DEVELOPMENT OF AN AUTOMATIC DETECTION SYSTEM FOR MEASURING PAVEMENT CRACK DEPTH ON FLORIDA ROADWAYS

Written by:

Jian John Lu, PhD., P.E.
Associate Professor

Xiaoyu Mei
Research Assistant

and

Manjriker Gunaratne, Ph.D., P.E.
Professor

Transportation Program
Department of Civil and Environmental Engineering
University of South Florida
Tampa, Florida
(813) 974-5817

Presented to:
Florida Department of Transportation

A Project Sponsored by:
Florida Department of Transportation

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Cracking has an adverse effect on pavement performance, and hence it is an important criterion for maintenance intervention. However, statistically reliable detection of the extent of cracking can also be one of the major difficulties encountered when implementing a pavement management system. The main difficulty is that there is no reliable way to directly obtain the information on pavement crack depth without destructing the pavement structure. This report summarizes a research project sponsored by Florida Department of Transportation (FDOT) to develop a system that can automatically and dynamically measure pavement cracks and estimate crack depth without destructing the pavement structure. The principle used in the project was to use high-accuracy laser sensors to measure the crack opening geometric including crack width, crack edge slopes, and measurable crack depth. With these obtained data and a neural network model developed in the project, the depth of the crack can be statistically estimated. Based on the evaluation results, it was found that the system developed in the project can detect pavement crack depth with a statistically reliable accuracy.

To detect crack depth, two steps are needed: crack identification and crack depth estimation. At the early stage of the project, several approaches were proposed. These proposed approaches included static ultrasonic method, dynamic ultrasonic sensor method, radar sensor method, ground penetrating radar method, and so on. Based on preliminary laboratory experiments and literature search and review, it was concluded that these methods were not able to detect crack depth with a statistically reliable accuracy or practical applicability. However, it was found that a combination of high-accuracy laser sensors and estimating models could produce satisfactory results.

To dynamically identify a crack, two laser sensors were used to minimize the detection errors. An algorithm called Partial Cross Correlation (PCC) was developed to enhance the crack detection ability. This algorithm was evaluated through field experiments and proven that the detection performance of PCC was much better than the approaches used in past research studies for identifying pavement cracks.

To automatically measure the crack opening geometric, one of the key elements is the measurement of longitudinal displacement of the measuring system. A distance sensor was used to measure the longitudinal displacement. However, field tests found that the sampling rate of the distance sensor had certain effect on the measurement accuracy. Thus, a scan-rate-effect-canceling model was developed based on field experimental data and modeling results. With the model, the accuracy of the longitudinal displacement measurement was significantly improved.

With the obtained crack opening geometric characteristics and the information on the pavement section such as the average daily traffic, pavement life cycle, pavement age, and other pavement related information, a neural network model was used to estimate the crack depth. The database used for the neural network model development was comprised of two parts: one was the distance sensor reading including the geometric characteristics of the crack; the other included pavement related variables. The crack information data were obtained from 95 pavement sections with the system developed in the project and a static ultrasonic measuring system which can statically measure crack depth with a reliable accuracy. The pavement related information data were obtained from a database provided by the FDOT. In the model development, different network architectures and different training algorithms were investigated and tested. An optimal architecture was determined based on the tests of different model architectures.

The system was implemented and installed in a manually operated push-car. This system now can measure crack depth at a walking speed. The current system consists of two high-accuracy laser sensors, a longitudinal distance measuring sensor, a portable computer with an interface to communicate with the sensors, and a comprehensive model software used to estimate pavement crack depth using the readings from these sensors and the pavement section related information. The system can only be used to measure transverse crack depth since the developed neural network model was based on field data of transverse cracks. To measure longitudinal crack depth, longitudinal crack data are needed for modeling purpose.

Although the system is now operated at walking speed, it can be operated in a much faster speed close to 55 mph if adequate modifications are implemented to increase the sampling interval and the sampling rate by using a more powerful computer such as an industry computer.

CHAPTER 1. INTRODUCTION

BACKGROUND

Since the beginning of Pavement Management System (PMS) implementation in the 1970's, one of the key components in a PMS has been the pavement surface condition survey [1, 2]. Cracking performance is one of the pavement condition data that need to be surveyed. Once initiated, cracking increases in severity and extent and allows water to penetrate the pavement. The water will further accelerate the rate of pavement deterioration. Thus, to determine the timing and cost of pavement maintenance, the information on pavement crack condition is needed.

Collection of cracking data is difficult and time consuming because a manual survey has to be involved in the process. Due to the nature of the subjective survey, it is very difficult to obtain results that are accurate, repeatable, and reproducible [3]. Thus, there is a need to automate the cracking survey process to improve safety and achieve more objective and consistent data of pavement cracking. Crack surveys are performed to characterize crack conditions in terms of type, severity, and extent. Compared with other characteristics of the crack, such as location, length, and width, crack depth is the most difficult one to be detected. On the other hand, the most serious problem affecting the service life of a pavement is the formation of cracks due to disruptive stresses from the original design or unexpected chemical, physical, or mechanical loading. Thus, crack depth is a frequently used factor when reconstruction is performed on pavement surface. The assessment technique that can detect crack depth is vital to provide effective maintenance of pavements.

The techniques of detecting pavement cracks have made great progress since later 1980's, and have been implemented on a production basis in last 4 to 5 years. The majority of the production work to date has been performed by Roadwave Corp.'s ARAN/Wise Crax system, IMS's PAVUE system, SES's ROSAN system, and GIE's LASER VISION system [2, 4]. However, the main measurements of these systems only focus on the crack

location, crack length, and crack width. Even though some systems can perform so-called crack depth measurement, few of them can effectively measure the real depth of cracks that are not vertical to the pavement surface. As a result of the literature search and survey performed in the project, it can be concluded that the existing methods for the measurement of pavement crack can be divided into two classes: destructive testing (DT) and nondestructive testing (NDT) methods. The DT method, i.e. core sampling, is both time consuming and resource consuming. Unfortunately, through a survey, it was found that the DT method is nearly the single method adopted by all the state departments of transportation (DOTs) that use crack depth as a factor to evaluate the pavement performance. The NDT methods include: (1) contact methods using impact echo or ultrasound approaches [5, 6, 7, 8], (2) non-contact methods using laser sensors [3, 9], ground-penetrating radar (GPR) [10, 11, 12], and image classification techniques [2].

EXISTING METHODS FOR MEASURING CRACK DEPTH

In the first stage of this project, all of the DOTs in USA were contacted to survey the methods used to measure crack depth in their PMS. All of the available databases and literature were reviewed to evaluate the relevant technologies that are feasible for application in this project. In addition, web sites were searched to identify whether or not existing techniques were available to automatically detect crack depth.

DOT Survey

In the survey, the following questions were asked to all of the DOTs:

- Do you measure pavement crack depth in your department of transportation?
- What methods/systems do you use for pavement crack depth measurement?
- Do you have any comments or suggestions for the automatic measurement of pavement crack depth?

Over 60% of the DOTs in USA responded to the survey, in which five DOTs did measure pavement crack depth for pavement condition evaluation in their PMS. Some other states

rate the severity of cracking solely based on the crack width. It was concluded from the survey that the core sampling method was the only method that was being used by DOTs to collect crack depth for their PMS. From the survey results, it was found that most DOTs agree that it is necessary to automatically detect pavement crack depth.

Literature Review

In this study, various methods were investigated. However, from the literature review, it was found that there are no commercially available systems that can be used to dynamically measure pavement crack depth.

Overview

Currently, there is no satisfactory method to automatically detect pavement surface crack depth. The complexity of the crack creation and propagation mechanism makes the problem more difficult. Nevertheless, increasingly state-of-art technologies have been applied in this field with promising experiment results [4, 6, 9, 11, 13]. It is rather difficult to systematically classify methods for crack measurement due to the different crack shapes and crack dimensions. However, almost all the methods to measure crack depth depend on the following physical phenomena:

- Reflection (with large reflectors)
- Scatter (with small reflectors)
- Diffraction (at the peaks)
- Wave conversion (e.g. on surface cracks)

From these physical phenomena, useful information about crack depth might be evaluated from:

- The echo pulse (shape, spectrum)

- The time of flight

- The probe position (directional characteristics)

Each of the methods introduced previously has different levels of impact on realizing pavement crack depth detection. A detailed introduction will be given in the following sections. The classification and relationship of these methods is illustrated in Fig. 1-1.

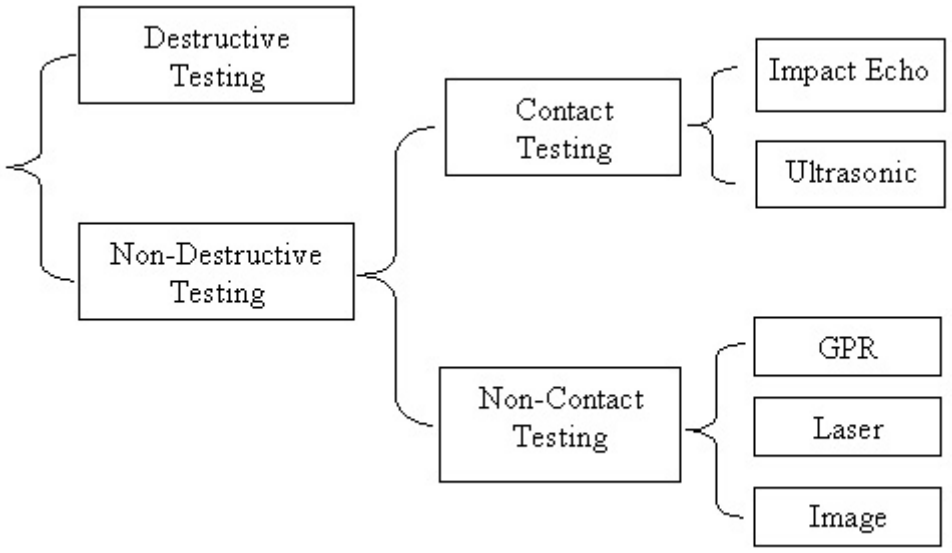


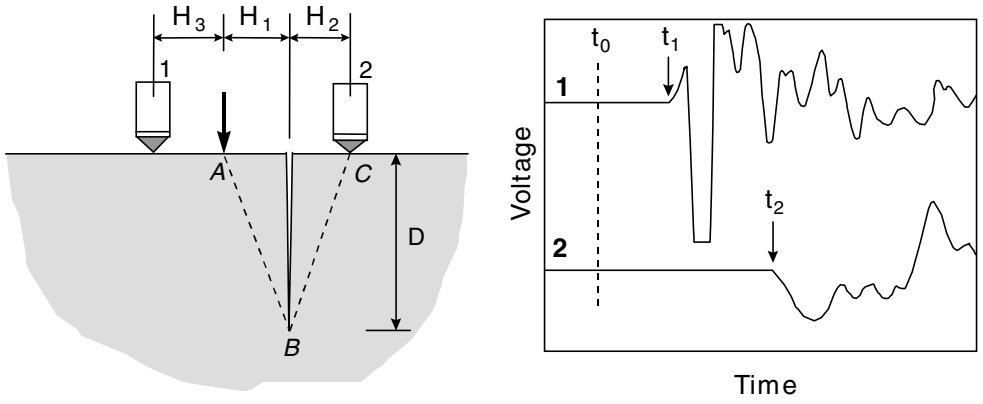
Figure 1-1. Crack Detection Method Classification

Contact Measuring Technology

There are several existing NDT methods that can be used to measure surface opening crack depth. Among these methods, impact echo [6, 7, 8] and ultrasonic pulse echo [5] are widely used. For the impact-echo method as shown in Fig. 1-2, receiving transducers (sensors 1 and 2 shown in Fig. 1-2 (a)) are used to monitor the surface displacements caused by the arrival of wave reflections from internal defects and external boundaries. Recorded displacement waveforms (shown in Fig. 1-2 (b)) can be analyzed either in the time domain or in the frequency domain. Since its development in the mid-1980s, the impact-echo method has been used successfully for measuring the thickness of, and detecting flaws in plate-like structures, such as bridge decks, slabs, and walks. At present,

for the determination of the depth of surface-opening cracks in concrete plates, a relatively mature test scheme is available.

