A NOVEL SYSTEM OF PAVEMENT CRACKING DETECTION ALGORITHMS USING 1MM 3D SURFACE DATA

By

AONAN ZHANG

Bachelor of Civil Engineering
Southwest Jiaotong University
Chengdu, China
2008

Master of Civil Engineering
Southwest Jiaotong University
Chengdu, China
2010

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF PHILOSOPHY
December, 2015

A NOVEL SYSTEM OF PAVEMENT CRACKING DETECTION ALGORITHMS USING 1MM 3D SURFACE DATA

Dissertation Approved:

Dissertation Adviser Dr. Kelvin C. P. Wang

Committee Member Dr. Stephen A. Cross

Committee Member Dr. Xiaoming Yang

Outside Committee Member Dr. Gary G. Yen

ACKNOWLEDGEMENTS

Firstly, I am very grateful to my advisor, Dr. Kelvin C.P. Wang, for his valuable guidance and long-lasting encouragements on the completion of this dissertation. Dr. Wang offers me great opportunities to learn advanced computer programming technology and numerous techniques as well as methodologies related with Automated Pavement Survey. Without such an experience, this dissertation could not be completed. Dr. Wang's rigorous attitude and unlimited passion on researches have been and will be motivating me towards future achievements in my whole career.

I would like to thank Dr. Gary Yen, Dr. Stephen Cross and Dr. Xiaoming Yang for being my committee members and giving me valuable instructions and suggestions. I would like to thank Dr. Gary Yen and Dr. Stephen Cross again for inspiring me with the systematic knowledge lectured at their classes and encouraging me to study independently to explore solutions for the class assignments.

I would like to thank my wife Lanlin Mao for completely understanding and supporting me in every bit of my life. I appreciate the helps and love from her that accompanied me to go through the tough time.

I sincerely thank Weiguo Gong for his substantial efforts to train me as a computer programmer. I gained much knowledge on computer programming following his instructions.

I also thank Dr. Joshua Qiang Li, Dr. Cheng Chen and Justin Thweatt for their helps and advices in the past four years.

My thanks also go to my lab mates: Shi Qiu, Wenjuan Wang, Guangwei Yang, Ran Ji, You Zhan and Yue Fei. They gave me many inspirations via discussions and presentations.

Last but definitely not the least, I am forever grateful to my parents, relatives, teachers and friends in China. Their loves and helps are the great source of enlightenments for me.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: AONAN ZHANG

Date of Degree: DECEMBER, 2015

Title of Study: A NOVEL SYSTEM OF PAVEMENT CRACKING DETECTION ALGORITHMS USING 1MM 3D SURFACE DATA

Major Field: Civil Engineering

ABSTRACT: Pavement cracking is one of the major concerns for pavement design and management. There have been rapid developments of automated pavement cracking detection in recent years. However, none of them has been widely accepted so far due to lack of capability of maintaining consistently high detection accuracy for various pavement surfaces. Using 1mm 3D data collected by WayLink Digital Highway Data Vehicle (DHDV), an entire system of algorithms, which consists of Fully Automated Cracking Detection Subsystem, Interactive Cracking Detection Subsystem and Noisy Pattern Detection Subsystem, is proposed in this study for improvements in adaptability, reliability and interactivity of pavement cracking detection.

The Fully Automated Cracking Detection Subsystem utilizes 3D Shadow Simulation to find lower areas in local neighborhood, and then eliminates noises by subsequent noise suppressing procedures. The assumption behind 3D Shadow Simulation is that local lower areas will be shadowed under light with a certain projection angle. According to the Precision-Recall Analysis on two real pavement segments, the fully automated subsystem can achieve a high level of Precision and Recall on both pavement segments.

The Interactive Cracking Detection Subsystem implements an interactive algorithm proposed in this study, which is capable of improving its detection accuracy by adjustments based on the operator’s feedback, to provide a slower but more flexible as well as confident approach to pavement cracking detection. It is demonstrated in the case study that the interactive subsystem can retrieve almost 100 percent of cracks with nearly no noises.

The Noisy Pattern Detection Subsystem is proposed to exclude pavement joints and grooves from cracking detection so that false-positive errors on rigid pavements can be reduced significantly. This subsystem applies Support Vector Machines (SVM) to train the classifiers for the recognition of transverse groove, transverse joint, longitudinal groove and longitudinal joint respectively. Based on the trained classifiers, pattern extraction procedures are developed to find the exact locations of pavement joints and grooves.

Non-dominated Sorting Genetic Algorithm II (NSGA-II), which is one of multi-objective genetic algorithms, is employed in this study to optimize parameters of the fully automated subsystem for the pursuing of high Precision and high Recall simultaneously. In addition to NSGA-II, an Auxiliary Prediction Model (APM) is proposed in this study to assist NSGA-II for faster convergence and better diversity.

Finally, CPU-based and GPU-based Parallel Computing Techniques, including Multi-GPU, GPU streaming, Multi-Core and Multi-Threading are combined in this study to increase the processing speed for all computational tasks that can be synchronous.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 INTRODUCTION

1.1 Background

Transportation system is of vital importance to the modern society. Over 3.9 million miles of roadway, which consists of the national highway infrastructure network of the United States, is an essential and extremely important component of the transportation system (Banks, 2002). Substantial funds are invested to numerous highway rehabilitation and maintenance projects in order to ensure the pavement serviceability in each year. Pavement preservation is essentially recognized as a data-driven process where pavements data play an obligatory role in optimizing the strategies and prioritizing resource allocation (Paterson and Scullion, 1990).

In general, pavement data could include pavement type, materials, geometries, layer configurations, pavement textures, surface roughness, and pavement distresses and so on. The status of pavements has explicit or implicit impacts on ride quality, deterioration rate, pavement remaining life, safety and other functions. In pavement engineering, these data sets have been utilized with other external information in situ such as climate records and traffic data almost throughout the entire life cycle of pavements: design, construction, maintenance, rehabilitation and management (Haas and Hudson, 1978; Huang, 1993; TRB, 2001; Wang, 2011).

Some of these data are static and retrievable in the inventory database where the information was kept when the pavements were initially constructed or overlaid; they are commonly called as-built information. The remainders, which are pavement performance data, are dynamic and need to be updated from the field on a regular basis to ensure timely decisions. Due to the massive mileages of the pavement system, the task to update the state of pavement performance has always been very challenging. Manual survey had been the dominant approach to collect pavement evaluation data for decades until the widespread application of modern sensors and advancements of computer science in the 1980s. Consequently, the automated survey systems started to be recognized in the industry (Haas and Hendrickson, 1990; Paterson, 1994; Cheng and Miyojim, 1997; Wang, 2000; Fu et al., 2011). Comparing to manual surveys, automated surveys are less labor-intensive, safer, more objective, more efficient and more repeatable. However, only 30 states in the US had been collecting pavement surface images for data extraction by 2004 (Mcghee, 2004). Even today, few agencies are dependent on automated systems of distress surveys for production level work. Instead, a number of agencies rely on semi-automated approach with human intervention because of low repeatability and reliability of the automated systems. Regardless of which method to be used for data collection, the real concern of highway agencies is whether the survey can generate accurate and precise results in a consistent manner.

For manual survey, trained engineers or operators are to rate the physical pavement features of interest based on requirements of the protocols. However, subjective tendency is a concern: it is challenging to retain the data consistency of the evaluation due to the long-duration of work and the disagreement between raters. As a result, errors are prone to occur during the data processing phase (Morian et al., 2002; Wang et al., 2011). In contrast, the automated systems have the advantage to eliminate inconsistence and human-errors in the data processing phase as long as the design and integration of the data collection system are fully validated.

1.2 Problem Statement

The advanced modern sensors and laser technology have made the automated high quality data collection no longer challenging. The remaining critical problem is the automated data processing systems that can produce highly reliable detection, classification, evaluation and reporting for decision makings in pavement engineering. However, it is extremely difficult to develop software solutions which could automatically produce highly reliable results even with decades' endeavors on worldwide basis. The difficulties lie in two aspects. First of all, as a matter of fact, pavement surface distresses have numerous and unpredictable presences on diversely textured pavement surfaces. It is challenging to fully recognize them even with human interaction. Secondly, there is no protocol for distress classification and measurements that has been widely adopted by agencies for network level surveys. Therefore, in the automated recognition, the output quality is highly dependent on the protocols used, which indicates the selection of a proper protocol could be a key issue (Wang et al. 2011). Past experiences manifest that the emphasis to improve the reliability of automated recognition system is to make the protocols easier and more practical to be followed by computer algorithms. As an alternative solution, some pavement engineers have customized protocols for image-based automated recognition systems but, unfortunately, none of them has been widely adopted due to various reasons (Paterson, 1994; Wang and Elliot, 1999; Wang et al., 2000; Raman et al., 2004; Fu et al., 2011).

Pavement cracking is the fracture appearing on the pavement surface due to material fatigue, repeated loading, climatic deteriorations or other factors. It is one of the most significant pavement distresses that have direct impacts on pavement safety and ride quality, as well as a major indicator in diagnosing deterioration for all types of pavements (Paterson, 1994; Lee and Kim, 2005). Accurate cracking survey is necessary as taking proper and timely countermeasures could prevent cracking from further deterioration. However, cracking is the most difficult type of data to be detected automatically, compared to other data such as rut depth, ride quality, texture and global positioning (Timm and McQueen, 2004). For the past two decades, numerous algorithms have been developed for automated cracking detection. However, the current automated cracking detection algorithms have various limitations, and none of the algorithms has been widely accepted (Wang et al., 2007; Tsai, 2010; Zhang, 2013). Particularly, no current automated detection algorithms can maintain consistently high detection accuracy for various roads due to the diversity and complexity of pavement surfaces. Meanwhile, researchers have not paid sufficient attention to data operator's involvement in processing. In other words, the current automated algorithms have not provided a systematic methodology for users to improve the detection performance based on their experience and observations. As a result, few studies have shown a system of algorithms that can allow operators to improve the detection accuracy and reduce the noise percentage until they are satisfied.

1.3 Objectives

The objective of this study is to propose an efficient and practical computer-aided system with multiple algorithms that can implement pavement cracking detection at different automation levels based on the users' acceptance and tolerance level. The objective of the Fully Automated Cracking Detection Subsystem is to develop algorithms that are capable of detecting cracks with high accuracy for many cases without human's intervention. In order to provide an interface for the operators to conduct interactive detection based on their acceptance and tolerance level, the objective of the Interactive Cracking Detection Subsystem is to utilize computer-aided interaction between the operators and interactive algorithms for improvements in detection. Despite the interactive detection may consume a long time, it is anticipated that the Interactive Cracking Detection Subsystem can fully satisfy the users' requirements on detection accuracy. Finally, based on the classifiers trained by Support Vector Machines (SVM), the Noisy Pattern Detection Subsystem is to detect pavement joints and grooves, which are the two major noisy patterns on rigid pavements that can seriously affect the accuracy of cracking detection. Once the pavement joints and grooves are detected, they will be excluded from the cracking detection results. The exclusion of pavement joints and grooves will largely reduce false-positive errors of cracking detection on rigid pavements. Thus, the automation levels and objectives of the three detection subsystems can be concluded in Table 1.1.

Table 1.1. Automation levels and objectives of the three proposed subsystems

Subsystems		Automation Level	Objectives
Fully Automated Cracking Detection subsystem		Fully Automated	Detect the majority of cracking without operator’s intervention
Interactive Cracking Detection subsystem	Bottom Level	Automated but requiring training	Detect the majority of cracking after operator’s training on selected samples
	Top Level	Semi-automated	Find missed cracks or delete noises based on operator’s acceptance level
Noisy Pattern Detection Subsystem		Fully Automated	Detect pavement joints and grooves, and exclude the detected noisy patterns from cracking detection

Unlike the Interactive Cracking Detection Subsystem, the Fully Automated Cracking Detecting Subsystem has no feedback as reference during its implementation. It is therefore necessary to optimize the parameters of the fully automated system so that high detection accuracy can be achieved for many cases. Based on representative cracked pavement data, the integration of Non-dominated Sorting Genetic Algorithm II (NSGA II) and Auxiliary Prediction Model (APM) is utilized to optimize the parameters of the fully automated subsystem for best compromise between the two conflicting optimization goals: high Precision and high Recall. Eventually, combinations of CPU-based and GPU-based Parallel Computing Techniques are implemented to greatly increase the speed of all computational tasks involved in detection algorithms that can be synchronous.

1.4 Organization of Dissertation

This dissertation is organized as following:

Chapter 2 provides literature review on pavement cracking detection algorithms, pavement joint and groove detection algorithms and Multi-objective Genetic Algorithms.

Chapter 3 describes the Fully Automated Cracking Detection Subsystem based on 3D Shadow Simulation and Noise Suppressing Algorithms.

Chapter 4 utilizes NSGA-II to optimize the primary parameters involved in Fully Automated Cracking Detection Subsystem. In addition to NSGA-II, the APM is proposed in this chapter to assist NSGA-II for faster convergence and better diversity.

Chapter 5 develops the Interactive Cracking Detection Subsystem. The interactive algorithms are proposed to improve the detection performance according to the operator's feedback. The two-level detection is developed to implement Interactive Detection at different automation levels.

Chapter 6 proposes the Noisy Pattern Detection Subsystem for the detection of pavement joints and grooves. This subsystem applies SVM algorithms to optimize the four classifiers for the recognition of transverse groove, transverse joint, longitudinal groove and longitudinal joint respectively via the supervised learning on numerous training samples. Based on the four SVM classifiers, pattern extraction procedures are proposed to find the locations of pavement joints and grooves.

Chapter 7 adopts combinations of CPU-based and GPU-based Parallel Computing Technology to increase the speed of all computational tasks involved in the detection algorithms that can be synchronous.

Chapter 8 lists the conclusions and recommendations for future work.

Chapter 2 LITERATURE REVIEW

2.1 Pavement Cracking Recognition

Over the recent decades, many algorithms have been developed to advance the automated pavement cracking recognition. Multilayer feed-forward neural-network classifier and a two-stage piecewise linear neural network classifier were proposed by Kaseko et al. to perform automated detection, classification and quantification of cracks on video images (1994). Skeleton Analysis, Real-time Thresholding, Fuzzy Set theory and Neural Network techniques were adopted by Cheng et al. (1998; 1999; 2001; 2003) for pavement cracking recognition. Huang and Xu utilized image processing algorithms to seek for cracks starting from divided image cells to clusters of linear pattern based on grayscale information and morphological presences (2006). Wang et al. (2007) applied wavelet edge detection based on à trous algorithm to detect pavement distress with the consideration of multi-scale. Nguyen et al. utilized Conditional Texture Anisotropy (CTA) to detect pavement defects. Based on the CTA results, cracks, joints and bridged defects were classified using Multi-layer Perceptron Neural Network (MLPNN) (2009). Tsai et al. evaluated the performances of six existing image segmentation algorithms for pavement cracking detection using a scoring measure. It was demonstrated that the dynamic optimization-based method which combines global and local information generated good results (2010).

Cord and Chambon employed a supervised learning method based on Adaptive Boosting (AdaBoost) for Texture Pattern Recognition, with the purpose of distinguishing roads with defects from roads without defects (2010). Beamlet transform-based approach was proposed by Ying and Salari to detect and classify cracks in partitioned small image windows (2010). Nejad and Zakeri incorporated a radon neural network into wavelet modulus to build a multi-resolution neural network for pavement cracking classification (2011). An adaptive road crack detection system was developed by Gavilan et al. (2011), for which linear SVM-based classifier is used to distinguish pavement types and then the output is applied to tune parameters for cracking detection. There are also other researches that implement SVM-based classifiers for pavement cracking detection (Li et al., 2009; Moussa and Hussain, 2011; Daniel et al., 2014). Mathavan et al. (2012) integrated Texture Analysis and Kohonen Feature Map for segmentation of pavement images which can provide help to pavement cracking detection. Nishikawa et al. presented cracking detection algorithms based on Genetic Programming for concrete surface. The sequence and combination of several image filters were evolved following genetic processes in order to search the best multi-sequential image filters. Re-extraction of missed cracks and elimination of residual noises were conducted recursively at local regions using the best multi-sequential image filters until specific criteria are satisfied (2012). Zhang et al. introduced Matched Filtering algorithm to pavement cracking detection using predesigned rotated filters to match crack features by shape, orientation and intensity (2013). Lee et al. developed algorithms to detect cracks from concrete surface. Global binarization and local binarization based on Shape Analysis were both considered for the enhancement of detection performance. Based on the detection results, crack width, length and orientation were computed by specific algorithms. Afterwards, a pattern recognition approach based on Neural Network was applied finally to classify transverse cracks, longitudinal cracks and diagonal cracks. (2013). Gabor Filters were employed by Zalama et al. to extract visual features from pavement images for detection and classification of longitudinal and transverse cracks (2014). Li et al. adopted Back Propagation Neural Network

(BPNN) to classify linear cracks and alligator cracks based on the result processed by a framework of image processing algorithms. Additionally, the linear cracks were further categorized into longitudinal cracks and transverse cracks according to the orientations (2014).

The vast majority of published developments on automated cracking surveys are based on 2D pavement surface visual data. In the last few years, 3D laser imaging technology has become the most powerful and sophisticated data acquisition platform for pavement surface data. 3D pavement elevation data is nearly invulnerable to ambient lighting conditions and noisy patterns, such as lane markings, tire marks and oil-spills, in terms of cracking detection, whereas visual data from 2D systems contain these various noises and unwanted information which must be processed separately, sometime with only limited successes. A recent study demonstrated that 3D pavement data can produce consistent results when detecting cracks even under different lighting conditions (Tsai and Li, 2012). Peng et al. employed Multi-seeding Fusion algorithm for automated cracking detection on 1mm resolution 3D data (2014). Most recently, Huang et al. combined 2D image and 3D information based on Dempster-Shafer theory for pavement cracking detection. It was concluded that the combination of 2D and 3D data sets as input data achieved higher accuracy and lower recognition error rate than using 2D images only (2014).

2.2 Multi-objective Optimization

With respect to performance optimization of the Fully Automated Cracking Detection Subsystem, the percentage of pixels are detected correctly and the percentage of pixels are detected incorrectly are essentially the two primary objectives that need to be evaluated during the optimization. Thus, the optimization of the Fully Automated Cracking Detection Subsystem eventually evolves to a multi-objective optimization problem.

For a multi-objective optimization problem, the optimization objectives are generally conflicting with each other. Therefore, there does not exist a single solution that simultaneously optimizes each objective. In contrary, there exist a number of Pareto-optimal solutions that are non-dominated with each other in terms of the objectives. Multi-objective Evolutionary Algorithms (MOEAs) are popular approaches to producing Pareto-optimal solutions for a multi-objective optimization problem. Many of current MOEAs belong to the class of Multi-objective Genetic Algorithms, which simulate the process of natural selection and mainly consist of the four components: population initialization, fitness assignment, recombination (crossover and mutation) and environmental selection. Konak et al. overviewed and summarized most of the well-known and credible Multi-objective Genetic Algorithms as listed below (2006):

- Vector Evaluated GA (VEGA);
- Multi-objective Genetic Algorithm (MOGA);
- Niched Pareto Genetic Algorithm (NPGA);
- Weight-based Genetic Algorithm (WBGA);
- Random Weighted Genetic Algorithm (RWGA);
- Non-dominated Sorting Genetic Algorithm (NSGA);
- Strength Pareto Evolutionary Algorithm (SPEA);
- Improved SPEA (SPEA2);
- Pareto-Archived Evolution Strategy (PAES);
- Pareto Envelope-based Selection Algorithm (PESA);
- Region-based Selection in Evolutionary Multi-objective Optimization (PESA-II);
- Fast Non-dominated Sorting Genetic Algorithm (NSGA-II);
- Multi-objective Evolutionary Algorithm (MEA);
- Micro-GA;
- Rank-Density Based Genetic Algorithm (RDGA);
- Dynamic Multi-objective Evolutionary Algorithm (DMOEA);

Konak et al. fully discussed the advantages and disadvantages of each aforementioned algorithm. Among these algorithms, NSGA-II (Deb et al., 2002), SPEA (Zitzler and Thiele, 1998), SPEA2 (Zitzler et al., 2001), RDGA (Lu and Yen, 2003) and DMOEA (Yen and Lu, 2003) perform better than others in terms of elitism and diversity preservation mechanism. But NSGA-II is comparatively simpler to be implemented than SPEA, SPEA2, RDGA and DMOEA. Comparisons of Multi-objective Genetic Algorithms are also found in other literatures (Zitzler and Thiele, 1999; Zitzler et al., 2000; Deb, 2001; Coello et al., 2007). However, numerous comparison studies manifest an overall better behavior of NSGA-II and SPEA2.

2.3. Noisy Pattern Detection

Pavement joints and grooves are the two major noisy patterns on rigid pavements in terms of cracking detection. They must be excluded from cracking detection, otherwise countless false-positive errors will be introduced. There are limited researches on automated detection of pavement grooves and joints. With respect to automated pavement groove identification, Wang and Davis proposed an automated groove identification program to evaluate the configurations of transverse grooves on airport pavements (2013). Sequentially, Beam-bridging filter was also introduced by Wang and Hayhoe to detect transverse grooves on runways (2013). However, these groove identification algorithms are limited to transverse grooves on airport pavements. Later on, Wang et al. suggested a template matched algorithm assisted by moving average filter for identification of longitudinal pavement grooves with 1mm resolution 3D data (2014). But these procedures are mainly for a general measurement on the longitudinal grooves by analyzing some sampled transverse profiles. Such a measurement is different from the detection for which the locations of all grooves need to be found.

On the other hand, an automated pavement joint detection algorithm was developed by Nguyen et al. based on Conditional Texture Anisotropy (CTA). In order to separate joints from cracks detected by CTA, Multi-layer Perceptron Neural Network (MLPNN) is suggested by Nguyen et al. to classify three types of detected defects: Joints, Cracks and Bridged Defects (2009). Although 100% accuracy on joint classification was reported in their research, the CTA is only conducted on 2D pavement images, and its performance on 3D pavement images is still unknown. More importantly, pavement groove, which is another major noisy pattern on rigid pavements and may introduce false-positive errors more than pavement joint, however is not considered in their research. Gavilan et al. proposed a combination of Canny Edge Detector and Hough Transform to detect pavement joints on 2D images. Firstly, the Canny Edge Detector is operated to locate dark edges. Then, Hough Transform is applied to extract straight lines from the edge detection results. It was reported that such a procedure yielded an overall Precision 90.34% and Recall 84.17% on 278 joint samples (2011). This procedure states a clear route to pavement joint detection. But it is invented for joint detection on 2D pavement images, and its performance on 3D pavement images has not been examined. Moreover, the Canny Edge Detector is merely a primitive media in detection, and its performance is very limited for object detection on pavements (Zhang et al., 2013). Furthermore, the reported Recall on 278 samples seems unsatisfactory, and improvements on Recall should be desired. No other automated approaches to separating pavement joints from pavement cracks have been found, although detection of joint faulting and other joint deteriorations has been discussed (Tsai, 2012; Hoegh, 2013; Wang, 2014).

2.4 Summary

In a summary, there are tremendous researches on pavement cracking detections. However, none of the researches has been generally recognized and broadly applied. For one reason, the pavement data involved in the researches follow various formats and standards as they are collected by different vendors. Due to the proprietary issues, no public benchmark data are available so far for commonly wared evaluations on the existing algorithms. For another reason, no researches have demonstrated consistently high detection accuracies for various roads even with their own data. There is no evidence showing that a flexible level of detection accuracy based on the user’s preference can be guaranteed.

Secondly, although there are several researches on the identification of pavement grooves and joints, they are restricted at different aspects and mainly based upon 2D images or airport pavements. Particularly, no literatures show a comprehensive system of algorithms that can detect transverse grooves, transverse joints, longitudinal grooves and longitudinal joints simultaneously but with clearly separated classifications. Therefore, one objective of this study is to develop an automated system of algorithms that can achieve these features from a comprehensive perspective.

Lastly, there are systematic developments on Multi-objective Optimization algorithms. Many of them are intensively tested and proved to be efficient for varied cases. Therefore, the optimization of the Fully Automated Cracking Detection Subsystem just needs to pick up one of the efficient Multi-objective Optimization algorithms.

Chapter 3 FULLY AUTOMATED CRACKING DETECTION SUBSYSTEM

3.1 Introduction to Data Collection System

All 3D data used in this study are collected by Digital Highway Data Vehicle (DHDV) developed by WayLink Systems Corporation. With twenty years' development, DHDV has been evolved into a sophisticated system that can collect full-lane data at a highway speed up to 60MPH (about 100 km/h). The vertical resolution of 3D data collected by DHDV is 0.3 mm, while the resolution in both longitudinal and transverse directions is 1mm. Figure 3.1 shows the exterior and interior appearance of DHDV. A representative frame of 3D data collected by DHDV at 60MPH is illustrated in Figure 3.2.

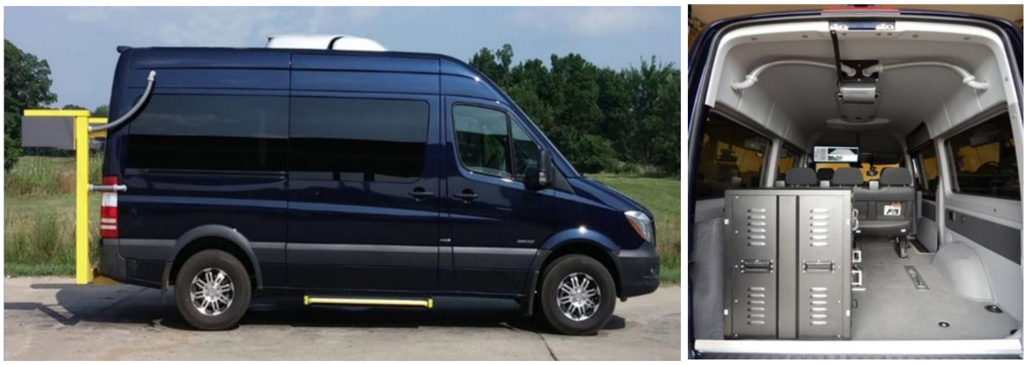


Figure 3.1. WayLink DHDV

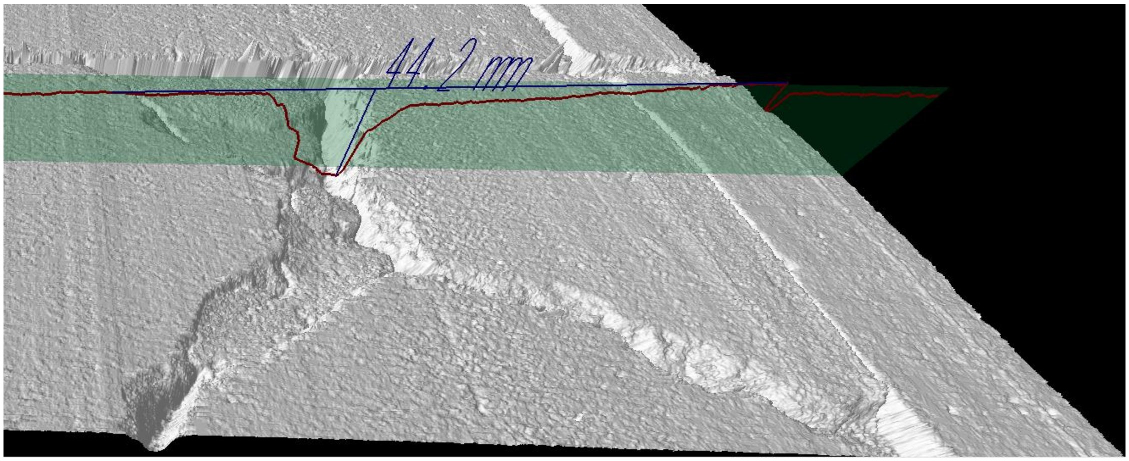


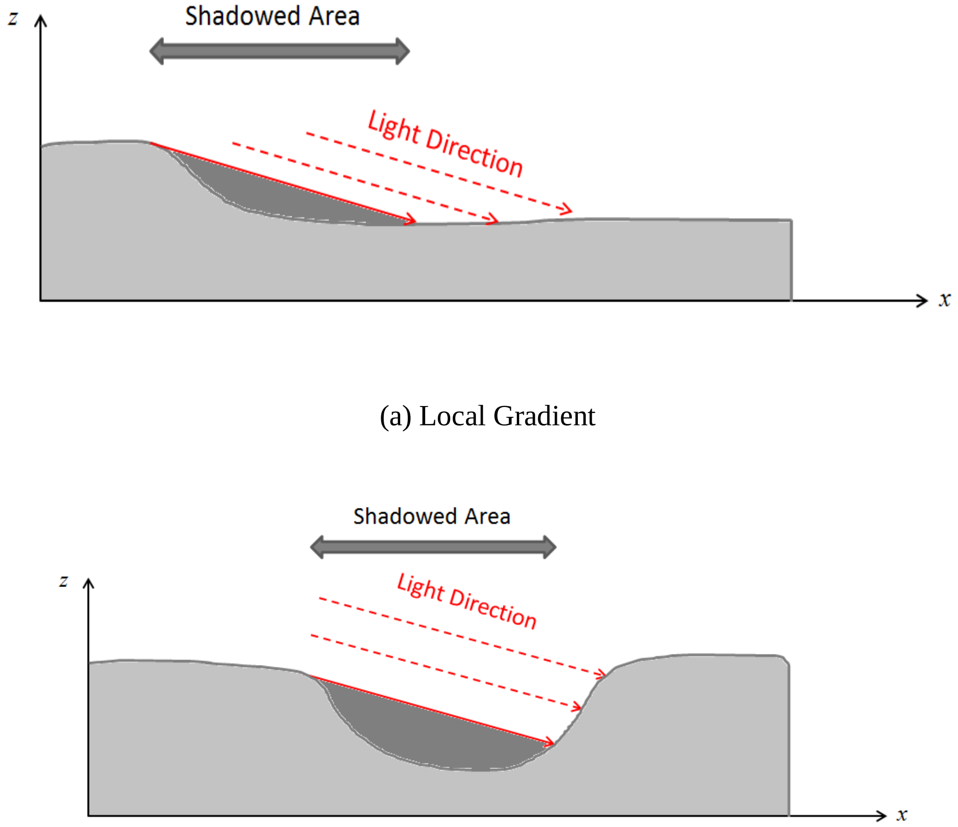
Figure 3.2. Representative 3D Pavement Surface at 60MPH

3.2 Fully Automated Cracking Detection Algorithms

3.2.1 3D Shadow Simulation

Over decades, a common approach to automated cracking detection is essentially the use of various filters and pre-processing algorithms in order to transform the original data into a comparatively simpler domain. In particular, the original data will be transformed into a binary domain, where suspected pixels are labeled as “0” and the others are labeled as “1”. As a result, the failure in detecting true cracks occurs when the original information is lost during the transformation. On the other hand, noises are introduced when the unwanted information is maintained during the transformation. The capability of a transformation algorithm lies in not only the correctness of transforming the original data, but also the degree that how much useful information is able to be extracted. In this study, a natural idea is proposed to shed light on the following two perspectives: 1) as the crack is a pattern that has lower height than the local neighborhood, the transformation should be able to capture local lower areas correctly; 2) the degree of extraction can be diverse and be controlled easily by setting certain constraints.

3D Shadow Simulation is proposed in this study to extract local lower areas by modeling the projection of natural sunlight. It is assumed that the sunlight comes from an infinitely long distance so that the sunlight beams are parallel everywhere. Meanwhile, no diffusions and reflections are considered in the modeling. Based on the assumption, local gradient and local dip are the two types of spatial occurrences that can be shadowed, as illustrated in Figure 3.3.

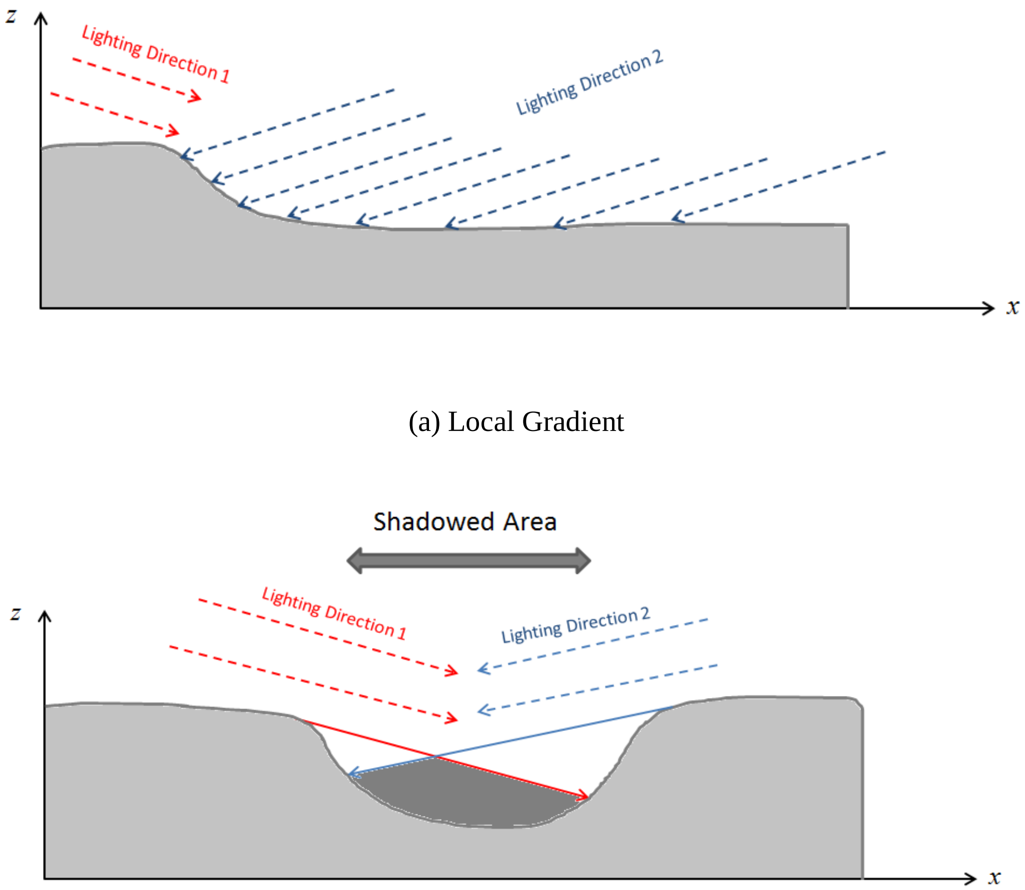


(a) Local Gradient

(b) Local Dip

Figure 3.3. Shadowing of Local Gradient & Local Dip

Since pavement crack is a long strip pattern constituted by local dips, local gradients should be discarded by combing the projections with two opposite lighting directions. Here, "opposite" means the two lighting directions are opposite to each other in the xy plane. As illustrated in Figure 3.4, local gradients do not present any shadowed areas under opposite lighting directions, while local dips could be shadowed under lighting of opposite directions.



(a) Local Gradient

(b) Local Dip

Figure 3.4. Composite Shadowing of Local Gradient & Local Dip under Lighting of Opposite Directions

It is clear that the shadowed area under opposite lighting directions is lower than the local neighborhood. And the cracks are hided partially or entirely in the shadowed areas. The percentage of true cracks being shadowed is dependent on the projection angle. The projection angle is defined as the angle between xy plane and the projection direction, illustrated in Figure 3.5.

