B. Computation Speed

The greatest advantage of the proposed algorithm with respect to the state-of-the-art algorithm is a vast reduction in computation time. The computation speed for processing the CDA-PES pavement image dataset has been considered. The dataset consists of 68 images of size 0.65 megapixels. The proposed algorithm took on average 1.15 seconds per image. The distribution of processing times (figure 6) shows that this is a skewed distribution. The median computation time is 0.52 seconds. The minimal path search step was the most computationally heavy step in the algorithm for images with cracks.

Figure 7 demonstrates the speed improvement over the TV algorithm. The proposed algorithm has a 36 times faster average computation speed and a 58 times faster median computation speed. When compared on the same image, the proposed algorithm performed 68, 32 and 27 times faster in an image with no cracking, one longitudinal crack and alligator cracking respectively.

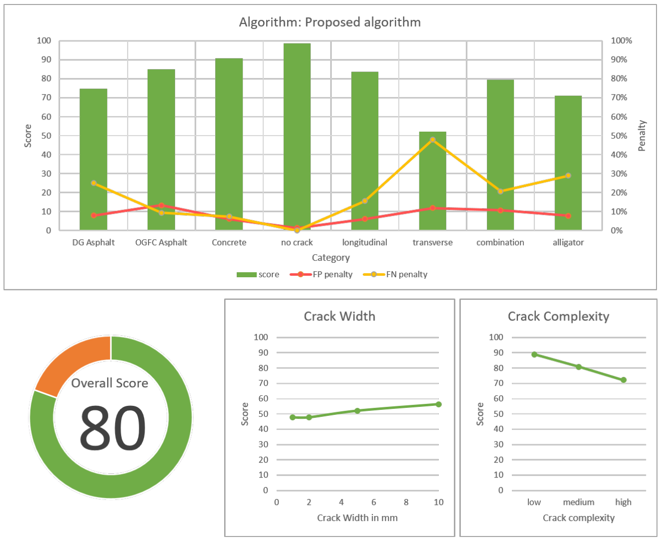


Fig. 5: Dashboard for proposed algorithm

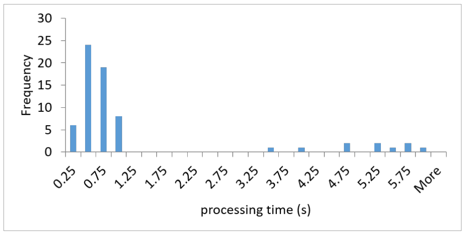


Fig. 6: Distribution of computation time of proposed algorithm

IV. Conclusion

This paper presented a fast and accurate crack detection algorithm. The core contribution to crack detection research is a novel preliminary crack segmentation method and a fast method for connecting the disjoint crack pattern as an alternative to tensor voting. These contributions provide a method to automatically obtain the input points for the fast-marching algorithm, which has been found to be very accurate in finding the crack path between between two given points on a crack. The algorithm presented in this paper offers a robust, accurate and fast crack detection algorithm which will have a strong beneficial impact on the widespread adoption of automated road condition surveys, making road condition surveys safer and more efficient.

The following future recommendations are made:

1) Although the proposed algorithm has a good overall performance, the CDA-PES analysis reveals that alligator cracking and transverse cracking are the most difficult cases for the proposed algorithm. Further research is recommended to further improve the performance in these categories.

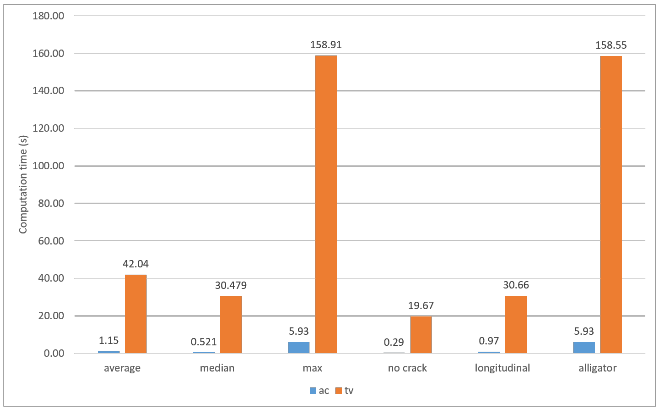


Fig. 7: Comparison of computation time

2) Parallel and GPU processing is recommended to further improve the speed of the algorithm.
3) The application of the proposed crack detection algorithm in other fields can be explored. Crack detection has applications in bridge inspection, building information modeling, manufacturing and medical imaging.
4) The crack objects already store useful information about the crack pattern. This crack pattern information can have applications in crack classification. The possibility of using the crack object data for crack classification should be explored.

REFERENCES

[1] Y.-C. J. Tsai, A. Chatterjee, and C. Jiang, “Challenges and lessons from the successful implementation of automated road condition surveys on a large highway system,” in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 2031–2035.

[2] GDOT, “Pavement condition evaluation system,” Georgia Department of Transportation, Tech. Rep., 2007.

[3] H. N. Koutsopoulos, I. E. Sanhouri, and A. B. Downey, “Analysis of segmentation algorithms for pavement distress images,” *Journal of transportation engineering*, vol. 119, no. 6, pp. 868–888, 1993.

[4] Y. Huang and B. Xu, “Automatic inspection of pavement cracking distress,” *Journal of Electronic Imaging*, vol. 15, no. 1, pp. 013 017–013 017, 2006.