For this method, a transient stress pulse (applied at point A) is introduced into a structure by mechanical impact at a point on the surface. This pulse travels into the structure as dilatational (P-) waves, distortional (S-) waves and along the surface as Rayleigh (R) waves. P-waves are of primary importance in impact-echo testing because the displacements caused by P-waves are much larger than those caused by S-waves at points located close to the impact point. When an impact is performed adjacent to a surface-opening crack, the presence of the crack changes the pattern of wave propagation. When portions of the P-wavefronts are incident on the crack tip, diffraction occurs.



(a) Testing Configuration (b) Output Waveforms from the Transducers

Figure 1-2. Time-of-Flight Technique for Crack Depth Detection

Given the wave speed and the geometric relationship between transducers (H_1, H_2, and H_3), crack depth can be estimated by means of calculating the time-of-flight through a developed equation. For a vertical surface-opening crack, the crack depth can be accurately determined by the method mentioned above. However, if the depth of inclined or curved cracks is to be estimated, the positions of the sensors (H_1, H_2, and H_3) should be adjusted so that the diffracted wave can be clearly identified.

Ultrasonic techniques are well established for measuring wave velocities and transit times, from which the material's elastic properties, sample dimensions, and anomaly properties can be determined. There are many methods such as scattered pulse, time-of-flight, spectroscopic analysis and so on. As a contact NDT technique, ultrasonic technique is the same as impact-echo technique in essentials. At present, there are many commercial systems using these two methods to perform NDT, such as LLC's Impact-Echo Test System and James' V-Meter Mark 2.

Non-Contact Measuring Technology

Ground Penetrating Radar (GPR) is a well-established method of using radio waves to detect objects and determine their distance (range) from echoes they reflect. GPR is a noninvasive and nondestructive tool that has been used to map subsurface conditions in a variety of applications. In recent years, several investigators have attempted to use GPR to detect subsurface problems including cracking [1, 10, 11, 12]. GPR systems generate short pulses of electromagnetic energy that penetrates pavement surface. Reflections from changes between these layers are detected and displayed. The primary components of a GPR system are illustrated in Fig. 1-3. Two types of GPR systems are commercially available: monostatic and bistatic. The former uses a single antenna for transmission and reception. The latter uses two separate but identical antennas.

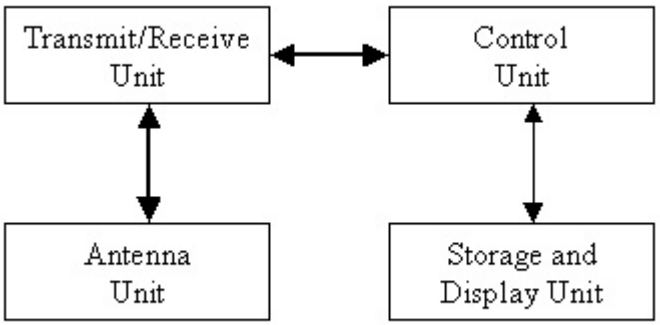


Figure 1-3. Major Components of a GPR System

The transmit/receive unit consists of a transmitter for signal generation, a receiver for signal detection and timing electronics for synchronizing the transmitter and receiver. The control unit is the operator interface that controls the overall operation of the radar

system, sending the received data to the data storage and display unit. At the control unit, the reflected signals are represented by a waveform of voltage changes as a function of time. These waveforms are the signals that are stored and displayed. One display technique is to graphically stack sequential waveforms to create a profile of horizontal distance over the pavement surface as a function of time. This profile is a depiction of impedance changes as a function of horizontal survey travel along the surface of the ground and radar signal travel-time into the ground, and thus, represents an anomaly map in radar space. Fig. 1-4 shows a typical output picture of a commercially available GPR.

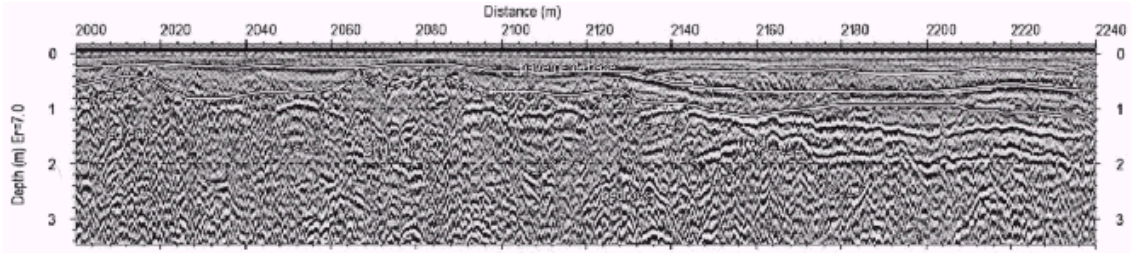


Figure 1-4. Pavement Profile from GPR

To obtain the quantitative information of the cracking, the interpretation of measured reflected waves is a key. Up to now, most of the commercially available GPR cannot provide accurate methods for obtaining detailed information. Thus, the detecting process usually requires the interaction of a skilled user.

Laser has its own advantages such as high accuracy and high measurement speed [3, 4, 9]. In early 1990's, Texas State Department of Highways and Public Transportation developed a laser system to identify pavement cracks. The hardware of the system includes the Selcom Laser Probes, the Motorola open-ended VME architecture, and a Compaq portable personal computer. The software implements two crack detection algorithms and crack reporting procedures. This system uses distance measurement to estimate crack width. The measurement is based on a triangulation principle as illustrated in Fig. 1-5. In order to identify pavement cracks, two algorithms were adopted: Autocorrelation Difference Method (Codiff) and Running-Mean Downup Method (Downup). The Codiff method is more sensitive to the severity of cracking. The Downup method can provide good results for all kinds of cracking levels. The basic idea for the

Downup method is that a crack can usually be identified by a sharp negative slope followed by a positive slope. The algorithm computes the slope of the crack by taking the difference between each averaged point of a base length. The Downup method can estimate both crack width and depth and can be executed sufficiently faster to be suited to real time application. Because the Selcom sensor can provide high-speed light source control, the system affords real time processing at speeds up to 60mph. However, based on the measurement principle mentioned above, the system function is limited by laser's inherent limitation: It cannot detect all crack characteristics.

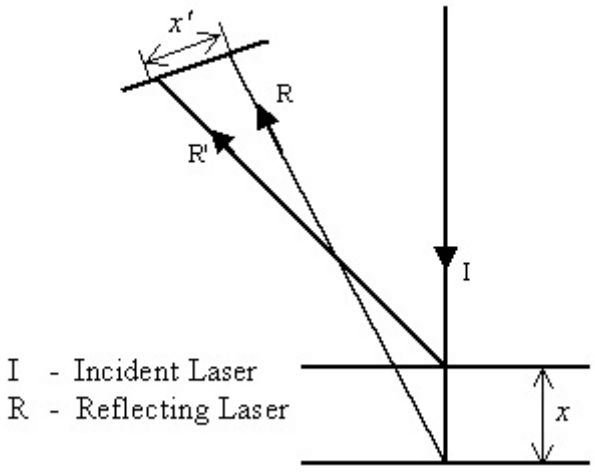


Figure 1-5. Triangulation Measurement Principle

Currently, there are several commercially available laser systems to detect pavement cracks, such as GIE's LASER VISION system and SES's ROSAN system. It is worthwhile to mention the LASER VISION system of GIE (LVI). With a combination of triangulation and defocusing technology, the LVI employs three laser beams to measure the 3-D distress characteristics of road surface at highway speed. To cover a standard highway lane with a width of 3.6m, the system uses six laser sensors to make both transversal measurements and longitudinal measurements. Due to the huge amount of data, a parallel processing architecture array of Digital Signal Processors (DSP) is adopted. The system architecture is shown in Fig. 1-6. In the system, each acquired road profile is processed in real-time and every event detected (e.g. cracks) is stored with its

characteristics: position, width and depth, in which crack and rut data allow a graphical reconstruction of 3-D images of specified road sections.

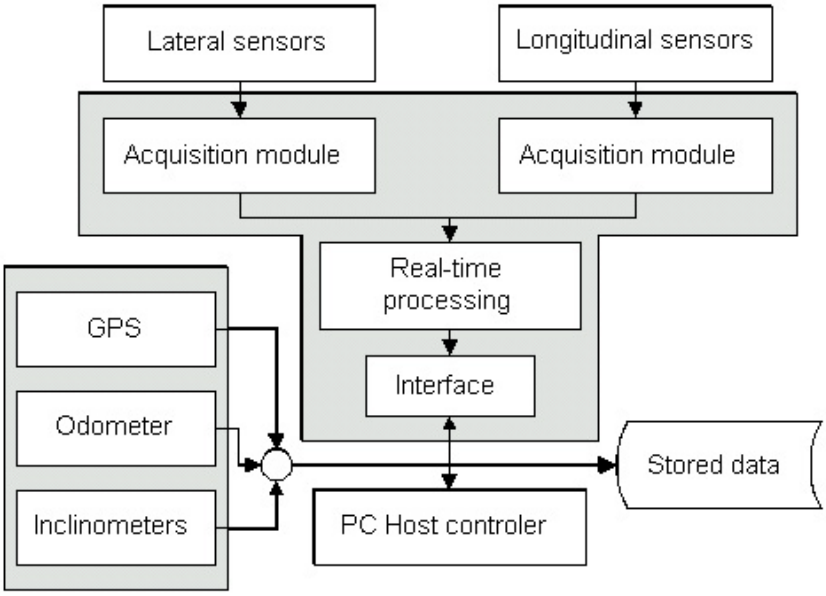


Figure 1-6. Architecture of the Laser Vision System

Image Processing is another crack detection technology that is being widely used [2]. Although the automated systems vary, most employ an image-capture process that uses artificial light and high-resolution cameras to record pavement surface images in either an analog or digital format. If the image is captured in analog format, it will be digitized. The digital image is then analyzed with image processing software that evaluates the changes in pixel brightness, looking for the contrast created by crack edges. Once a crack is detected, then additional image processing logic is used to capture the length of the entire crack and its width. However, up to now, even if with the most avant-garde technology, it is nearly impossible to quantify the crack depth through the changes in brightness.