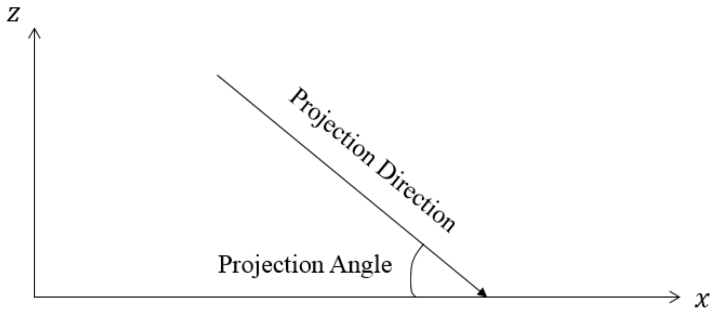


Figure 3.5. Projection Angle

When the projection angle is 90 degree, nothing can be shadowed. On the other hand, all pixels can be shadowed if the projection angle is 0 degree. Therefore, the projection angles between 0-90 degrees will generate a diverse transformations of the original data into the binary map. Such a transformation is controlled only by a single parameter: the projection angle, which implies the advantage of 3D Shadow Simulation.

Certainly, there could be many pairs of opposite lighting directions in the xy plane, as illustrated in Figure 3.6.

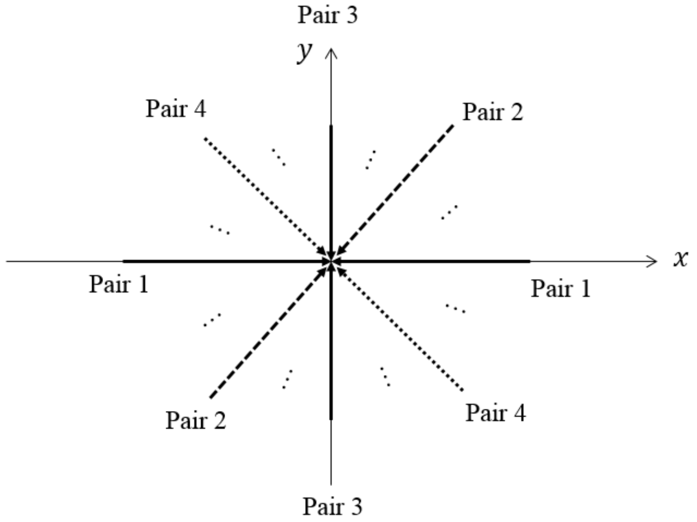
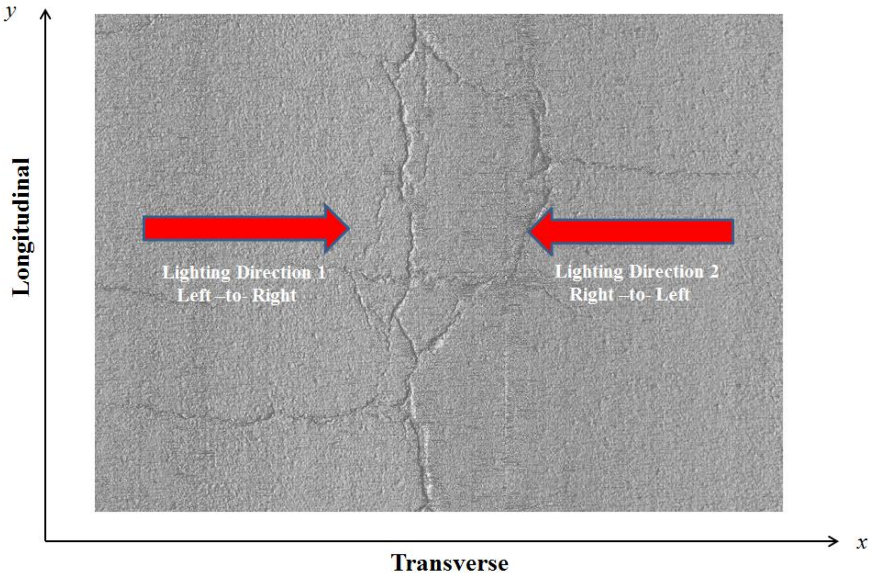
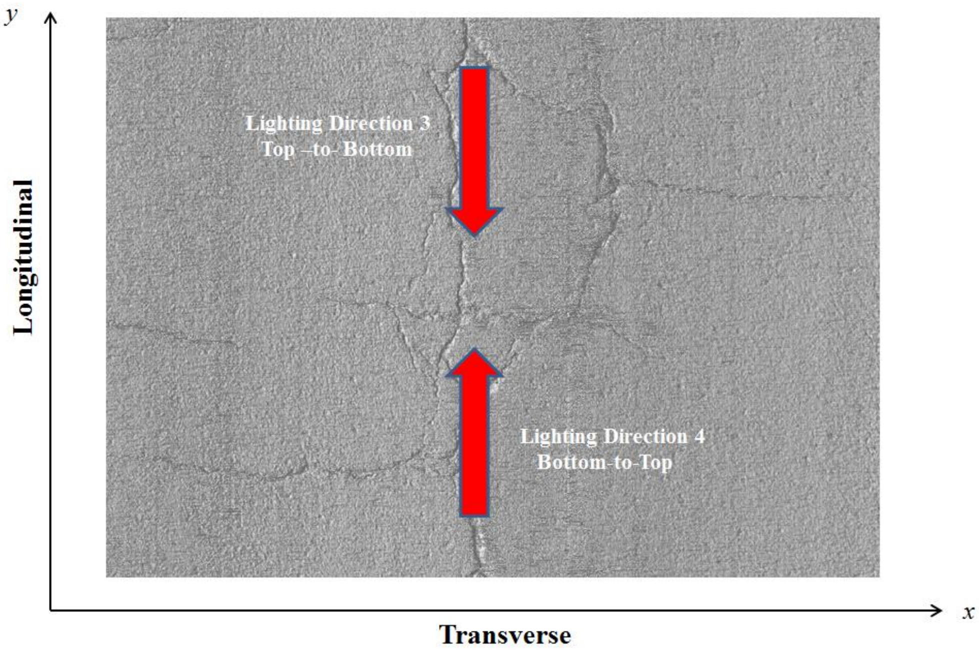


Figure 3.6. Pairs of Opposite Directions in XY Plane

In the xy plane, the optimized pair of lighting directions should be perpendicular to the growing direction of a crack. However, cracks on pavement surface always present various growing directions. Instead of using excessive pairs of lighting directions that magnifies the computational complexity and increases processing time, two pairs of lighting directions are suggested in this study to implement tasks of finding longitudinal-dominated cracks and transverse-dominated cracks respectively. According to the projection directions of each pair in xy plane, the two pairs of opposite lightings are called Transverse Lighting and Longitudinal Lighting, as illustrated in Figure 3.7.



(a) Transverse Lighting



(b) Longitudinal Lighting

Figure 3.7. Transverse Lighting & Longitudinal Lighting

It could be determined that Transverse Lighting is devised mainly for the detection of longitudinal-dominated cracks, while Longitudinal Lighting is designated to detect transverse-dominated cracks.

Denote $\boldsymbol{u} = (u_x, u_y, u_z)$ as the normalized vector in the lighting direction, which subjects to:

$$u_z \geq 0 \tag{3.1}$$

The inequality (3.1) indicates the light always comes above the xy plane, or parallel to the xy plane. In order to determine which pixels can be shadowed, it is necessary to define the light tracing model.

As the sunlight beams are assumed to be parallel everywhere, the sunlight imaginarily starts from each pixel and travels along the lighting direction until it is out of the data domain. Therefore, each pixel can be consider as the origin of a light beam. Denote $p(i, j)$ as the height of pixel (i, j),

$\boldsymbol{P}_{ij} = (i, j, p(i, j))$ as the 3D point presented by pixel (i, j), and $\boldsymbol{l}(i, j)$ as the light beam whose origin is $\boldsymbol{P}_{ij}$. Given the sampling resolution Δs, a collection of 3D points $\boldsymbol{\Omega}_{i,j} = \{\boldsymbol{P}_{ij}^{(1)}, \boldsymbol{P}_{ij}^{(2)}, \boldsymbol{P}_{ij}^{(3)} \dots, \boldsymbol{P}_{ij}^{(N)}\}$ can be sampled along the light beam $\boldsymbol{l}(i, j)$. That is:

$$\boldsymbol{P}_{ij}^{(k)} = \boldsymbol{P}_{ij} + k \cdot \Delta s \cdot \boldsymbol{u} \tag{3.2}$$

Where: $\boldsymbol{P}_{ij}^{(k)}$ is the kth 3D point sampled along $\boldsymbol{l}(i, j)$;

The x and y components of $\boldsymbol{P}_{ij}^{(k)}$ can approximate the location of a certain pixel in xy plane, while the z component of $\boldsymbol{P}_{ij}^{(k)}$ reflects the height of light beam $\boldsymbol{l}(i, j)$ at that pixel. In order to determine which pixels are visited by the light beam $\boldsymbol{l}(i, j)$, the x and y components of $\boldsymbol{P}_{ij}^{(k)}$ are rounded to the nearest integers such that they can be associated with a specific pixel in the xy plane. In other words, $\boldsymbol{P}_{ij}^{(k)}$ is approximated as:

$$\boldsymbol{P}_{ij}^{(k)} = round\left(\boldsymbol{P}_{ij}^{(k)} \cdot \boldsymbol{e}_x\right)\boldsymbol{e}_x + round\left(\boldsymbol{P}_{ij}^{(k)} \cdot \boldsymbol{e}_y\right)\boldsymbol{e}_y + \left(\boldsymbol{P}_{ij}^{(k)} \cdot \boldsymbol{e}_z\right)\boldsymbol{e}_z \tag{3.3}$$

Where: $\boldsymbol{e}_x$, $\boldsymbol{e}_y$ and $\boldsymbol{e}_z$ are the unit vectors for the x axis, y axis and z axis respectively.

In equation (3.2), the sampling resolution Δs should be small enough such that no pixels will be ignored. The sampling resolution Δs is set as $\Delta s = 1$ in this study. Thereby the following inequality holds:

$$\begin{cases} \Delta s \cdot \boldsymbol{u} \cdot \boldsymbol{e}_x = u_x \leq 1 \\ \Delta s \cdot \boldsymbol{u} \cdot \boldsymbol{e}_y = u_y \leq 1 \end{cases} \tag{3.4}$$

The inequality (3.4) indicates no pixels will be missed with the sampling resolution $\Delta s = 1$.

With the light tracing model, a pixel could be visited by multiple light beams starting from different pixels. Therefore, $\boldsymbol{P}_{ij}^{(k)}$ is modified as $\boldsymbol{P}_{ij}^{(i'j')}$ to denote the 3D sample point at pixel

(i',j') of the light beam $\boldsymbol{l}(i,j)$. For simplicity, Light Height $h_{ij}(i',j')$ is defined as the height of light beam $\boldsymbol{l}(i,j)$ at pixel (i',j'). Thus:

$$h_{ij}(i',j') = \boldsymbol{P}_{ij}^{(i'j')} \cdot \boldsymbol{e}_z \tag{3.5}$$

For a pixel that is influenced by multiple light beams, only the Maximum Light Height at that pixel is retained. That is:

$$h_{max}(i',j') = \max\{h_{i_1 j_1}(i',j'), h_{i_2 j_2}(i',j'), \dots, h_{i_m j_m}(i',j')\} \tag{3.6}$$

Where: $h_{i_k j_k}(i',j')$ is the height of kth light beam $\boldsymbol{l}(i_k,j_k)$ at pixel (i',j').

Finally, a pixel is shadowed if the Maximum Light Height at that pixel is greater than its own height. That is:

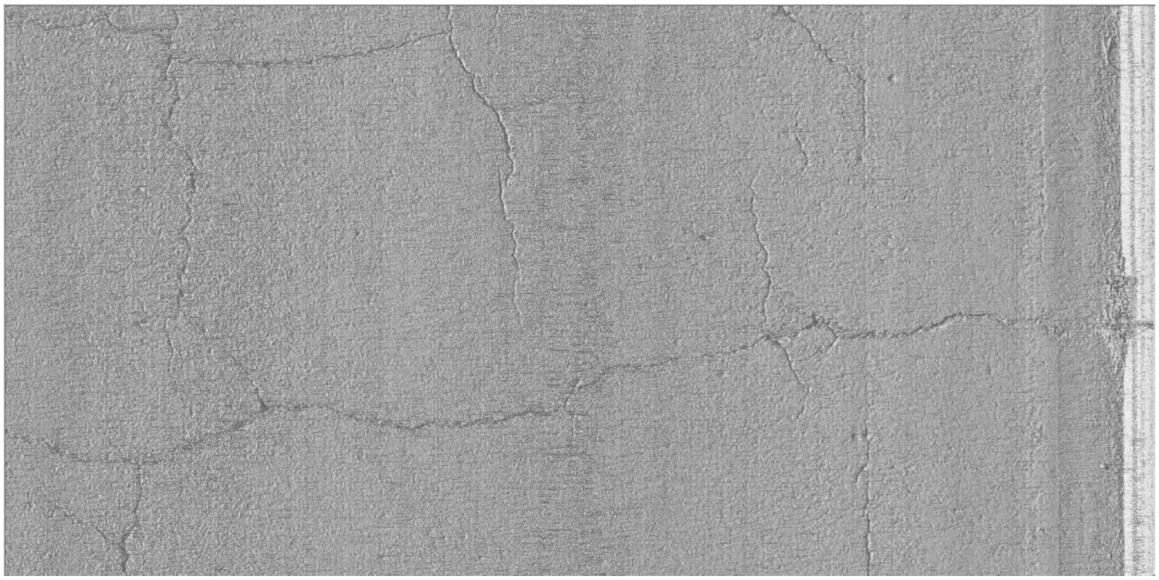
$$B(i,j) = \begin{cases} 0 & if\ h_{max}(i,j) > p(i,j) \\ 1 & otherwise \end{cases} \tag{3.7}$$

Where: $B(i,j)$ is the value of pixel (i,j) in the binary map.

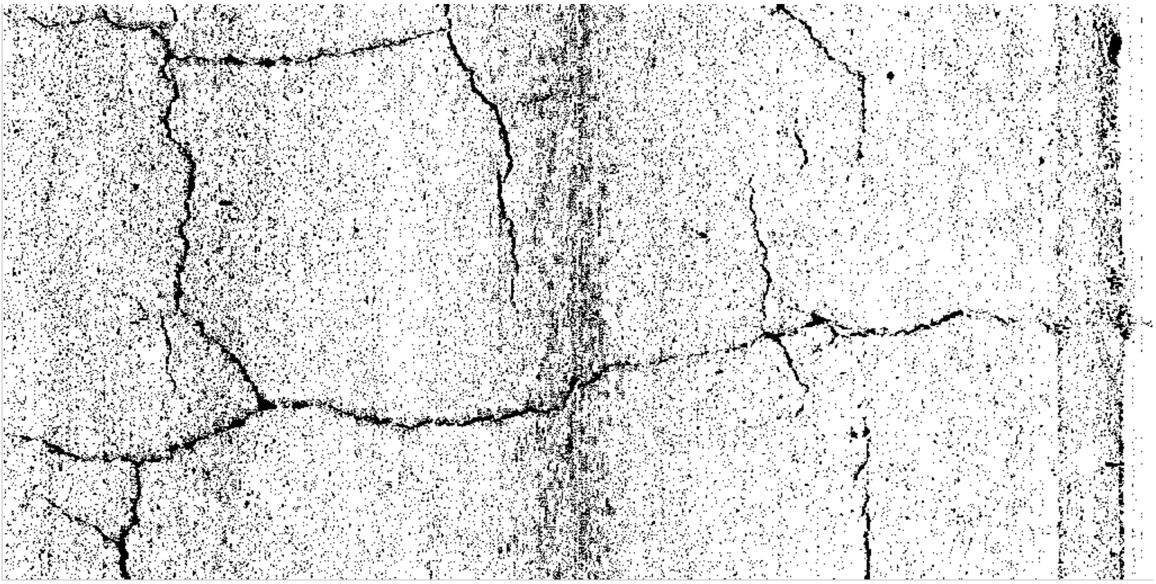
Based on the aforementioned methodology, the procedures to generate shadow map are concluded below, where shadow map is the binary map in which shadowed pixels have the value “0” and the others have the value “1”.

- Initialize the maximum light height at each pixel as $h_{max}(i,j)$=0;
- For each pixel (i,j), start tracing of the light beam $\boldsymbol{l}(i,j)$ until it is out of the data domain;
- Determine the pixels visited by $\boldsymbol{l}(i,j)$ and update the corresponding maximum light heights at their locations;
- Assign binary value for each pixel by referring to equation (3.7).

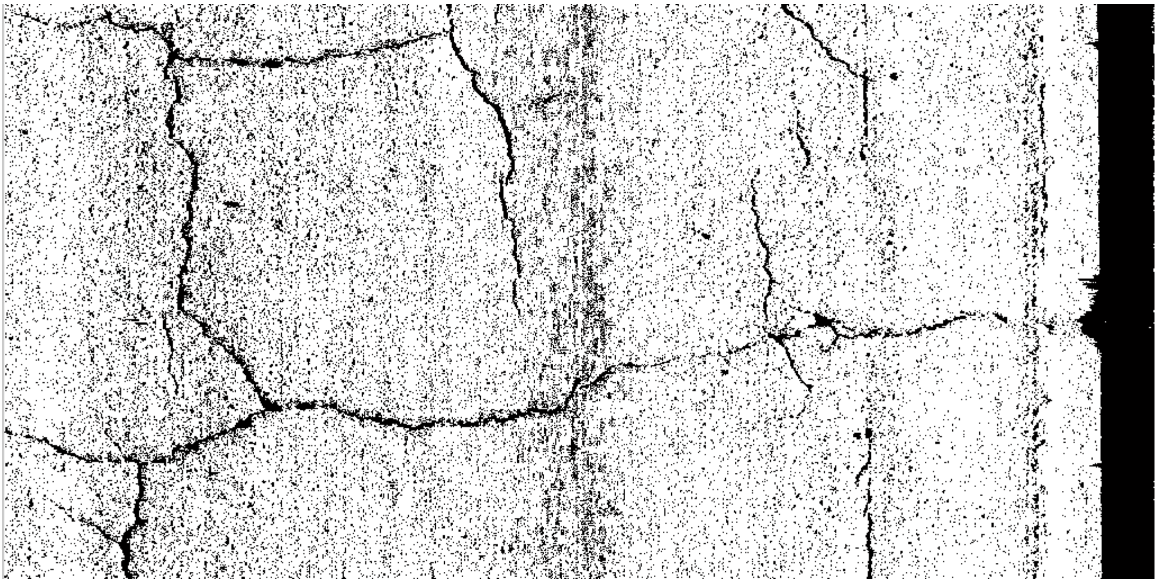
Examples of shadowing on 3D pavement surface under four individual lighting directions are shown in Figure 3.8.



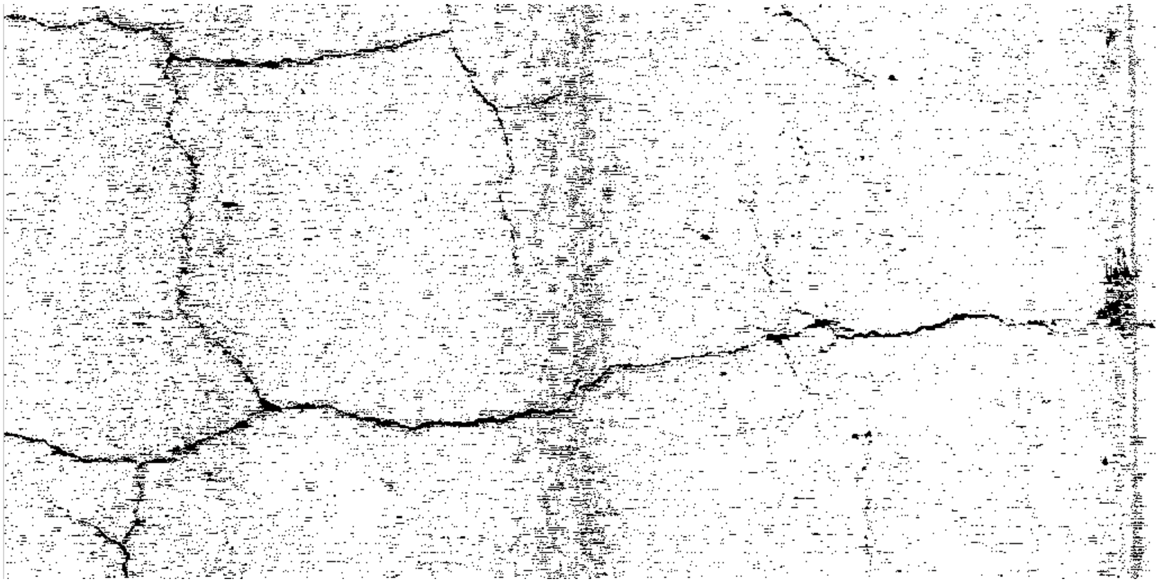
(a) Original 3D Data



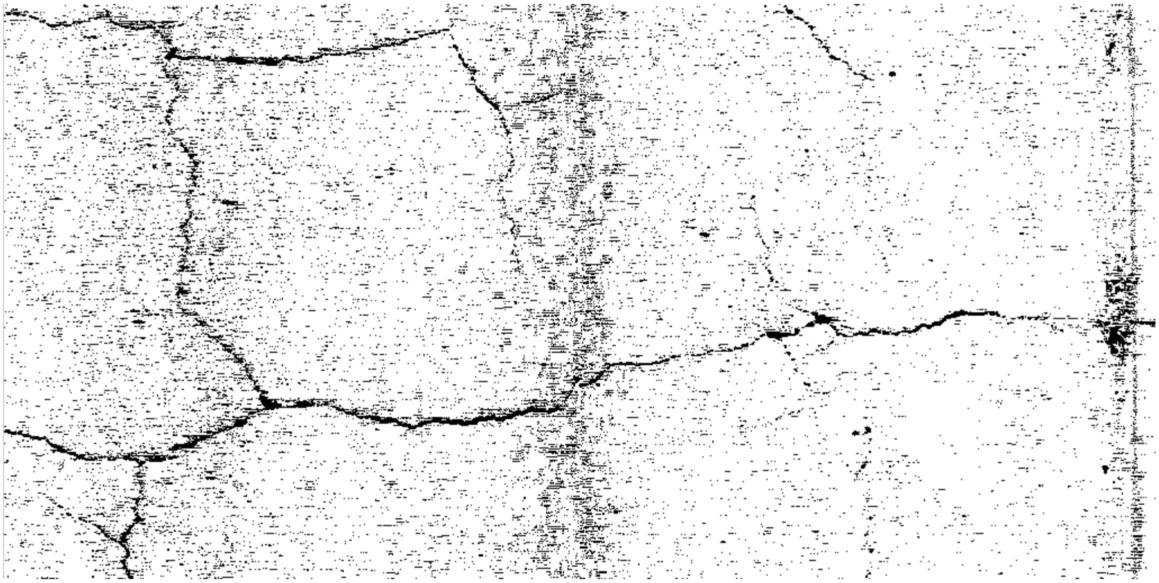
(b) Shadow Map 1 (Lighting Direction: Left-to-Right)



(c) Shadow Map 2 (Lighting Direction: Right-to-Left)



(d) Shadow Map 3 (Lighting Direction: Top-to-Bottom)



(e) Shadow Map 4 (Lighting Direction: Bottom-to-Top)

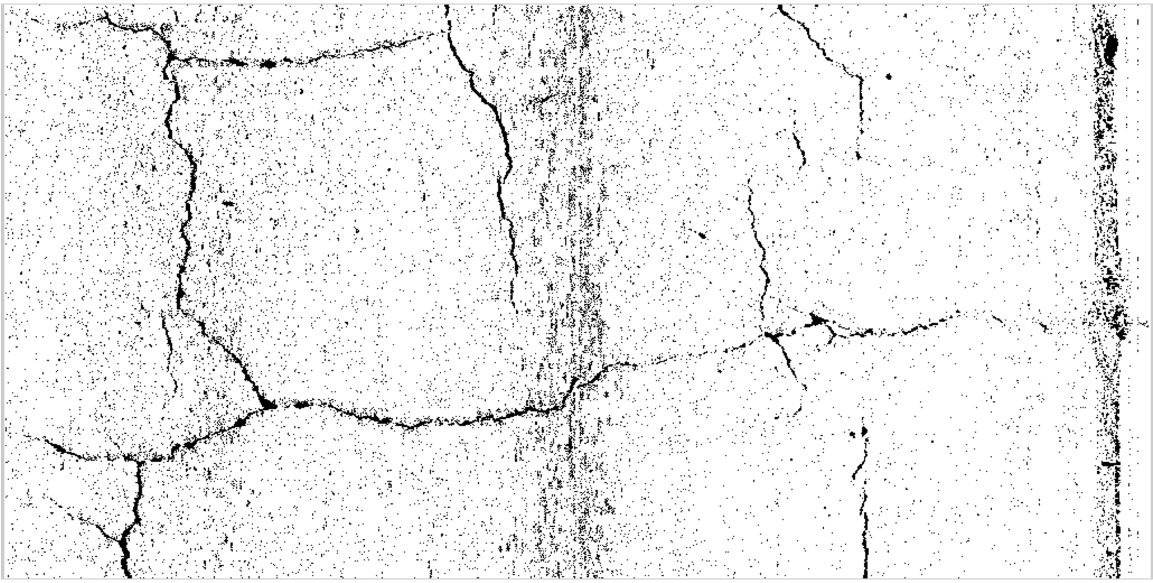
Figure 3.8. Shadowing on 3D Pavement Surface with 4 Lighting Directions

In order to only retain local dips, shadow maps under opposite lighting directions are combined together to formulate a composite shadow map where pixels shadowed under both directions are assigned with value “0”, and the others are labeled with value “1”. That is:

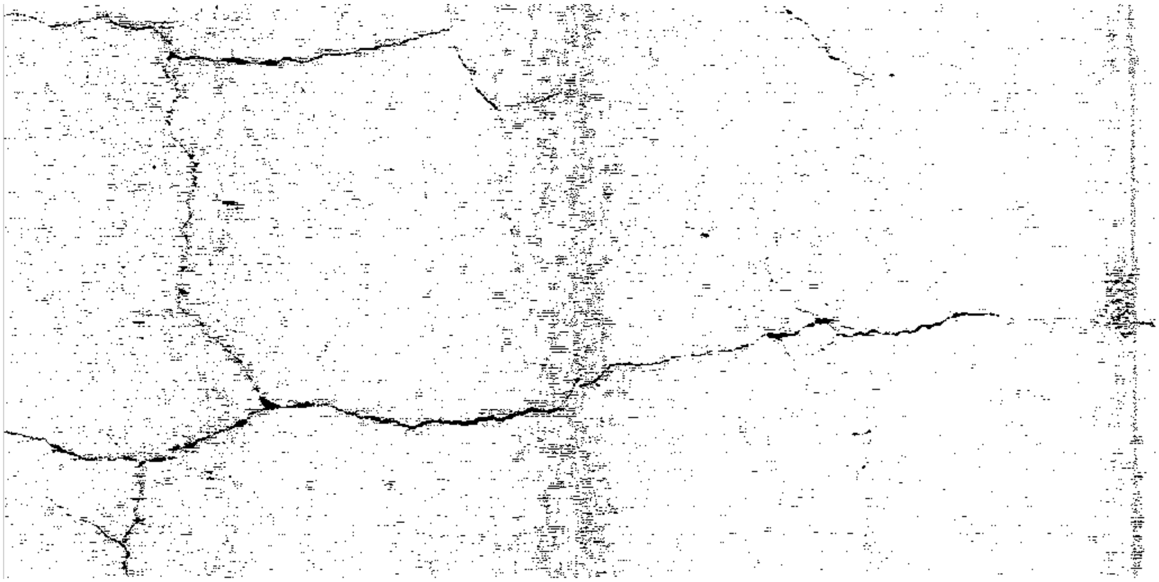
$$B_c(i,j) = \begin{cases} 0 & if\ B(i,j) = 0\ and\ \bar{B}(i,j) = 0 \\ 1 & otherwise \end{cases} \tag{3.8}$$

Where: $B_c(i,j)$ is the binary value of pixel (i,j) on the composite shadow map; $B(i,j)$ and $\bar{B}(i,j)$ are the binary values of pixel (i,j) on the two shadow maps with opposite lighting.

The composite shadow maps of the example data generated by Transverse Lighting and Longitudinal Lighting are shown in Figure 3.9.



(a) Composite Shadow Map by Transverse Lighting



(b) Composite Shadow Map by Longitudinal Lighting

Figure 3.9. Composite Shadow Maps by Transverse Lighting and Longitudinal Lighting

It can be observed from Figure 3.9 that longitudinal-dominated cracks have clearer presence and better continuity on the composite shadow map generated by Transverse Lighting. On the contrary, transverse-dominated cracks are more sustainable under Longitudinal Lighting. In order to integrate longitudinal-dominated cracks and transverse-dominated cracks together for latter analyses, the composite shadow maps under Transverse Lighting and Longitudinal Lighting are integrated as:

$$I_c(i,j) = \begin{cases} 0 & if\ B_c(i,j) = 0\ or\ \bar{B}_c(i,j) = 0 \\ 1 & otherwise \end{cases} \tag{3.9}$$

Where: $I_c(i,j)$ is the binary value of pixel (i,j) on the final integrated shadow map; $B_c(i,j)$ and $\bar{B}_c(i,j)$ are the binary values of pixel (i,j) on the two composite shadow map generated by Transverse Lighting and Longitudinal Lighting.

3.2.2 Noise Suppressing Algorithms

When the integrated shadow map is obtained, the remaining task is to extract linear patterns from the integrated shadow map. In this study, the following two criteria are used for the implementation of noise suppressing.

- A crack should be a continuously developed linear pattern whose length is much greater than its width;
- The length of a crack should be greater than the defined minimum length.

Before the implementation of noise suppressing, Clustering is applied to label collected pixels in the integrated shadow map such that further analyses can be conducted on cracking clusters rather than individual pixels. A common image processing technique, Blob Extraction (alternatively Connected-component Labeling), is adopted in this study to divide resulting binary images into connected subsets using the two-pass algorithm (Shapiro and Stockman, 2001). The first pass of the clustering algorithm is to assign temporary labels and record equivalences, while the second pass is to reassign temporary labels as the smallest labels using the Union-Find structure which is designed for “Union” operation (Merging two sets into one) and “Find” operation (determining which set a specific element belongs to). An example of clustering is shown in Figure 3.10, where collected pixels are grouped in the same cluster and assigned with identical color.

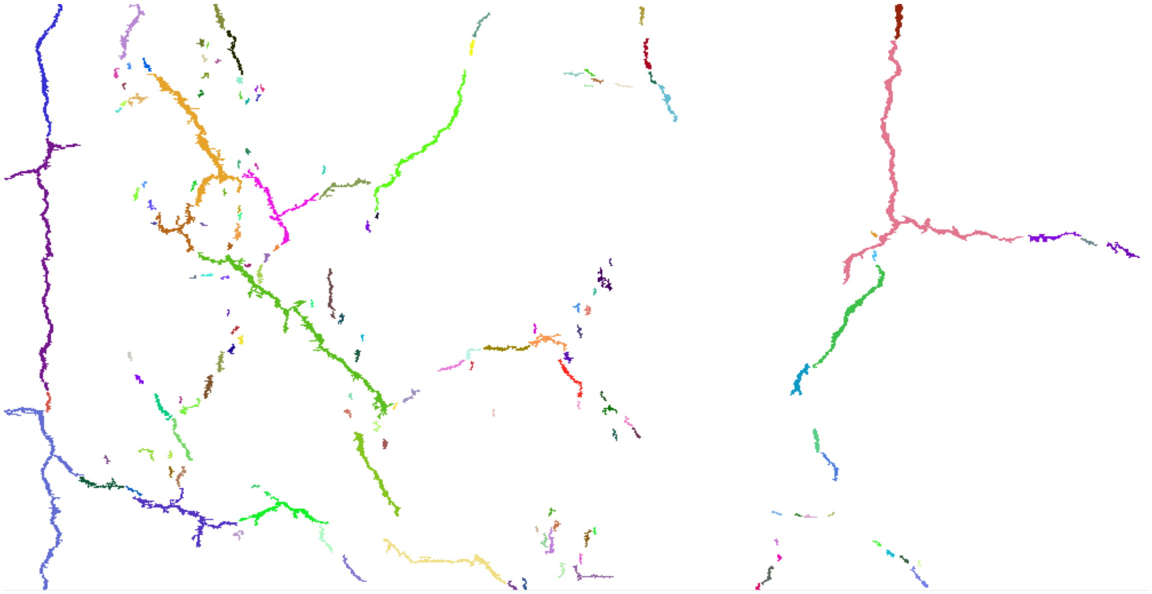


Figure 3.10. Example of Clustering

Based on the clustering result, Linear Pattern Analysis is developed as a noise suppressing algorithm to provide some help in removing clusters which are very likely to be noises due to unsatisfied linearity. For Linear Pattern Analysis, the following features of a crack are considered in the evaluation: 1) a crack is a long strip pattern; 2) the width distribution of a crack in a small local area can be regarded as uniform; 3) the growing direction of a crack in a small local area does not change dramatically.

Three indices are used to analyze the pattern of a potential cracking cluster: score of slimness, score of width uniformity and score of direction uniformity. Scoring slimness is essentially to calibrate the ratio of the total length of the potential cracking cluster to its average width into the range from 0 to 100. Based on preliminary investigations, the ratio of the total length of the crack to its average width is greater than 20 for most of pavement cracks. Thus, the score of slimness of a potential cracking cluster is defined as below:

$$S_{i,1} = \min(100, \frac{L_i}{\overline{W}_i} \times \frac{100}{R_{threshold}}) \tag{3.10}$$

Where: $S_{i,1}$- Score of slimness of i^{th} cracking cluster; L_i- Total length of i^{th} cracking cluster; $\overline{W}_i$-Average width of i^{th} cracking cluster; $R_{threshold}$- The threshold above which the ratio of the total length of a cracking cluster to its average width is completely acceptable, $R_{threshold} = 20$ is adopted in this paper.