[5] B. J. Lee, H. Lee *et al.*, “Position-invariant neural network for digital pavement crack analysis,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 19, no. 2, pp. 105–118, 2004.

[6] F. Roli, “Measure of texture anisotropy for crack detection on textured surfaces,” *Electronics Letters*, vol. 32, no. 14, pp. 1274–1275, 1996.

[7] T. S. Nguyen, S. Begot, F. Duculty, and M. Avila, “Free-form anisotropy: A new method for crack detection on pavement surface images,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 1069–1072.

[8] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[9] H. Zhao, G. Qin, and X. Wang, “Improvement of canny algorithm based on pavement edge detection,” in *Image and Signal Processing (CISP), 2010 3rd International Congress on*, vol. 2. IEEE, 2010, pp. 964–967.

[10] A. Cubero-Fernandez, F. J. Rodriguez-Lozano, R. Villatoro, J. Olivares, and J. M. Palomares, “Efficient pavement crack detection and classification,” *EURASIP Journal on Image and Video Processing*, vol. 2017, no. 1, p. 39, 2017.

[11] S. Mallat and S. Zhong, “Characterization of signals from multiscale edges,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 7, pp. 710–732, 1992.

[12] A. Cuhadar, K. Shalaby, and S. Tasdoken, “Automatic segmentation of pavement condition data using wavelet transform,” in *Electrical and Computer Engineering, 2002. IEEE CCECE 2002. Canadian Conference on*, vol. 2. IEEE, 2002, pp. 1009–1014.

[13] J. Zhou, P. Huang, and F.-P. Chiang, “Wavelet-based pavement distress classification,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1940, pp. 89–98, 2005.

[14] J. Zhou, P. S. Huang, and F.-P. Chiang, “Wavelet-based pavement distress detection and evaluation,” *Optical Engineering*, vol. 45, no. 2, pp. 027 007–027 007, 2006.

[15] H. Cheng, J. Wang, Y. Hu, C. Glazier, X. Shi, and X. Chen, “Novel approach to pavement cracking detection based on neural network,” *Transportation Research Record: Journal of the Transportation Research Board*, no. 1764, pp. 119–127, 2001.

[16] H.-D. Cheng, “Automated real-time pavement distress detection using fuzzy logic and neural network,” in *Nondestructive Evaluation Techniques for Aging Infrastructure and Manufacturing*. International Society for Optics and Photonics, 1996, pp. 140–151.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *arXiv preprint arXiv:1511.00561*, 2015.

[18] H. Oliveira and P. L. Correia, “Automatic crack detection on road imagery using anisotropic diffusion and region linkage,” in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 274–278.

[19] D. Zhang, Q. Li, Y. Chen, M. Cao, L. He, and B. Zhang, “An efficient and reliable coarse-to-fine approach for asphalt pavement crack detection,” *Image and Vision Computing*, vol. 57, pp. 130–146, 2017.

[20] A. Zhang, K. C. Wang, R. Ji, and Q. J. Li, “Efficient system of cracking-detection algorithms with 1-mm 3d-surface models and performance measures,” *Journal of Computing in Civil Engineering*, vol. 30, no. 6, p. 04016020, 2016.

[21] Q. Zou, Y. Cao, Q. Li, Q. Mao, and S. Wang, “Cracktree: Automatic crack detection from pavement images,” *Pattern Recognition Letters*, vol. 33, no. 3, pp. 227–238, 2012.

[22] J. Huang, W. Liu, and X. Sun, “A pavement crack detection method combining 2d with 3d information based on dempster-shafer theory,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 29, no. 4, pp. 299–313, 2014.

[23] H. Guan, J. Li, Y. Yu, M. Chapman, H. Wang, C. Wang, and R. Zhai, “Iterative tensor voting for pavement crack extraction using mobile laser scanning data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1527–1537, 2015.

[24] C. Jiang, “A crack detection and diagnosis methodology for automated pavement condition evaluation,” Ph.D. dissertation, Georgia Institute of Technology, 2015.

[25] M. Avila, S. Begot, F. Duculty, and T. S. Nguyen, “2d image based road pavement crack detection by calculating minimal paths and dynamic programming,” in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 783–787.

[26] R. Amhaz, S. Chambon, J. Idier, and V. Baltazart, “Automatic crack detection on two-dimensional pavement images: An algorithm based on minimal path selection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2718–2729, 2016.

[27] Q. Li, Q. Zou, D. Zhang, and Q. Mao, “Fosa: F* seed-growing approach for crack-line detection from pavement images,” *Image and Vision Computing*, vol. 29, no. 12, pp. 861–872, 2011.

[28] N. Strisciuglio, G. Azzopardi, and N. Petkov, “Detection of curved lines with b-cosfire filters: A case study on crack delineation,” in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2017, pp. 108–120.

[29] V. Baltazart, P. Nicolle, and L. Yang, “Ongoing tests and improvements of the mps algorithm for the automatic crack detection within grey level pavement images,” in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 2016–2020.

[30]

[31] Y.-C. Tsai and A. Chatterjee, “Comprehensive, quantitative crack detection algorithm performance evaluation system,” *Journal of Computing in Civil Engineering*, vol. 31, no. 5, p. 04017047, 2017.