Summary

A DT method is fundamentally different from a NDT method. The DT method is time-consuming and resource-consuming. Besides, it is difficult to determine the exact location and extent of the coring.

Currently, the NDT method is a popular means for estimating pavement crack depth. For contact-type NDT method, the impact stress or ultrasound methods can estimate crack depth with relatively high accuracy. They can determine the depth of vertical, inclined, curved, and air- or water- filled surface-opening cracks. However, these methods need the accurate value of wave speed in the medium (pavement) and can only be statically operated, which is very time-consuming. Since the wave speed is not an absolute constant in pavement materials, the application of these types of methods is very limited.

As a non-contact NDT method, GPR technologies overcome many inherent shortcomings of the contact methods. With some quick scanning techniques such as color intensity plots (produced by commercial radar software), GPR technologies can enable high-speed field inspection. However, most of the current commercially available systems are limited to estimating the location and gross quantities of the deterioration. Few of them can provide a more accurate method for obtaining detailed information. In many cases, an expert is needed to locate the problem areas. Sometimes, the manual interpretation system is disappointing. A second drawback of GPR technologies is their heavy system configuration, which makes it very expensive. In order to perform the on-site and real-time data interpretation, it is necessary to adopt a new approach to signal processing.

As compared to the rich information about internal features given by GPR, a laser system is limited to analysis of only the visible parts (i.e. the vertical part of a crack). This limitation is due to the laser's short-wave characteristics. However, the descriptive results of a laser sensor are much more clear than the implicit outputs of a GPR. If the accurate data set from a laser sensor is adequately used and models for estimating crack depth are developed, the detection of pavement crack depth will become possible.

Image processing technique is the only way that provides the picture of the crack in its entirety. Perhaps, with the adding of some more effective models, a combined approach can make it possible to provide a comprehensive report about pavement cracks. Nevertheless, among all of the methods mentioned above, image processing is apparently the least effective in detecting crack depth characteristics.

In the field of NDT, most techniques are applied to structural cracks or cracks in metal materials. Although some techniques have been used for crack evaluation of metallic materials, most of these methods are only valid for homogenous and isotropic materials. This study focuses on the cracks in the pavement surface, which is mainly composed of nonmetallic materials such as concrete. It is neither homogenous nor isotropic. But if the method is applied in a low frequency case (the wavelength is much larger than the size of the aggregate particles in pavement) and averaged values are used, pavement may be considered homogenous and globally isotropic materials. Therefore, the NDT techniques can be conditionally applied to pavement applications as measurement references to calibrate the neural network models for estimating pavement crack depth. The details will be discussed later.

RESEARCH OBJECTIVES

The main purpose of this project was to develop an automatic system to estimate pavement crack depth on Florida roadways. The main objectives of this project were:

- To search and review the existing literature databases to identify technologies and methodologies that have the potential for automatic pavement crack depth measurement;
- To perform laboratory and field experiments to further prove the feasibility of selected technologies; and
- To develop a prototype that can be used to measure pavement crack depth on Florida roadways.

RESEARCH APPROACH

Preliminary Experiments

Before the research approach was finalized, a series of preliminary experiments were conducted to select the best detecting method applicable to the system requirements, including sensor accuracy, sensor resolution, sensor detecting speed, sampling rate,

feasibility to measure crack depth, hardware cost, etc. The detailed results of the preliminary experiments were documented in an interim report of this project. Based on the preliminary experimental results, a practically feasible approach was recommended and submitted to FDOT project manager for review and approval. The research approach is described in the following sections.

Approach Description

Although some existing technologies may provide certain possibilities to measure pavement crack depth, due to the special characteristics of the pavement crack and the basic requirements of the detecting system, direct use of these technologies cannot satisfy the feasibility requirements. A feasible way to automatically and dynamically identify a crack and estimate its depth is to use high-speed and high-accuracy sensors to obtain the microscopic characteristics of the crack opening, including width of the crack opening, the downward slopes on both sides of the crack, and the measurable crack depth. With the measurements and pavement related information, a neural network model was used to statistically estimate pavement crack depth. The basic concept of the system architecture design is illustrated in Fig. 1-7.

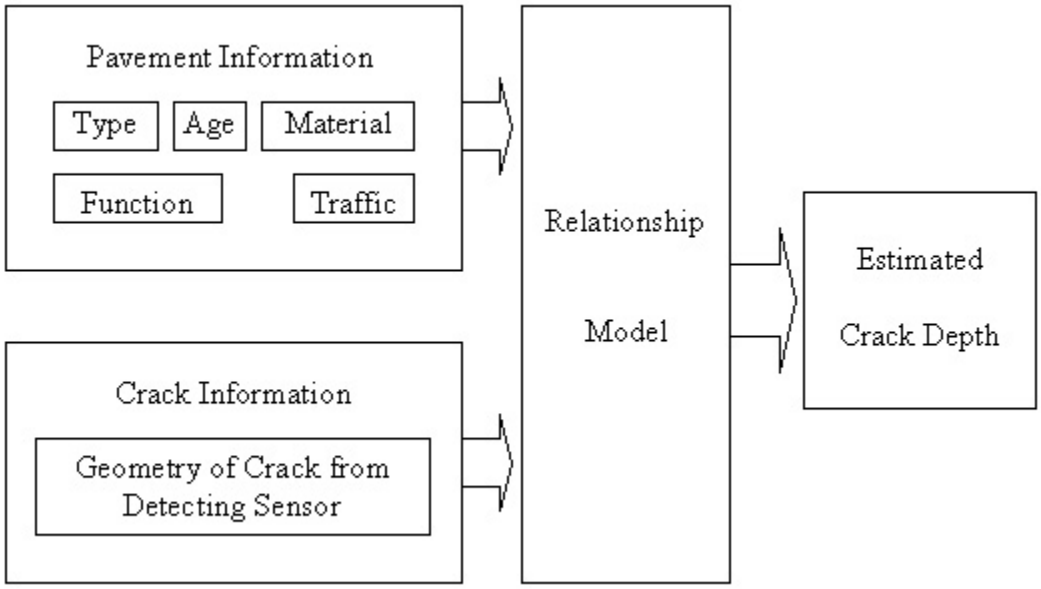


Figure 1-7. System Architecture for Measuring Pavement Crack Depth

Crack Information

Crack information includes width of the crack opening, the downward slopes of both sides of the crack, and the physically measurable crack depth. These data are measured by two laser probes (sensors) and one distance sensor. Several models (algorithms) are used to ensure the accuracy of these data.

Pavement Information

Pavement information includes pavement type, age, materials, roadway function, and average daily traffic. These data can be obtained from FDOT database. The main purpose of using the pavement information is to provide additional information (in addition to crack information) to the relationship model (the model used to statistically estimate crack depth) so that the estimation of crack depth is reliable.

Relationship Model

The relationship model is the one used to statistically estimate crack depth. For this purpose, crack information and pavement information are used as the model input. After careful review and assessments of available model types, the neural network model was selected in the project as the relationship model. Development of the model (including model calibration and training) was performed in the project. To develop the model, two types of data were needed. The first type refers to crack information and pavement information (model inputs), and the second type refers to real crack depth. The first type of data was obtained through the measurements by laser sensors and distance sensor and from the FDOT database. The second type of data was measured by a contact-type NDT system with relatively good accuracy.

Model Development

Development of the model was divided into two phases. In the first phase, two types of data were collected. The first type of data included crack information, such as width of the crack opening, the downward slopes on both sides of the crack, and the measurable crack depth. These data were measured by the two laser probes (sensors) and one

distance sensor. The second type of data included the corresponding crack depth measured by a contact-type NDT system with relatively good accuracy and pavement information including pavement type, age, materials, roadway function, and average daily traffic. In the second phase, these two types of data were used for model development. To develop the models for estimating crack depth, the neural network model was selected. During the model development process, part of data was used for model training, part of data was used for model testing, and the rest of the data was used for model validation. In fact, many neural network model structures were trained and tested to determine the best one in terms of estimating errors. With the final model, the data for validation were used to justify the model quality.

REVIEW OF NEURAL NETWORK APPLICATIONS

In the field of pavement engineering, advances in computer technology have led to the use of more complex, and arguably more sophisticated approaches to develop optimizing strategies using the pavement serviceability-performance concept. A characteristic feature of pavement performance models is the prediction of pavement performance based on a set of easily and cheaply measurable characteristics, which may yield large savings to the agencies responsible for the pavements. Even though researchers traditionally use multiple linear regression techniques, no single prediction model applies to all pavements due to the high variability in the number and type of pavement characteristics measured by each agency for its pavements, as well as the pavement performance indicator. As to the crack evaluation, each agency determines its own criteria for its PMS. Recently, applications of neural networks in this area have become increasingly popular. Many research studies regarding the comparison between the neural network model and the traditional statistical model have been done, and the advantages of the neural network over statistical models were summarized [14]. However, all the neural network applications in pavement performance prediction focus only on the macroscopic effects of the cracks. The modeling of the microscopic characteristics for cracks can be found in the field of material science. Although the crack formation mechanism in material science is very different from that of pavement engineering, from the point of

view of the neural network application, it is also a good reference to this study. The following section summarizes the major achievements in these two different fields.

Macroscopic Models

The application of artificial neural networks (ANN) has been introduced recently in the field of pavement performance prediction. Most studies are concentrated either on the comparison between the ANNs and regression methods or the effect of the parameter and architecture on the prediction performance. Owusu-Ababio investigated the effect of neural network topology on flexible pavement cracking prediction for the Connecticut Department of Transportation (ConnDOT) [15]. The database used for this study was extracted from the distress survey files of the ConnDOT. The network inputs consisted of ESAL, pavement surface age, and pavement surface thickness, where ESAL is a converted form of traffic volume. Analysis of variance conducted to examine the effect of pavement regional location on the amount of cracking indicated no significant differences between the regions with different environmental variables. Based on the analysis, it was concluded that a one-hidden-layer topology might be sufficient in achieving satisfactory results in cracking prediction, and increasing the number of layers might not add any significant benefit to the performance of the model.

Another attempt of applying ANN was made by Attoh-Okine to examine the effect of gradient-descent parameters (namely, momentum and learning rate) on pavement performance models [16]. In this study, real pavement condition and traffic data and specific architecture were used to investigate the effect of learning rate and momentum on backpropagation algorithm neural network trained to predict flexible pavement performance. It was concluded that an extremely low learning rate around 0.001-0.005 combination and momentum between 0.5-0.9 did not give satisfactory results for the specific data set and the architecture used.