For the score of width uniformity, the width of each profile in a cracking cluster is compared to the width of next profile, if the difference between the two widths exceeds a certain degree, the current profile will be counted as an invalid profile. The number of invalid profiles will be accumulated as:

$$N_{i,invalid} = \sum_{j=1}^{N_i-1} E(i,j) \tag{3.11}$$

Subject to:

$$E(i,j) = \begin{cases} 0, & if \;\; \frac{\max(W_{i,j}\,,\;W_{i,j+1}) - \min(W_{i,j}\,,\;W_{i,j+1})}{\min(W_{i,j}\,,\;W_{i,j+1})} \le \frac{\min(W_{i,j}\,,\;W_{i,j+1})}{2} \\ 1, & if \;\; \frac{\max(W_{i,j}\,,\;W_{i,j+1}) - \min(W_{i,j}\,,\;W_{i,j+1})}{\min(W_{i,j}\,,\;W_{i,j+1})} > \frac{\min(W_{i,j}\,,\;W_{i,j+1})}{2} \end{cases} \tag{3.12}$$

Where: $N_{i,invalid}$-Number of invalid profiles in i^{th} cracking cluster; N_i-Number of profiles in i^{th} cracking cluster; $E(i,j)$ –Indicator of invalid profile for the j^{th} profile in i^{th} cracking cluster; $W_{i,j}$-Width of j^{th} profile in i^{th} cracking cluster;

With the number of invalid profiles, the score of width uniformity for the i^{th} potential cracking cluster $S_{i,2}$ can be derived by:

$$S_{i,2} = \frac{N_{i,invalid}}{N_i - 1} \times 100 \tag{3.13}$$

With respect to the score of direction uniformity, the current growing direction at a cracking profile is pointing from the center of last profile to the center of current profile. Then, the current growing direction will be compared to the next growing direction pointing from the center of current profile to the center of next profile. If the angle between the two growing directions is greater than a certain threshold, the current growing direction will be counted as an invalid growing direction. Therefore, the number of invalid growing directions is accumulated as below:

$$N_{i,invalid}{}' = \sum_{j=2}^{N_i-1} E'(i,j) \tag{3.14}$$

Subject to:

$$E'(i,j) = \begin{cases} 0, \ \ if \ \ \frac{(\boldsymbol{Pt}_{i,j} - \boldsymbol{Pt}_{i,j-1}) \cdot (\boldsymbol{Pt}_{i,j+1} - \boldsymbol{Pt}_{i,j})}{\|(\boldsymbol{Pt}_{i,j} - \boldsymbol{Pt}_{i,j-1})\| \times \|(\boldsymbol{Pt}_{i,j+1} - \boldsymbol{Pt}_{i,j})\|} \geq \cos\beta_0 \\ 1, \ \ if \ \ \frac{(\boldsymbol{Pt}_{i,j} - \boldsymbol{Pt}_{i,j-1}) \cdot (\boldsymbol{Pt}_{i,j+1} - \boldsymbol{Pt}_{i,j})}{\|(\boldsymbol{Pt}_{i,j} - \boldsymbol{Pt}_{i,j-1})\| \times \|(\boldsymbol{Pt}_{i,j+1} - \boldsymbol{Pt}_{i,j})\|} < \cos\beta_0 \end{cases} \tag{3.15}$$

Where: $N_{i,invalid}{}'$-Number of invalid growing directions in i^{th} cracking cluster; N_i-Number of profiles in i^{th} cracking cluster; $E'(i,j)$ –Indicator of invalid growing direction at j^{th} profile in i^{th} cracking cluster; $\boldsymbol{Pt}_{i,j}$-Center point of j^{th} profile in i^{th} cracking cluster; β_0-The allowed maximum angle between two successive growing directions, $\beta_0 = \frac{\pi}{12}$ is used in the paper.

Similarly, the score of direction uniformity for the i^{th} potential cracking cluster $S_{i,3}$ can be computed as:

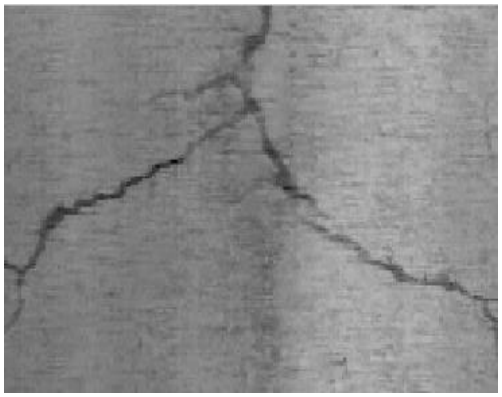
$$S_{i,3} = \frac{N_{i,invalid}{}'}{N_i - 2} \times 100 \tag{3.16}$$

The weighted sum of the three indices is finally applied to give an overall score for the linearity of the cracking cluster. Consequently, cracking clusters which follow the statement below will survive in Linear Pattern Analysis, otherwise they will be eliminated.

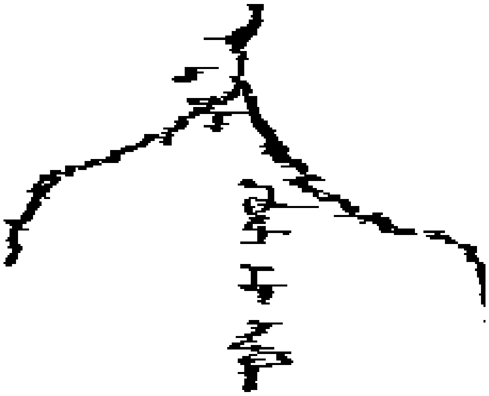
$$F_i = \alpha_1 S_{i,1} + \alpha_2 S_{i,2} + \alpha_3 S_{i,3} \geq F_{min} \tag{3.17}$$

Where: F_i-Linearity Score of i^{th} cracking cluster; $S_{i,1}$-Score of slimness of the i^{th} cracking cluster; $S_{i,2}$-Score of width uniformity of i^{th} cracking cluster; $S_{i,3}$-Score of direction uniformity of i^{th} cracking cluster; α_1, α_2, α_3-Weight factors, $\alpha_1 = \alpha_2 = \alpha_3 = 0.33$ is used in this study; F_{min}-Minimal linearity score, $F_{min} = 80$ is adopted in this study.

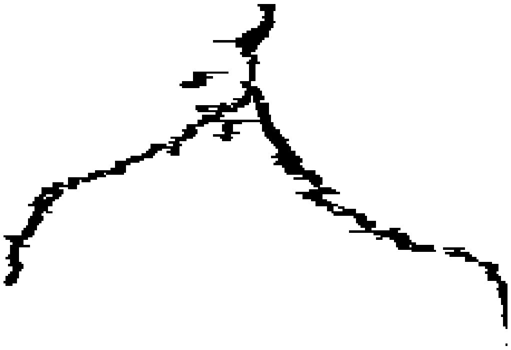
Figure 3.11 shows an example of some noisy patterns that are eliminated through Linear Pattern Analysis.



(a) Original 3D Image



(b) Result Before Linear Pattern Analysis



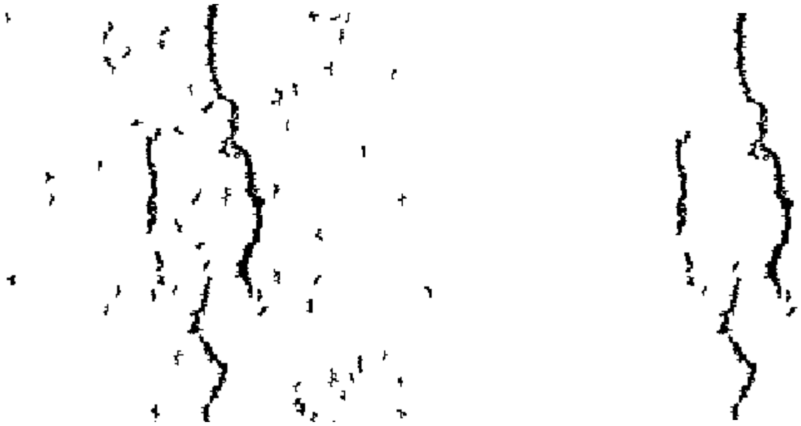
(c) Result After Linear Pattern Analysis

Figure 3.11.Linear Pattern Analysis

According to the definition of a crack described in the AASHTO Designation (AASHTO, 2013), the minimum length of a crack is 25mm. Therefore after Linear Pattern Analysis, if there are any tiny cracks that satisfy both of the following conditions, they will be considered as noises and then eliminated.

- The length of the crack is less than 25mm;
- The crack is isolated within the local neighborhood of a certain size.

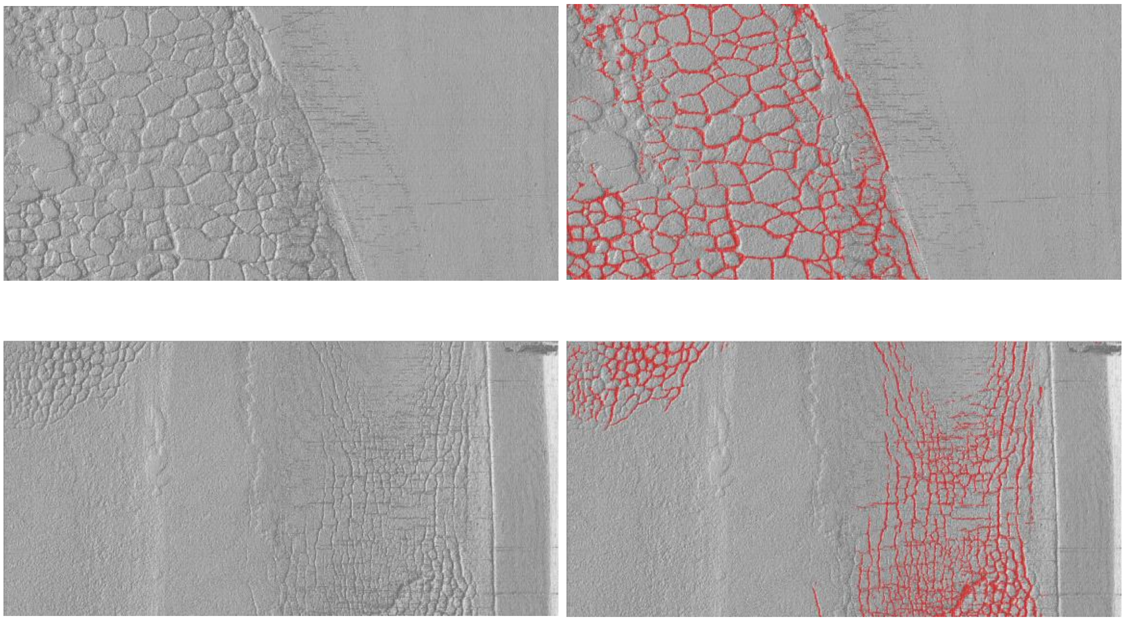
It should be noted that the second condition is set to protect those small parts of an acceptable crack from elimination. An example of eliminating tiny cracks is shown in Figure 3.12.

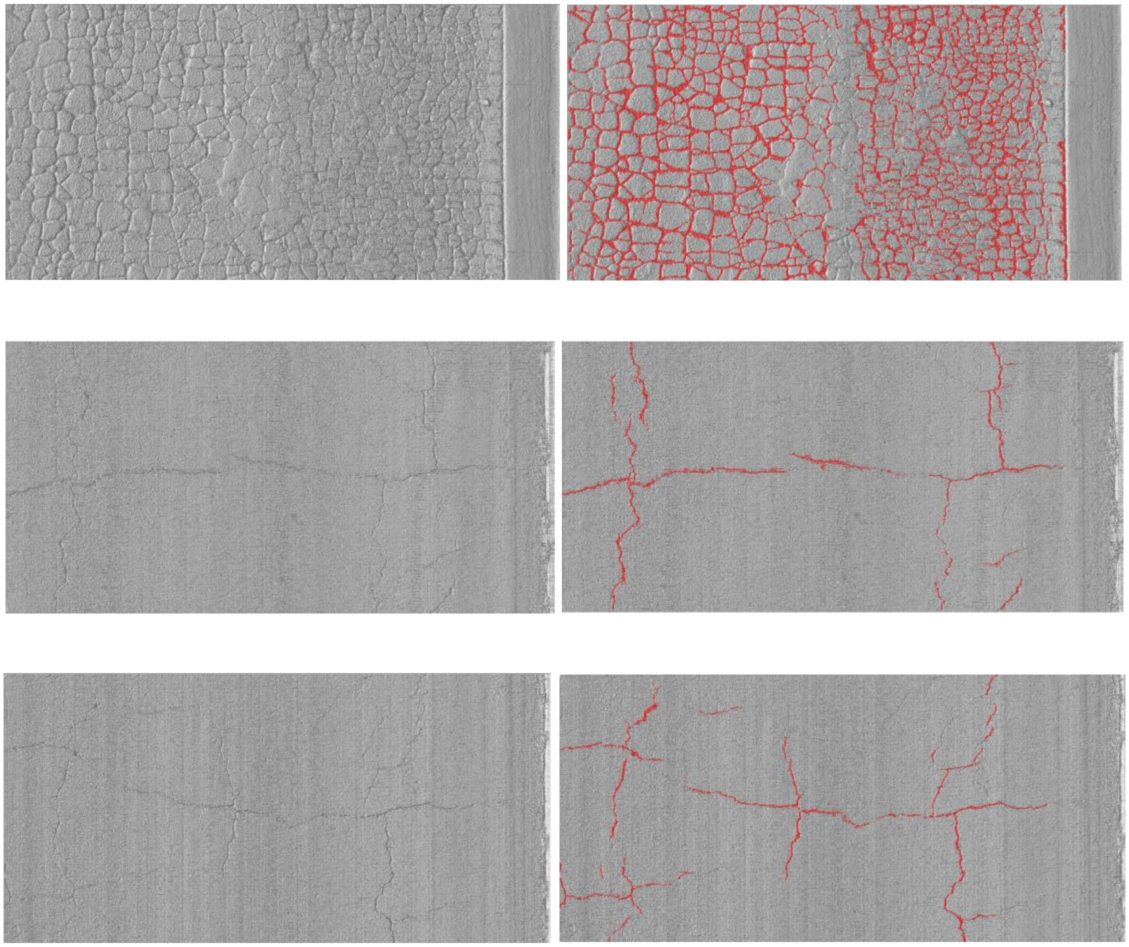


(a) Before Elimination of Tiny Cracks (b) After Elimination of Tiny Cracks

Figure 3.12. Elimination of Tiny Cracks

In a summary, the Fully Automated Cracking Detection Subsystem applies 3D Shadow Simulation to transform the original 3D data into a binary map, followed by the use of noise suppressing algorithms to eliminate unwanted patterns from the binary map. Some typical examples of pavement cracks detected by Fully Automated Cracking Detection Subsystem are shown in Figure 3.13.





(a) Original 3D Data

(b) Results

Figure 3.13. Typical Examples by Fully Automated Cracking Detection Subsystem

3.3 Case Study

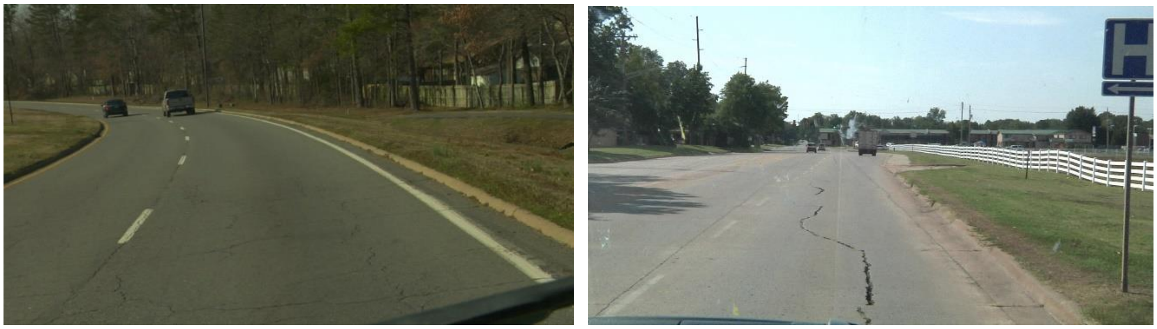
For the validation of the Fully Automated Cracking Detection Subsystem, a case study on real 3D pavement data is conducted in this chapter. Pavements of different levels of cracking severity are chosen for performance evaluation. As the projection angle has a great impact on the 3D Shadow Simulation, an appropriate set of projection angles for Transverse Lighting and Longitudinal Lighting is determined and fixed for the case study based on prior knowledge and preliminary testing. In the case study, the projection angle for both Transverse Lighting and Longitudinal Lighting is fixed as 60 degree.

3.3.1 Selected Pavement Segments

Two pavement segments which have substantial amount and diverse types of pavement cracking are selected for case study. The first pavement segment (Pavement I) is from State Highway AR100-EB in Arkansas, while the second pavement segment (Pavement II) comes from the Hall of Fame Road located in Stillwater, Oklahoma. Both segments are 300m long. The cracking severity levels of the two segments are listed in Table 3.1. And the right-of-view of the selected pavement segments is shown in Figure 3.14.

Table 3.1. Cracking Severity Levels of Pavement I and Pavement II

Pavement ID	Location	Length (m)	Level of Cracking Severity
Pavement I	Arkansas, US	300	Low & Medium
Pavement II	Oklahoma, US	300	Medium & High



(a) Pavement I

(b) Pavement II

Figure 3.14. Right-of-way View of the Selected Pavement Segments

3.3.2 Precision-Recall Analysis

In order to evaluate the performance of pavement cracking detection algorithms, the concept of Precision-Recall analysis is utilized in this study as the evaluation methodology. Precision-Recall analysis was first introduced in the area of Information Retrieve (IR) as an evaluation metric (Fawcett, 2006; Davis and Goadrich, 2006). The calculation of parameters in this analysis is shown in equation (3.18) ~ (3.21), where True Positive, True Negative, False Positive and False Negative are four detection outcome scenarios that constitute the *confusion matrix* of a detection result.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{3.18}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3.19}$$

$$False\ Positive\ Rate\ (FPR) = \frac{False\ Positive}{False\ Positive + True\ Negative} \tag{3.20}$$

$$False\ Omission\ Rate\ (FOR) = \frac{False\ Negative}{False\ Negative + True\ Negative} \tag{3.21}$$

In pavement cracking detection, True Positive pixels are the cracking pixels classified correctly as belonging to one of the cracking clusters, while False Positive pixels are the non-cracking pixels classified incorrectly as belonging to one of the cracking clusters. On the other hand, True Negative pixels represent non-cracking pixels that are classified correctly as not belonging to any cracking cluster. And False Negative pixels stand for cracking pixels that are classified incorrectly as not belonging to any cracking cluster but should have been. Therefore, as referring to equation (3.18) ~ (3.21), Precision is a measure of how many of the detected pixels are classified correctly as cracks, while Recall reflects the percentage of cracking pixels detected by the detection algorithm. In addition, False Positive Rate (FPR) and False Omission Rate (FOR) stand for the percentage of False Positive error (Type I error) and the percentage of False Negative error (Type II error) respectively. In a general view, False Positive errors will be reduced as the Precision is increased, while False Negative errors will also be reduced with a higher Recall.

Proceeding to Precision-Recall Analysis, the ground truth of all cracks on the two pavement segments are marked manually. The results of Precision-Recall Analysis on the two selected segments are illustrated in Table 3.2 and Table 3.3.

Table 3.2. Performance of Fully Automated Cracking Detection Subsystem on Pavement I

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	91.68	91.09	0.28	0.30
2	20-40	89.30	86.58	0.35	0.45
3	40-60	93.39	88.60	0.30	0.54
4	60-80	86.45	87.85	0.34	0.30
5	80-100	94.43	89.87	0.15	0.29
6	100-120	87.13	90.16	0.34	0.25
7	120-140	92.01	85.77	0.15	0.29
8	140-160	89.39	86.71	0.18	0.23
9	160-180	95.73	94.59	0.11	0.14
10	180-200	88.32	92.47	0.34	0.21
11	200-220	89.26	85.99	0.33	0.44
12	220-240	95.10	84.45	0.14	0.51
13	240-260	91.83	88.01	0.29	0.45
14	260-280	90.74	86.25	0.30	0.46
15	280-300	94.68	91.87	0.15	0.24
Overall	0-300	**91.30**	**88.69**	0.25	0.34

Table 3.3. Performance of Fully Automated Cracking Detection Subsystem on Pavement II

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	93.06	91.93	0.27	0.32
2	20-40	93.20	90.70	0.28	0.40
3	40-60	94.47	92.08	0.25	0.37
4	60-80	89.43	90.30	0.22	0.20
5	80-100	87.57	88.26	0.18	0.17
6	100-120	94.56	89.50	0.09	0.19
7	120-140	93.84	88.53	0.15	0.29
8	140-160	92.89	93.11	0.13	0.12
9	160-180	96.25	90.01	0.10	0.27
10	180-200	91.39	87.40	0.14	0.21
11	200-220	90.65	89.59	0.32	0.37
12	220-240	92.63	91.63	0.15	0.17
13	240-260	94.38	87.41	0.09	0.21
14	260-280	91.75	93.14	0.28	0.23
15	280-300	93.78	88.47	0.11	0.22
Overall	0-300	**92.66**	**90.14**	0.18	0.25

It can be observed from Table 3.2 and Table 3.3 that the fully automated detection algorithms achieves high Precision and high Recall for both segments. For Pavement I, the average Precision is 91.30%, while the average Recall is 88.69%. Comparatively, the fully automated detection algorithms reach higher Precision (92.66%) and Recall (90.14%) on Pavement II, which may be due to the higher severity and better continuity of cracks on Pavement II. Moreover, the performance of the fully automated detection algorithms on all reporting sections does not change with a wild range.

3.4 Summary

In this chapter, 3D Shadow Simulation is proposed to transform the original 3D data to a binary map where local lower areas are shadowed and labeled as “0”, and local higher areas are not shadowed and labeled as “1”. The pairing of opposite lighting directions is conduced to differentiate local dips (potential parts of cracks) from local gradients. Transverse Lighting, which is a pair of opposite lighting in the transverse direction, serves to find the longitudinal-dominated cracks. On the other hand, Longitudinal Lighting, which is a pair of opposite lighting in the longitudinal direction, is mainly to preserve transverse-dominated cracks. In addition to 3D Shadow Simulation, a sequence of noise suppressing algorithms is utilized to further eliminate noises from shadow maps. The case study on two pavement segments with diverse cracks demonstrates the good potential of the proposed fully automated cracking detection algorithms.

The advantage of 3D Shadow Simulation is that diverse solutions can be derived by adjusting a single parameter: the projection angle. One significant implication of such diversity is that all cracks could be shadowed with a low enough projection angle, no matter how vague the contrast between the crack and the local background is. Once all cracks could be shadowed correctly, a robust system of noise suppressing algorithms will maintain a large portion of true cracks and thus have a high level of success.

Furthermore, the selection of an appropriate projection angle is mainly associated with cracking width and cracking depth. As a matter of fact, the widths and depths of cracks on pavement surface have limited range, which means this fully automated detection subsystem is capable of maintaining high accuracy for many cases once the proper projection angles are assigned.

Chapter 4 OPTIMIZATION OF FULLY AUTOMATED CRACKING DETECTION ALGORITHMS USING GENETIC ALGORITHM

Genetic Algorithm is an artificial intelligence that simulates the process of natural selection, where elite solutions are maintained through generations while the others are eliminated. Genetic Algorithm is applied frequently for search problems in a large decision space. According to the number of objectives, Genetic Algorithms can be divided into two categories: Single Objective Genetic Algorithms and Multi-objective Genetic Algorithms. Since the goal for the optimization of a cracking detection algorithm is eventually to maximize Precision and Recall simultaneously, Multi-objective Genetic Algorithms are discussed in this study.

The objective of this chapter is to adopt a proper Multi-objective Genetic Algorithm that has adequate efficiency for optimization of the fully automated detection algorithm. The fundamental idea behind Multi-objective Genetic Algorithms is the evolutionary search of non-dominated solutions with high level of diversity along the pareto-optimal front. As mentioned in Section 2.2, Non-dominated Sorting Genetic Algorithm II (NSGA II) and Improved Strength Pareto Evolutionary Algorithm (SPEA2) are the two of most representative Multi-objective Genetic Algorithms. Since NSGA II has a lower complexity than SPEA2, NSGA II is adopted in this chapter. NSGA II possesses the capability in both preserving elites and maintaining diversity.

Through the systematic emphasis on the selection of a better non-dominated front, NSGA II can progress to the optimal solutions front by front. Additionally, the diversity is also preserved through the Crowded-Comparison approach proposed in NSGA II which penalizes crowed solutions and values isolated solutions (Deb et al., 2002).

4.1 NSGA II with Auxiliary Prediction Model

In addition to NSGA II, Auxiliary Prediction Model (APM) is proposed in this study to improve the convergence and diversity of solutions. The NSGA II evaluates groups of individual solutions in terms of convergence and diversity at each generation. However, it never predicts at each generation by using the information why a group is better than others in terms of dominance or diversity. Therefore, the APM is devised in this study to address three considerations:

- Why the first front is better than other fronts;
- Why some solutions are better than others in terms of diversity;
- How a potentially better group can be predicted by analyzing the difference between an existing better group and an existing worse group.

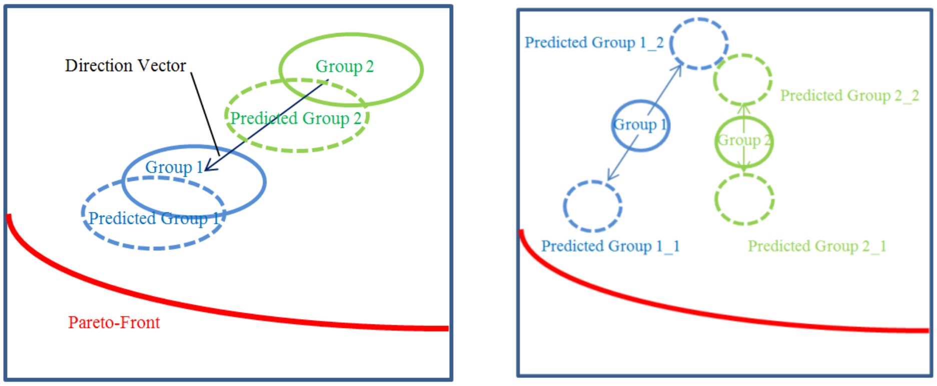
Since the convergence and diversity are of same importance to the final solutions, the APM essentially is divided into two separate models: Convergence Prediction Model and Diversity Prediction Model. The former is used to predict a group that may have better dominance, while the latter is applied to predict a group that has a potentially broader distribution.

4.1.1 Convergence Prediction Model

With respect to the number of non-dominated fronts, there could be two cases through generations:

- Case I: Individuals are located at multiple fronts;
- Case II: Entire population are located at the first front.

The Convergence Prediction Model is illustrated in Figure 4.1.



(a) Case I

(b) Case II

Figure 4.1. Convergence Prediction Model

As illustrated in Figure 4.1(a), when there are multiple fronts, the first front is treated as Group 1, and the other fronts are regarded as Group 2. A direction vector in the decision space pointing from Group 2 to Group 1 will be computed. Afterwards, two groups will be predicted ahead of the two original groups in the computed direction. The two predicted groups will potentially have better dominance than the original groups. Denote $\boldsymbol{O_1}$ as the centroid of Group 1, $\boldsymbol{O_2}$ as the centroid of Group 2. Then, the direction vector $\boldsymbol{V}$ pointing from $\boldsymbol{O_2}$ to $\boldsymbol{O_1}$ is:

$$\boldsymbol{V} = \boldsymbol{O_1} - \boldsymbol{O_2} \tag{4.1}$$

Then, the predicted solutions under Case I can be expressed as:

$$\boldsymbol{Q} = \{\boldsymbol{Q_i} | \boldsymbol{Q_i} = \boldsymbol{T_i} + \varepsilon \boldsymbol{V}\} \tag{4.2}$$

Where: $\boldsymbol{Q}$ is the predicted population; $\boldsymbol{Q_i}$ is the ith individual solution in the predicted population; $\boldsymbol{T_i}$ is the ith individual solution in the original population; ε ∈ (0,1] is a random variable that determines how far the predicted population are projected ahead.

Once the prediction is completed, the predicted groups will compete with the original groups. Only the best individuals from all groups will survive after the competition. Such a competition is the critical procedure to maintain the best individuals.

However, as illustrated in 4.1(b), if the entire population are located at the first front, no certain better direction could be learned. In this case, a pair of random opposite directions will be chosen for each individual randomly selected from the original solutions. Afterwards, for each randomly selected individual solution, two solutions will be predicted ahead along the two selected random directions respectively. That is:

$$\begin{cases} \boldsymbol{Q_i} = \boldsymbol{T_k} + \boldsymbol{V}_{r,k} \\ \boldsymbol{Q_j} = \boldsymbol{T_k} - \boldsymbol{V}_{r,k} \end{cases} \tag{4.3}$$

Where: $\boldsymbol{Q_i}$ and $\boldsymbol{Q_j}$ are the pair of predicted solutions; $\boldsymbol{T_k}$ is the kth individual in the randomly selected population from which the two predicted solutions are projected; $\boldsymbol{V}_{r,k}$ is one of the two opposite random vectors selected for $\boldsymbol{T_k}$.

The randomness of the selection on predicting directions and original solutions allows the entire population more opportunities to escape from the local optimality. Moreover, the dual predicting directions emphasize the search in opposite directions. It should be stressed again that all predicted solutions will participate in the competition with original solutions, where only the best solutions are preserved. If the original solutions are already located at the Pareto-optimal front, the predicted groups will not win during the competition. Otherwise, once a predicted solution survives after completion, any other worse individuals could move towards it for potentially better solutions.

4.1.2 Diversity Prediction Model

Similarly, as illustrated in Figure 4.2, if a group of solutions has better diversity than the other group, the direction vector in the decision space pointing from the worse group to the better group will be derived. Then, new groups will be predicted ahead of original groups in the predicted direction.

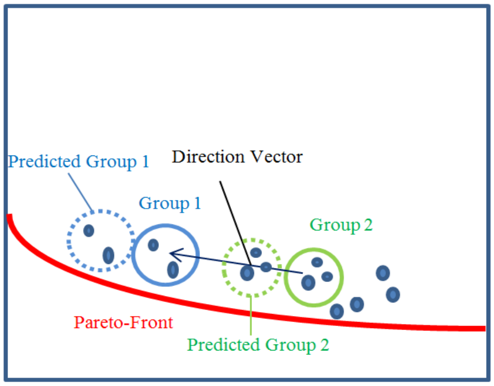


Figure 4.2. Diversity Prediction Model

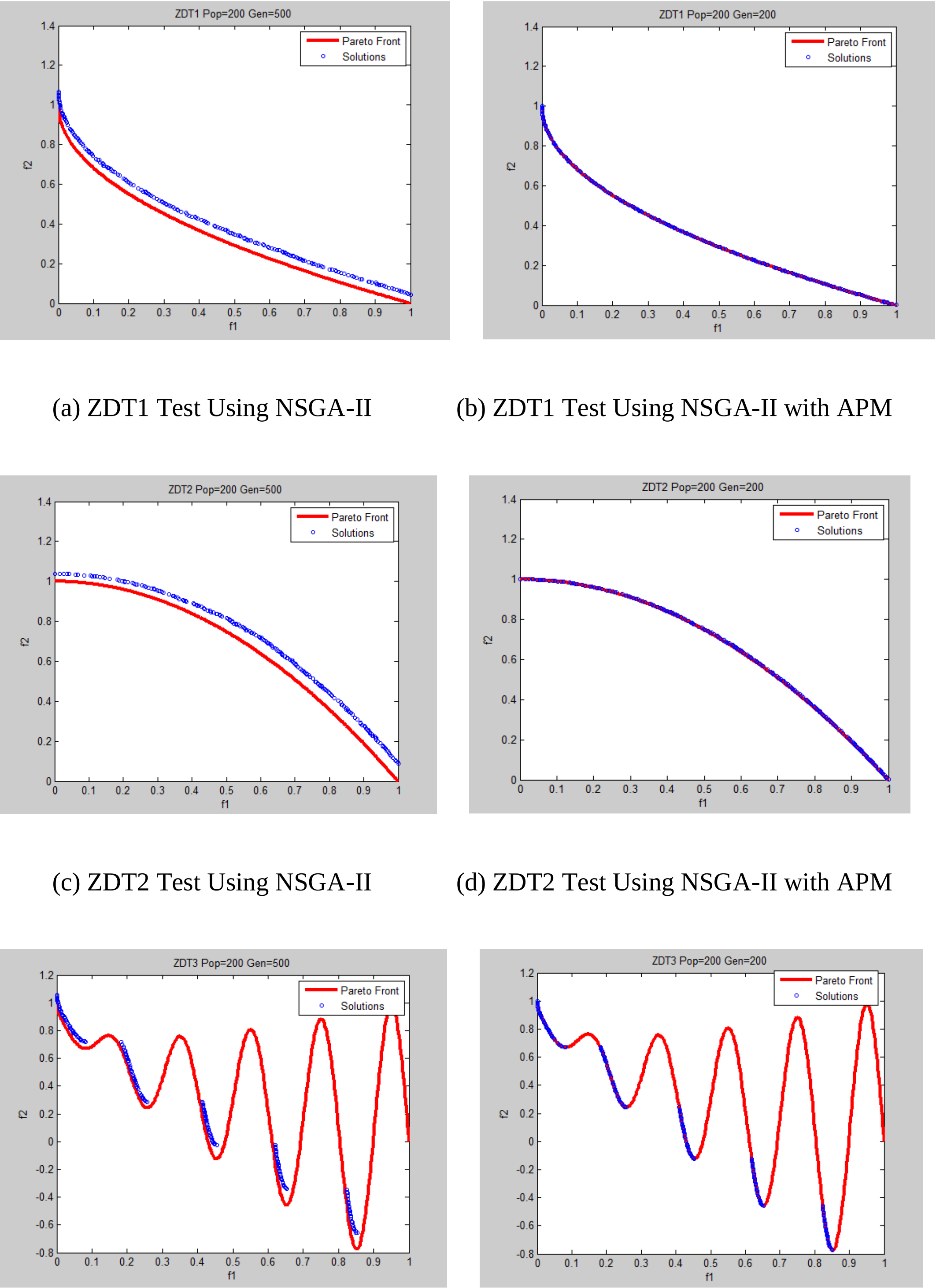
Suppose the Group 1 is the better half of population in terms of diversity, while Group 2 is the other half of worse diversity. Then the prediction vector and predicted population can be obtained using equation 4.1 and 4.2. If Group 1 and Group 2 are identical in the measurements of diversity, no further predictions are considered as necessary in this study.

In a summary, the NSGA II with APM is implemented as below:

```
Initialize Population;
Initial Non-dominated sorting;
For Each Generation
    Non-dominated sorting;
    Tournament selection;
    Generating Offspring;
    Non-dominated sorting on the extended population, including original population and the offspring;
    Population Update (Replacement);
    Convergence Prediction on the updated population;
    Diversity Prediction on the updated population;
    Non-dominated sorting on the integrated population, including updated population, predicted populations either for convergence or for diversity;
    Final Population Update (Replacement);
end
```

4.1.3. Performance of NSGA-II with APM on Benchmark Functions

In order to evaluate the performance of NSGA-II with APM, some benchmark functions with known optimality are needed. Five benchmark data from ZDT Test Suite, ZDT1, ZDT2, ZDT3, ZDT4 and ZDT6, are adopted in this study to compare the performance of NSGA-II with the performance of NSGA-II with APM (Zitzler et al., 2000). Each of the ZDT functions introduces different tangible difficulties in locating the true optima for optimization algorithms. This is the reason why they are widely accepted for performance evaluations. Particularly, each of the ZDT functions contains two optimization objectives, which is in accordance with the goal of this study. The comparison results are shown in Figure 4.3.