Microscopic Models

In the field of non-destructive testing (NDT), neural network models are currently used to evaluate the crack sizing, crack opening load, and crack propagation in various pavement

materials. Zgonc et al incorporated a neural network for crack sizing classification [17]. In his study, a finite element (FE) program generated a set of back - and forward - scattered signals for cracks of various lengths. Data obtained in this way were used to train a neural network classifier that categorized the data according to the crack length. The application of ANN provided improved reliability and produced results in a short time. The results showed that the ANN could be further utilized to estimate crack sizes from the ratio between the reflected ultrasonic signals and through-transmitted signals.

Kang et al developed a one-hidden-layer neural network to determine the crack opening load from differential displacement signal curves, in which 100 data points of the signal were used as the network inputs [18]. In order to examine the measurement accuracy and precision of the neural network method, computer simulation was extensively performed for various combinations of crack opening levels and signal-to-noise ratios.

Issues

From these studies described above, it can be seen that ANN is an applicable tool for predicting pavement performance macroscopically and microscopically. On the other hand, it is evident that most applications use the backpropagation method to train the networks with one-hidden-layer architecture. With different applications and databases, parameters of the neural network models will vary. Neural network models have been known to produce wide variations in their predictive properties for even small changes in the network design. Therefore, it is necessary to enhance the performance as much as possible. Although early work shows that the application of neural network did yield much more improvement in the related areas, some problems do exist as described in what follows:

- *When should the training be stopped?* Using a high-speed computer, it will take just a short time to train the network for thousands of epochs. With continual network training and suitable algorithms, the error rate may be reduced to any desired minimum. However, in this situation, the network can become overfitting to the training set data. Further, it may cause bad performance on new data. In most of the past studies, the network training was stopped when a desired

accuracy was attained. In fact, it is impossible to know what accuracy the network will gain. Therefore, this kind of pre-determined accuracy will usually cause either overfitting or under-fitting.

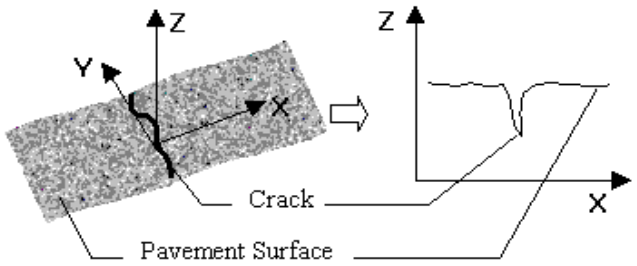
- *Is it necessary to do the comparison between different algorithms?* Although the standard backpropagation method is widely used, it is not the only choice. Compared with many new algorithms, its convergence is very slow. It is also tedious to tune the learning rate and the momentum in practice.

- *Is the data preprocessing necessary?* According to the literature review, it can be seen that many applications directly use the original data as network inputs. The quality of the database may limit the network performance. Especially in macroscopic modeling area, many data are obtained from manual surveys. It is difficult to gain a representative database without data preprocessing. On the other hand, there may exist correlated data in the original database, which will further reduce the network efficiency.

CHAPTER 2. SYSTEM DESIGN

BASIC CONCEPT

This research was to build a feasible system capable of detecting pavement surface crack depth in an automatic and real-time mode with high efficiency and high accuracy. At the same time, the system should be economic and practical in field applications. Thus, a prototype system was designed and implemented. With a personal laptop computer, the system has a friendly interface for data collection, processing, and management. The system is capable of providing an accurate description of pavement surface profile and automatic detecting of pavement surface cracks. Fig. 2-1 shows the basic parameters collected by the system and their geometric relationship. To describe a crack, two sources of information are required: X-direction and Z-direction. Therefore, the system design focused on how to obtain these two sources of data.



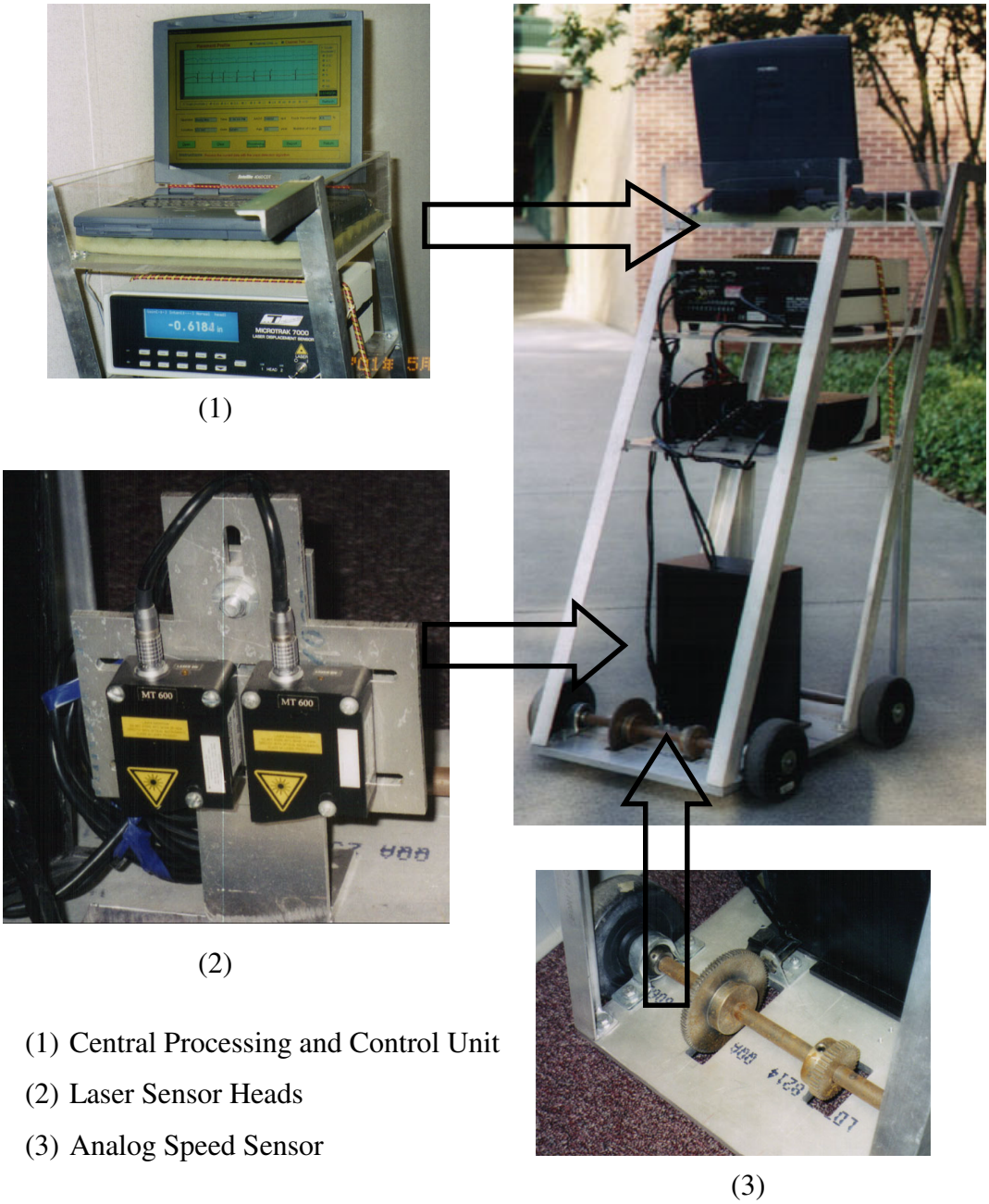
Note: X = Longitudinal Direction (Driving Direction)
Y = Transversal Direction
Z = Vertical Direction

Figure 2-1. Crack Detection Geometric Description

HARDWARE DESCRIPTION

The prototype and its main components are shown in Fig. 2-2. Compared with other sensors, a laser sensor has its own advantages such as high accuracy and high measurement speed. However, its measurement is based on a triangulation principle (as shown in Fig. 1-5). Thus, there is an angle between incident laser beam and reflecting laser beam. The smaller the angle, the higher the detection ability is. To increase the

sensitivity of the laser sensor for detecting a crack, the sensor should have a very small beam diameter or spot size so that even a tiny crack can be detected. In this project, two Microtrak 7000 laser sensors were selected. These sensors have a high resolution and high sampling rate. The specification of the laser measurement unit is listed in Tab. 2-1. With the dual-channel capability (two laser sensors), two sets of data can be obtained. The two-channel structure design can significantly increase the crack detection ability.



(1)

(2)

(3)

(1) Central Processing and Control Unit
(2) Laser Sensor Heads
(3) Analog Speed Sensor

Figure 2-2. Pictures of the Prototype and its Components

Table 2-1. Specification of the Laser Measurement Unit

Measuring Range	3.0 in.
Resolution	200 μin.
Accuracy	± 0.1% of range
Sampling Frequency	100 kHz
Spot Size	0.006 × 0.010 in.
Head Type	Diffuse
Analog Output	- 10 to + 10 VDC

To obtain X-direction displacement information, a speed sensor is used. Through some simple procedure, speed measurement can be transformed to displacement quantity, i.e. X-direction information. An analog-to-digital converter (NI-DAQ Card 1200) scans and samples the analog readings from the two laser sensors and the speed sensor. The NI DAQ Card 1200 is a low-cost, low power analog input, digital output, digital I/O, and timing I/O card for PCs equipped with a Type II PC Card slot. The small size and light weight of the NI-DAQ Card 1200 coupled with its low-power consumption make this card ideal for use in portable computers, making portable data acquisition practical in field applications. Fig. 2-3 shows the connections between the card and the measurement units. The card contains a 12-bit, successive - approximation analog-to-digital converter with eight inputs. A laptop computer processes the digital output of the card and obtain results. The system architecture diagram is shown in Fig. 2-4.