(a) ZDT1 Test Using NSGA-II

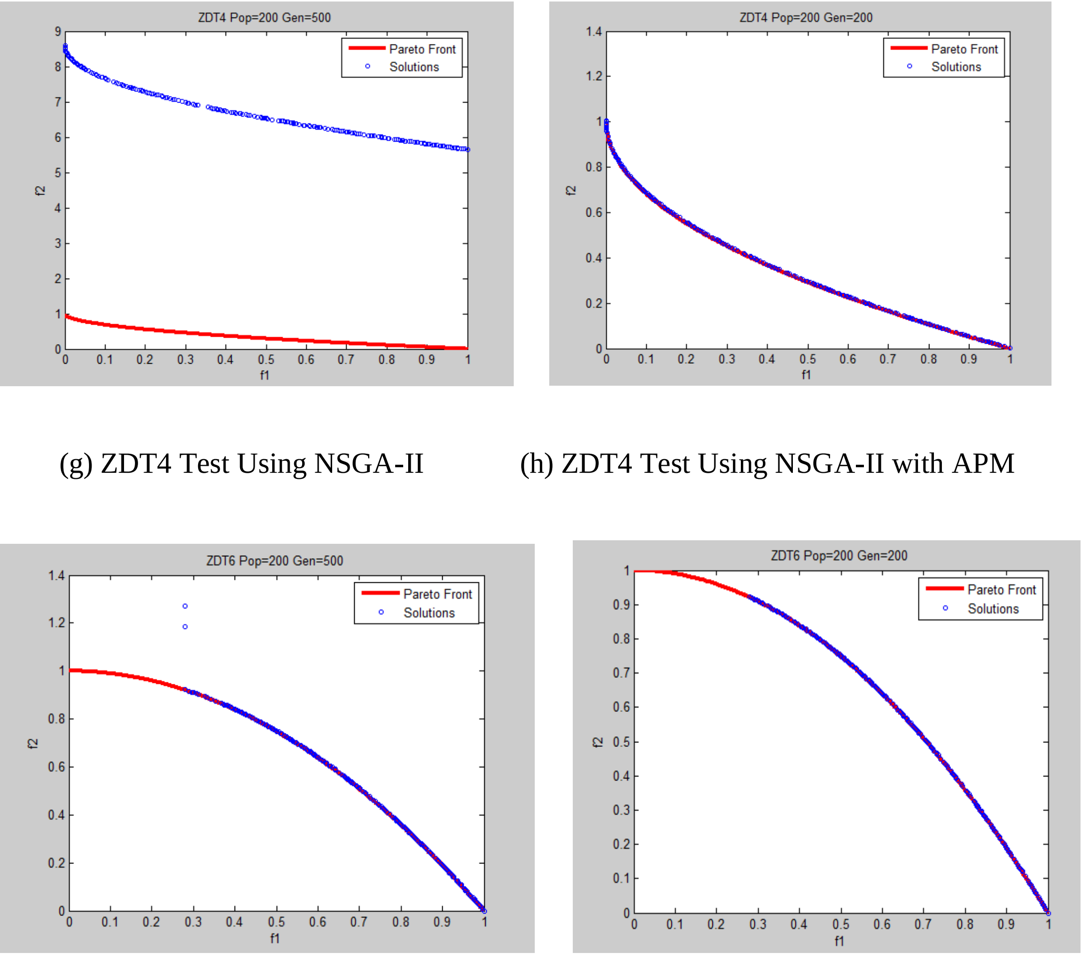
(b) ZDT1 Test Using NSGA-II with APM

(c) ZDT2 Test Using NSGA-II

(d) ZDT2 Test Using NSGA-II with APM

(e) ZDT3 Test Using NSGA-II

(f) ZDT3 Test Using NSGA-II with APM



(g) ZDT4 Test Using NSGA-II

(h) ZDT4 Test Using NSGA-II with APM

(i) ZDT6 Test Using NSGA-II

(j) ZDT6 Test Using NSGA-II with APM

Figure 4.3. Comparison between NSGA-II and NSGA-II with APM

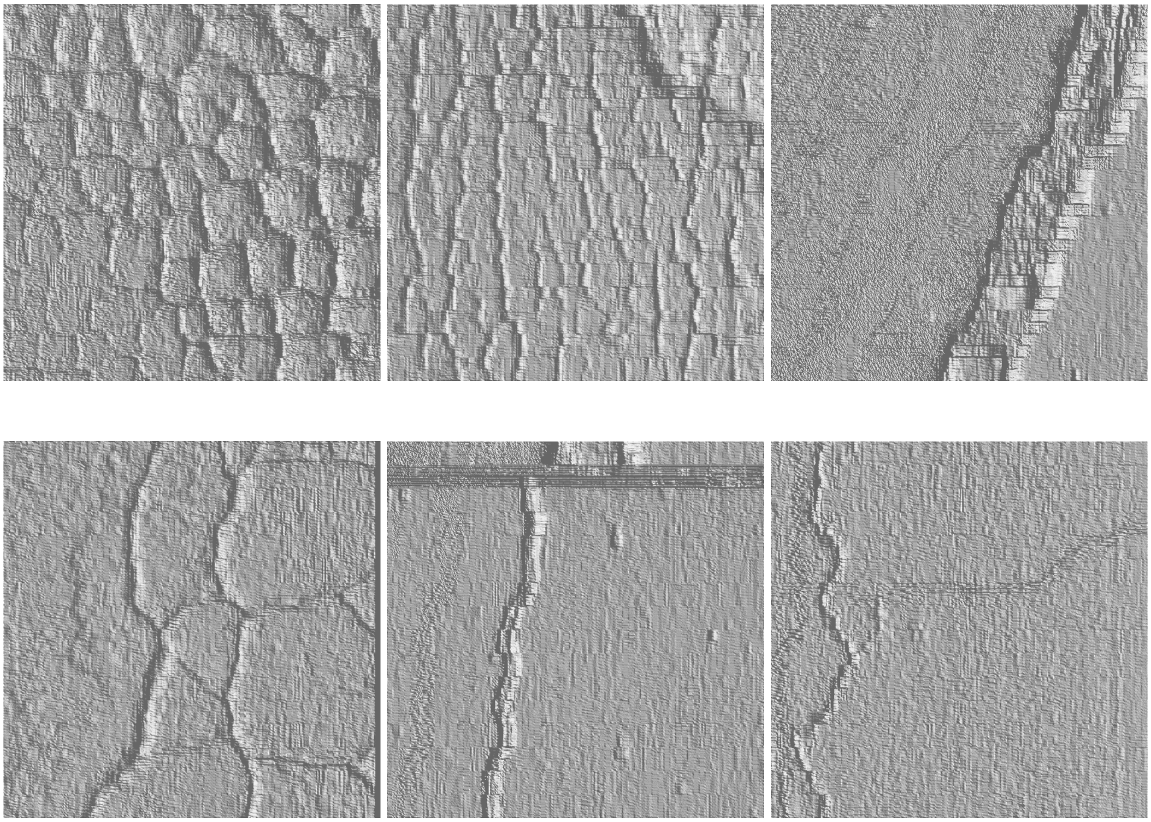
It is found from Figure 4.3 that NSGA-II with APM converges faster than NSGA-II, as NSGA-II with APM evolves with fewer generations but achieves much better results. In the meantime, NSGA-II with APM finds true optimal and diverse solutions successfully on all benchmark functions without getting stuck at local optima.

4.2 Optimization on Fully Automated Cracking Detection Algorithm

As mentioned before, maximization of Precision and maximization of Recall are the two optimization objectives in this study. For most cracking detection scenarios, Detection of high Precision often results in low Recall, while detection of high Recall yields low precision due to the introduced noises. Therefore, a robust cracking detection algorithm should maintain high Precision and high Recall simultaneously at an acceptable level.

4.2.1 Samples for Optimization

In order to increase overall performance of the fully automated detection subsystem on various pavement surfaces, the samples are collected from different pavement segments representing diverse surface textures and various cracking attributes. In this study, 100 pavement samples in size of 512×512 are involved in the evaluation of overall Precision and Recall. And some of them are illustrated in Figure 4.4. Ground Truth of cracking on all samples is marked manually such that Precision and Recall can be computed directly by a pixel-to-pixel comparison.



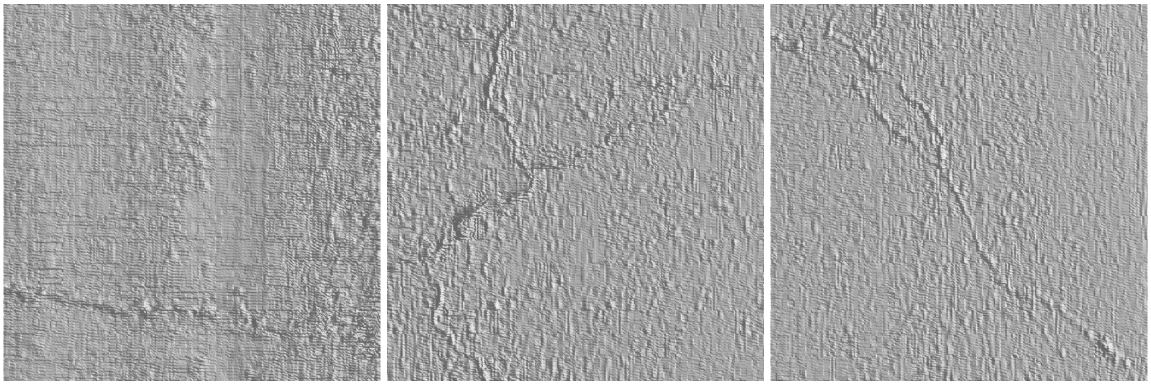


Figure 4.4. Representative Samples for Optimization

4.2.2 Decision Variables and Fitness Evaluation

The two major parameters of the fully automated detection algorithm, Transverse Lighting Angle and Longitudinal Lighting Angle, are selected as the decision variables for optimization. The range of each decision variable is determined by prior knowledge and listed in Table 4.1.

Table 4.1. Range of Decision Variables

Transverse Lighting		Longitudinal Lighting Angle	
Minimum	Maximum	Minimum	Maximum
40º	80º	40º	80º

The Fitness Evaluation Functions are the overall Precision and Recall of the detection on all samples. Denote TP_i as the number of true positive pixels in the detection result on ith sample, FP_i as the number of false positive pixels in the detection result on ith sample, and FN_i as the number of false negative pixels in the detection result on ith sample. Based on equations 3.18 and 3.19, the Fitness Evaluation Functions can be written as:

$$F_{precision} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FP_i)} \tag{4.4}$$

$$F_{recall} = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N}(TP_i + FN_i)} \tag{4.5}$$

Where: N is the number of samples.

4.2.3 Optimization Results

The optimized solutions are obtained by designating 100 individuals to evolve through 30 generations. Figure 4.5 shows the optimization result.

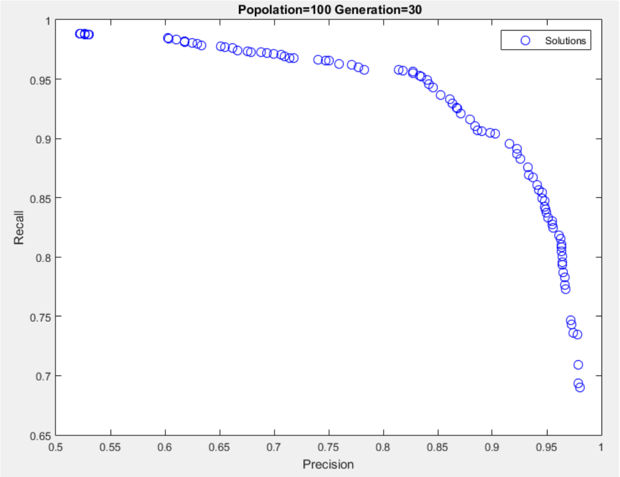


Figure 4.5. Optimization Result by NSGA II with APM

It could be found from Figure 4.5, the NSGA II with APM generates diverse solutions progressing toward the right upper corner. In order to utilize the optimization result in practice, a final solution which reachs high Pricison and high Recall simultaneously is selected from the diverse non-dominated solutions. As illlustrated in Figure 4.6, the Precision and Recall of the final solution are 90.29% and 90.37% respectively. In addition, the associated Transverse Lighting Angle is 57.46 degree, while the associated Longitudinal Lighting Angle is 66.99 degree.

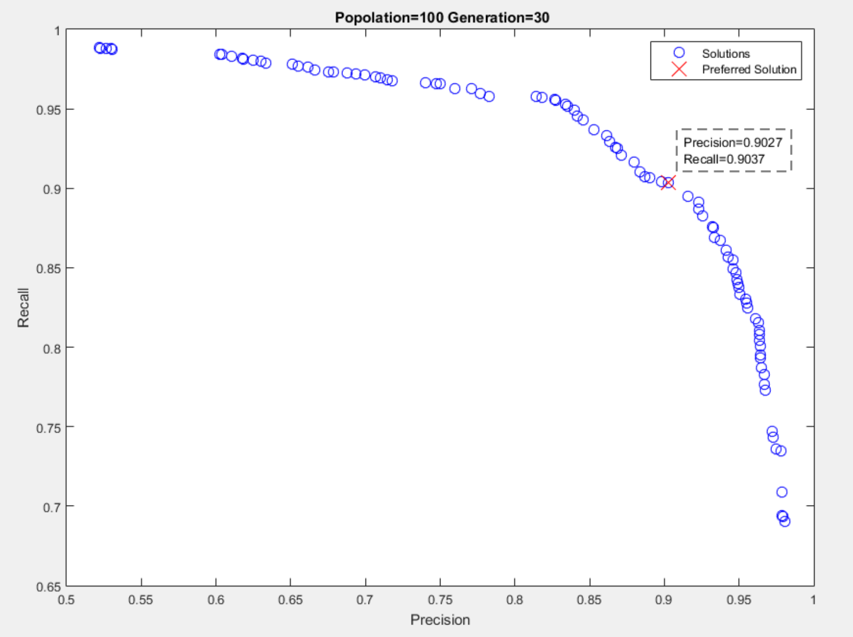


Figure 4.6. Preferred Solution

Using the optimized Transverse Lighting Angle 57 degree and Longitudinal Lighting Angle 67 degree, a comparison of detections on the same pavement segments discussed in Section 3.3 is conducted. Table 4.2 and Table 4.3 show the detection results using optimized lighting angles.

Table 4.2. Performance of Fully Automated Cracking Detection Subsystem on Pavement I Using Optimized Lighting Angles

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	91.94	93.94	0.27	0.20
2	20-40	94.76	94.02	0.17	0.20
3	40-60	91.15	92.94	0.43	0.33
4	60-80	88.49	85.46	0.28	0.36
5	80-100	88.70	85.23	0.32	0.43
6	100-120	89.51	89.95	0.27	0.26
7	120-140	92.11	85.98	0.15	0.29
8	140-160	91.60	85.79	0.14	0.25

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
9	160-180	92.36	95.23	0.21	0.13
10	180-200	90.52	93.64	0.28	0.18
11	200-220	87.96	83.49	0.36	0.52
12	220-240	95.05	85.36	0.15	0.48
13	240-260	91.93	87.95	0.29	0.45
14	260-280	91.49	87.74	0.27	0.41
15	280-300	95.01	95.83	0.15	0.12
Overall	0-300	**91.51**	**89.50**	0.25	0.31

Table 4.3. Performance of Fully Automated Cracking Detection Subsystem on Pavement II Using Optimized Lighting Angles

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	93.21	95.75	0.27	0.17
2	20-40	93.01	90.83	0.29	0.39
3	40-60	94.23	95.27	0.27	0.22
4	60-80	88.75	85.24	0.22	0.31
5	80-100	86.39	85.15	0.19	0.21
6	100-120	92.90	93.83	0.13	0.11
7	120-140	94.16	90.91	0.14	0.23
8	140-160	95.22	93.98	0.08	0.11
9	160-180	92.60	93.95	0.21	0.17
10	180-200	95.18	87.71	0.07	0.21
11	200-220	93.40	89.79	0.22	0.36
12	220-240	92.14	91.78	0.16	0.17

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
13	240-260	94.08	87.66	0.09	0.21
14	260-280	93.83	91.75	0.20	0.28
15	280-300	94.82	88.12	0.09	0.22
Overall	0-300	**92.93**	**90.78**	0.18	0.22

Comparing Table 4.2-4.3 with Table 3.2-3.3, it is found that the optimized lighting angles yield slight improvements on the overall Precision and Recall, even though they generate worse results in some reporting sections.

4.3 Summary

In a summary, NSGA II with APM is adopted in this chapter to optimize the primary parameters, of the fully automated detection algorithms based on two objectives: high Precision and high Recall. APM is proposed in this chapter as a complementary procedure to predict individual solutions that are likely to have better dominance and diversity in each generation. It is demonstrated in the comparison study on the five ZDT benchmark functions that NSGA II with APM converges much faster than NSGA II only. It is also shown that the NSGA II with APM eventually obtains diverse non-dominated solutions by evaluating the overall Precision and Recall of detections on the 100 representative cracking samples. Lastly, the optimized parameters yield general improvements in the detection on the same pavement segments discussed in Section 3.3.

The reliability of GA optimization greatly depends on the diversity of the samples selected for fitness evaluation. The collected samples in this chapter reflects different levels of severity and complexity of pavement cracks, which may ensure sufficiently high detection performance for many cases. However, due to complexity of pavement surface and the vast formulations of pavement cracking, the selected samples in this chapter are still inadequate to reflect various situations of pavement cracks in real world. Therefore, the performance of the proposed fully automated algorithms cannot be predicted and guaranteed under a larger scope of pavement data. In order to ensure a higher level of confidence, an interactive detection subsystem which supplies the mechanism to retrieve pavement cracks with much higher level of confidence will be discussed in next chapter.

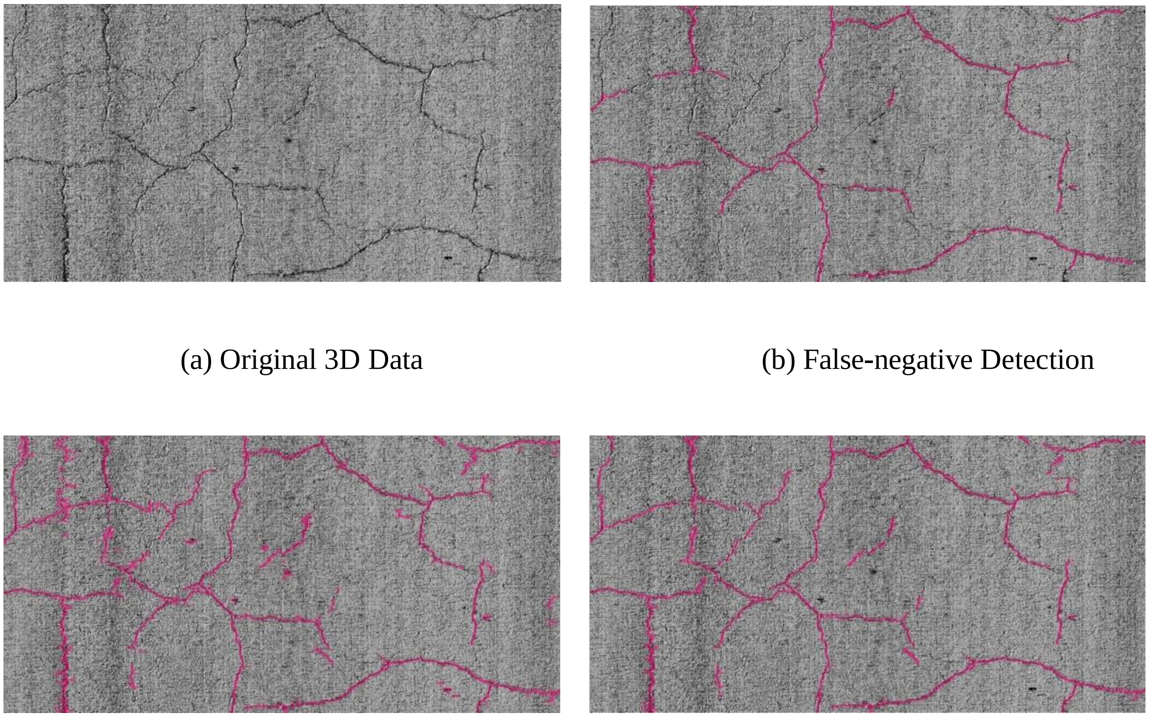
Chapter 5 INTERACTIVE CRACKING DETECTION SUBSYSTEM

Various private and public endeavors in the last three decades on worldwide basis vastly underestimated the challenges and difficulties of developing fully automated cracking survey solutions for field production. A common problem for the current fully automated cracking detection algorithms is that consistently high detection accuracy is not guaranteed for large number of pavement sections due to unpredictable uncertainties and variations on pavement surfaces. Developers of the current automated cracking detection algorithms did not pay sufficient attention to data operator's involvement in processing. In other words, the current automated algorithms have not provided a systematic methodology for users to improve the detection performance based on their experience and observations.

An interactive cracking detection algorithm, which can substantially improve its accuracy according to operator' feedback, is proposed in this chapter. Based on the interactive detection algorithm, a computer program is also developed to implement two-level interactive detection. The bottom level is automated and can be used to find the majority of cracking. The top level is assisted by the users and can be applied for semi-automated refinement. In particular, the top level can be used to find missed cracks or delete noises within any user-defined region.

5.1 3D Algorithms with Interactive Cracking Detection

At pixel level, “ground truth” of cracking observed by different crack analysis persons may not be the same. Particularly, different observers might not have the same agreements on the presence and severity of fine cracks. For example, a 2-pixel wide crack in reality may seem to have dissimilar widths (i.e. 1-pixel wide or 3-pixel wide) for different observers. However, with respect to individual’s manual processing for cracking, the research categorizes three scenarios of detection performance. The detection results could be False-positive, False-negative or Acceptable. False-positive Detection means the percentage of noises exceeds acceptance level of the observer. False-negative Detection means the percentage of undetected cracks is higher than the acceptance level of the observer. Finally, the percentage of noises and the percentage of undetected cracks satisfy the performance requirement of analysis when the detection is acceptable. Figure 5.1 shows the three performance scenarios as basis of the proposed interactive algorithm.



(a) Original 3D Data

(b) False-negative Detection

(c) False-positive Detection

(d) Acceptable Detection

Figure 5.1. Detection Scenarios

The appropriate interaction between the manual operator and the interactive algorithm is that the algorithm should improve itself with high level of success when the operator makes decisions on the detection result: false-negative or false-positive. In other words, the interactive algorithm should perform more negatively if the result is false-positive. On the other hand, it should be more positive if the result is regarded as false-negative. The response of an interactive detector to the observed results can be expressed as:

$$\begin{cases} m^* \to m^* \oplus, & if\ m^*(\boldsymbol{I}) = \boldsymbol{I}_n \\ m^* \to m^* \ominus, & if\ m^*(\boldsymbol{I}) = \boldsymbol{I}_p \end{cases} \tag{5.1}$$

Where: m^*-Interactive detector; $\oplus$ -Adjustment of interactive detector to be more positive; $\ominus$-Adjustement of interactive detector to be more negative; $\boldsymbol{I}$-Input Image; $\boldsymbol{I}_n$-Output image with false-negative results; $\boldsymbol{I}_p$ –Output image with false-positive results.

The biggest challenge for an interactive algorithm is how to guarantee the high level of correct response to the manual operator's feedback. For example, an algorithm with many parameters would be lack of certainties in adjusting itself efficiently due to the complexity of adjusting multiple parameters at the same time to improve performance. Therefore, it is desired that an interactive algorithm could have as few tunable parameters as possible. Ideally, the interactive algorithm could only have one sensitive parameter so that it can have more straightforward interaction with observers.

5.1.1 Design of Single Primary Parameter

The proposed interactive algorithm is to extract cracks and suppress noises effectively while having only one sensitive parameter that can be used for desirable interaction with manual operators. 3D pavement surface data represent the height information of pavement surface. The cracking area is always lower than the local non-cracking area. In terms of image processing, the cracking area in a rendered 3D image is darker than the local background, as illustrated in Figure 10. The term Minimal Contrast is introduced in this paper as the primary parameter of the proposed interactive algorithm. Given a local neighborhood $\boldsymbol{A}$, if the potential cracking area $\boldsymbol{C}$ within this neighborhood has an average no greater than the average of non-cracking area $\boldsymbol{B}$ within the neighborhood minus Minimal Contrast m_c, then the area $\boldsymbol{C}$ will be recognized as cracking area, which can be expressed as below.

$$\mathbf{C} = \begin{cases} \mathbf{0} & if\ \overline{p_c} \le \overline{p_b} - m_c \\ \mathbf{1} & if\ \overline{p_c} > \overline{p_b} - m_c \end{cases} \quad \text{subject to: } \begin{cases} \boldsymbol{B} \cup \boldsymbol{C} = \boldsymbol{A} \\ \boldsymbol{B} \cap \boldsymbol{C} = \boldsymbol{\phi} \end{cases} \tag{5.2}$$

Where: $\mathbf{0}$- Cracking Area; $\mathbf{1}$-Non-cracking Area; $\overline{p_c}$-Average of Area **C**; $\overline{p_b}$-Average of Area **B**; m_c – Minimal Contrast; $\boldsymbol{\phi}$ - Empty Set.

For each 3D image, the original 3D data is normalized to [0, 255], which is the range of a gray-scale image. Thus, the possible range of Minimal Contrast m_c is also from 0 to 255. When

Minimal Contrast is set as $m_c = 0$, all areas no brighter than the local background will be detected as cracking areas. On the contrary, almost no areas will be recognized as cracking areas when Minimal Contrast reaches up to 255. For any given 3D image, there could be an optimum Minimal Contrast that can produce acceptable detection result. The manual operator can tell the algorithm if the detection is false-positive or false-negative to allow the algorithm to improve itself by tuning the parameter, until the result is satisfying or the best performance has been achieved. As manifested by equation (5.2), Minimal Contrast m_c is a straightforward parameter to be used for interaction. If the detection result is false-positive, the Minimal Contrast will be increased for more negative detection. Otherwise, it will be decreased for more positive detection.

5.1.2 Line Scanning on 3D Surface

The first procedure of proposed algorithms is to implement the Line Scanning method once the Minimal Contrast has been set to identify potential crack locations based on the 3D information of pavement surface height. This Line Scanning method is not the same as that used in line scan digital cameras which are based on 2D based image acquisition. In this paper Line Scanning is conducted at two directions: transverse direction and longitudinal direction. All local minima and maxima are marked from a transverse or longitudinal line based on the following specifications.

$$\begin{cases} \boldsymbol{K}_{i,min} = \{\boldsymbol{I}(i,y) | I(i,y-1) > I(i,y), I(i,y+1) > I(i,y)\} \\ \boldsymbol{K}_{i,max} = \{\boldsymbol{I}(i,y) | I(i,y-1) < I(i,y), I(i,y+1) < I(i,y)\} \\ \boldsymbol{K}_{j,min}^{T} = \{\boldsymbol{I}(x,j) | I(x-1,j) > I(x,j), I(x+1,j) > I(x,j)\} \\ \boldsymbol{K}_{j,max}^{T} = \{\boldsymbol{I}(x,j) | I(x-1,j) < I(x,j), I(x+1,j) < I(x,j)\} \end{cases} \tag{5.3}$$

Where: $\boldsymbol{K}_{i,min}$-local minima at i^{th} longitudinal image line; $\boldsymbol{K}_{i,max}$-local maxima at i^{th} longitudinal image line; $\boldsymbol{K}_{j,min}^{T}$- local minima at j^{th} transverse image line; $\boldsymbol{K}_{j,max}^{T}$- local maxima at j^{th} transverse image line; $\boldsymbol{I}(i,y)$-Image pixels at i^{th} longitudinal image line; $\boldsymbol{I}(x,j)$-Image pixels at j^{th} transverse image line.

In this chapter, the total of successive pixels that start at a local minimum and outreach to the two closest local maxima located at the two sides of the local minimum is treated as a potential cracking profile, either being longitudinal or transverse:

$$\begin{cases} \mathbf{C}_i(m)' = \{\boldsymbol{I}(i,y) | y_{max}^{(i,m)} < y < \bar{y}_{max}^{(i,m)}\} \\ \mathbf{C}_j^T(m)' = \{\boldsymbol{I}(x,j) | x_{max}^{(j,m)} < x < \bar{x}_{max}^{(j,m)}\} \end{cases} \tag{5.4}$$

Subject to:

$$\begin{cases} y_{max}^{(i,m)} < y_{min}^{(i,m)} < \bar{y}_{max}^{(i,m)} \\ x_{max}^{(j,m)} < x_{min}^{(j,m)} < \bar{x}_{max}^{(j,m)} \\ \left(i, y_{min}^{(i,m)}\right) \epsilon \boldsymbol{K}_{i,min}, \left(x_{min}^{(j,m)}, j\right) \epsilon \boldsymbol{K}_{j,min}^T \\ \left(i, y_{max}^{(i,m)}\right), \left(i, \bar{y}_{max}^{(i,m)}\right) \epsilon \boldsymbol{K}_{i,max} \\ \left(x_{max}^{(j,m)}, j\right), (\bar{x}_{max}^{(j,m)}, j) \epsilon \boldsymbol{K}_{j,max}^T \end{cases}$$

Where: $\mathbf{C}_i(m)'$- m^{th} longitudinal potential cracking profile at i^{th} longitudinal line, including the m^{th} local minimum at i^{th} longitudinal line; $\mathbf{C}_j^T(m)'$- m^{th} transverse potential cracking profile at j^{th} transverse line, including the m^{th} local minimum at j^{th} transverse line; $\left(i, y_{min}^{(i,m)}\right)$– m^{th} local minimum at i^{th} longitudinal image line; $\left(x_{min}^{(j,m)}, j\right)$-$m^{th}$ local minimum at j^{th} transverse image line; $\left(i, y_{max}^{(i,m)}\right), \left(i, \bar{y}_{max}^{(i,m)}\right)$- Two local maxima at i^{th} longitudinal line that are closest to $\left(i, y_{min}^{(i,m)}\right)$; $\left(x_{max}^{(j,m)}, j\right), (\bar{x}_{max}^{(j,m)}, j)$ - Two local maxima at j^{th} transverse image line that are closest to $\left(x_{min}^{(j,m)}, j\right)$.

Figure 5.2 shows an example of potential cracking profile. By referring to equation (5.2), potential cracking profiles satisfying the first condition will be kept for latter analyses; otherwise they will be eliminated through Line Scanning. The local background used for comparison with a potential cracking profile is a set of neighboring pixels situated besides the two local maxima, in other words the two local lips, which can be expressed as below.

$$\begin{cases} \boldsymbol{B_i}(m) = \{\boldsymbol{I}(i,y) | y_{max}^{(i,m)} - \sigma \le y \le y_{max}^{(i,m)} \ or \ \bar{y}_{max}^{(i,m)} \le y \le \bar{y}_{max}^{(i,m)} + \sigma\} \\ \boldsymbol{B_j^T}(m) = \{\boldsymbol{I}(x,j) | x_{max}^{(j,m)} - \sigma \le x \le x_{max}^{(j,m)} \ or \ \bar{x}_{max}^{(j,m)} \le x \le \bar{x}_{max}^{(j,m)} + \sigma\} \end{cases} \tag{5.5}$$

Where: $\boldsymbol{B_i}(m)$-Background used for comparison with m^{th} potential cracking profile at i^{th} longitudinal line; $\boldsymbol{B_j^T}(m)$-Background used for comparison with m^{th} potential cracking profile at j^{th} transverse line; σ-the size of background used for comparison ($\sigma = 0$ adopted in this study).

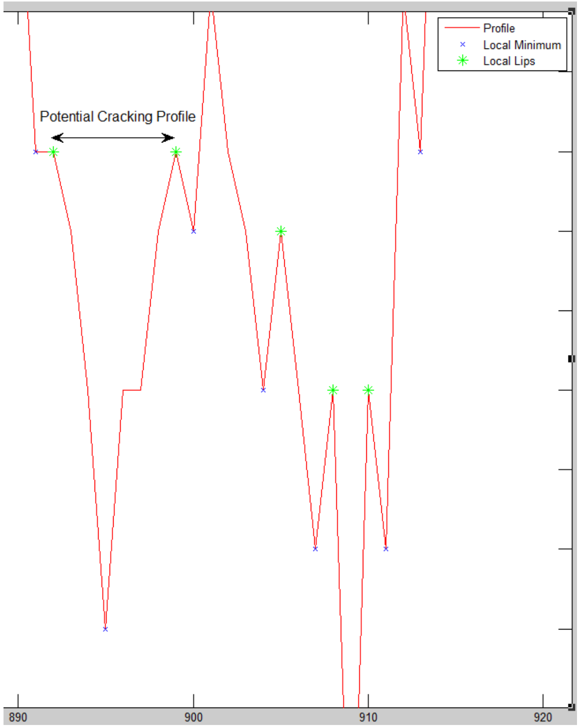
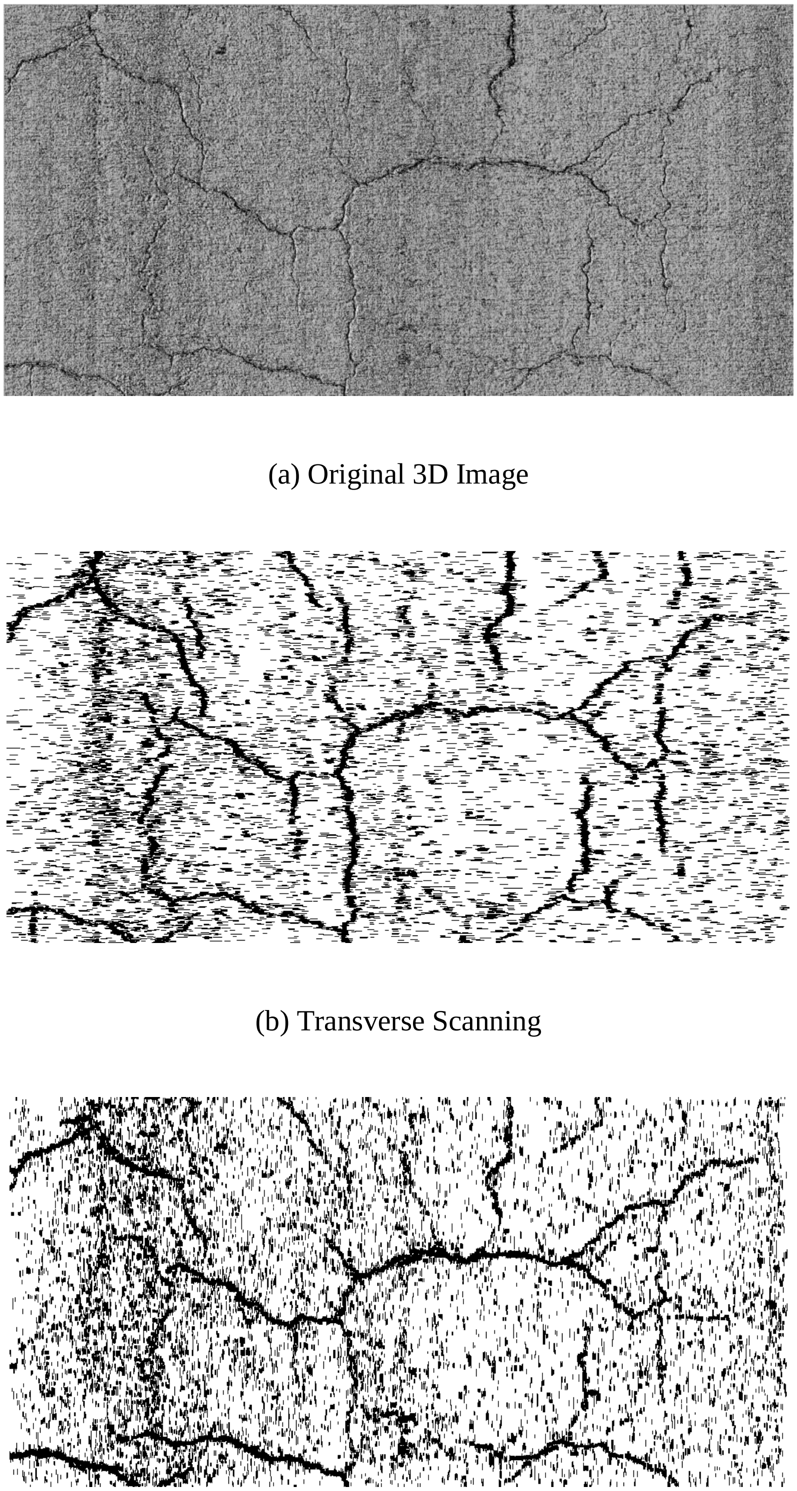


Figure 5.2. Potential Cracking Profile

In general, Transverse Scanning is mainly responsible for finding longitudinal-dominated cracks, while Longitudinal Scanning is applied mainly for transverse-dominated cracks. Certainly, scanning could be implemented at multiple directions in order to find more diagonal cracks. However for digital image, the processing time greatly increases when irregular directions are considered. And it has been tested that Line Scanning in both transverse and longitudinal directions is sufficient to locate all cracks with proper Minimal Contrast in most cases. Figure 5.3 shows an example of Line Scanning.



(a) Original 3D Image

(b) Transverse Scanning

(c) Longitudinal Scanning

Figure 5.3. Line Scanning

5.1.3 Thinning

Since the cracking profiles located by Line Scanning are lines starting from the local minima to the local lips at two sides, they may not necessarily be the true cracking profiles representing actual cracking widths. Therefore, Thinning is needed to find true profiles based on the results by Line Scanning. There are many existing algorithms for thinning. However, the main objective of current thinning algorithms is to skeletonize a thick pattern to a thin structure with a specific fixed width, particularly one-pixel wide for many cases. As a matter of fact, the width of a crack varies along with its extension. Therefore, each potential cracking profile should be examined so as to retain the correct width distribution of a crack. This paper proposes a different thinning procedure to reduce redundant pixels from Line Scanning results through the examination on all potential profiles maintained after Line Scanning. It is assumed that the cracking profiles terminate at pixels where image intensity change sharply. To be more specific, a cracking profile is supposed to start at a local minimum and outreach to the two winged pixels where local maximal gradients present.

Similarly, Thinning is also implemented at both transverse and longitudinal directions. Since transverse thinning and longitudinal thinning are conducted separately, the image gradients are computed separately and only reflect intensity changes in transverse direction or in longitudinal direction. That is:

$$\begin{cases} G(x,y) = \left|\frac{I(x,y)-I(x,y-1)}{y-(y-1)}\right| = |I(x,y) - I(x,y-1)| \\ G^T(x,y) = \left|\frac{I(x,y)-I(x-1,y)}{x-(x-1)}\right| = |I(x,y) - I(x-1,y)| \end{cases} \tag{5.6}$$

Where: $G(x,y)$-Longitudinal gradient at pixel (x,y); $G^T(x,y)$ -Transverse gradients at pixel (x,y);

It should be noticed that the gradients only need to be computed at potential cracking pixels identified by Line Scanning. According to equation (5.4) and the assumption on a true cracking profile, the thinned potential cracking profiles can be described as:

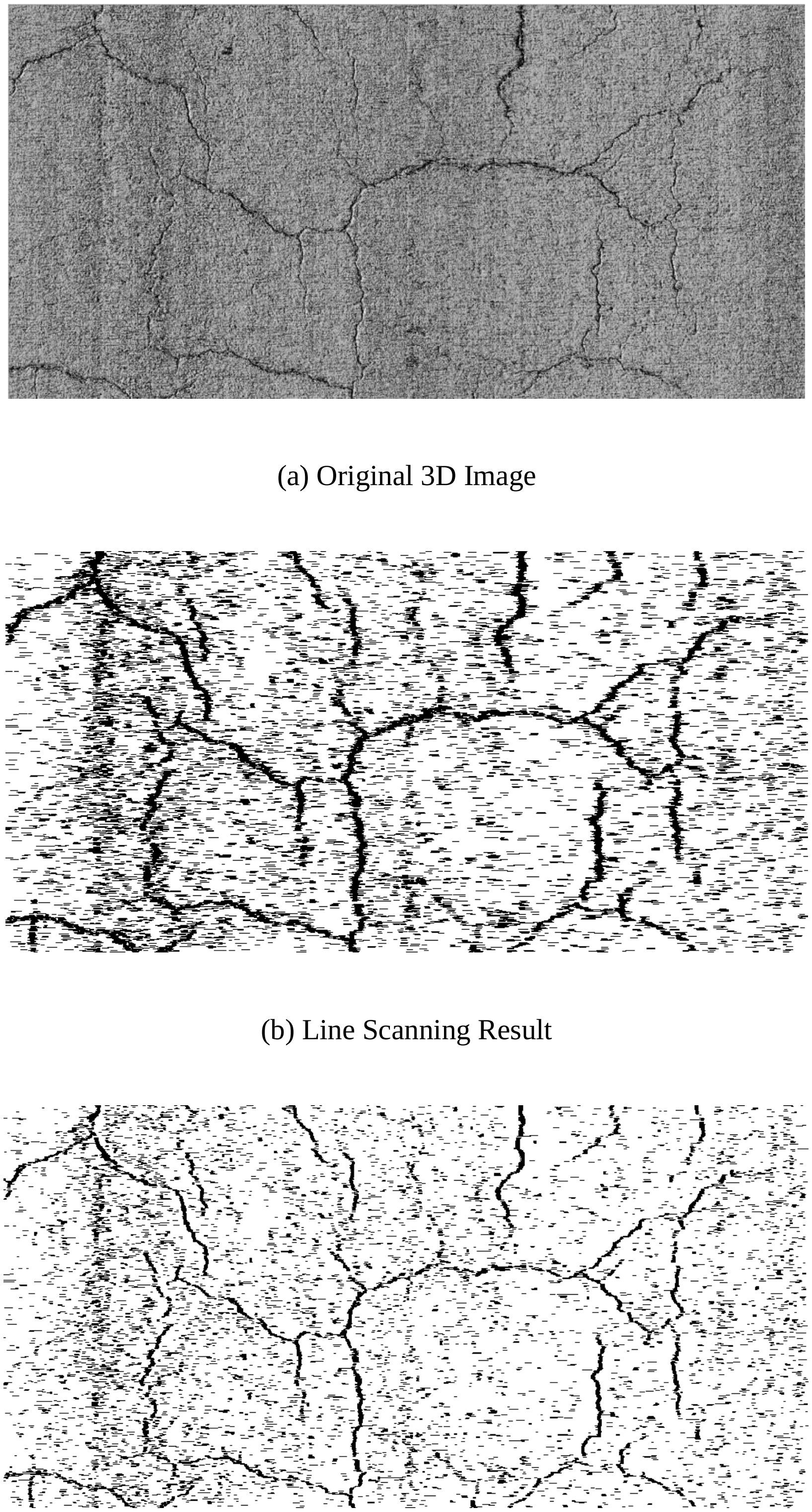
$$\begin{cases} \mathbf{C}_i(m) = \{\boldsymbol{I}(i,y)|\ y_{G,max}^{(i,m)} < y < \bar{y}_{G,max}^{(i,m)}\} \\ \mathbf{C}_j^T(m) = \{\boldsymbol{I}(x,j)|\ x_{G,max}^{(j,m)} < x < \bar{x}_{G,max}^{(j,m)}\} \end{cases} \tag{5.7}$$

Subject to:

$$\begin{cases} y_{max}^{(i,m)} < y_{G,max}^{(i,m)} \leq y_{min}^{(i,m)} < \bar{y}_{G,max}^{(i,m)} < \bar{y}_{max}^{(i,m)} \\ x_{max}^{(j,m)} < x_{G,max}^{(j,m)} < x_{min}^{(j,m)} < \bar{x}_{G,max}^{(j,m)} < \bar{x}_{max}^{(j,m)} \\ \forall y_0 \epsilon [y_{max}^{(i,m)}, y_{min}^{(i,m)}]: G(i,y_0) \leq G\left(i, y_{G,max}^{(i,m)}\right) \\ \forall y_0 \epsilon \left[y_{min}^{(i,m)}, \bar{y}_{max}^{(i,m)}\right]: G(i,y_0) \leq G\left(i, \bar{y}_{G,max}^{(i,m)}\right) \\ \forall x_0 \epsilon \left[x_{max}^{(j,m)}, x_{min}^{(j,m)}\right]: G^T(x_0,j) \leq G^T\left(x_{G,max}^{(j,m)}, j\right) \\ \forall x_0 \epsilon \left[x_{min}^{(j,m)}, \bar{x}_{max}^{(j,m)}\right]: G^T(x_0,j) \leq G^T\left(\bar{x}_{G,max}^{(j,m)}, j\right) \end{cases}$$

Where: $\mathbf{C}_i(m)$-m^{th} thinned potential cracking profile at i^{th} longitudinal line; $\mathbf{C}_j^T(m)$-m^{th} thinned potential cracking profile at j^{th} transverse line; $\left(i, y_{G,max}^{(i,m)}\right)$, $\left(i, \bar{y}_{G,max}^{(i,m)}\right)$- Two pixels of maximal longitudinal gradients located at the two sides of $\left(i, y_{min}^{(i,m)}\right)$; $\left(x_{G,max}^{(j,m)}, j\right)$, $\left(\bar{x}_{G,max}^{(j,m)}, j\right)$- Two pixels of maximal transverse gradients located at the two sides of $\left(x_{min}^{(j,m)}, j\right)$.

Figure 5.4 illustrates improvements achieved by Thinning on locating cracking profiles.



(a) Original 3D Image

(b) Line Scanning Result

(c) Thinning Based on Line Scanning Result

Figure 5.4. Thinning

5.1.4 Clustering and Connecting

For the proposed interactive algorithm, Clustering is applied to label collected pixels in binary images after Thinning has been applied to 3D pavement surface data so that further analyses can be implemented on cracking clusters rather than on scanned cracking profiles. The Blob Exaction Algorithm (Shapiro and Stockman, 2001), which is a common technique and discussed in section 3.2.2, is also adopted in this chapter to label connected pixels after Thinning.

It has been observed that some fine cracks do not have desirable continuity in original images, as illustrated in Figure 5.5.

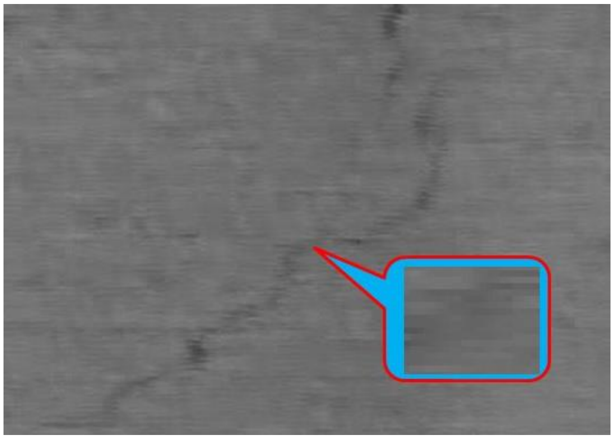


Figure 5.5. Discontinuity of Fine Crack

Generally, the discontinuity problem can have many reasons, such as inadequate data resolution, discontinuity of real cracks or even noises from data acquisition system. In addition, since the original image is assigned with single Minimal Contrast value, some true cracking profiles where local contrasts are smaller than the Minimal Contrast will thus be ignored during the process of Line Scanning, which also results in discontinuity problem.

In order to solve discontinuity problem, a specific algorithm is developed in this study to look for connectable neighbors at the ends of a potential cracking cluster. First of all, the connecting algorithm will label branches for each potential cracking cluster, as illustrated in Fig.5.6.

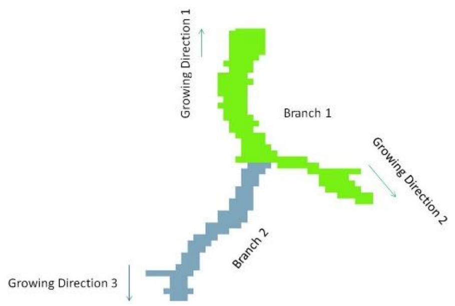


Figure 5.6. Braches of a Cracking Cluster

Secondly, the algorithm will determine the terminals of each branch where no other branch is connected. Then, the growing directions of all unconnected terminals will be derived based on the following equation.

$$\boldsymbol{D}_{k,n} = \frac{\sum_{i=0}^{Nt-1}[\boldsymbol{Pt}_n^{(k)}(end-i)-\boldsymbol{Pt}_n^{(k)}(end-i-1)]}{N_t} \tag{5.8}$$

Where: $\boldsymbol{D}_{k,n}$- Growing direction of k^{th} branch at n^{th} terminal; $\boldsymbol{Pt}_n^{(k)}(end-i)$ – Center point of i^{th} foregoing profile from the terminal; N_t- Number of profiles used to compute growing direction.

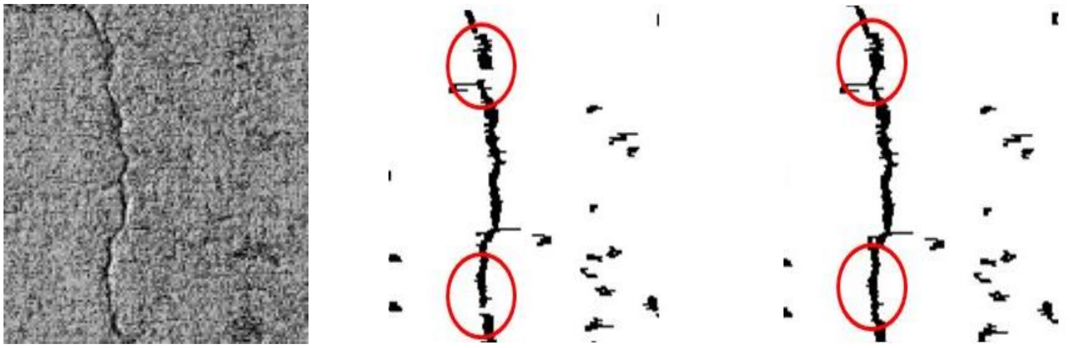
A small but adequate value of N_t will emphasize the local tendency of the growing direction. $N_t = 5$ is used in this paper based on testing on numerous samples. With the growing direction of a branch terminal, a virtual growing of the branch starts at the terminal. Virtual points are first predicted along the growing direction as:

$$\boldsymbol{Pv}_n^{(k)}(end+i) = \boldsymbol{Pt}_n^{(k)}(end) + i\widehat{\boldsymbol{D}}_{k,n},\ 0 < i \leq N_v \tag{5.9}$$

Where: $\boldsymbol{Pv}_n^{(k)}(end+i)$- i^{th} point predicted in the growing direction starting from the n^{th} terminal of k^{th} branch; $\widehat{\boldsymbol{D}}_{k,n}$-Normalized growing direction; N_v-Number of predicted points.

For i^{th} predicted point, the original profile, which is closest to the predicted point and scanned by Line Scanning without comparing to background for noise filtering, will be thinned by the aforementioned thinning algorithm and then adopted as the i^{th} predicted profile. Afterwards, the algorithm will implement a recursive process of virtual growing until one of the following stopping criteria is satisfied: 1) i^{th} predicted profile has been connected with other branches; 2) i^{th} predicted profile has lost connection to $(i-1)^{th}$predicted profile; 3) $i = N_v$.

If the virtual growing is successful in connecting the examined branch into other branches, all predicted profiles will be added to the detection result, otherwise they will be discarded. The number of predicted points N_v determines positivity of the connecting algorithm. As a result, it should be assigned with a small value to avoid false-positive errors. $N_v = 5$ is adopted in this paper. Meanwhile, it should be addressed that predicted profiles for longitudinal-dominated branches are all transverse, while predicted profiles for transverse-dominated braches are all longitudinal. Figure 5.7 shows an example of connecting two parts of the same crack.



(a) Original 3D Image (b) Disconnected Cracking (c) Connected Cracking

Figure 5.7. Connecting

5.1.5 Neighborhood Analysis

Minimal Contrast is employed in Line Scanning for finding potential cracking profiles. However, after clustering, it can be applied for recognition of potential cracking clusters. For Neighborhood

Analysis, each cluster is compared with the local background around the cluster being examined, as illustrated in Fig.5.8.

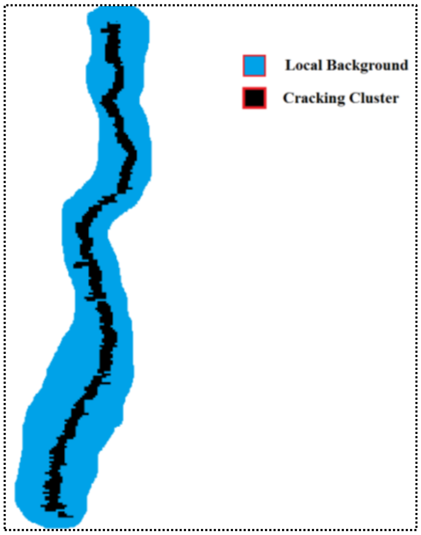


Figure 5.8. Local Neighborhood for a Cracking Cluster

Denote the longitudinal-dominated cracking cluster and transverse-dominated cracking cluster as:

$$\begin{cases} \boldsymbol{C}^{(k)} = \left\{\boldsymbol{C}_{(1)}, \boldsymbol{C}_{(2)}, \dots, \boldsymbol{C}_{(N)}\right\} \\ \boldsymbol{C}^{T,(k)} = \{\boldsymbol{C}^{T}_{(1)}, \boldsymbol{C}^{T}_{(2)}, \dots, \boldsymbol{C}^{T}_{(N^T)}\} \end{cases} \tag{5.10}$$

Where: $\boldsymbol{C}^{(k)}$-k^{th} longitudinal-dominated cracking cluster; $\boldsymbol{C}^{T,(k)}$- k^{th} transverse-dominated cracking cluster; $\boldsymbol{C}_{(i)}$-i^{th} longitudinal cracking profile in $\boldsymbol{C}^{(k)}$; $\boldsymbol{C}^{T}_{(i)}$-i^{th} transverse cracking profile in $\boldsymbol{C}^{T,(k)}$; N-Number of longitudinal cracking profiles in $\boldsymbol{C}^{(k)}$; N^T-Number of transverse cracking profiles in $\boldsymbol{C}^{T,(k)}$.

Then, the local background used for Neighborhood Analysis is then defined in the study as:

$$\begin{cases} \boldsymbol{B}^{(k)} = \left\{\boldsymbol{B}_{(1)}, \boldsymbol{B}_{(2)}, \dots, \boldsymbol{B}_{(N)}\right\} \\ \boldsymbol{B}^{T,(k)} = \{\boldsymbol{B}^{T}_{(1)}, \boldsymbol{B}^{T}_{(2)}, \dots, \boldsymbol{B}^{T}_{(N^T)}\} \end{cases} \tag{5.11}$$

Subject to:

$$\begin{cases} \boldsymbol{B}_{(i)} = \{\boldsymbol{I}(x_i, y) | y_i - \varepsilon w_i \leq y < y_i \ or \ \bar{y}_i < y \leq \bar{y}_i + \varepsilon w_i, \ y_i \leq \ \bar{y}_i\} \\ \boldsymbol{B}_{(i)}^T = \{\boldsymbol{I}(x, y_i^T) | x_i^T - \varepsilon w_i^T \leq x < x_i^T \ or \ \bar{x}_i^T < x \leq \bar{x}_i^T + \varepsilon w_i^T, \ x_i^T \leq \bar{x}_i^T\} \end{cases}$$

Where: $\boldsymbol{B}^{(k)}$-Local background around $\boldsymbol{C}^{(k)}$; $\boldsymbol{B}^{T,(k)}$-Local background around $\boldsymbol{C}^{T,(k)}$; $\boldsymbol{B}_{(i)}$-i^{th} longitudinal background profile in $\boldsymbol{B}^{(k)}$; $\boldsymbol{B}_{(i)}^T$-i^{th} transverse background profile in $\boldsymbol{B}^{T,(k)}$; (x_i, y_i) , $(x_i, \bar{y}_i)$- Two ends of longitudinal cracking profile $\boldsymbol{C}_{(i)}$; (x_i^T, y_i^T) , $(\bar{x}_i^T, y_i^T)$- Two ends of transverse cracking profile $\boldsymbol{C}_{(i)}^T$; w_i-Width of longitudinal cracking profile $\boldsymbol{C}_{(i)}$; w_i^T-Width of transverse cracking profile $\boldsymbol{C}_{(i)}^T$; ε-Parameter used to determine the size of local background.

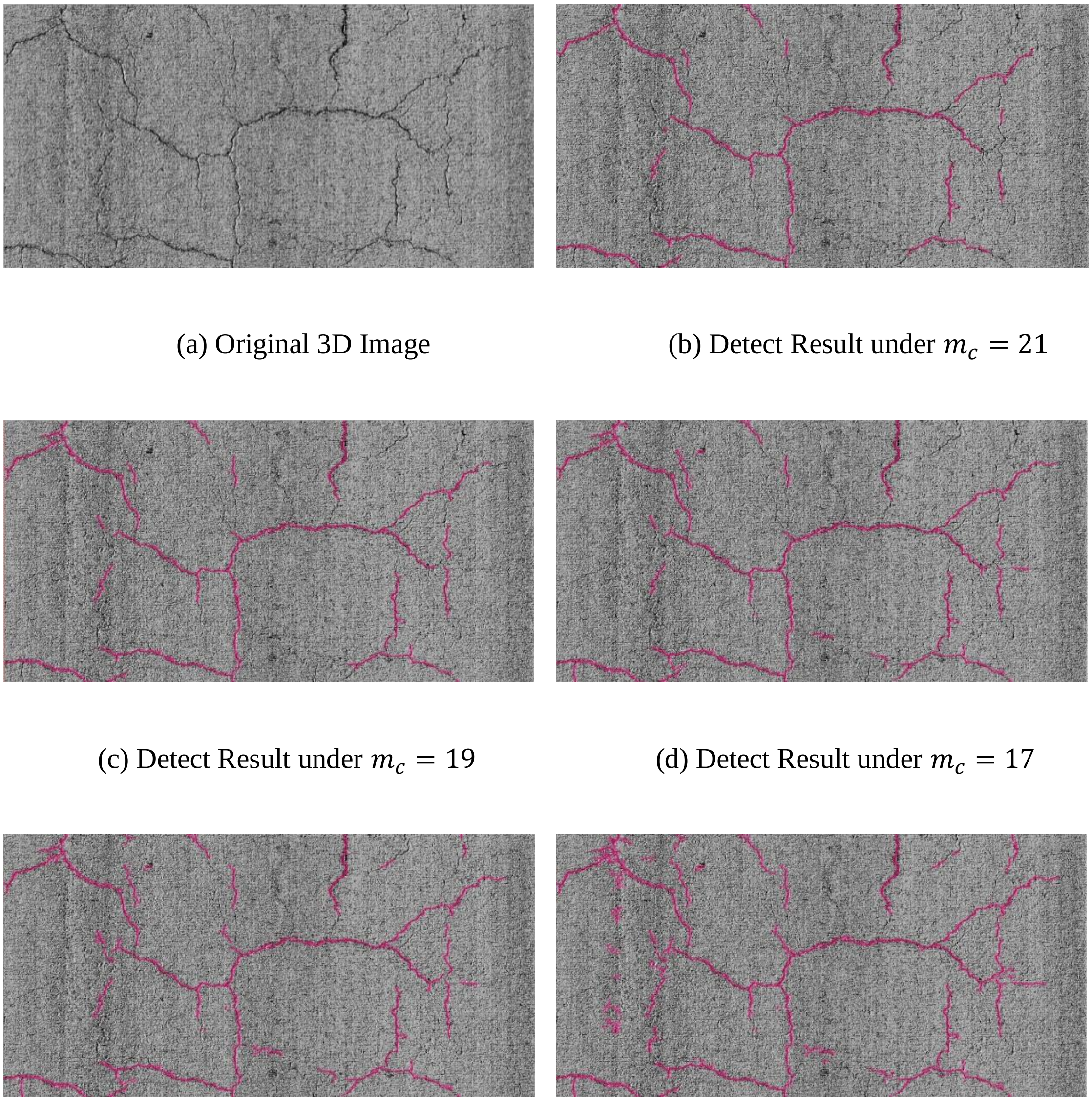
It is indicated in equation (5.11) that the amount of local background extracted near a cracking profile is associated with the width of that profile, because of the tendency that local comparison between the cracking and the background usually diminishes when the cracking width decreases. Attentions may be paid to the appropriate size of the local neighborhood. The size of local neighborhood should be large enough to have adequate background for comparison. However, the local neighborhood cannot be excessively large; otherwise the local comparison will be attenuated.

There could be many strategies to determine the size of the local neighborhood for Neighborhood Analysis. However, a practical strategy should be derived based on endeavors of optimization with vast 3D data. Thousands of 3D images reflecting diverse pavement textures and various cracking presences have been studied in order to determine the strategy for assigning appropriate size of the local neighborhood enclosing the potential cracking cluster. It has been determined that the parameter $\varepsilon = 1$ preformed desirably for most cases and hence is adopted in the paper to extract the local background for Neighborhood Analysis. Once the local neighborhood is determined, it would be straightforward to verify if the corresponding cluster is cracking or not simply by referring to equation (5.2).

5.1.6 Linear Pattern Analysis and Elimination of Tiny Cracks

Noises can still remain after Neighborhood Analysis, particularly when a small Minimal Contrast is introduced. As discussed in Section 3.2.2, Linear Pattern Analysis can removes patterns with low scores in linearity, which provides further rejection on noises. Similar to the length limitation mentioned in Section 3.2.2, if a cracking cluster is isolated and its length is less than 25mm, it will be treated as noise and thus eliminated.

In a summary, the proposed interactive algorithm employs Minimal Contrast as the primary parameter to examine cracking areas line by line and then cluster by cluster, while applying other procedures to suppress noises. Figure 5.9 shows the overall performances of the proposed interactive algorithm with five successively decreased Minimal Contrasts. It is demonstrated that there is an evident evolution in general of the proposed algorithm from being false-negative to false-positive when the Minimal Contrast decreases. Therefore, the proposed algorithm has good potential in responding to manual operator’s feedback to select the proper parameter value to substantially improve the Precision and Recall of cracking detection.



(a) Original 3D Image

(b) Detect Result under $m_c = 21$

(c) Detect Result under $m_c = 19$

(d) Detect Result under $m_c = 17$

(e) Detect Result under $m_c = 15$

(f) Detect Result under $m_c = 13$

Figure 5.9. Detection Results with Five Successively Decreased Minimal Contrasts

5.2 Application of Interactive Cracking Detection Algorithm

The interactive cracking detection is a process of evolving the cracking detector into a more sophisticated one according to the operator’s feedback so that the best performance or acceptable results can be achieved. To express the evolution of an interactive cracking detector in a time manner, equation (5.1) can be developed as:

$$\begin{cases} m^*_{(0)} = m^* & at\ time\ 0 \\ m^*_{(k)} = \begin{cases} m^*_{(k-1)} \oplus & if\ m^*_{(k-1)}(\boldsymbol{I}) = \boldsymbol{I}_n \\ m^*_{(k-1)} \ominus & if\ m^*_{(k-1)}(\boldsymbol{I}) = \boldsymbol{I}_p \\ m^*_{(k-1)}\ and\ evolution\ stops & if\ m^*_{(k-1)}(\boldsymbol{I}) = \boldsymbol{I}_a \end{cases} & at\ time\ k \end{cases} \tag{5.12}$$

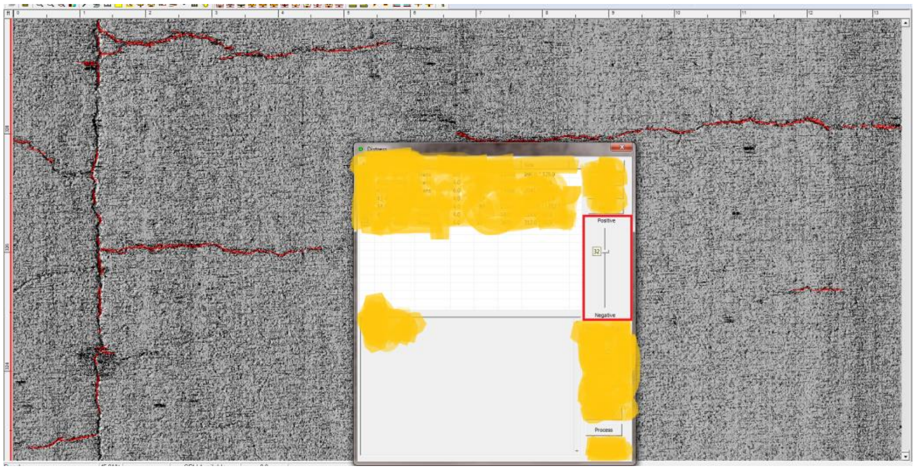
Where: m^*-Initial detector; $m^*_{(k)}$-Evolved Detector at time k; $\boldsymbol{I}_a$-Output image with acceptable results.

A computer program is developed based on the methodology of interactive detection in order to implement two-level's detection mentioned earlier in the study. Detection at the bottom level is to find the majority of pavement cracks automatically after a preferred Minimal Contrast has been selected by the users for the sections being analyzed, while the objective of detection at the top level is to find cracks missed by the bottom level or delete noises generated by the bottom level with users' assistance. To be more specific, the two levels of detection are called Automated Detection and Assisted Detection respectively in the study.

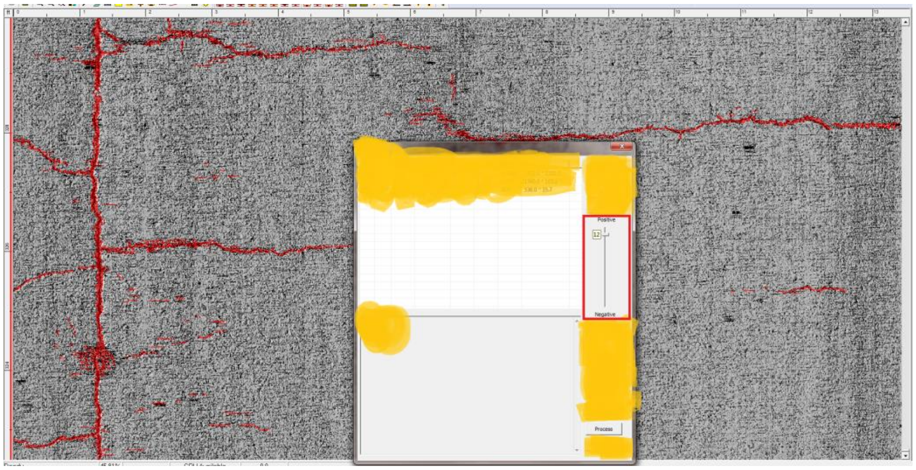
5.2.1 Automated Detection

For Automated Detection, users are able to divide the entire road into multiple sections. Consequently, several representative images or even all of images within a section can be selected for training. Users can control a slider bar designed in the computer program to tell the algorithm to perform more positively or more negatively by reviewing the immediate detection results. Afterwards the algorithm will adjust the Minimal Contrast according to users' feedback. Once an acceptable performance has been achieved through training, the trained Minimal Contrast will then be applied to the whole section.

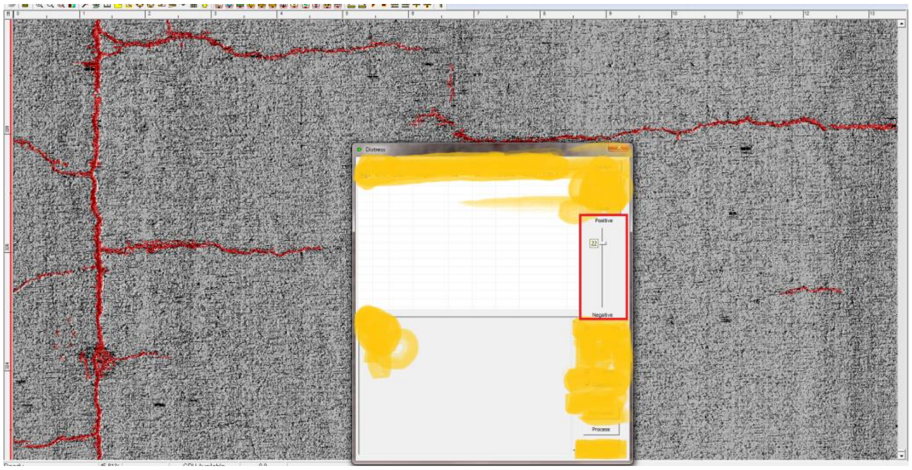
In the computer program, the section size of pavement is flexible so that the training can be conducted for the entire road, or for several images, or even for each image. When the section size decreases, the detection could be more adaptive and thus produce better results. However, the time for users to accomplish training increases with smaller sections. Therefore, users' tolerance and acceptance levels have a major impact on the detection performance of the interactive algorithm. Fig.13 shows the performance scenarios of Automated Detection in the proposed computer program.



(a) False-negative Automated Detection



(b) False-positive Automated Detection

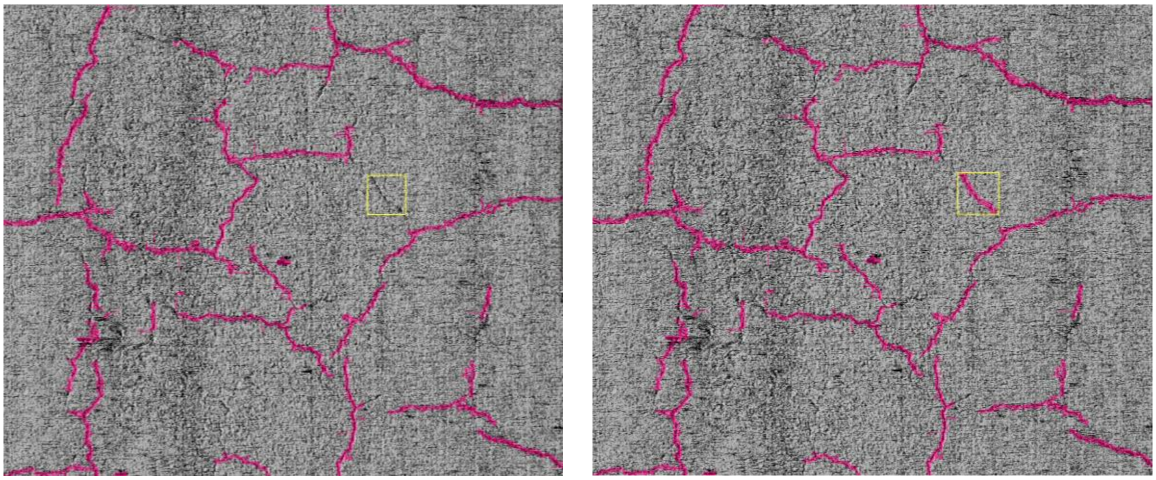


(c) Acceptable Automated Detection

Figure 5.10. Performance Scenarios of Automated Detection in Proposed Computer Program

5.2.2 Assisted Detection

Based on the results by Automated Detection, Assisted Detection can be implemented to find cracks missed by Automated Detection or even delete noises by increasing Minimal Contrast. For Assisted Detection, users can draw any region of interest and repeat interactive detection within the defined region until the best performance has been achieved. Figure 5.11 shows a missed part of cracking found by Assisted Detection, where the rectangle represents the user-defined region.