SOFTWARE DESCRIPTION

The system software is written in Visual Basic 6.0 and runs on the Microsoft Windows 98 platform. Based on the operation platform, the function of the software interface can be classified into low-level and high-level tasks. With the NI-DAQ application-programming interface (API), which comes with NI-DAQ card, the software completes

some low-level hardware operations such as system initialization and data scanning. The architecture and function of the software are summarized in Fig. 2-5. The major function is logically divided into four modules:

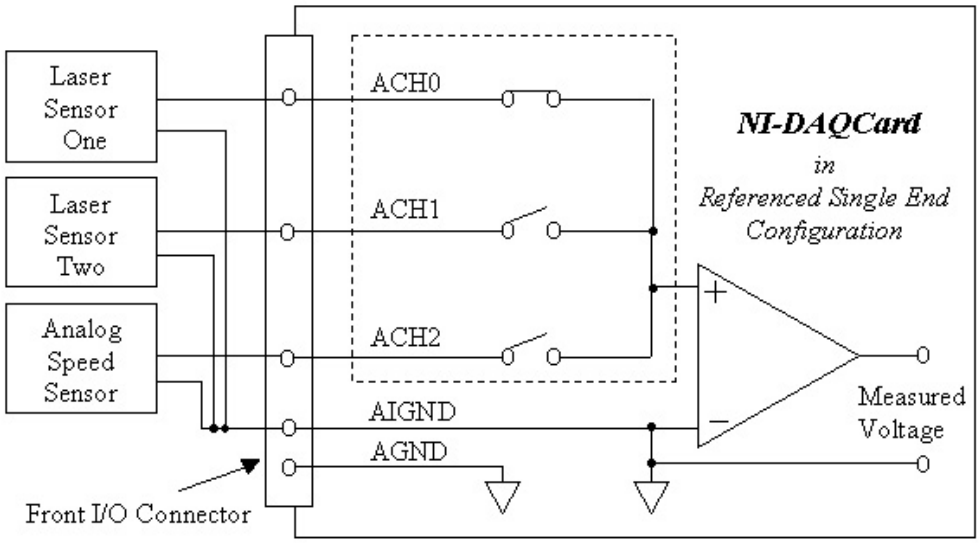


Figure 2-3. Connections between Measurement Sensors and the NI-DAQ Card 1200

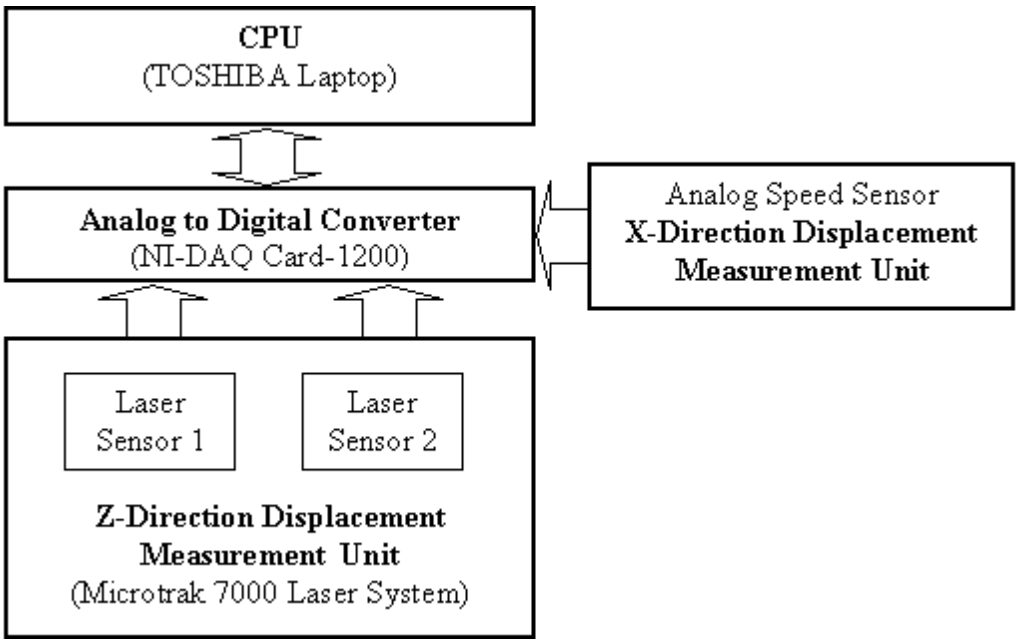


Figure 2-4. System Hardware Architecture

Data Sampling: The main task of this part is the communication between CPU and the NI-DAQ Card that converts the analog outputs of the sensors into digital signals. To automatically locate cracks, speed signal is converted into distance signal which is then converted into digital signal by the NI-DAQ Card.

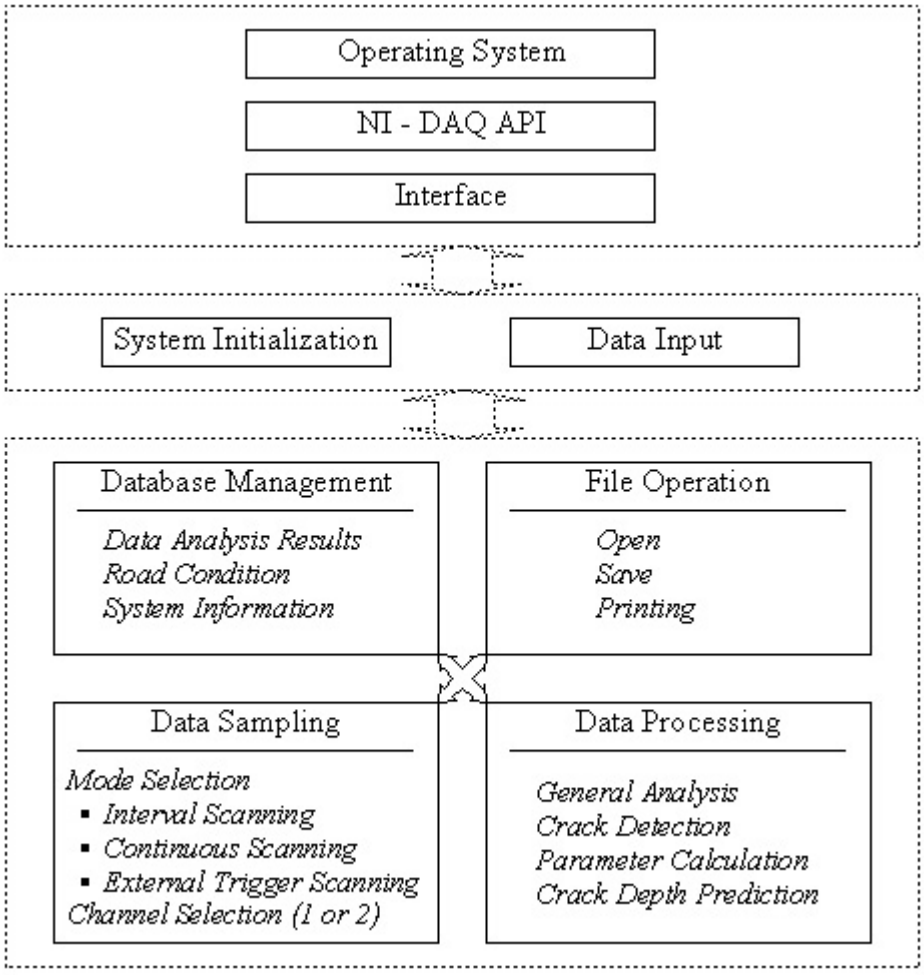


Figure 2-5. System Software Architecture and Function Diagram

Data Processing: The distance signal is processed by a Scan-Rate-Effect-Canceling model to calibrate the X-direction distance measurement to enhance the system accuracy. The digital signals from the laser sensors are preprocessed by a high-pass filter to filter out some low-frequency noise. The preprocessed signals are then processed by the crack-detection-algorithm (called Partial-Cross-Correlation algorithm) to identify cracks and the corresponding locations as well as crack opening geometry parameters. Meanwhile,

with the pavement section relevant information, the preprocessed signals are processed by the neural network model to estimate corresponding crack depths. The data-processing unit is the key part of the software. The flow chart of this unit is shown in Fig. 2-6.

Database Management: The final crack report including information on crack location, crack opening geometry parameters, and crack depth is stored in a database. The database also contains some other information related to data collection, such as date, pavement section ID, and road conditions.

File Operation: All the original data and processed data can be saved as fixed file formats for later processing and analysis. The formats are compatible with other general window software such as MS Excel.

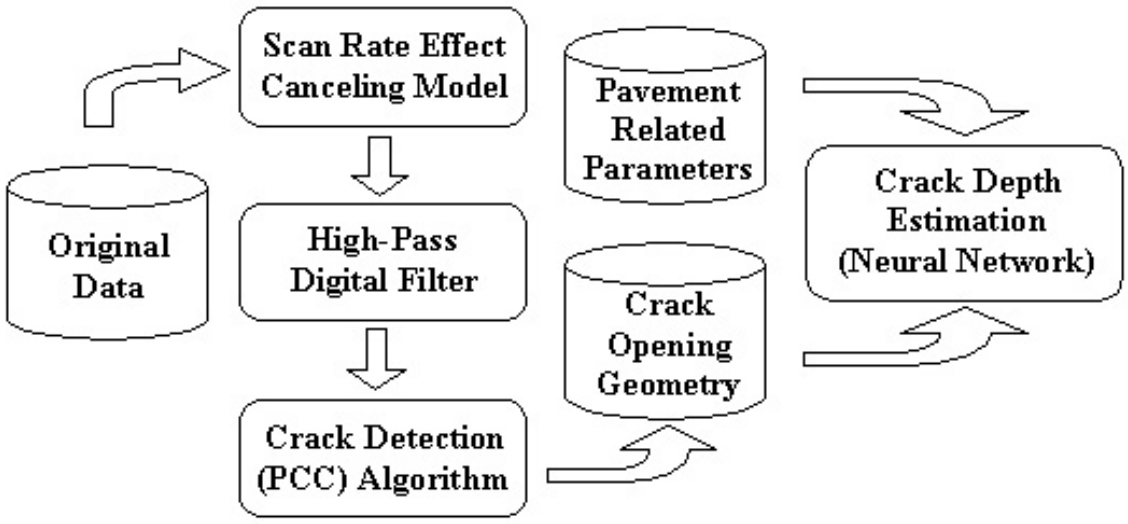


Figure 2-6. Data Processing Flow Chart of the Software

MODEL FOR CRACK DEPTH ESTIMATION

Basic Concept

Currently, there is no technique available to directly measure pavement surface crack depth with reasonable accuracy. Based on the preliminary experiments conducted in the research project, the indirect method (using neural network model and basic measurements from the laser sensors and speed sensor to estimate crack depth) was

proposed by the research team and approved by the project manager. The modeling approach can be conceptually expressed by the following equation:

$$Crack\ Depth = f\left(AADT, Age, Truck\ Percentage, Opening\ Geometry...\right) \quad (2\text{-}1)$$

where

AADT	=	Average Annual Daily Traffic;
Age	=	Age of the Pavement Section;
Truck Percentage	=	Percentage of Truck in the Total Traffic;
Opening Geometry	=	Crack Opening Geometric Characteristics such as width, depth, and slopes.

Inspired by the structure of the human brain, ANN method has been widely applied to fields such as pattern classification, signal processing, coding, forecasting, control, etc., because of their ability to solve cumbersome or intractable problems by learning directly from the data. Use of ANN has been introduced recently in the field of pavement engineering. As stated previously, research studies involving the use of ANN in pavement engineering have concentrated either on demonstration of ANN in pavement condition and performance prediction or comparison between ANN and regression methods. In this project, a neural network was developed to map the relationship between the measures from the system and the actual crack depth.