(a) Automated Detection Result

(b) Missed Cracking Found by Assisted Detection

Figure 5.11. Missed Cracking Detected by Assisted Detection

It is proposed in the study that there always exists a non-cracking area $\boldsymbol{B'}$ adjacent to the cracking area $\boldsymbol{C}$ that has an average $\overline{p_b{}'}$ greater than the average of the cracking area $\overline{p_c}$. If the non-cracking area $\boldsymbol{B_u}$ within the user-defined region $\boldsymbol{A_u}$ approaches to $\boldsymbol{B'}$, the following statement holds.

$$\lim_{\boldsymbol{B_u}\to\boldsymbol{B'}} \overline{p_{bu}} - \overline{p_c} = \overline{p_b{}'} - \overline{p_c} > 0 \tag{5.13}$$

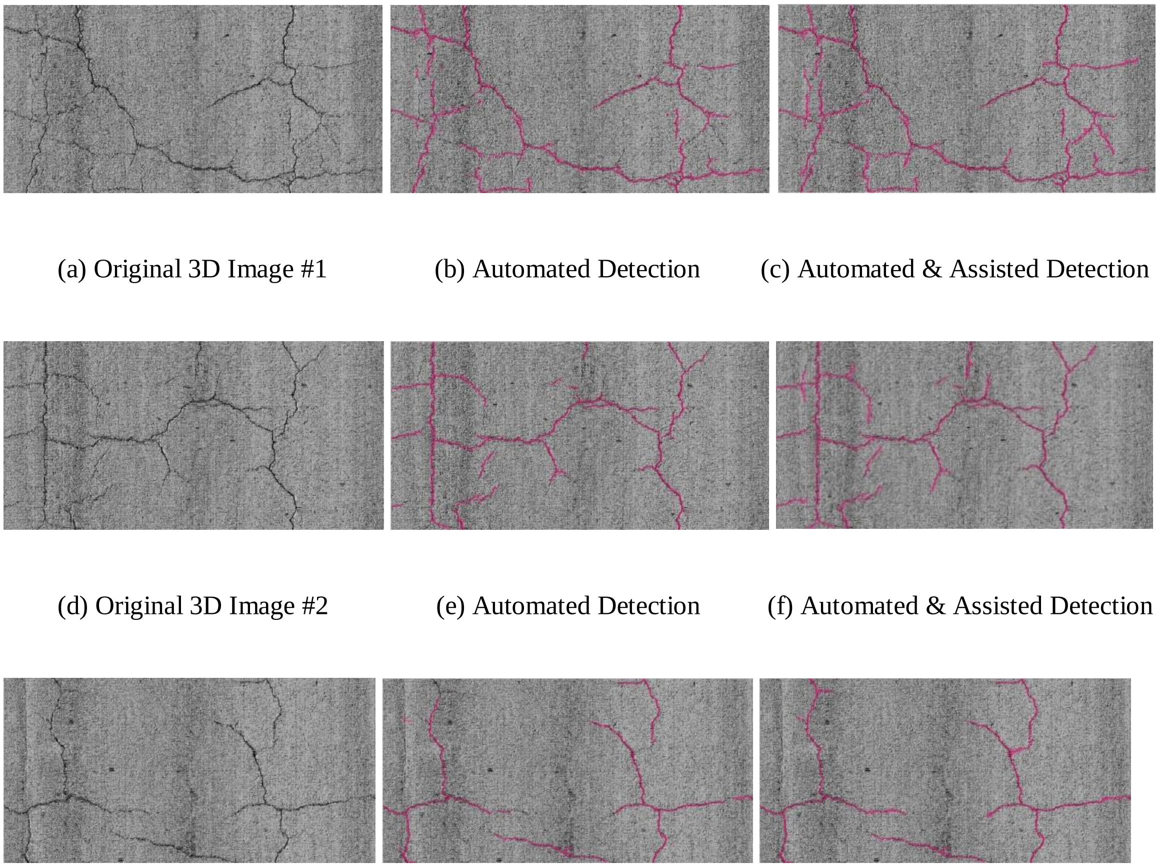
Subject to:

$$\begin{cases} \boldsymbol{B_u} \cup \boldsymbol{C} = \boldsymbol{A_u} \\ \boldsymbol{B_u} \cap \boldsymbol{C} = \boldsymbol{\phi} \end{cases} \tag{5.14}$$

Where: $\overline{p_{bu}} - Average\ of\ \boldsymbol{B_u}$.

The above statement indicates the cracking area within the user-defined region can definitely be detected with an appropriate positive Minimal Contrast, when the user-defined region contains any non-cracking area that has an average greater than the average of the cracking area. In order to increase the efficiency in finding fine cracks, the original 3D data at the defined region will be renormalized to [0,255], followed by increased contrast between fine cracks and the background with high probability when the region size decreases. The extreme case will be only one cracking profile with some amount of background is selected for Assisted Detection. For such a scenario, the contrast between the cracking profile and the background can be maximized by renormalization so that the cracking profile will certainly be located correctly with proper Minimal Contrast, given that the background is brighter than the cracking profile.

On the other hand, noises produced by Automated Detection can also be deleted with high Minimal Contrast, particularly when it is set to be 255. Therefore, the Assisted Detection, which is the top level of interactive detection, provides an approach to retrieving all cracks and eliminating all noises by substantially increasing the detection adaptability. The remaining issue is merely how much time the users would like to spend on refining the results of Automated Detection. Figure 5.12 shows some examples for comparison between Automated Detection and the combination of Automated Detection and Assisted Detection. It is demonstrated that Assisted Detection achieves substantial improvements over the results processed by Automated Detection.



(a) Original 3D Image #1 (b) Automated Detection (c) Automated & Assisted Detection

(d) Original 3D Image #2 (e) Automated Detection (f) Automated & Assisted Detection

(g) Original 3D Image #3 (h) Automated Detection (i) Automated & Assisted Detection

Figure 5.12. Comparison between Automated Detection and the Integration of Automated Detection and Assisted Detection

5.3 Case Study

The same pavement segments mentioned in Section 3.3 are also selected in this chapter to evaluate the performance of the Interactive Cracking Detection Subsystem. For both segments, there are two strategies implemented for the Automated Detection. The first strategy is to apply identical trained Minimal Contrast to the entire segment. The other one is to employ three different trained Minimal Contrasts to three evenly divided subsections respectively. Table 5.1 lists the Minimal Contrasts used for the two strategies. Based on the results of Automated Detection, Assisted Detection is then applied to retrieve true cracks missed by Automated Detection and delete false-detected noises.

Table 5.1. Minimal Contrasts used for Automated Detection

Section ID	Location (m)	m_c for Strategy I		m_c for Strategy II	
		Pavement I	Pavement II	Pavement I	Pavement II
1	0-100	9.4	19.0	13.0	14.0
2	100-200			8.6	20.0
3	200-300			7.4	21.2

The performances of Automated Detection Strategy I and Strategy II are shown from Table 5.2 to Table 5.5, where the performance metrics are reported per twenty meters. It can be observed from Table 5.2 and Table 5.3 that Strategy I is capable of detecting 85.06% of the cracks on Pavement I with a detection precision 86.91% on average, while Strategy II can detect 91.11% of the cracks with an average detection precision 92.70%.

With respect to Pavement II, Strategy I retrieves 86.16% of the cracks with detection precision 87.34%. And Strategy II improves both Precision and Recall almost in every reporting section, comparing to the results of Strategy I. Eventually, the overall Precision and Recall on Pavement II are 90.79% and 90.31% respectively.

The findings indicate the proposed interactive cracking detection subsystem is able to achieve good performance with appropriate training, and its performance even gets better with smaller section size for training and detection.

Table 5.2. Performance of Automated Detection Strategy I for Pavement I

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	89.21	82.69	0.32	0.60
2	20-40	79.03	89.18	0.95	0.36
3	40-60	95.81	89.21	0.19	0.52

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
4	60-80	81.88	88.74	0.82	0.30
5	80-100	87.64	89.12	0.34	0.37
6	100-120	89.09	91.19	0.42	0.23
7	120-140	89.80	87.58	0.21	0.27
8	140-160	85.92	87.84	0.34	0.22
9	160-180	91.06	80.61	0.23	0.50
10	180-200	83.45	85.44	0.59	0.41
11	200-220	88.76	85.44	0.45	0.43
12	220-240	93.16	82.77	0.19	0.56
13	240-260	75.07	89.27	1.46	0.41
14	260-280	79.42	86.28	0.85	0.48
15	280-300	96.61	55.17	0.07	1.23
Overall	0-300	**86.90**	**85.06**	0.50	0.45

Table 5.3. Performance of Automated Detection Strategy II for Pavement I

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	84.82	95.08	0.56	0.17
2	20-40	89.83	92.69	0.35	0.24
3	40-60	95.85	94.00	0.20	0.28
4	60-80	92.56	93.25	0.25	0.17
5	80-100	92.46	94.67	0.23	0.17
6	100-120	95.33	92.87	0.11	0.19
7	120-140	92.93	91.80	0.15	0.17
8	140-160	93.36	93.55	0.11	0.12

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
9	160-180	87.17	91.79	0.39	0.21
10	180-200	94.36	89.00	0.14	0.31
11	200-220	95.21	89.57	0.16	0.29
12	220-240	94.06	86.80	0.17	0.44
13	240-260	95.89	89.79	0.15	0.38
14	260-280	92.96	88.99	0.23	0.36
15	280-300	93.99	80.36	0.15	0.56
Overall	0-300	**92.70**	**91.11**	0.22	0.27

Table 5.4. Performance of Automated Detection Strategy I for Pavement II

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	87.90	82.97	0.45	0.67
2	20-40	86.53	84.21	0.56	0.67
3	40-60	89.03	84.74	0.49	0.71
4	60-80	89.38	83.62	0.21	0.34
5	80-100	90.41	82.33	0.12	0.25
6	100-120	85.06	84.29	0.26	0.28
7	120-140	89.03	88.43	0.28	0.29
8	140-160	84.37	90.82	0.30	0.16
9	160-180	87.76	86.08	0.33	0.38
10	180-200	85.95	85.66	0.24	0.24
11	200-220	86.41	88.49	0.49	0.40
12	220-240	87.66	87.91	0.26	0.25

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
13	240-260	87.45	86.29	0.21	0.23
14	260-280	88.14	89.09	0.41	0.37
15	280-300	85.07	87.53	0.29	0.23
Overall	0-300	**87.34**	**86.16**	0.33	0.37

Table 5.5. Performance of Automated Detection Strategy II for Pavement II

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	88.77	89.99	0.45	0.39
2	20-40	90.45	93.94	0.42	0.26
3	40-60	89.58	89.78	0.49	0.48
4	60-80	92.54	93.27	0.16	0.14
5	80-100	89.67	86.27	0.14	0.20
6	100-120	92.46	91.84	0.13	0.15
7	120-140	92.21	94.32	0.20	0.14
8	140-160	94.45	91.61	0.10	0.15
9	160-180	86.29	89.54	0.39	0.29
10	180-200	88.99	88.14	0.18	0.20
11	200-220	89.54	90.72	0.37	0.33
12	220-240	92.46	87.99	0.15	0.25
13	240-260	92.33	88.62	0.12	0.19
14	260-280	89.21	89.09	0.36	0.37
15	280-300	92.88	89.54	0.13	0.20
Overall	0-300	**90.79**	**90.31**	0.25	0.25

Table 5.6 and Table 5.7 show the performance of the integration of Automated Detection and Assisted Detection. It can be concluded from Tables 5.3, 5.5, 5.6 and 5.7 that the involvement of Assisted Detection significantly increases the detection precision and particularly the recall. The recall increased by Assisted Detection for every section finally reaches up to almost 100 percent, which means almost all cracks can be detected by the integration of Automated Detection and Assisted Detection. The reason why the detection precisions for all sections are not as close as recalls to 100 percent is that some false-detected pixels attached to the true cracking were not recognized and eliminated by the observers due to difficulties in distinguishing them from the true cracking pixels. However, it was carefully examined that no obvious noises remain after the Assisted Detection. Most significantly, it is also found that all noises can be deleted once they are recognized by the observers.

Table 5.6. Performance of the Integration of Automated Detection and Assisted Detection on Pavement I

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	95.15	99.83	0.18	0.01
2	20-40	95.20	99.94	0.16	0.00
3	40-60	97.36	99.98	0.13	0.00
4	60-80	95.08	99.90	0.14	0.00
5	80-100	95.39	99.88	0.14	0.00
6	100-120	96.69	100.00	0.08	0.00
7	120-140	94.36	99.92	0.13	0.00
8	140-160	94.20	99.85	0.10	0.00
9	160-180	91.61	99.72	0.25	0.01
10	180-200	94.76	99.29	0.15	0.03
11	200-220	95.35	99.52	0.15	0.02

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
12	220-240	95.47	99.49	0.14	0.02
13	240-260	96.65	99.83	0.14	0.01
14	260-280	95.00	98.80	0.17	0.04
15	280-300	94.70	99.72	0.16	0.01
Overall	0-300	**95.14**	**99.75**	0.15	0.01

Table 5.7. Performance of the Integration of Automated Detection and Assisted Detection on Pavement II

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
1	0-20	99.35	99.97	0.03	0.00
2	20-40	98.31	99.93	0.07	0.00
3	40-60	97.98	99.88	0.10	0.01
4	60-80	99.00	99.64	0.02	0.01
5	80-100	99.43	99.91	0.01	0.00
6	100-120	99.45	99.79	0.01	0.00
7	120-140	97.13	100.00	0.07	0.00
8	140-160	97.67	100.00	0.04	0.00
9	160-180	96.42	99.92	0.10	0.00
10	180-200	96.41	99.73	0.06	0.00
11	200-220	98.72	99.80	0.05	0.01
12	220-240	97.09	99.41	0.06	0.01
13	240-260	96.94	99.33	0.05	0.01
14	260-280	96.70	99.67	0.12	0.01
15	280-300	98.49	99.92	0.03	0.00

Reporting Section ID	Location (m)	Precision (%)	Recall (%)	FPR (%)	FOR (%)
Overall	0-300	**97.94**	**99.79**	0.05	0.00

In a summary, it has been demonstrated in the case study that the proposed interactive cracking detection subsystem can produce much better results with smaller section size, and the integration of Automated Detection and Assisted Detection is successful to detect almost all cracks and eliminate almost all noises. However, the processing time for Assisted Detection increases substantially compared with Automated Detection, especially when the missed cracks are scattered everywhere or mixed with noises. Therefore, the segmentation of pavements being processed by Automated Detection is critical for reducing the processing time. Specifically, pavement images with same surface type, similar pavement textures or noise levels should be grouped together for better detection performance.

5.4 Summary

In this chapter, interactive cracking detection algorithms that can substantially improve its cracking detection performance by adjustments according to the observer's feedback are proposed. Since the correct response to observer's feedback is critical to the performance of the interactive algorithm, this chapter exposes a series of novel methodologies to accomplish the interaction between the detection system and the observer with high level of success. Based on the proposed algorithms, a computer program with friendly user interface is developed to establish the interactive detection system and implement two levels of detection: Automated Detection and Assisted Detection. The primary objective of Automated Detection is to find the majority of pavement cracking automatically after a preferred Minimal Contrast is trained by the users for the sections being analyzed. In complement to Automated Detection, Assisted Detection provides a semi-automated approach to retrieving missed cracks and eliminating noises in any user-define region. Particularly, the interactive cracking detection can be repeated until the detection results are satisfactory or the best performance has been achieved. According to Precision-Recall Analysis on two selected pavement segments where pavement cracks with diverse presences and different levels of severity are observed, the two-level detection shows the capability of finding almost all of the cracks while maintaining nearly no noises.

Chapter 6 NOISY PATTERN DETECTION SUBSYSTEM

Pavement joint and groove are the two primary noisy patterns on rigid pavements that can introduce serious false positive errors in cracking detection, as they can be regarded as manmade "cracks" on rigid surfaces. Without protecting pavement cracks from joints and grooves, the cracking detection results would be unfeasible in terms of Precision. Therefore, the recognition and exclusion of pavement joints and grooves are critical for the cracking detection on rigid pavements.

Support Vector Machine (SVM) based algorithms are proposed in this chapter to recognize pavement joints and grooves. SVM is a powerful supervised learning model that has been widely used in the field of classification and regression. There are many researches using SVM to classify pavement cracks (Li et al., 2009; Gavilan et al., 2011; Moussa and Hussain, 2011; Daniel et al., 2014). However, limited effort on the application of SVM for the recognition of pavement joints and grooves has been shown.

On the basis of a sufficiently sophisticated SVM classifier that is capable of capturing the classification complexity, Feature Extraction is eventually the most important procedure to address the accuracy of classification. Therefore, this study proposes a methodology to extract features in relation to pavement grooves and joints which could yield high classification accuracy.

Instead of using multi-class SVM algorithms which increase the computational complexity, two-class SVM algorithms are applied in this chapter to distinguish pavement joints and pavement grooves separately from other patterns.

6.1 Kernel Function Based SVM

Denote $\{\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}\}$ as the two-class data set and $y_i \in \{-1,1\}$ as the corresponding class label of $\boldsymbol{x_i}$. The decision boundary can be found by solving the following optimization problem (Boser et al., 1992; Cortes and Vapnik, 1995):

$$\min_{\boldsymbol{w},b,\xi} \quad \frac{\|\boldsymbol{w}\|^2}{2} + C\sum_{i=1}^{n}\xi_i \tag{6.1}$$

Subject to:

$$\begin{cases} y_i(\boldsymbol{w}^T \emptyset(\boldsymbol{x_i}) + b) \geq 1 - \xi_i \\ \xi_i \geq 0 \end{cases} \tag{6.2}$$

Where: $\boldsymbol{w}$ is the normal vector of the hyperplane; b is the offset of the hyperplane; C is the tradeoff parameter between error and margin; ξ_i is a “slack” variable that represents the classification error of $\boldsymbol{x_i}$; $\emptyset$ is the selected kernel function;

The first term in (6.1) represents the objective to maximize the margin between the two classes, while the second term is to minimize the errors if the hyperplane cannot fully separate the two classes. In practical use of SVM, the kernel function is specified to transform the original data to a hyperspace such that a non-linear operation in the original space is identical to a linear operation in the hyperspace. The commonly used kernel functions are listed below (Hsu et al., 2010):

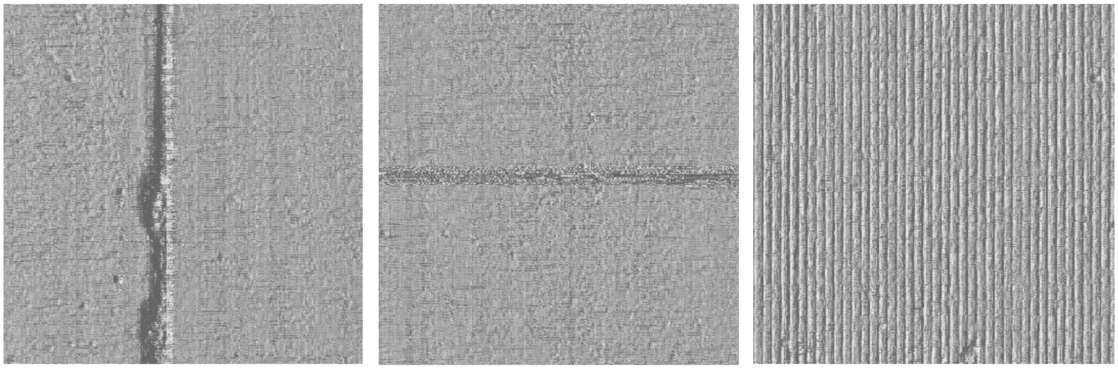
- Linear: $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i^T x_j}$;
- Polynomial: $K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\gamma \boldsymbol{x_i^T x_j} + r)^d, \gamma > 0$;

- Radial Basis Function (RBF): $K\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \exp\left(-\gamma\left\|\boldsymbol{x_i} - \boldsymbol{x_j}\right\|^2\right), \gamma > 0$;
- Sigmoid: $K\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right) = \tanh(\gamma \boldsymbol{x_i^T} \boldsymbol{x_j} + r)$;

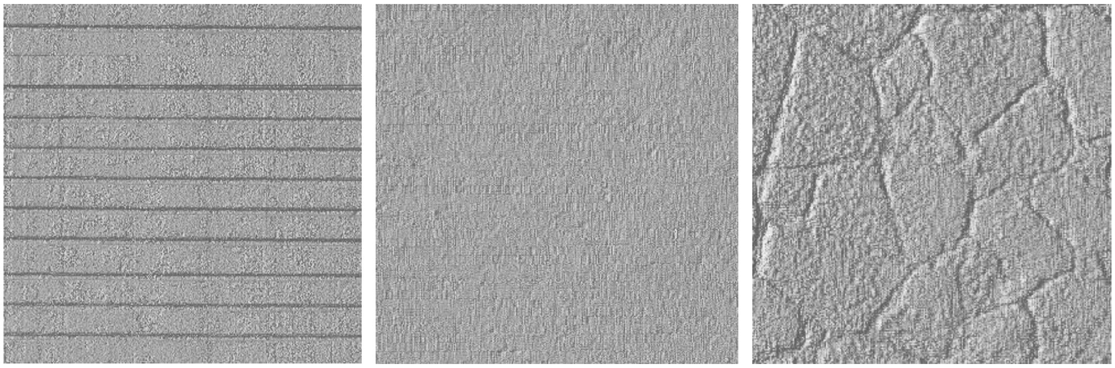
Among the kernel functions, RBF is recommended by many researches as the first choice. Hsu et al. have identified the reasons for choosing RBF in a great detail (2010). The significant advantage of RBF which is highly valued in this study is its versatility in capturing the complexity of the decision boundary, either for nonlinear or linear cases. Thus, it is anticipated in this study that the RBF based SVM could be sophisticated enough to find the complex relations between the feature data and the class labels.

6.2 Sample Collection

The objective of the classification in this chapter is to differentiate pavement joints and pavement grooves from other patterns observed frequently on pavement surfaces. Based on the principal direction, pavement joints can be subdivided into transverse joints and longitudinal joints. Similarly, pavement grooves can be categorized as transverse grooves and longitudinal grooves. Thus, *transverse joint*, *longitudinal joint*, *transverse groove*, *longitudinal groove* and the *others* are the five types of patterns defined in the study for the recognition of pavement joints and grooves, as illustrated in Figure 6.1.



(a) Longitudinal Joint (b) Transverse Joint (c) Longitudinal Groove



(d) Transverse Groove (e) Others (Undamaged Surface) (d) Others (Cracked Surface)

Figure 6.1. Pattern Types Defined for the Recognition of Pavement Joints and Grooves

In order to implement two-class SVM algorithms, it is necessary to define different training model for each pattern to be recognized. For instance, all of transverse joint samples will be labeled as Class 1, while the other four patterns are labeled as Class 2. Therefore, there will be four different classification models for the recognition of transverse joint, longitudinal joint, transverse groove and longitudinal groove respectively, as illustrated in Table 6.1.

Table 6.1. Classification Models for the Recognition of Different Pattern Types

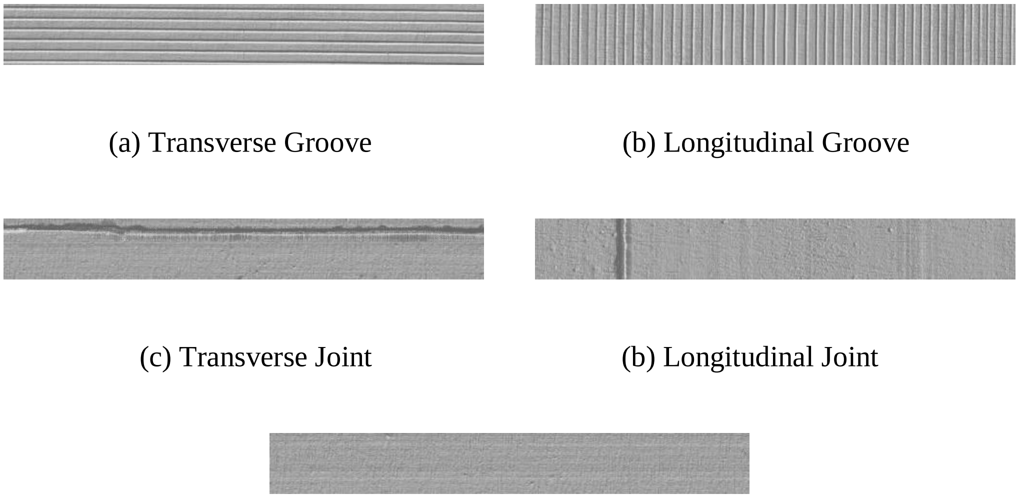
Recognition Objective	Classification Model	
	Class 1	Class 2
Transverse Joint	Transverse Joint	Longitudinal Joint, Transverse Groove, Longitudinal Groove and Others
Longitudinal Joint	Longitudinal Joint	Transverse Joint, Transverse Groove, Longitudinal Groove and Others
Transverse Groove	Transverse Groove	Transverse Joint, Longitudinal Joint, Longitudinal Groove and Others
Longitudinal Groove	Longitudinal Groove	Transverse Joint, Longitudinal Joint, Transverse Groove and Others

The only difference between a transverse joint and a longitudinal joint, or between a transverse groove and a longitudinal groove, is the principal direction. In order to fully consider the principal direction of a pattern so that its features can be extracted more successfully, the sampling region for the recognition of transverse patterns is a long strip in transverse direction. Similarly, the sampling region for the recognition of longitudinal patterns is a long strip in longitudinal direction. The sampling sizes for the recognition of longitudinal patterns and transverse patterns are listed in Table 6.2.

Table 6.2. Sampling Sizes for Transverse Patterns and Longitudinal Patterns

Transverse Patterns		Longitudinal Patterns	
Transverse Joint	Transverse Groove	Longitudinal Joint	Longitudinal Groove
Sample Width: 2048 mm Sample Height: 256 mm		Sample Width: 256 mm Sample Height: 2048 mm	

For simplicity, the samples collected for the recognition of transverse patterns are called Transverse Samples, while the samples collected for the recognition of longitudinal patterns are called Longitudinal Samples. It should be noted that either Transverse Samples or Longitudinal Samples collected in this chapter contain the five types of patterns defined previously. Some representative Transverse Samples and Longitudinal Samples are shown in Figure 6.2 and Figure 6.3 respectively.



(a) Transverse Groove

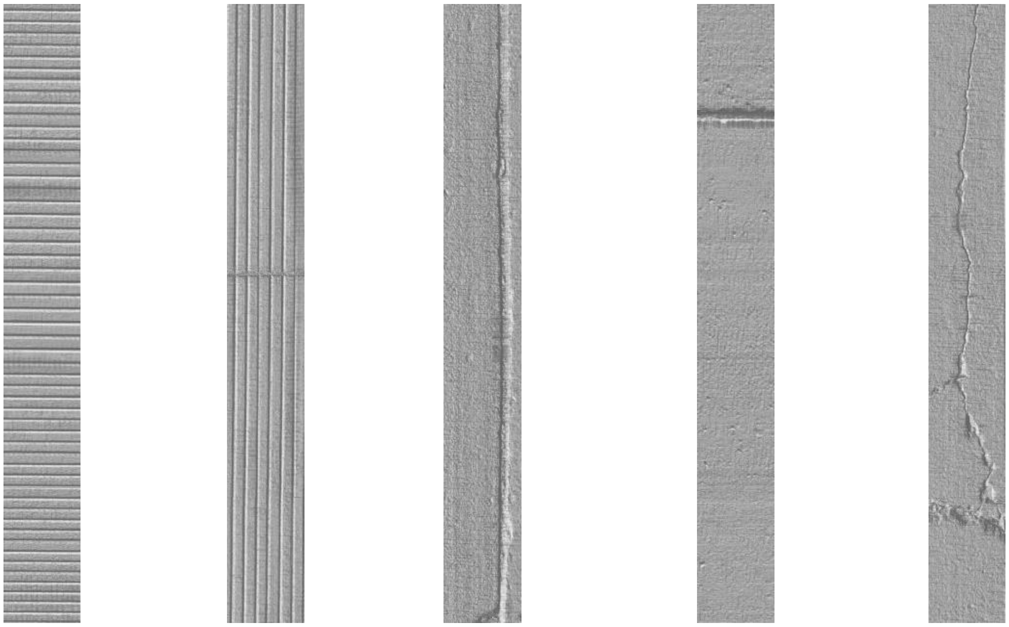
(b) Longitudinal Groove

(c) Transverse Joint

(b) Longitudinal Joint

(e) Others

Figure 6.2. Representative Transverse Samples



(a) Trans. Groove (b) Long. Groove (c) Long. Joint (d) Trans. Joint (e) Others

Figure 6.3. Representative Longitudinal Samples

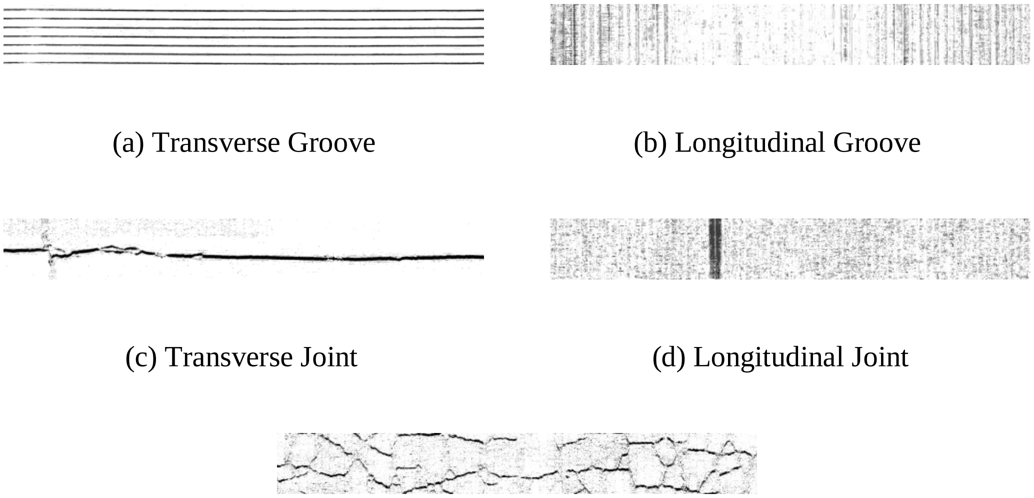
In this chapter, 1000 Transverse Samples and 1000 Longitudinal Samples are collected with a careful supervision on the sampling diversity. Half of the samples will be involved in the training of SVM classifiers, and the other half will be used for testing the accuracy of the trained classifiers.

6.3 Feature Extraction

6.3.1 Complexity Reduction Using 3D Shadow Simulation

The features extracted from collected samples play a critical role in the success of a SVM classifier. The complexity of the original data of a 3D sample creates a lot of difficulties in extracting meaningful and effective features. Therefore, this study utilizes 3D Shadow Simulation discussed in Chapter 3 to reduce the complexity of the original data. After 3D Shadow Simulation, the original data can be transformed to a binary domain, where the pixels with smaller heights are shadowed and labeled as “0”, and other pixels with greater heights are not shadowed and thus marked as “1”. In this way, the complexity of the original data can be reduced significantly. With respect to data loss through the transformation, since both pavement groove and pavement joint are lower the local background, a large portion or even entire of them should remain as long strip shadowed patterns if the lighting direction is correct and the projection angle is low enough. In other words, it is feasible to maintain the pattern of pavement joint or pavement groove with an appropriate lighting. In order to preserve the pattern of a pavement joint or pavement groove in large portion, the projection angle is fixed as 45 degree for both Transverse Lighting and Longitudinal Lighting, which is tested as sufficiently low for most of cases.

As discussed in Chapter 3, Transverse Lighting performs better in preserving longitudinal-dominated patterns, while Longitudinal Lighting preserves more of transverse-dominated patterns. Therefore, Transverse Lighting is only conducted on Longitudinal Samples, and Longitudinal Lighting is merely applied to Transverse Samples for the reduction of complexity. Some typical examples of 3D Shadow Simulation on Transverse Samples and Longitudinal Samples are shown in Figure 6.4 and Figure 6.5.

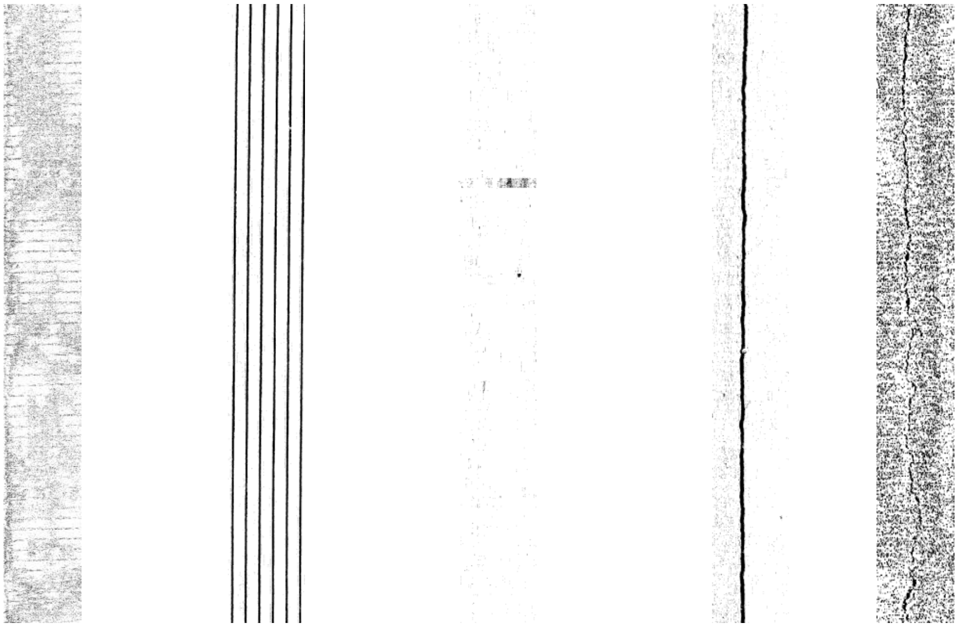


(a) Transverse Groove (b) Longitudinal Groove

(c) Transverse Joint (d) Longitudinal Joint

(e) Others

Figure 6.4. Longitudinal Lighting Results on Transverse Samples



(a) Trans. Groove (b) Long. Groove (c) Trans. Joint (d) Long. Joint (e) Others

Figure 6.5. Transverse Lighting on Longitudinal Samples

It can be observed from Figures 6.4 and 6.5 that transverse patterns, including Transverse Groove and Transverse Joint, are maintained desirably under Longitudinal Lighting. Such a consequence is also found on longitudinal patterns when Transverse Lighting is applied. Moreover, the intervention of 3D Shadow Simulation not only preserves patterns whose principal directions are close to the wanted orientation, but also attenuates other patterns whose principal directions is far from the wanted orientation, which broadens the gap between patterns of different principal directions. Based on 3D Shadow Simulation, the features to be extracted from the shadow maps are discussed in the following sections.

6.3.2 Density

Density is defined as the percentage of shadowed pixels on a sample, which can be written as:

$$d_i = \frac{NS_i}{w_i \cdot h_i} \tag{6.3}$$

Where: d_i is the Density of ith sample; NS_i is the number of shadowed pixels in ith sample; w_i is the width of ith sample; h_i is the height of ith sample;

The Density can be an important indicator for the boundary between pavement joints and grooves. Additionally, it is also helpful for the exclusion of some unwanted patterns with too few or too many shadowed pixels.

6.3.3 Average Width

The Average Width is defined as the weighted average width of all shadowed clusters. Here, the shadowed clusters are obtained using the Blob Extraction algorithm mentioned in Chapter 3 and Chapter 5.

$$\omega_i = \sum_{j=1}^{m_i} \frac{NS_{i,j}}{NS_i} \cdot \omega_{i,j} \tag{6.4}$$

Where: ω_i is the weighted average width of shadowed clusters in ith sample; m_i is the number of shadowed clusters in ith sample; $NS_{i,j}$ is the number of shadowed pixels of jth shadowed cluster in ith sample; $\omega_{i,j}$ is the average width of jth shadowed cluster in ith sample.

Since the width of a pavement joint is usually greater than the width of a pavement groove, the Average Width can be a useful feature to be learned in order to separate pavement joints from pavement grooves.

6.3.4 Average Angle

Likewise, the Average Angle in this chapter is defined as the weighted average angle of all shadowed clusters. That is:

$$\theta_i = \sum_{j=1}^{m_i} \frac{NS_{i,j}}{NS_i} \cdot \theta_{i,j} \tag{6.5}$$

Where: θ_i is the weighted average angle of shadowed clusters in ith sample; $\theta_{i,j}$ is the angle of jth shadowed cluster in ith sample.

The participation of Average Angle places considerations on the range of pattern angles, which may result in a distinct boundary between wanted orientations and unwanted orientations. For instance, the angle of a longitudinal joint is around 90 degree, while the angle of a transverse joint is close to 0 degree.

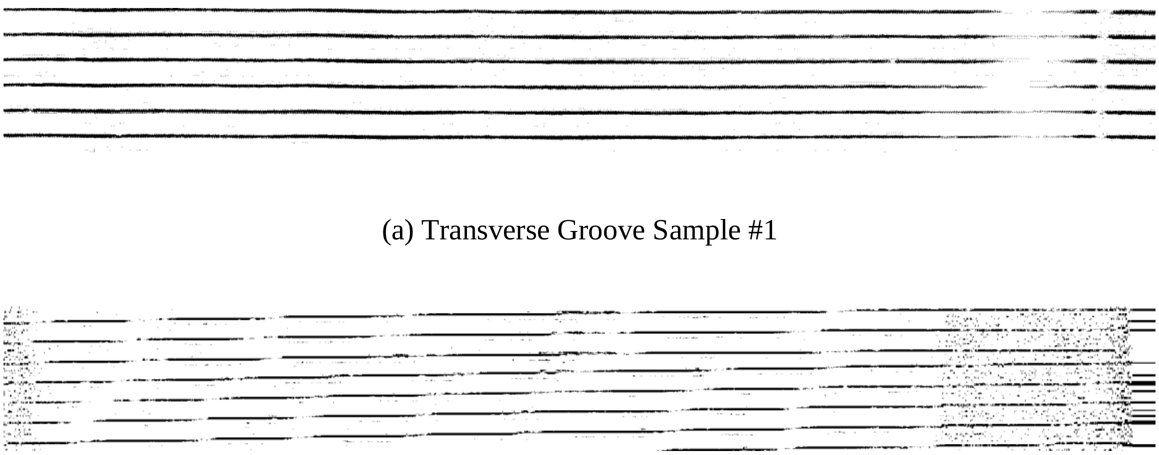
6.3.5 Number of Long Strip Patterns

The Number of Long Strip Patterns is proposed in this study as an important feature to recognize pavement joints and pavement grooves. As a matter of fact, pavement grooves are a bunch of long strip patterns paralleled even in a small region. On the contrary, pavement joint is only a single long strip pattern laid alone within a much larger region. Therefore, the Number of Long Strip Patterns can serve in splitting pavement joints and pavement grooves by showing the repetitions of long strip patterns within a small region. In this study, a long strip pattern is defined as the pattern whose length exceeds a certain threshold, which can be expressed by:

$$l_{i,j} > \lambda \cdot \max(w_i, h_i) \tag{6.6}$$

Where: $l_{i,j}$ is the length of jth long strip pattern found on ith sample; λ is the threshold value.

$\lambda = 0.5$ is used in this chapter, which means a strip pattern whose length is greater than half length of the longer edge of the sample will be treated as a long strip pattern. It should be noted that some portion of a pavement joint or a pavement groove may not be kept on the binary map due to the data loss during 3D Shadow Simulation or even the discontinuity of the original 3D data. As a result, a whole pavement joint or groove might be divided to several shadowed clusters due to the discontinuity. Figure 6.6 shows the discontinuity of some pavement grooves.



(a) Transverse Groove Sample #1

(b) Transverse Groove Sample #2

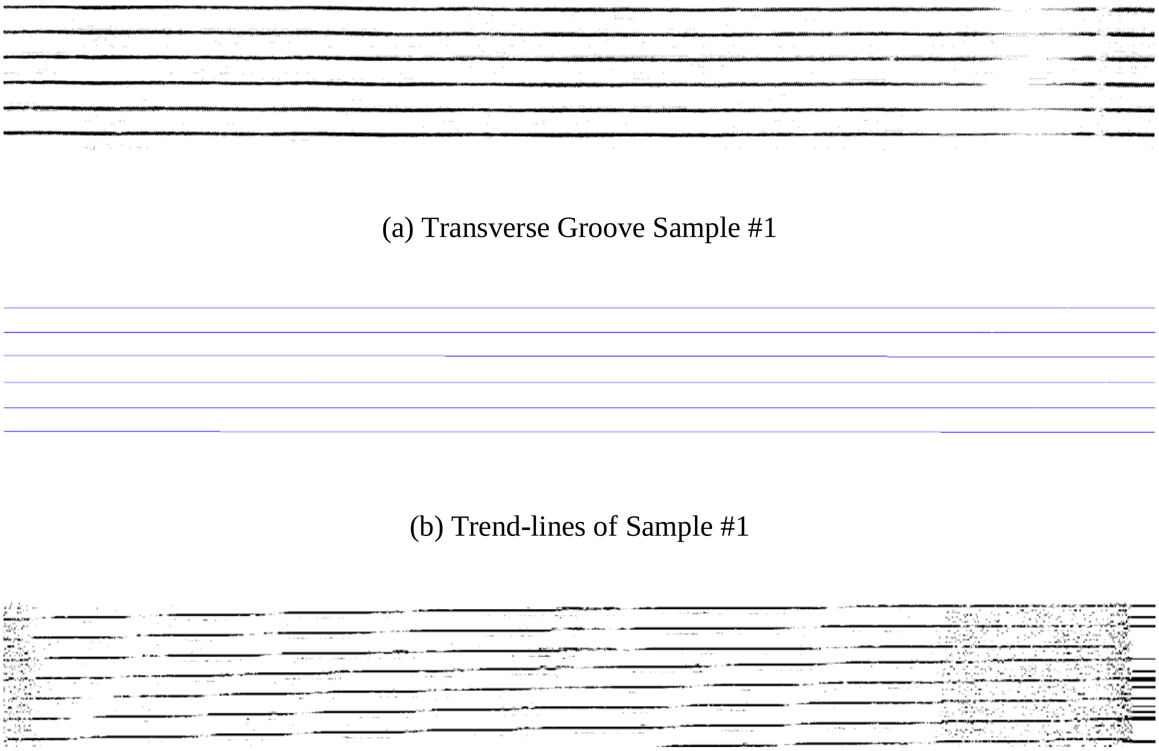
Figure 6.6. Representative Discontinuity Problems

The following procedures are proposed to seek for long strip patterns in their entireties for the solution of discontinuity.

- For each shadowed cluster, draw a line that penetrates through the entire shadowed cluster and travels across the entire sample until it is out of the sample domain.
- Count the shadowed pixels on the line plotted for each cluster. If the number of shadowed pixels on the line satisfies the inequality described in (6.6), this line will be recorded as a possible trend-line of a long strip pattern;

- At the first step, some pixels may be recounted by the lines plotted for different shadowed clusters, which could result in overestimation on the number of long strip patterns. Thus, at this step, all recorded trend-lines will be examined such that no two trend-lines are intersected with each other within the sample domain. Once two trend-lines are intersected, only the one whose direction is closer to the preferred orientation (e.g. 90 degree for longitudinal patterns and 0 degree for transverse patterns) will be preserved.
- Count the trend-lines survived eventually as the *Number of Long Strip Patterns*.

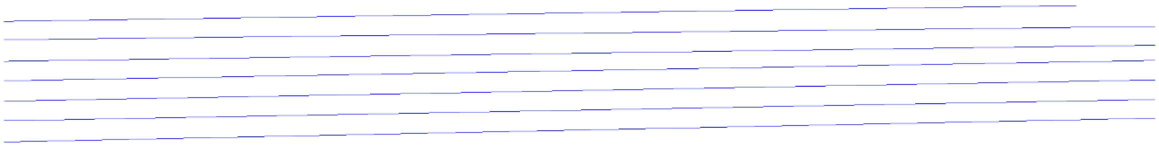
Using the above procedures, Figure 6.7 shows the final trend-lines of the two samples cited in Figure 6.6.



(a) Transverse Groove Sample #1

(b) Trend-lines of Sample #1

(c) Transverse Groove Sample #2



(d) Trend-lines of Sample #2

Figure 6.7. Trend-lines Obtained by the Proposed Procedures

In a summary, Density, Average Width, Average Angle and Number of Long Strip Patterns are the four basic features proposed in this chapter for the recognition of pavement joints and grooves.

6.4 Training and Testing

A well-known online SVM library, *LIBSVM*, is adopted in this section for the training and testing of SVM classifiers (Chang and Lin, 2011). As mentioned in Section 6.1, RBF is chosen as the kernel function for the SVM classifiers. Therefore, there are essentially two parameters to be considered in the process: γ and C.

In order to retrieve a good set of γ and C and avoid over-fitting problem, v-fold Cross-Validation and Grid Search are integrated together as a fast procedure to obtain a noble set of γ and C which yields the best prediction accuracy on “unknown” data (Hsu et al., 2010). In the v-fold cross-validation, the training data are divided into v subsets. Afterwards, one subset is predicted based on the training on other subsets. $v = 5$ is used in this chapter. For the Grid Search, numerous values of γ and C which increase exponentially (i.e. $2^{-10}, 2^{-9}, \dots, 2^{9}, 2^{10}$) are tried to obtain the best set of γ and C. A typical result of Grid Search is illustrated in Figure 6.8.

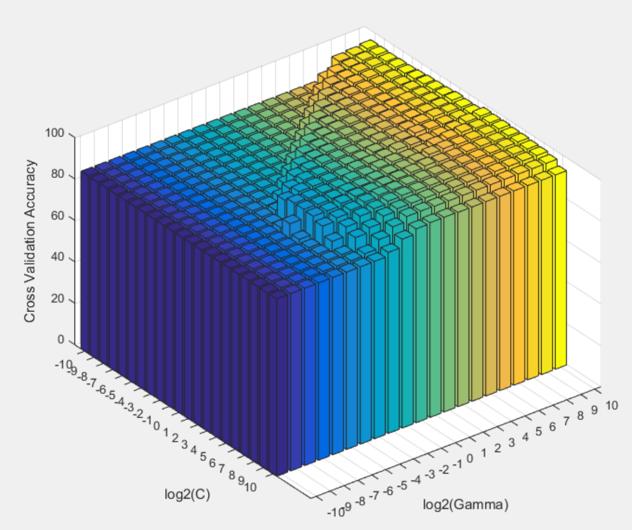


Figure 6.8. Typical Result of Grid Search

Half of the collected Transverse Samples (500 samples) are involved in the training for the recognition of Transverse Grooves and Transverse Joints. Similarly, 500 Longitudinal Samples are learned in the training for the recognition of Longitudinal Grooves and Longitudinal Joints. Before training starts, the features of training samples are extracted by the methods described in Section 6.3. Then, the class labels are assigned according to the classification models listed in Table 6.1 for different recognition objectives. The features extracted for the four recognition objectives are plotted in a one-versus-one manner in Figure 6.9, Figure 6.10, Figure 6.11 and Figure 6.12 respectively. Note that the features *Average Width*, *Average Angle* and *Number of Long Strip Patterns* have been scaled into a range around [0, 1].

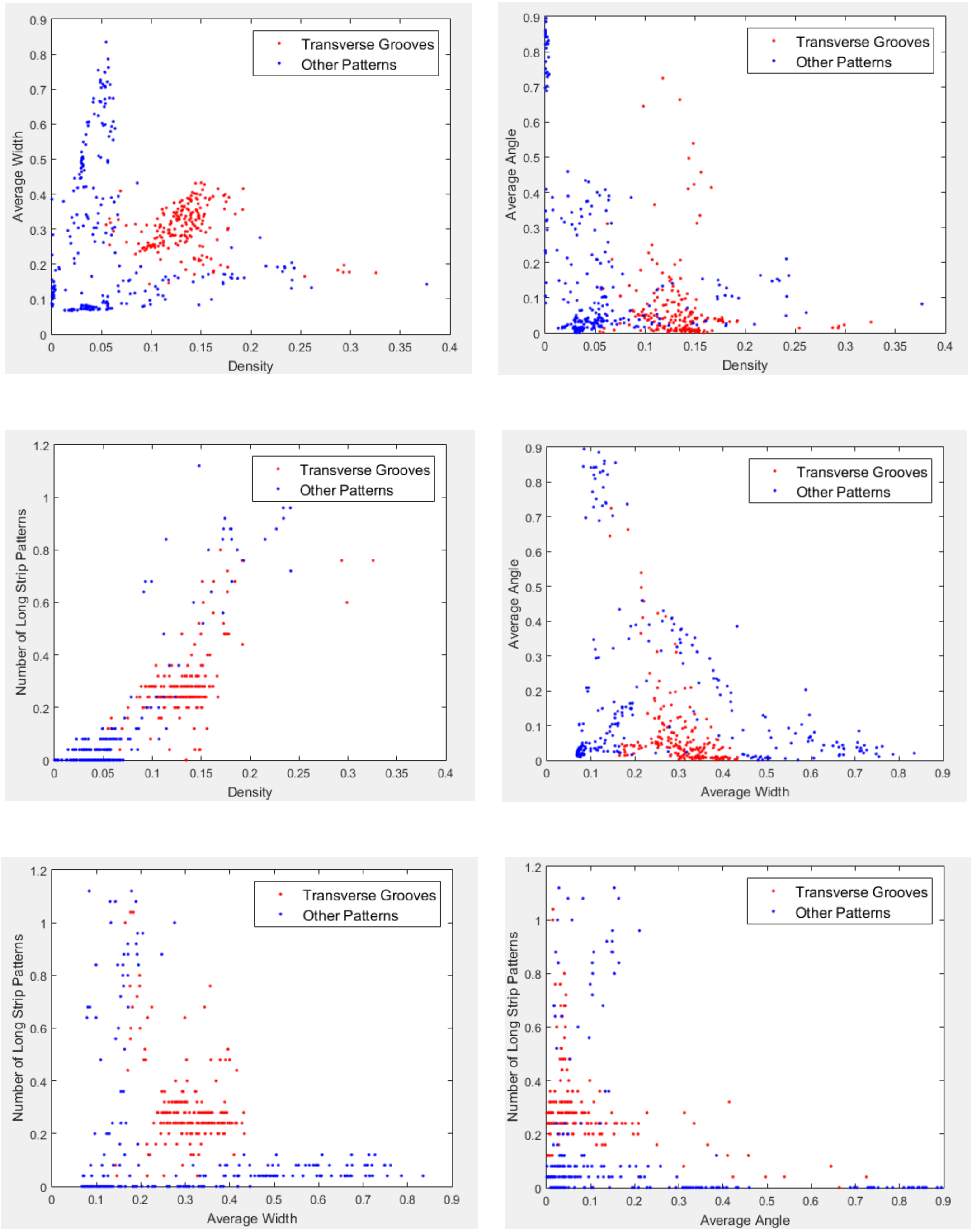


Figure 6.9. Extracted Features for the Recognition of Transverse Groove

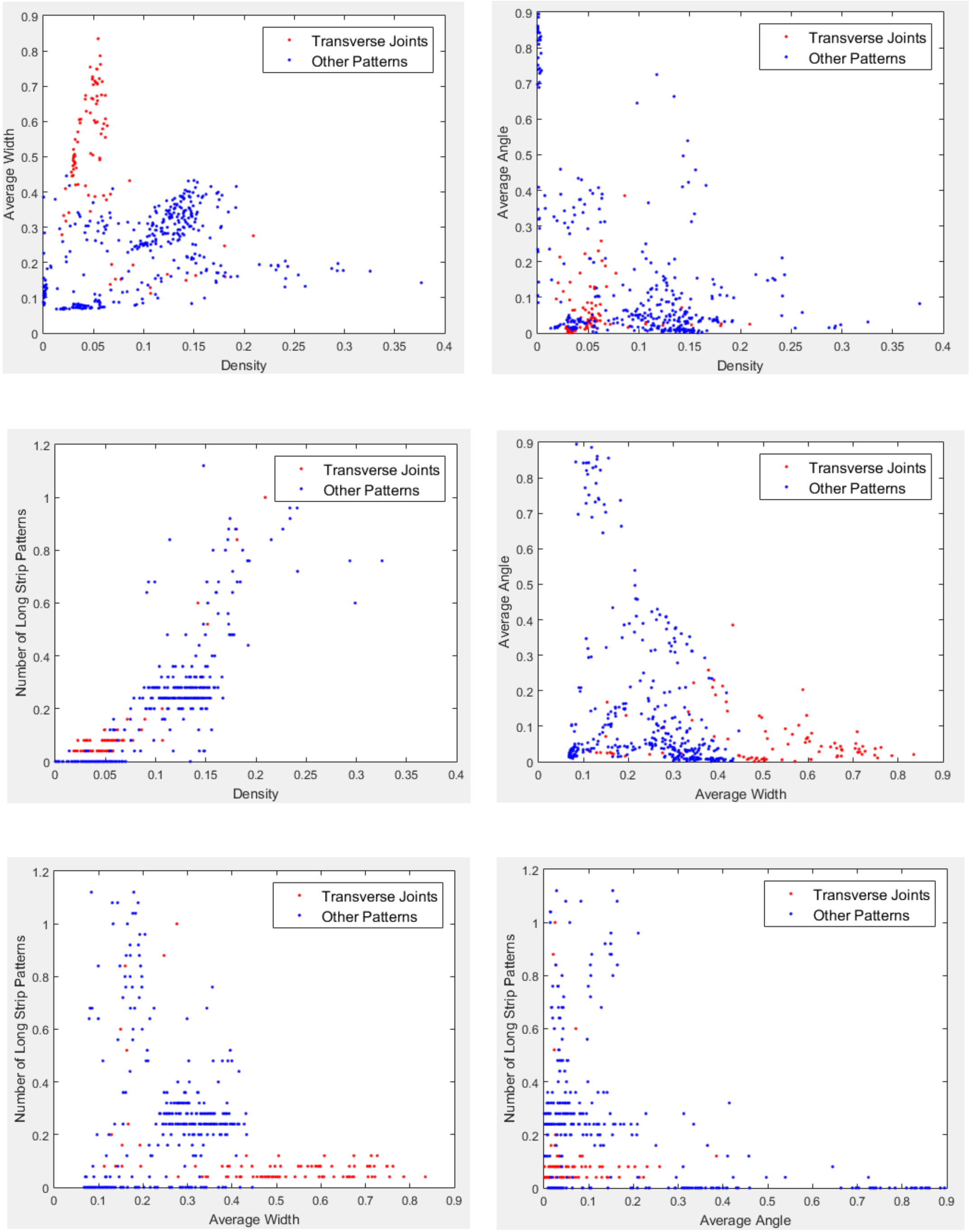


Figure 6.10. Extracted Features for the Recognition of Transverse Joint

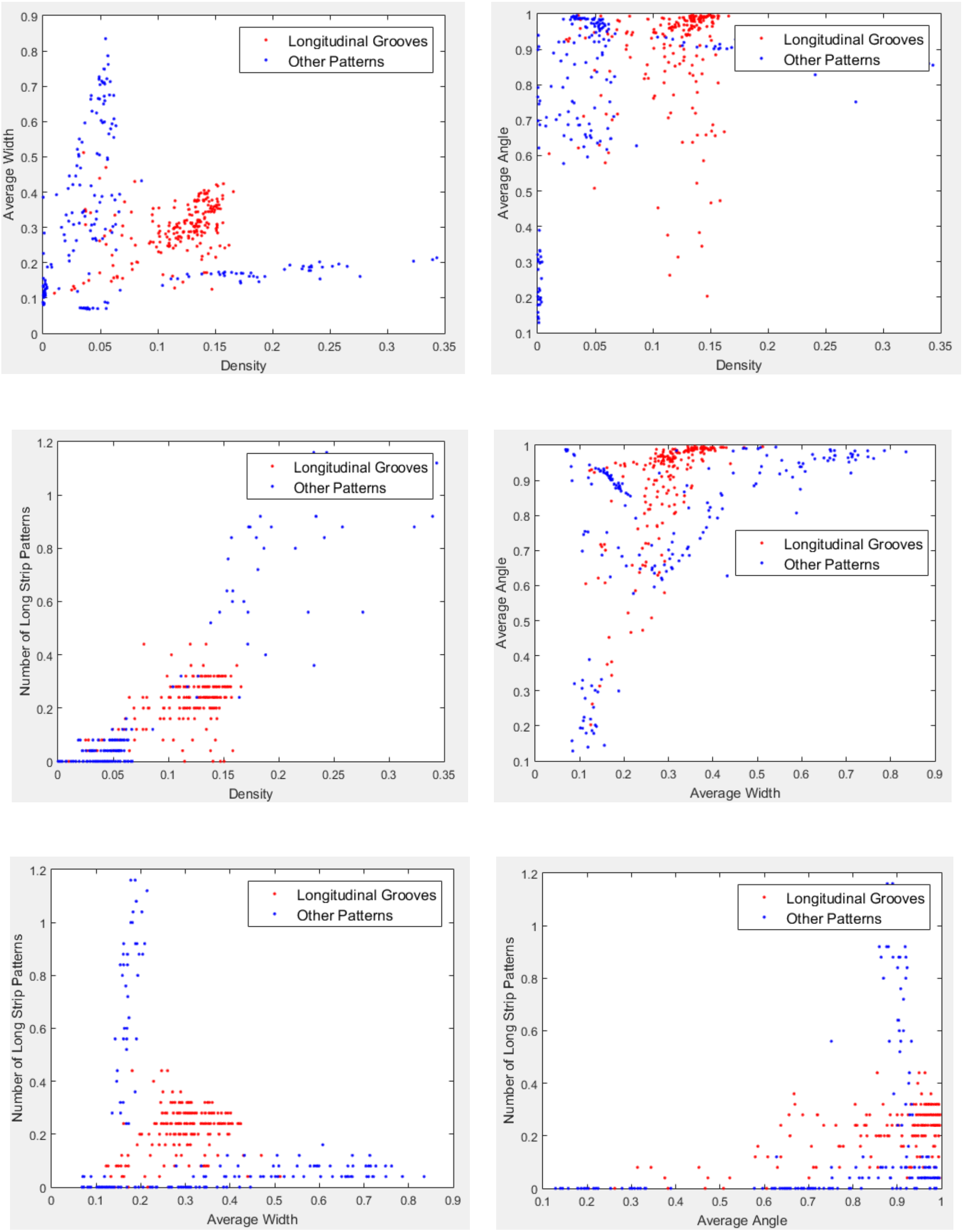


Figure 6.11. Extracted Features for the Recognition of Longitudinal Groove

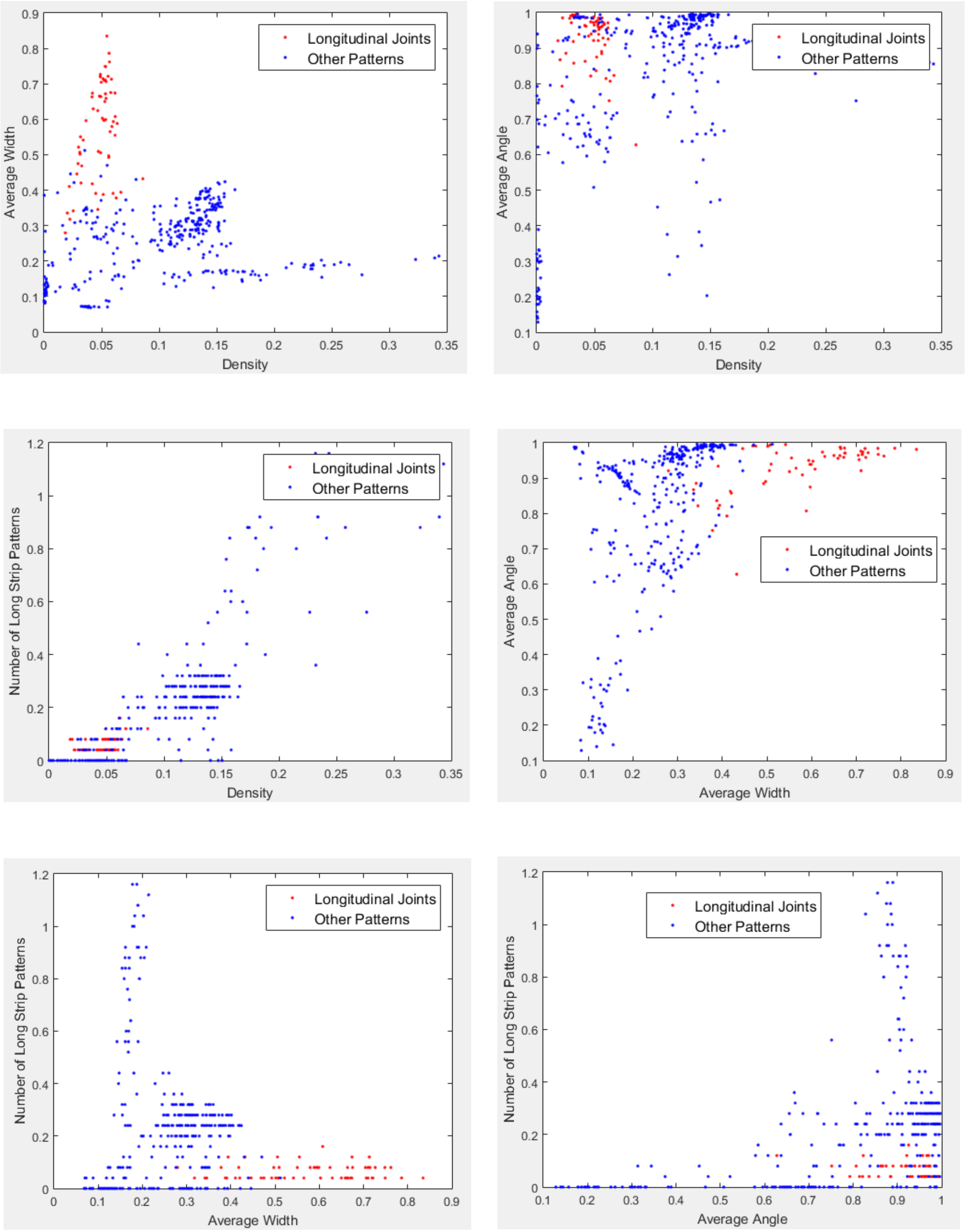


Figure 6.12. Extracted Features for the Recognition of Longitudinal Joint

It is revealed in Figures 6.9-6.12, the two-class data for each recognition objective are separable, even though there are some data may be misclassified and the hyperplane seems to be complicated. For each recognition objective, the γ and C are trained respectively using Cross Validation and Grid Search. Then, the training accuracies for the four recognition objectives are listed in Table 6.3.

Table 6.3. Training Accuracies for the Four Recognition Objectives

Recognition Objective	Training Accuracy (%)
Transverse Groove	99.6
Transverse Joint	97.2
Longitudinal Groove	99.4
Longitudinal Joint	99.6

It has been demonstrated in Table 6.3 that the training accuracy is consistently high for each of the recognition objectives. Based on the Support Vectors selected via the training, 500 Transverse Samples and 500 Longitudinal Samples which are different from the samples used in training are employed to test the efficiency of selected Support Vectors. Again, the 500 Transverse Samples are used for the recognition of transverse joints and transverse grooves, while the 500 Longitudinal Samples are used for the recognition of longitudinal joints and longitudinal grooves. The testing accuracies for the four recognition objectives are shown in Table 6.4.

Table 6.4. Testing Accuracies for the Four Recognition Objectives

Recognition Objective	Testing Accuracy (%)
Transverse Groove	98.6
Transverse Joint	97.0
Longitudinal Groove	93.8

Longitudinal Joint	98.6

According to Table 6.4, the selected Supported Vectors achieve high prediction accuracies on the unknown samples for each of the four recognition objectives, which implies the efficiency and reliability of the trained SVM classifiers.

6.5 Pattern Extraction

The classification discussed in previous section is fundamentally to assign a label to a sample indicating whether the target pattern occurs on that sample. However as referring to Table 6.2, each collected sample is a data block of the 3D surface and the exact location of the target pattern is still unknown. Therefore, the following pattern extraction procedures are proposed in this section to locate the target pattern for the practical use of classification outputs.

- Determine the Recognition Objective (Transverse Groove, Transverse Joint, Longitudinal Groove or Longitudinal Joint);
- Input the Supported Vectors associated with the Recognition Objective;
- Sample Data using the size discussed in Table 6.2;
- Apply 3D Shadow Simulation on the sample data to obtain the binary map
- Extract Features from the binary map;
- Classify the sample using the input Support Vectors;
- If the sample is label as belonging to the class of the target pattern, pick up shadowed clusters situated on the Trend-lines learned during Feature Extraction. The picked shadowed clusters will be preserved as the parts of the target pattern. Other shadowed clusters will thus be eliminated.

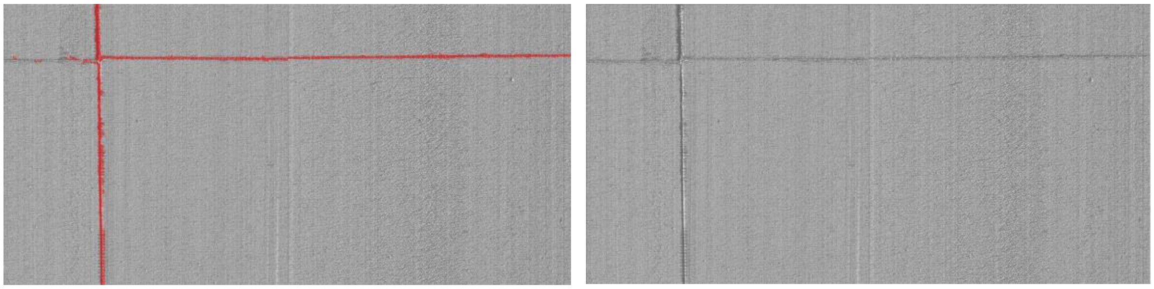
Based on the pattern extraction procedures, Precision-Recall Analysis is conducted on the 1000 testing samples (500 Transverse Samples and 500 Longitudinal Samples), where the ground truths are marked manually. The Precision and Recall are listed in Table 6.5. It is demonstrated in

Table 6.5 that the methodology of Pavement Joint & Groove Detection based on the pattern extraction procedures and four SVM classifiers eventually progresses to the accomplishment in high Precision and high Recall on all testing samples collected from the real roads.

Table 6.5. Precision-Recall Analysis Results on Testing Samples

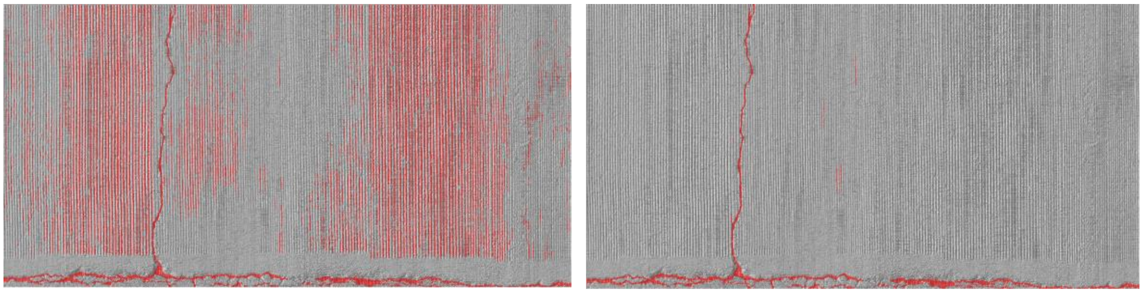
Sample Type	Number of Samples	Precision (%)	Recall (%)	FPR (%)	FOR (%)
Transverse Samples	500	98.72	95.86	0.07	0.22
Longitudinal Samples	500	98.83	94.68	0.08	0.36
Overall	1000	98.78	95.27	0.07	0.29

The Noisy Pattern Detection Subsystem is finally embedded into a computer program, where the detected pavement joints or pavement grooves will be excluded from the cracking detection results. Figure 6.13 shows the performance of Noisy Pattern Detection Subsystem on some typical examples.

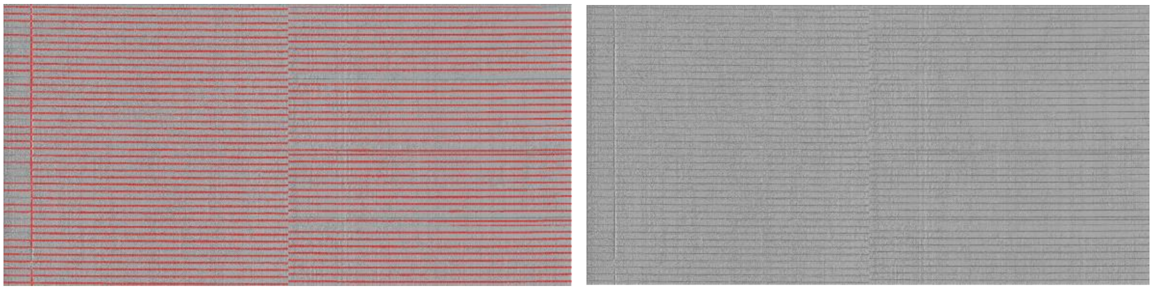


(a) Original Cracking Detection Result #1

(b) Corrected Cracking Detection Result #1



(c) Original Cracking Detection Result #2 (d) Corrected Cracking Detection Result #2



(e) Original Cracking Detection Result #3 (f) Corrected Cracking Detection Result #3

Figure 6.13. Cracking Detection Results Corrected by Noisy Pattern Detection Subsystem

6.6 Summary

In a summary, a comprehensive system of algorithms based on SVM are proposed in this chapter for the detection of pavement joints and grooves. In order to differentiate transverse patterns (transverse joints and transverse grooves) from longitudinal patterns (longitudinal joints and longitudinal grooves), each sample is extracted from 3D pavement surfaces as a long strip region whose principal directions are identical to the principal direction of the target pattern, either transverse or longitudinal. 3D Shadow Simulation is then applied to transform the original 3D samples to the associated binary maps for the reduction of data complexity. During the 3D Shadow Simulation, Transverse Samples are only assigned with Longitudinal Lighting, while Longitudinal Samples are only assigned with Transverse Lighting. *Density*, *Average Width*, *Average Angle* and *Number of Long Strip Patterns* are the four features extracted from the binary maps. Later on, two-class SVM algorithms using Radial Basis Function are implemented to train the four classifiers for four different recognition objectives (transverse groove, transverse joint, longitudinal groove and longitudinal joint) based on the features learned from the training samples. It is demonstrated that the trained SVM classifiers achieve high accuracy in categorizing both training and testing samples. Based on the trained SVM classifiers, pattern extraction procedures are developed not only to classify the input samples, but also to find the location of target patterns on the input samples once they are labeled as containing the target patterns. Finally, Precision-Recall Analysis are conducted on all the testing samples to evaluate the performance of the proposed algorithms on the detection of pavement joints and grooves. It has been shown that the proposed algorithms based on SVM classifiers are successful in detecting pavement joints and grooves on all the testing samples with a high level of Precision and Recall.