Artificial Neural Network

ANNs are collections of mathematical models that emulate some of the observed properties of biological nervous systems and draw on the analogies of adaptive biological learning. A neural network resembles the brain in two respects [19]:

- Knowledge is acquired by the network through a learning process; and
- Interneuron connection strengths known as synaptic weights are used to store the knowledge.

The key element of the ANN paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing

elements that are analogous to neurons and are tied together with weighted connections that are analogous to synapses. A neuron is an information-processing unit that is fundamental to the operation of a neural network. It consists of three basis elements: a set of synapses or connection links, an adder for summing the input signals and an activation function of limiting the amplitude of the output of a neuron. The model of a neuron is shown in the Fig. 2-7.

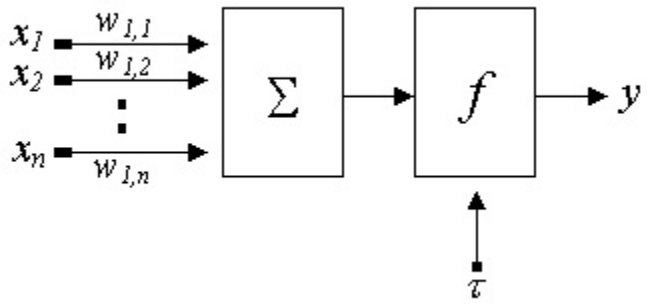


Figure 2-7. Mathematical Model of a Neuron

In mathematical terms, a neuron may be described by the following equation:

$$y = f\left(\sum_{1}^{d} w_i x_i - \tau\right) = f\left(\underline{w} \cdot \underline{x} - \tau\right) \tag{2-2}$$

The single output y is related to the multiple inputs $[x_1, x_2..., x_d]$, encoded as a column vector $\underline{x}$, and to the strengths of the synaptic connections (weights) $w=[w_1, w_2..., w_d]$, and firing threshold τ through a memoryless activation function f, which defines the output of a neuron in terms of the activity level at its input. With different applications, the activation function may assume different forms. Choice of f that reflects thresholding behavior of a neuron is a sign function as defined in the following equation:

$$f(x) = sgn(x - \tau) = \begin{cases} 1, & if\ x \geq \tau; \\ -1, & otherwise. \end{cases} \tag{2-3}$$

For graded responses, a so-called sigmoidally shaped non-linearity is adopted, i.e. a function that is continuously differentiable, increasing, and has a range of (0, 1). The following equation (Eq. 2-4) is an example of this form, called logistic function:

$$f(x) = \frac{1}{1+e^{-\alpha x}} \tag{2-4}$$

Actually, most applications of neural networks are related problems either of approximating to given functions or to noisy data sets [20]. In practice, the functions will be specified not by algorithms but by a table or a training set T consisting of n argument-value pairs, i.e. a d-dimensional argument $\underline{x}$ and an associated target value t that is the goal. Then, t will be approximated by a network output. The function to be constructed will be fitted to the following equation:

$$T = \left\{ \left(\underline{x}_i, t_i \right) : i = 1 : n \right\} \tag{2-5}$$

Usually, the training set T is considered noisy and the goal is not to reproduce it exactly but rather to construct a network function that produces a smoothened reconstruction that generalizes (learns) well to new function values. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANN as well [19]. Learning typically occurs by example through training, or exposure to a set of input/output data where the training algorithm iteratively adjusts the connection weights (synapses). Multi-class problems may also be solved by having a number of neurons operating in parallel. With the structure described above, the ANN method offers the following useful properties and capabilities:

Computation: In principle, neural networks can compute any computable function, i.e., they can do everything a normal digital computer can do. A neural network, made up of an interconnection of neurons, is itself nonlinear, which gives it an advantage for dealing with complex, real-world problems.

Learning and Training: In practice, neural networks are especially useful for classification and function approximation/mapping problems which are tolerant of some imprecision, which have lots of training data available, but to which hard and fast rules

(such as those that might be used in an expert system) cannot easily be applied. Almost any finite-dimensional vector function on a compact set can be approximated to arbitrary precision by neural networks through a learning process.

Architectures and Algorithms for Designing ANN

Among many interesting properties of a neural network, the most significant one is the ability to learn from its environment and to improve its performance through learning. Learning is a process by which the free parameters of a neural network are adapted through a continuing process of stimulation by the environment in which the network is embedded. There are many kinds of learning algorithms that have been and are being developed. The two main kinds of learning algorithms are supervised and unsupervised:

Supervised Learning: The correct results (target values, desired outputs) are known and are given to the ANN during training so that the ANN can adjust its weights to match its outputs to the target values. After training, the ANN is tested by giving it only input values, not target values, and seeing how close it comes to outputting the correct target values.

Unsupervised Learning: The ANN is not provided with the correct results during training. Unsupervised ANN usually performs some kind of data compression, such as dimensionality reduction or clustering [21].

The distinction between supervised and unsupervised methods is not always clear-cut. An unsupervised method can learn a summary of a probability distribution; then, that summarized distribution can be used to make predictions. Furthermore, supervised methods come in two sub-varieties: auto-associative and hetero-associative. In auto-associative learning, the target values are the same as the inputs, whereas in hetero-associative learning, the targets are generally different from the inputs. Many unsupervised methods are equivalent to auto-associative supervised methods.

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. In general, there are two major kinds of network topology:

Feed-forward ANN: The connections between units do not form cycles. Feed-forward ANN usually produces a response to an input quickly. Most feed-forward ANN can be trained using a wide variety of efficient conventional numerical methods.

Feedback or recurrent ANN: There are cycles in the connections. In some feedback ANN, each time an input is presented, the ANN must iterate for a potentially long time before it produces a response. Feedback ANN are usually more difficult to train than feed-forward ANN.

With different architectures and learning algorithms, diverse neural networks may be realized. In this study, a supervised multi-layer feed-forward network, called multi-layer perception (MLP), was developed to estimate actual crack depth from some field measured quantities and pavement section related information. There are one or more hidden layers to intervene between the external input and the network output in a multi-layer network. Its architecture is illustrated in Fig. 2-8. Compared with other neural networks, a distinctive characteristic of a MLP is its ability to learn complex tasks by extracting progressively more meaningful features from the input patterns (vectors).

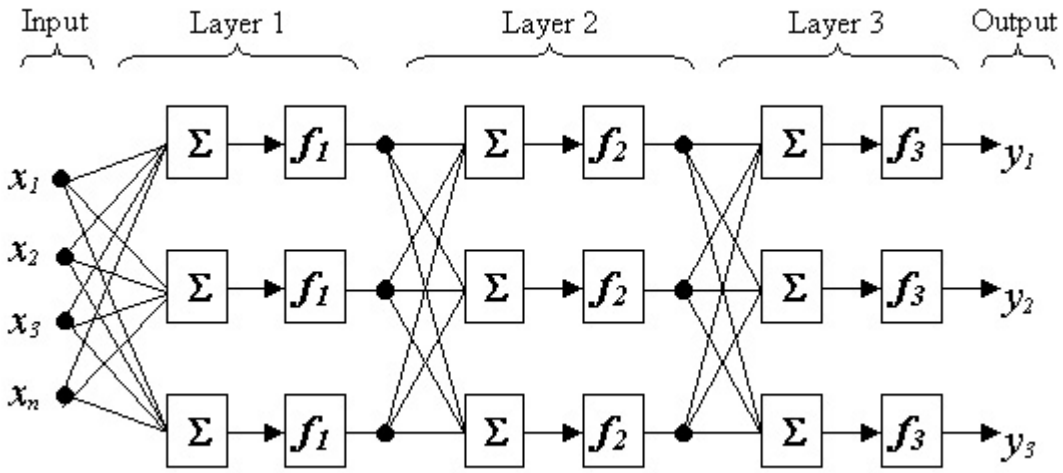


Figure 2-8. Multi-layer Feed-forward Network Architectures

Comparison between MLP and Other Statistical Methods

There is considerable overlap between the fields of neural networks and statistics [22]. Many methods in the statistical literature can also be used for flexible nonlinear modeling. These methods include: polynomial regression, Fourier series regression, local polynomial smoothing, multivariate adaptive regression splines (MARS) and projection pursuit, etc.

A MLP tends to be useful in the same situations as projection pursuit regression, i.e.:

- the number of inputs is fairly large,
- many of the inputs are relevant, and
- most of the predictive information lies in a low-dimensional subspace.

The main advantage of MLP over projection pursuit regression is that computing predicted values from MLP is simpler and faster [22]. In addition, MLP is better at learning moderately pathological functions than are many other methods with stronger smoothness assumptions.

Backpropagation Algorithm for Gradient Evaluation

As described in Eq. 2-5, the final objective of any neural network application is to select a net N so that the output $\underline{y}_i = N(\underline{x}_i, \underline{w})$ is close to the desired output t_i for the input $\underline{x}_i$. The notion of closeness on the training set T is typically formalized through an error of objective function or metric of the form written as follows:

$$\delta(n) = \frac{1}{2}\sum_{i=1}^{n}\left\|y_i - t_i\right\|^2 \tag{2-6}$$

The error is a function of weights, $\underline{w}$. The network training, therefore, is a nonlinear optimization procedure. In practice, the weights are usually trained by using an iterative gradient descent-based optimization routine called Backpropagation (BP) algorithm [19], in which the goal is to find a set of network weights that minimize the objective function. A feed-forward network trained by Backpropagation is called a BP network. The detailed

derivation of the algorithm can be found in many references [19, 23, 24]. Here, only the vital relationships are summarized.

In a BP network, there are two kinds of signals, which are identified as Function Signals and Error Signals. The function signal is an input signal (stimulus) that comes in at the input end of the network, i.e. $\underline{x_i}$ in Eq. 2-5. The function signal propagates forward through the network, and emerges at the output end of the network as an output signal, i.e. $\underline{y_i}$ in Eq. 2-6. An error signal originates at an output neuron of the network, i.e. $\delta(n)$ in Eq. 2-6. It propagates backward through the network.

The computation of the back-propagation algorithm consists of two distinct passes: forward pass and backward pass. In the forward pass, the synaptic weights remain unaltered throughout the network. The function signal appearing at the output of neuron *j* can be computed through the following equation:

$$y_j(n) = \varphi\left(\sum_{i=0}^{p} w_{ji}(n) y_i(n)\right) \tag{2-7}$$

where

p = the total number of inputs applied to neuron j,

$w_{ji}(n)$ = the synaptic weight connecting neuron i to neuron j, and

$y_i(n)$ = the input signal of neuron j.

The function signals of the network are computed on a neuron-by-neuron basis until the end of the network. Then, the final output is compared with the target value, i.e. t_i in Eq. 2-5 to obtain the error signal for the relative output neuron.