Chapter 7 PARALLEL COMPUTING TECHNIQUES FOR PAVEMENT CRACKING DETECTION

The increase of processing speed is always a fundamental necessity in real world applications where massive data flow are required. The 3D pavement surface with 1mm resolution is unsurprisingly large even within a short section. For instance, the 3D pavement data for a 1 mile long and 4 meters wide lane has a giant size more than 12 gigabytes. Therefore, the speedup of pavement cracking detection algorithms will become much beneficial when a long road is to be analyzed. Regardless of the improvements on algorithms for the reduction of computational intensity, Parallel Computing is another approach to substantial time savings by completing many computational tasks simultaneously. The prerequisite of Parallel Computing is that the computational problem should be able to be broken into several parts that can be executed simultaneously. Over decades, there are tremendous developments in parallel computing technology which makes parallel computing techniques always the first choice when the processing time is necessarily concerned. Up to recently, parallel computing have been commonly acknowledged in numerous fields, including engineering, scientific applications and commercial applications (Grama et al., 2003). In terms of the hardware system, parallel computing can be executed on CPUs, GPUs or even a hybrid system of CPUs and GPUs.

The advantage of GPUs over CPUs is that GPUs have much more processors that can run simultaneously. However, the latency of a GPU processor is longer than the latency of a CPU processor, which means a single GPU processor runs slower than a single CPU processor. Additionally, most of the modern computers require the data transfer between CPUs and GPUs if the tasks are to be executed on GPUs. Although the GPUs are much faster in completing tremendous independent tasks concurrently, the data transfer consumes some amount of time due to the low bandwidth between GPUs and CPUs, which may affect the preferences on GPUs in some cases. Due to the different architectures of GPUs and CPUs, there are many factors, such as data size, computational intensity and independency level that can contribute to the choice of hardware system in which the parallel computing is performed. The objective in this chapter is not to explore any new parallel computing techniques, but to apply some of the existing parallel computing techniques executed on either GPUs or CPUs to accelerate all tasks in previous chapters that can be synchronous.

7.1 Parallel Computing Techniques on CPUs and GPUs

In order to implement parallel computing techniques, the hardware system has to supply a computing platform that can execute multiple tasks simultaneously. According to Flynn’s taxonomy, the computing platforms can be categorized into four classes based on the number of concurrent instructions and the number of data streams (Flynn, 1972):

- Single Instruction, Single Data (SISD): no parallelism in either instruction or data stream;
- Single Instruction, Multiple Data (SIMD): multiple data streams can be processed simultaneously under a single instruction;
- Multiple Instructions, Single Data (MISD): multiple instructions can be operated on a single data stream;
- Multiple Instructions, Multiple Data (MIMD): multiple instructions can be executed on multiple data streams.

Among the four classes, SIMD and MIMD have been the two most popular computing platforms so far. SIMD is the architecture used in modern GPUs, while most of modern CPUs belong to the class of MIMD.

With respect to the domain of CPU-based processes, there are two parallel computing techniques representing different levels of parallelism. In a single CPU Core system, Multiple threads can be created and share the resources on that core, which is known as Multi-Threading (MT) Technology. The MT Technology designates the shared execution engine to work on the threads simultaneously by interleaving threads to keep the CPU core as busy as possible so that maximum available resources can be used by threads. Nevertheless, only the instruction-level parallelism can be achieved via MT technology, and the performance gains merely come from the latency hiding. Unlike MT technology, Multi-Core (MC) Technology achieves true parallelism among threads by providing multiple and independent CPU cores on the same chip. With such architecture, threads from different CPU cores are not interleaved but executed independently at the same time. In this way, thread-level parallelism is attained (Akhter and Roberts, 2006). In some cases, the combination of MT technology and MC technology, which means to configure multiple threads on each CPU core, can be attempted for even faster performances.

GPU was originally invented to increase the speed solely for graphical computations. Based on the developments over decades, the modern GPUs now can be massively parallel with tremendous computational horsepower and suitable as well as programmable for various purposes. This brings about the intensive usage of GPUs in many fields, including Computational Finance, Manufacturing, Deep Learning, Animations, Fluid Dynamics and other industries which require intensive computations (NVidia, 2015). Currently, many types of GPU devices have been commercialized as a serial of products supplying diverse levels of computability. For example, some high level GPUs can achieve over than 5000 GFLOPS (a billion of floating-operations per second), while some low level GPUs can achieve less than 100 GFLOPS (NVidia, 2015).

Compared to CPUs, GPUs can reach much higher GFLOPS on average, which is the fancy feature of GPUs. Furthermore, more computing powers can be assigned when the hardware system is equipped with multiple GPU devices. As Big Data has been more and more concerned in recent years, GPUs with massive parallelism are being recognized as the future of computing. For efficient use of GPUs, there could be many parallel computing tactics to access GPUs. Some general techniques, such as GPU Streaming and Multi-GPU techniques, are adopted in this chapter.

7.2 Implementation of Parallel Computing Techniques

In this section, parallel computing techniques for the three detection subsystems are discussed respectively.

7.2.1 Parallel Computing for Fully Automated Cracking Detection Subsystem

3D Shadow Simulation is the most expensive procedure of the fully automated detection. The simulated light starting at a pixel needs to travel across the 3D image until it is out of the image. Such a process has to be repeated at each pixel. The computations become even more intensive when a lower projection angle is used. In order to reduce the processing time significantly, parallel computing techniques on a single GPU device is applied for the speedup of 3D Shadow Simulation. Each pixel is associated with a GPU thread. As the maximum light height at each pixel needs to be updated once when this pixel is visited by the light starting from another pixel, the *atomic max* function is used to avoid Race Condition (An undesirable situation occurs when two or more writing operations are executed at the same address). Table 7.1 lists the improvement by GPU techniques, where the processing time is the average time of trials on 10 different 3D images, and the tested GPU device is NVidia GeForce GT 750M (Computability 3.0). It should be noted the measured time for the succeeding algorithms discussed later in this section are all the average time of trials on 10 different 3D images. In addition, the 10 images tested for both of the two cracking detection subsystems are all different from the 10 images used for noisy pattern detection subsystem.

Table 7.1.Time Improvement on 3D Shadow Simulation by Single GPU

Image Size	Processing Time (milliseconds)		Speedup Factor
	Single CPU Thread	Single GPU (No Streaming)	
4096×2048	352	77	4.57

After 3D Shadow Simulation, the Blob Extraction is implemented on a single CPU thread. No parallel computing techniques are utilized because Blob Extraction is a highly dependent procedure. However, Multi-Core Techniques are applied after the clustering via Blob Extraction is completed. The cracking clusters are divided into several subsets. Each subset is examined by the noise suppressing algorithms discussed in Section 3.2.2 on each CPU core independently. Table 7.2 illustrates the time improvement achieved by Multi-Core Techniques, where 12 identical CPU cores (Intel Core^{TM} I7-4930K @3.40 GHz) are involved.

Table 7.2. Time Improvement on Noise Suppressing Algorithms in Fully Automated Subsystem by Multi-Core Techniques

Image Size	Processing Time (milliseconds)		Speedup Factor
	Single CPU Thread	12 CPU Cores	
4096×2048	215	23	9.35

The overall time improvement on fully automated subsystem is shown in Table 7.3.

Table 7.3. Overall Time Improvement on Fully Automated Subsystem by Combined Parallel Computing Techniques

Image Size	Processing Time (milliseconds)		Speedup Factor
	Single CPU Thread	Combined Parallel Computing Techniques	
4096×2048	580	113	5.13

7.2.2 Parallel Computing for Interactive Cracking Detection Subsystem

In the interactive cracking detection subsystem, Line Scanning is the most expensive procedure. However, the distinct advantage of Line Scanning is that each scan line is independent. Furthermore, each pixel in Line Scanning can be technically independent as the search of a cracking profile can be conducted independently at each of the pixels belong to that cracking profile, although the search is repeated at these pixels. Therefore, GPU-based parallel computing is very suitable for Line Scanning in either transverse or longitudinal direction. Instead of assigning a GPU thread to each independent line, each pixel is assigned with a GPU thread to take advantage of the huge number of GPU threads. Similarly, each pixel in Thinning can also be technically independent as the thinning of each cracking profile can be repeated at every pixel belongs to that cracking profile. Therefore, the Line Scanning and Thinning can be implemented on the GPU device in a sequential order. Furthermore, the combination of GPU Streaming and Multi-GPU techniques are utilized to provide a two-layer GPU-based computing platform, as illustrated in Figure 7.1.

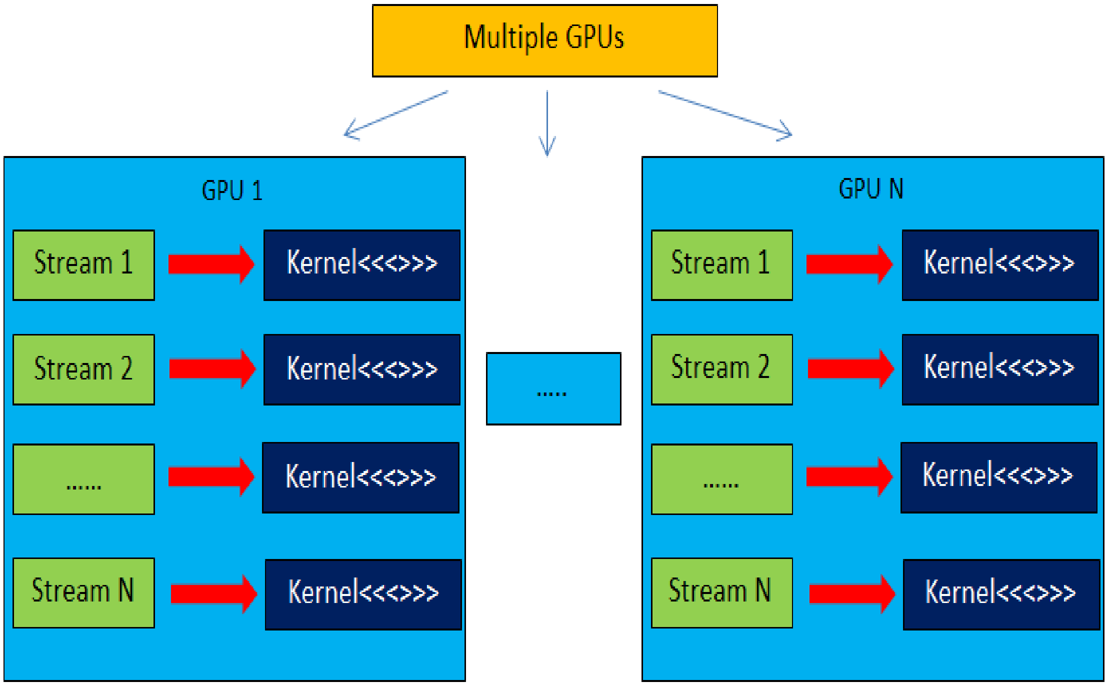


Figure 7.1. Two-Layer GPU-based Computing Platform

First of all, the independent scan lines are evenly divided into $N_{GPU} \times N_{Stream}$ subsets, where N_{GPU} is the number of GPU devices to be used and N_{Stream} is the number of streams to be designated on each GPU device. Then, every pixel in each subset is assigned with a GPU thread. Finally, Multi-Core technology is applied to trigger the multiple GPU devices and handle data transfers simultaneously. The time improvement by such a computational scheme is illustrated in Table 7.4, where 2 identical GPU devices (NVidia GeForce GTX TITAN Black, Computability: 3.5) and 2 identical CPU cores (Intel Core^{TM} I7-4930K @3.40 GHz) are tested.

Table 7.4. Time Improvement on Line Scanning and Thinning by Multi-Core & Two-Layer GPU-based Techniques

Image Size	Processing Time (milliseconds)		Speedup Factor
	Single CPU Thread	2 CPU Cores + 2 GPUs + 4 Streams on Each GPU	
4096×2048	628	28	22.42

Similarly, Blob Extraction is used to label cracking clusters after Line Scanning and Thinning, for which no parallel computing are conducted. Afterwards, the rest procedures are analyses on cracking clusters. Therefore, Multi-Core techniques are applied for the rest procedures using the same partition method mentioned in Section 7.2.1. Consequently, the reduction on processing time of the rest procedures is shown in Table 7.5, where 12 identical CPU cores (Intel Core^{TM} I7-4930K @3.40 GHz) are used.

Table 7.5. Time Improvement on Noise Suppressing Algorithms in Interactive Subsystem by Multi-Core Techniques

Image Size	Processing Time (milliseconds)		Speedup Factor
	Single CPU Thread	12 CPU Cores	
4096×2048	279	26	10.73

The overall time improvement on interactive cracking detection subsystem is listed in Table 7.6.

Table 7.6. Overall Time Improvement on Interactive Subsystem by Combined Parallel Computing Techniques

Image Size	Processing Time (milliseconds)		Speedup Factor
	Single CPU Thread	Combined Parallel Computing Techniques	
4096×2048	924	71	13.01

7.2.3 Parallel Computing for Noisy Pattern Detection Subsystem

The training on SVM classifiers for the recognition of pavement joints and grooves are completed offline. Thus, there is no strong reasons to accelerate the training by parallel computing techniques. However, the pattern extraction procedures discussed in Section 6.5 needs to be accelerated so as to increase the real-time efficiency. First of all, 3D Shadow Simulation is accelerated by the GPU-based techniques as mentioned in Section 7.2.1. Next, the combination of Multi-Core and Multi-Threading techniques is organized to reduce the processing time of the rest procedures, including regional data sampling, regional Blob Extraction, feature extraction, classification and trend-lines preservation. Table 7.5 lists the achieved time improvement, where 12 identical CPU cores (Intel Core™ I7-4930K @3.40 GHz) and 2 CPU threads are involved.

Table 7.7. Time Improvement on Remaining Procedures after 3D Shadow Simulation in Noisy Pattern Detection Subsystem by the Integration of Multi-Threading and Multi-Core Techniques

Image Size	Regional Sample Size	Sample Number	Processing Time (milliseconds)		Speedup Factor
			Single CPU Thread	12 CPU Cores + 2 CPU Threads	
4096×2048	256×2048 or 2048×256	16	541	62	8.73

Similarly, Table 7.8 illustrates the overall time improvement on noisy pattern detection subsystem achieved by the combination of multiple parallel computing techniques.

Table 7.8. Overall Time Improvement on Noisy Pattern Detection Subsystem by Combined Parallel Computing Techniques

Image Size	Processing Time (milliseconds)		Speedup Factor
	Single CPU Thread	Combined Parallel Computing Techniques	
4096×2048	917	151	6.07

7.3 Summary

In a summary, several combinations of CPU-based and GPU-based parallel computing techniques are organized in this chapter to reduce the processing time of the three detection subsystems significantly. For fully automated cracking detection subsystem, parallel computing techniques on single GPU device and Multiple CPU cores are employed, by which an overall speedup factor 5.13 is attained. With respect to the interactive cracking detection subsystem, Multi-GPU and GPU Streaming are integrated as a two-layer GPU-based computing structure which yields a satisfactory speedup about 20 times faster on Line Scanning and Thinning. Later on, Multi-Core technology leads other procedures to a performance which is nearly 10 times faster. The overall achieved speedup factor for the interactive cracking detection subsystem is 13.01. Finally, the integration of Multi-Threading and Multi-Core is deployed for all the procedures except 3D Shadow Simulation in the noisy pattern detection subsystem. Including the time for 3D Shadow Simulation, the noisy pattern detection subsystem is roughly 6 times faster with the assistance from parallel computing techniques.

The measured time in this chapter is based on specified 3D data, limited number of trials and particular hardware system. Differences in processing time may be found under a different environment. However, the powerfulness of parallel computing techniques has been clearly stated in the comparison to the use of a single CPU thread.

Chapter 8 CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

The immaturity of pavement cracking detection motivates the study on a practical system of multiple algorithms for pavement cracking detection from a comprehensive perspective. Using 1mm 3D pavement surface data, this study proposes three detection subsystems: Fully Automated Cracking Detection Subsystem, Interactive Cracking Detection Subsystem and Noisy Pattern Detection Subsystem, in order to address various needs in automation levels and detection accuracies on different types of pavement surfaces.

First of all, the Fully Automated Cracking Detection Subsystem is proposed as a fast approach to pavement cracking detection without any human intervention. The Fully Automated Cracking Detection Subsystem utilizes 3D Shadow Simulation to transform the original data to a binary map called as Shadow Map where potential cracking pixels are labeled as “0” and the others are labeled as “1”. Sequentially, a serial of noise suppressing algorithms are proposed to remove noises from the Shadow Map. The performance of the Fully Automated Cracking Detection Subsystem is fully examined by the Precision-Recall Analysis on two real pavement segments. It is demonstrated in the Precision-Recall Analysis that the Precisions on both segments are above 90 percent, while the Recalls on both segments are around 90 percent.

This indicates a high level of success in cracking detection when the complexity and diversity of pavement surfaces are considered. The significant advantage of 3D Shadow Simulation is its simplicity as it approximates and simplifies the projection of natural lights. In addition to its simplicity, 3D Shadow Simulation can provide diverse solutions by adjusting a single parameter: the projection angle, which makes the cracking detection feasible for various pavement surfaces and leads to a simpler control on the sensitivity of the detection. In order to increase the accuracy of the Fully Automated Cracking Detection Subsystem, NSGA-II which is a popular and efficient Multi-Objective Genetic algorithm is employed in this study to optimize the parameters of the fully automated algorithms. In addition to NSGA-II, APM is proposed in this study as a complementary procedure that can be integrated with NSGA-II for the pursuing of faster convergence and better diversity. It is demonstrated in the comparison study on five ZDT benchmark functions that the integration of NSGA-II and APM yields a much faster progress to the pareto-optimal front than using NSGA-II alone. According to the Precision-Recall Analysis on the same pavement segments using the optimized parameters, slight improvements on both Precision and Recall are found.

Secondly, Interactive Cracking Detection Subsystem is proposed in this study as a slower but more confident as well as flexible approach to pavement cracking detection. This subsystem uses Minimal Contrast as the single primary parameter that can be trained by the users through the monitoring on the immediate detection results. The interactive mechanism is fulfilled as a process of parameter adjustments according to the user’s feedback. To be more specific, a computer program is developed to provide an interface connecting the operator and the interactive algorithms. Once the operator tells the algorithms if the current detection results are false-positive, false-negative or acceptable, the algorithms will respond accordingly. Based on the computer program, a two-level detection can be managed. Automated Detection, which is the bottom level, can be used to detect the majority of cracks automatically after the operator has selected an appropriate Minimal Contrast for the sections being analyzed. Such a process can be repeated if the results are not acceptable from the operator’s point of view. In complement to Automated Detection, Assisted Detection, which is the top level, can be applied to retrieve undetected cracks and remove unwanted patterns within any user-defined region in a more subtle but much slower way. It has been demonstrated in the case study that the integration of Automated Detection and Assisted Detection can eventually locate nearly all of the cracks with almost no noises. The disadvantage of Interactive Cracking Detection Subsystem is the inevitable consumptions on time. In a general view, higher the requirement on detection accuracy is, longer the processing time will be. However, the Interactive Cracking Detection Subsystem indeed supplies a flexible and confirmative approach to finding almost 100 percent cracks without any noises. The significant implication of the Interactive Cracking Detection Subsystem is the detection of desired accuracy level can be guaranteed.

Thirdly, the Noisy Pattern Detection Subsystem is proposed for cracking detection on rigid pavements. Pavement joints and grooves are the two major noisy patterns on rigid pavements that seriously affect the detection accuracy. The four classifiers for the recognitions of transverse groove, transverse joint, longitudinal groove and longitudinal joint respectively are trained by RBF based SVM algorithms through the learning of numerous training samples. In order to extract feature efficiently, 3D Shadow Simulation is adopted to transform the original samples to binary maps where pavement joints and grooves may be preserved in a large portion. In this way, the complexity of the original data can be reduced significantly, and the features extracted from the binary maps could be more useful. It has been manifested in Chapter 6 that the four SVM classifiers all achieve high accuracies in classifying both training samples and testing samples. Based on the four classifiers, pattern extraction procedures are proposed to find the exact locations of pavement joints and grooves from the sampling area according to the classification outputs. From the Precision-Recall Analysis on the 1000 testing samples, it can be concluded that the proposed noisy pattern detection procedures generate a sufficiently high level of Precision and Recall. Once there are any pavement joints and grooves have been detected, they will be excluded from the cracking detection. In this way, the cracking detection accuracy on rigid pavements will be greatly improved.

Finally, numerous combinations of CPU-based and GPU-based parallel computing techniques are implemented to increase the processing speed of all computational tasks involved in this study that can be synchronous. It can be observed in Chapter 7 that most of the computational tasks gain a speedup by a factor over than 5.

8.2 Future Work

The Transverse Lighting Angle and Longitudinal Lighting Angle are the two primary parameters used in the Fully Automated Cracking Detection Subsystem. They have been optimized using NSGA-II with APM through the evaluations on the overall Precision and Recall of the detections on all selected 3D images. In other words, the optimized parameters may not be best parameters for some of the 3D images. There is lack of a robust procedure to determine the best parameters for each 3D image.

Similarly, the Minimal Contrast involved in the Interactive Cracking Detection Subsystem could also have different optimal values for different 3D images. No automated procedure is used to find the best Minimal Contrast for each 3D image during the Automated Detection. As a consequence, more time might be spent on the Assisted Detection in order to reach an acceptable solution.

Finally, the 3D Shadow Simulation is also involved in the Noisy Pattern Detection Subsystem. And the best lighting angles for different samples are still varied without certainty. Although a low lighting angle (45 degree) is fixed in this study and can serve to preserve a large portion of joint and groove patterns in many cases, it still is not be the best option for every 3D sample and thus could not contribute much to the adaptability of the algorithm.

All in all, a robust adaptive algorithm needs to be developed for the automated determination of optimal parameters for every individual 3D image. A recommendation for such an algorithm is to estimate the percentage of noises and the percentage of true patterns in a stochastic way. Furthermore, the original image can even be partitioned into several subsets. Then, the best parameters are estimated and applied independently for each subset. If such an adaptive procedure can be attained, the cracking detection on various pavement surfaces will be more successful.

REFERENCES

AASHTO (2013), Standard Practice for Quantifying Cracks in Asphalt Pavement Surface from Collected Images Utilizing Automated Methods, American Association of State Highway and Transportation Officials, Washington, D.C.

Akhter, S. & Roberts, J. (2006), *Multi-Core Programming: Increasing Performance through Software Multi-threading*, Intel Press.

Banks, J. (2002). *Introduction to transportation engineering*, 2nd Ed., McGraw-Hill, New York.

Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992), A Training Algorithm for Optimal Margin Classifiers, in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, ACM Press, 144-152.

Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1--27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

Chang, C. C. & Lin, C. J. (2011), LIBSVM : A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011, Software available at: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Cheng, H.D. & Miyojim, M. (1998), Automatic Pavement Distress Detection System, *Information Sciences*, 108(1-4), 219-240.

Cheng, H.D., Chen, J-R., Glazier, C. & Hu, Y.G. (1999), Novel Approach to Pavement Cracking Detection Based on Fuzzy Set Theory, *Journal of Computing in Civil Engineering*, 13(4), 270-280.

Cheng, H. D., Wang, J., Hu, Y.G., Glazier, C., Shi, X. J. & Chen, X. W. (2001), Novel Approach to Pavement Cracking Detection Based on Neural Network, in *Transportation Research Record 1764*, TRB, National Research Council, Washington, D.C., 119-127.

Cheng, H.D., Shi, J. & Glazier, C. (2003), Real-time Image Thresholding Based on Sample Space Reduction and Interpolation Approach, *Journal of Computing in Civil Engineering*, 17(4), 264-272.

Coello, C. A. C., Lamont, G. B. & Veldhuizen, D. A. V. (2007), *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer Science + Business Media, LLC, New York, NY.

Cord, A. & Chambon, S. (2010), Automatic Road Defect Detection by Textural Pattern Recognition Based on AdaBoost, *Journal of Computer-Aided Civil and Infrastructure Engineering*, 27(4), 244-259.

Daniel, A. & Preeja, V. (2014), A Novel Technique for Automatic Road Distress Detection and Analysis, *International Journal of Computer Applications*, 101(10), 18-23.

Davis, J. & Goadrich, M. (2006), The Relationship between Precision-Recall and ROC Curves, in *Proceedings of the 23rd international conference on Machine learning*, *ACM, Pittsburgh*, 233-240.

Deb, K. (2001), *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, Chichester, UK.

Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. (2002), A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

Fawcett, T. (2006), An introduction to ROC analysis, *Pattern Recognition Letters*, 27(8), 861-874.

Flynn, M. (1972), Some Computer Organizations and Their Effectiveness, *IEEE Transactions on Computers*, C-21(9), 948-960.

Fu, P., Harvey, J., Lee, J., & Vacura, P. (2011), New Method for Classifying and Quantifying Cracking of Flexible Pavements in Automated Pavement Condition Survey, *Transportation Research Record 2225*, Transportation Research Board, Washington, D. C., 99-108.

Gavilan, M., Balcones, D., Marcos, O., Llorca, D.F., Sotelo, M.A., Parra, I., Ocana, M., Aliseda, P., Yarza, P. & Amirola, A. (2011), Adaptive Road Crack Detection System by Pavement Classification, *Sensors Journal*, 11(10), 9628-9657.

Grama, A., Gupta, A., Karypis, G. & Kumar, V. (2003), *Introduction to Parallel Computing*, Second Edition, Addison Wesley, Boston.

Haas, R. & Hudson, W. (1978), *Pavement management system*, McGraw-Hill, Inc.

Haas, C. & Hendrickson, C. (1990), Computer-based Model of Pavement Surfaces, *Transportation Research Record 1260*, Transportation Research Board, Washington, D. C., 91-98.

Hoegh, K., Khazanovivh, L., Worel B. J. & Yu, H.T. (2013), Detection of Subsurface Joint Deterioration: Blind Test Comparison of Ultrasound Array Technology with Conventional Nondestructive Methods, *Transportation Research Record 2367,* TRB, National Research Council, Washington, D.C., 3-12.

Huang, J. P., Liu, W.Y. & Sun, X. M. (2014), A Pavement Crack Detection Method Combining 2D with 3D Information Based on Dempster-Shafer Theory, *Journal of Computer-Aided Civil and Infrastructure Engineering*. 29(4), 299-313.

Huang, Y. (1993), *Pavement Analysis and Design*, Prentice Hall, Englewood Cliffs, NJ.

Huang, Y.X. & Xu, B.G. (2006), Automatic Inspection of Pavement Cracking Distress, *Journal of Electronic Imaging*, 15(1), 013017.

Hsu, C. W., Chang, C. C. & Lin, C. J. (2010), *A Practical Guide to Support Vector Classification*, online available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf .

Kaseko, M.S., Lo, Z.P. & Ritchie, S.G. (1994), Comparison of traditional and neural classifiers for pavement crack detection, *Journal of Transportation Engineering*, 120(4), 552-569.

Konak, A., Coit, D. W., Smith, A. E. (2006), Multi-objective Optimization Using Genetic Algorithms: A tutorial, *Reliability Engineering and System Safety*, 91(9), 992-1007.

Lee, B.Y., Kim, Y. Y., Yi, S. & Kim, J. (2013), Automated Image Processing Technique for Detecting and Analysing Concrete Surface Cracks, *Structure and Infrastructure Engineering*. 9(6), 567-577.

Lee, H. & Kim, J. (2005), Development of a Crack Type Index, *Transportation Research Record 1940*, Transportation Research Board, Washington, D.C., 99-109.

Li, L., Sun, L.J., Ning, G. B. & Tan, S.G. (2014), Automatic Pavement Crack Recognition Based on BP Neural Network, *PROMET-Traffic & Transportation* ,26(1), 19-22.

Li, N., Hou, X. D., Yang, X. Y. & Dong, Y. F. (2009), Automation Recognition of Pavement Surface Distress Based on Support Vector Machine, *Second International Conference on Intelligent Networks and Intelligent Systems*, Tianjin, China, 346-349.

Lu, H. & Yen, G. G. (2003), Rank-density-based Multiobjective Genetic Algorithm and Benchmark Test Function Study, *IEEE Transactions on Evolutionary Computation*, 7(4), 325-343.

Mathavan, S., Rahman, M.M. & Kamal, K. (2012), Application of Texture Analysis and Kohonen Map for Region Segmentation of Pavement Images for Crack Detection, *Transportation Research Record 2304*, TRB, National Research Council, Washington, D.C., 150-157.

Mcghee, K. (2004), NCHRP synthesis 334: automated pavement distress collection techniques, a synthesis of highway practice, *National Academy Press*., Washington, D. C.

Morian, D., Stoeffels, S. & Firth, D. (2002), Quality Management of Pavement Performance Data, *2002 Pavement Evaluation Conference*, Roanoke, VA.

Moussa, G. & Hussain, K. (2011), A new technique for automatic detection and parameters estimation of pavement crack, *4th International Multi-Conference on Engineering and Technological Innovation*, Orlando, Florida, 11-16.

Nejad, M. F. & Zakeri, H. (2011), An Expert System based on Wavelet Transform and Radon Neural Network for Pavement Distress Classification, *Expert Systems with Applications*, 38(6), 7088-7101.

Nguyen, T. S., Avila, M. & Begot, S. (2009), Automatic Detection and Classification of Defect on Road Pavement Using Anisotropy Measure. *17th European Signal Processing Conference*, Glasgow, UK, 617-621.

Nishikawa, T., Yoshida, J., Sugiyama, T. & Fujino, Y. (2012), Concrete Crack Detection by Multiple Sequential Image Filtering, *Journal of Computer-Aided Civil and Infrastructure Engineering*. 27(1), 29-47.

NVidia (2015), *GPU-Accelerated Applications*, NVidia, Santa Clara, CA. Online available at: http://www.nvidia.com/content/gpu-applications/PDF/GPU-apps-catalog-mar2015.pdf .

NVidia (2015), *CUDA C Programming Guide*, NVidia, Santa Clara, CA. Online available at: https://docs.nvidia.com/cuda/cuda-c-programming-guide/ .

Paterson, W. & Scullion, T. (1990), Report INU77: Information systems for road management: draft guidelines on system design and data issues, *The World Bank*., Washington, D. C.

Paterson, W. (1994), Proposal of Universal Cracking Indicator for Pavements, *Transportation Research Record 1455*, Transportation Research Board, Washington, D. C., pp. 69-75.

Peng, B., Wang, K.C.P. & Chen, C. (2014), Automatic Crack Detection by Multi-Seeding Fusion on 1mm Resolution 3D Pavement Images, *Second Transportation & Development Congress, ASCE*, Orlando, FL, 543-552.

Raman, M., Hossain, M., Miller, R., Cumberledge, G., Lee, H. & Kang, K. (2004), Assessment of Image-based Data Collection and The AASHTO Provisional Standard for Cracking on Asphalt-surfaced Pavement, *Transportation Research Record 1889*, Transportation Research Board, Washington, D. C., 106-115.

Shapiro, L.G. & Stockman, G.C. (2001), *Computer Vision*, Prentice Hall, Upper Saddle River, New Jersey.

Timm, D. H. & McQueen, J. M. (2004), *A Study of Manual vs. Automated Pavement Condition Surverys*. Highway Research Center, Auburn University, Auburn, Alabama.

Transportation Research Board (TRB) (2001), Fulfilling the Promise of Better Roads, a Report of the TRB Long-term Pavement Performance Committee. *Transportation Research Board*, Washington, D. C.

Tsai, Y-C., Kaul, V. & Mersereau, R.M. (2010), Critical Assessment of Pavement Distress Segmentation Methods, *Journal of Transportation Engineering*, 136(1), 11-19.

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. *Journal of Transportation Engineering*, 138(5), 649-656.

Tsai, Y-C., Wu, Y., Ai, C. & Pitts, E. (2012), Critical Assessment of Measuring Concrete Joint Faulting Using 3D Continuous Pavement Profile Data, *Journal of Transportation Engineering*, 138(11), 1291-1296.

Vapnik, V. (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, NY.

Wang, K.C.P. (2000), Design and implementation of automated systems for pavement surface distress survey. *ASCE Journal of Infrastructure Systems*, 6(1), 24-32.

Wang, K.C.P. (2011), Automated Survey of Pavement Distress Based on 2D and 3D Laser Images, *Mack-Blackwell Transportation Center*, University of Arkansas.

Wang, K.C.P., Hou, Z. & Williams, S. (2011), Precision Test of Cracking Surveys with the Automated Distress Analyzer, *Journal of Transportation Engineering*, 137(8), 571-579.

Wang, K.C.P. & Elliott, R. (1999), Investigation of Image Archiving for Pavement Surface Distress Survey, *Mack-Blackwell Transportation Center*., University of Arkansas.

Wang, K.C.P., Li, Q. & Gong, W. (2007), Wavelet-Based Pavement Distress Image Edge Detection with à Trous Algorithm, in *Transportation Research Record 2024*, TRB, National Research Council, Washington, D.C., 73-81.

Wang, K.C.P., Li, L., Li, Q., Scofield, L.A. & Tikalsky, P. (2014), Automated Groove Identification and Measurement for Next Generation Concrete Surface (NGCS) Using 3D

Pavement Data at 1 mm Resolution, *Transportation Research Board 93rd Annual Meeting*, Washington D.C., 14.

Wang, K.C.P., Li, L. & Li, Q. (2014), Automated Joint Faulting Measurement Using 3D Pavement Texture Data at 1 mm Resolution, *Second Transportation & Development Congress, ASCE*, Orlando, FL, 498-510;

Wang, Q. & Davis, J. (2013), Airport pavement groove identification and analysis at NAPTF, *Advanced Materials Research*, 723, 1003-1010.

Wang, Q. & Hayhoe, G.F. (2013), Development and Implementation of a Beam-Bridging Filter for Use in Airport Groove Identification, *Transportation Research Record 2369*, TRB, National Research Council, Washington, D.C., 95-103.

Yen, G. G. & Lu, H. (2003), Dynamic Multiobjective Evolutionary Algorithm: Adaptive Cell-based Rank and Density Estimation, *IEEE Transaction on Evolutionary Computation*, 7(3), 253-274.

Ying, L. & Salari L. (2010), Beamlet Transform-Based Technique for Pavement Crack Detection and Classification, *Journal of Computer-Aided Civil and Infrastructure Engineering*, 25(8), 572-580.

Zalama, E., Gomez-Garcia-Bermejo, J., Medina, R. & Llamas, J. (2014), Road Crack Detection Using Visual Features Extracted by Gabor Filters, *Journal of Computer-Aided Civil and Infrastructure Engineering*, 29(5), 342-358.

Zhang, A., Li, Q., Wang, K.C.P. & Qiu, S. (2013), Matched Filtering Algorithm for Pavement Cracking Detection, in *Transportation Research Record 2367*, TRB, National Research Council, Washington, D.C., 30-42.

Zitzler, E., Deb, K. & Thiele, L. (2000), Comparison of Multiobjective Evolutionary Algorithms: Empirical results. *Evolutionary Computation*, 8(2), 173-195.

Zitzler, E., Laumanns, M. & Thiele, L. (2001), *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Swiss Federal Institute Technology, Zurich, Switzerland.

Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: a Comparative Case Study and the Strength Pareto Approach, *IEEE Transaction on Evolutionary Computation*, 3(4), 257-271.

VITA

AONAN ZHANG

Candidate for the Degree of

Doctor of Philosophy

Thesis: A NOVEL SYSTEM OF PAVEMENT CRACKING DETECTION ALGORITHM USING 1MM 3D SURFACE DATA

Major Field: Civil Engineering

Biographical:

Education:

Completed the requirements for the Doctor of Philosophy in Civil Engineering at Oklahoma State University, Stillwater, Oklahoma in December, 2015.
Completed the requirements for the Master of Science in Civil Engineering at Southwest Jiaotong University, Chengdu, China in 2010.
Completed the requirements for the Bachelor of Science in Civil Engineering at Southwest Jiaotong University, Chengdu, China in 2008.

Experience:
May 2012-present: Research Assistant, Oklahoma State University