In the backward pass, on the other hand, the signal starts at the output layer by passing the error signals backward through the network, layer by layer, to recursively adjust the weights between each layer according to a so-called delta rule, which is described in the following equation:

$$w_{k+1} = w_k - \alpha_k g_k \tag{2-8}$$

where, w_k is a vector of current network connections; g_k is the current gradient; and α_k is the learning rate. The gradient represents a sensitivity factor, determining the direction of search in weight space for the relative weight. It is computed from the error signals of all the neurons to which that hidden neuron is directly connected. With the different location of neuron *j*, the computation of the gradient varies. The recursive computation is continued, layer-by-layer, by propagating the changes to all synaptic weights made.

In brief, the Backpropagation algorithm can be summarized as follows:

Initialization: Start with a reasonable network configuration, and set all the synaptic weights and the activity level of the network.

Forward Computation: Present the network with the training examples. Compute the activation potentials and function signals of the network by proceeding forward through the network, layer by layer.

Backward Computation: Compute the local gradients of the network by proceeding backward, layer by layer. Adjust the synaptic weights of the network between each layer according to the delta rule mentioned above.

Iteration: Iterate the computation by presenting all the training examples to the network until the network stabilize their values and the average error over the entire training set is at a minimum or acceptably small value.

CHAPTER 3. DISTANCE MEASUREMENT

INTRODUCTION

The system developed in the project has two measuring units to obtain the microscopic longitudinal profile of the pavement surface. Laser measuring unit (with two laser sensors) measures the Z-direction distance, whereas a speed sensor collects X-direction information (as shown in Fig. 2-1). Based on field and laboratory test results, it was proven that the accuracy of laser measurement unit is high enough for this application. It can also be seen from the specifications listed in Tab. 2-1 that the resolution of the laser sensor is very small. However, the speed sensor used in the system has certain limitation if no calibrating model is used. The main reason for this is that the signal from the speed sensor needs to be converted to distance signal and the sampling rate of the speed sensor has direct impact on the accuracy of the speed sensor readings. In the project, field data were collected to calibrate the impact of the sampling rate on speed sensor accuracy. This chapter presents the details about the X-direction distance measurement and calibration.

PRINCIPLE OF THE DISTANCE MEASUREMENT

There are mainly two types of commercially available non-contact speed/displacement measurement sensors. One is Doppler radar sensor such as the Doppler Radar Speed Sensor from GMH Engineering and DRS1000 Speed Sensor by Datron Technology Limited. The other type is the magnetic speed sensor such as the MP1A Magnetic Pickup by Daytronic Corp., and 3010 series VRS Sensors by Invensys Control Systems. A permanent magnet is the heart of a magnet sensor and establishes a fixed magnetic field. An output signal is generated by changing the strength of this field. The alternating presence and absence of ferrous metal (gear tooth) varies the reluctance, or “resistance of flow” of the magnetic field, which dynamically changes the magnetic field strength and further changes the output signal. The Doppler radar sensors are easy to use, and they have a wide speed measurement range. Nevertheless, the price of the Doppler sensor is usually much higher than magnetic sensors. Based on the cost/performance comparison among many commercial magnetic sensors, the M12x1-180ASAw Analog Speed Sensor

by Sensor Solutions Corp. (as shown in Fig. 3-1) was selected as the X-direction distance measurement unit.

Figure 3-1. M12x1-180ASAw Analog Speed Sensor by Sensor Solutions Corp.

The analog speed sensor provides an analog output voltage proportional to the speed of a rotating gear or moving rack. Unlike a variable reluctance (VR) speed sensor that produces a sine wave, the Analog Speed Sensor produces a DC voltage that varies with speed. The full-scale frequency and response time of these sensors is set with a few internal components, so no external parts are needed. The relationship between the analog voltage output and the target rotating frequency is shown in Fig. 3-2. Based on the reading from the analog output, the frequency of the sensor can be obtained. The X-direction distance measurement can be illustrated by Fig. 3-3. Mathematically, the X-direction distance is obtained through the following equation:

$$D = \sum_{i=0}^{N} d_i \tag{3-1}$$

$$d_i = a \times v_i \times t_i \tag{3-2}$$

where

D = X-direction distance,

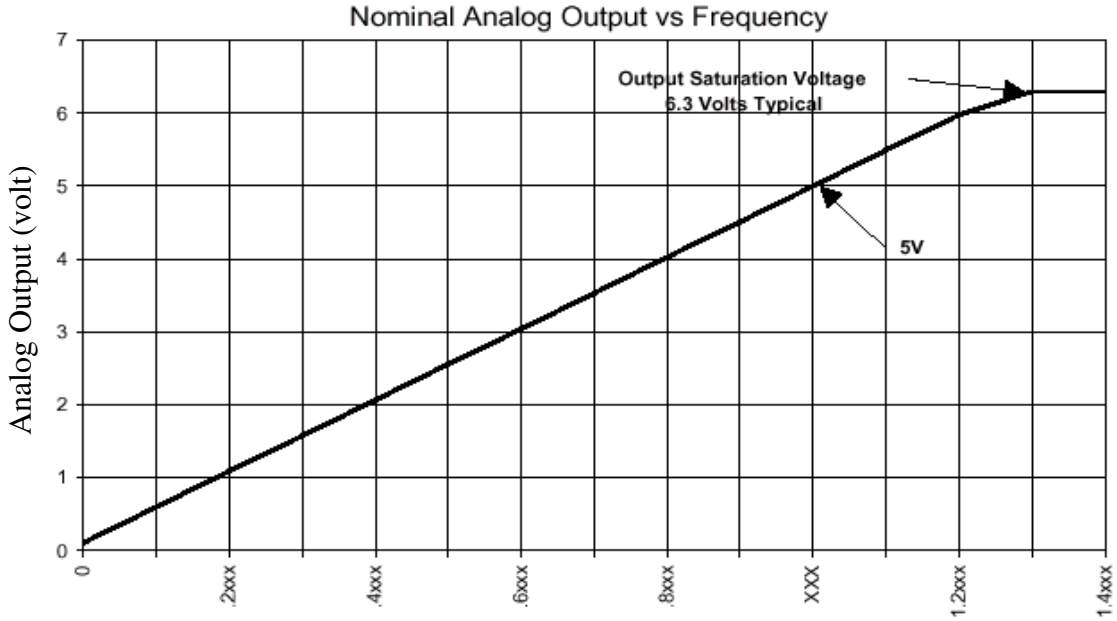
d_i = Sub-distance between each sampling,

a = Transformation factor between direct analog output of speed sensor and X-direction linear speed,

v_i = Direct analog output of the speed sensor, and

t_i = Time interval between each sampling, i.e. the reciprocal of the scan rate.

According to Fig. 3-3, the entire distance, D, is a summation of a series sub-distance, i.e. d_i, where, d_i is the operating distance in each DAQ Card sampling period.



xxx = Full Scale Frequency

Figure 3-2. Analog Speed Sensor Output - Target Rotating Frequency Relationship

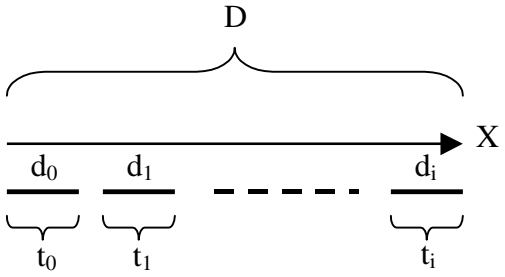


Figure 3-3. X - Direction Distance Measurement

To obtain distance information, uniform motion of the measuring system is assumed in each t_i. Because the time interval between each sampling period is very small (the scan rate range is between 1 kHz ~ 10 kHz), this assumption is feasible in real application.

In fact, in order to obtain the estimated distance information with reasonable accuracy, the model shown in Eq. 3-2 should be calibrated with field data including the distance data calculated from Eq. 3-2 and the real distance data measured by field engineers. Thus, field experiments were conducted in the project to calibrate the speed sensor so that distance data can be obtained by the system.

SPEED SENSOR CALIBRATION

Field Data Collection and Analysis

Through field experiments, it was found that the speed sensor could not provide accurate distance information if no calibrating model was used. Mainly, the measurement error of distance information is from three sources:

- Inaccurate demarcated frequency,
- System mechanism error, and
- Approximate calculation.

The first two sources belong to the system inherent error. Both of them come from the transformation factor, "*a*" shown in Eq. 3-2. The third one is due to the assumption of uniform motion in each sampling period. From Eq. 3-2, it can be seen that two variables may affect the synthetic error, i.e. system operating speed and analog-to-digital card scan rate. In order to evaluate the effects of operating speed and scan rate, three field experiments were done at different field locations and with different operating ranges. In each field experiment, the system was operated at low-speed, medium-speed, and high-speed, respectively. With a fixed scan rate *K*, two sets of data were obtained. The first set of data contained the actual distance, and the second set of data contained the distance data measured by the system. Then, more data were obtained by changing the scan rate

K. With these data, the relationship curves for speed effect and scan rate effect were obtained. Two sets of curves were selected as examples (as shown in Figs. 3-4 and 3-5). In the figures, D_a refers to the actual distance and D_m refers to the distance measured by the system. According to Fig. 3-4, the scan rate has a certain effect on the distance measurement. Thus, the speed sensor should be calibrated for the scan rate. According to Fig. 3-5, the operating speed has little effect on the X-direction distance measurement.

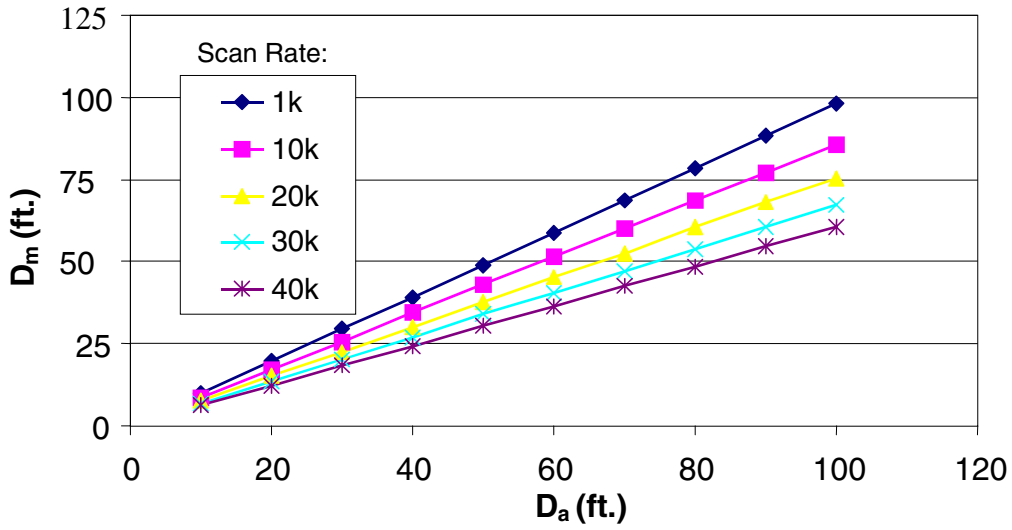


Figure 3-4. Relationships between D_m and D_a with Different Scan Rates

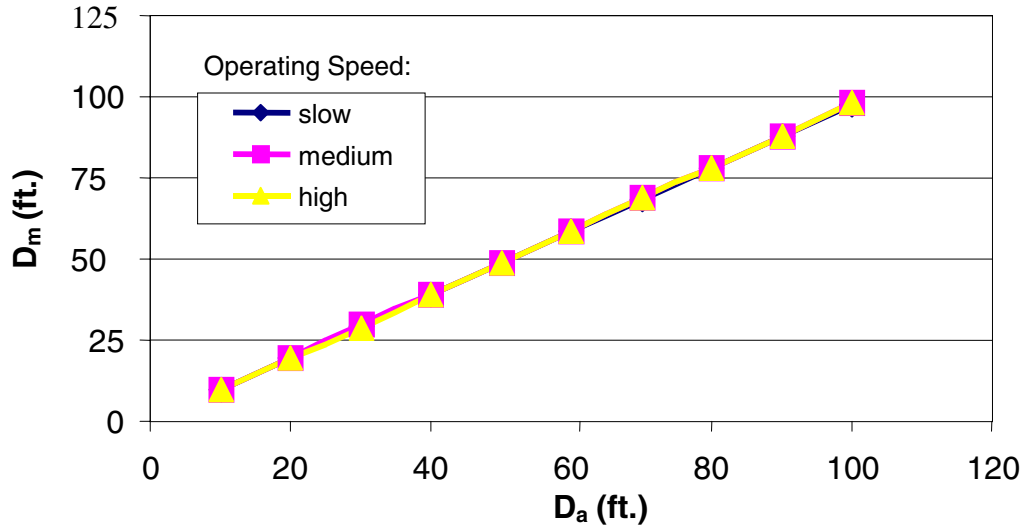


Figure 3-5. Relationships between Dm and Da with Different Operating Speeds

Calibration for Distance Measurement

Field experiments showed that the synthetic error is linear and its source is complex. Thus, the statistical method is more practical for canceling scan rate effect [5]. The basic concept for performing the calibration is to obtain the statistical relationship between the X-direction distance data and scan rates. For a given test section and a selected scan rate K, the distance measurement from the system D_m can be rectified into its corresponding actual distance D_a through Eq. 3-3:

$$D_a = f(K, D_m) \tag{3-3}$$

where, $f(.)$ is a function of K and D_m which can be obtained by curve-fitting methods. A linear function was used to fit $f(K, D_m)$. The function has the following form:

$$D_a = A(K) + B(K)D_m \tag{3-4}$$

where, $A(K)$ and $B(K)$ are parameter functions of K. To estimate these parameters, several test sections were selected for data collection with different scan rates. Based on Fig. 3-2, for a given scan rate, the parameters $A(K)$ and $B(K)$ can be obtained. Thus, a matrix was obtained as shown in Tab. 3-1.

Table 3-1. Field of Test Results Form for Scan Rate Effect Canceling

Test Section Number	Scan Rate (K)	A (K)	B (K)
1	x	x	x
2	x	x	x
…	…	…	…
M	x	x	x

Note: The element x in the table is a representative of the real value.

The relationship curves between parameters *A* and *B* and scan rate *K* are shown in Figs. 3-6 and 3-7. With the linear regression method, the following equation was obtained:

$$D_a = 0.0014\ K - 0.0644 + (0.0081K + 0.503)D_m \qquad (3\text{-}5)$$

Eq. 3-5 is the final Scan-Rate-Effect-Canceling (SEC) model. That is, with a given scan rate, the correct X-direction distance can be estimated through Eq. 3-5.

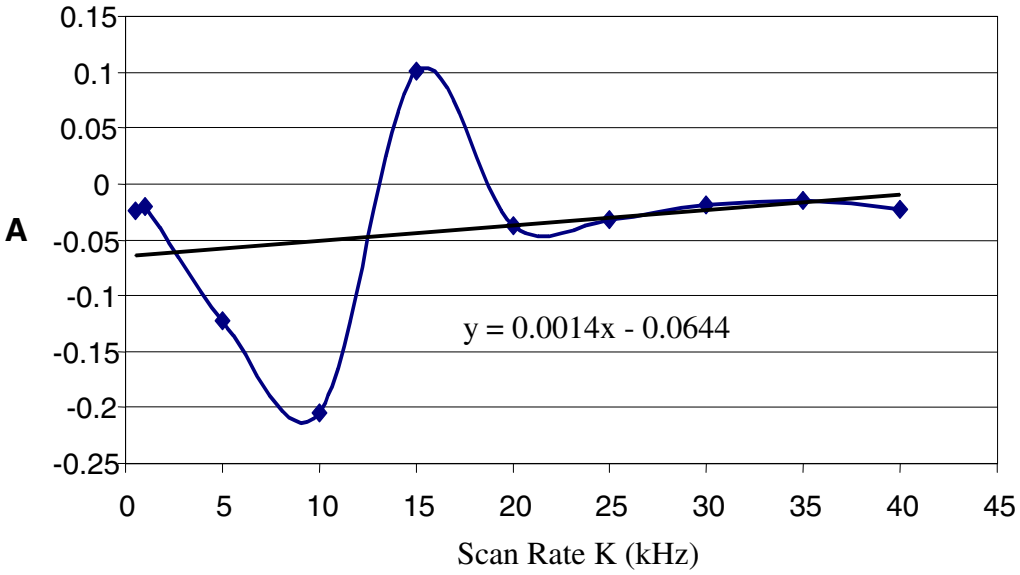


Figure 3-6. Scan-Rate-Canceling Model - Coefficient (A) versus Scan Rate (K)

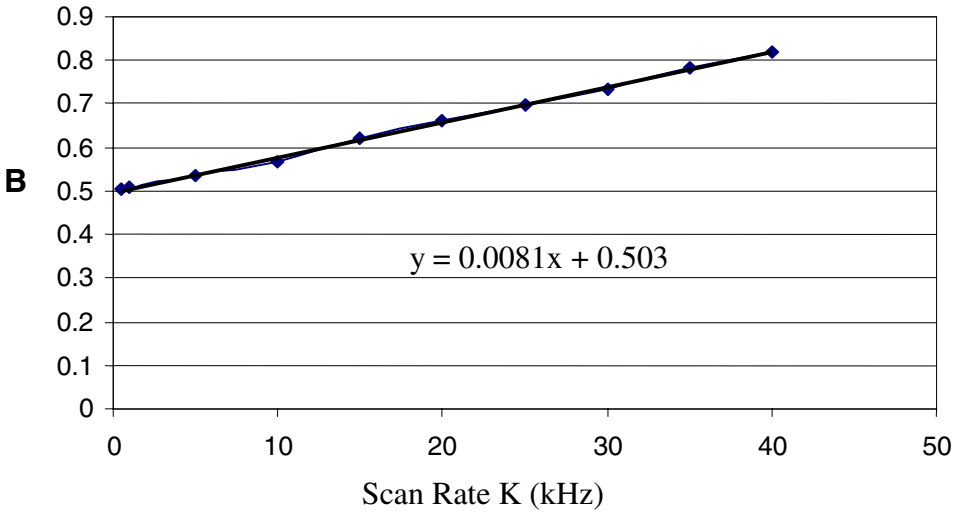


Figure 3-7. Scan-Rate-Canceling Model - Coefficient (B) versus Scan Rate (K)

Field Validation

To evaluate whether the calibrating model (SEC model) can provide distance data with reasonable accuracy, validation experiments were performed through field tests. In the validation experiments, real distance data were obtained and the distance data measured by the system and calibrated by the SEC model were recorded. Both distance data were compared to obtain the error and accuracy measures. Fig. 3-8 shows the distance measurement accuracy. From this figure, it can be concluded that the relative error of the distance measurement is less than 0.5%. Such an accuracy can ensure the quality of distance measurement.

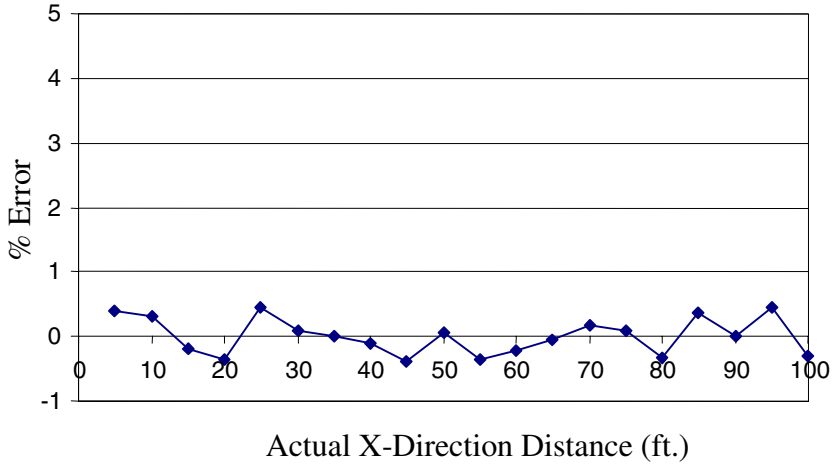


Figure 3-8. X-Direction Distance Measurement Accuracy

CHAPTER 4. CRACK IDENTIFICATION

INTRODUCTION

Accurate identification of cracks is the prerequisite to the automatic detection of crack depth. Performance of the algorithm to identify cracks directly affects the system efficiency and its real-time characteristics. However, selection of the algorithm and the corresponding software development depend on the hardware performance of the laser sensors. This chapter summarizes the development of the algorithm and evaluation of the algorithm.

PAST ALGORITHMS

In the past, several algorithms have been developed to identify pavement surface cracking [3], among which a method called Running-Mean Downup Method (Downup) produces better results [9]. The basic idea behind the Downup method is that a crack can usually be identified by a sharp negative (down) slope and followed by a sharp positive (up) slope. A running mean algorithm filters the noise of the data and establishes a reference plane. A parameter *mbar* denotes the number of points used for the running average in the Downup algorithm. The algorithm identifies the down and up slopes by means of a difference sequence. First, the algorithm computes the slope, called *diff*, by taking the difference between each averaged point for a base length *sbar*. Then, a preset threshold value is used to determine the crack location. To make sure that the algorithm performs well, two additional parameters are used to obtain more reasonable results, i.e. *width*, a maximum acceptable crack width and *tc*, a maximum acceptable crack depth. The Downup method can be executed sufficiently faster to be suited to real-time application. However, a common drawback of these existing algorithms including the Downup method is their complexity. Usually, there are too many parameters that need to be determined, and with different parameters the algorithm will produce different results. Especially, when used for pavements with different surface conditions, different sets of parameters are required. Such kind of parameter determination is difficult for practical applications.

PARTIAL-CROSS-CORRELATION ALGORITHM

Primarily, the complexity of the past algorithms is caused by the limitation of the laser sensor. With the use of a narrow laser beam to scan the pavement surface, massive amounts of information across the lane may be missed and many false cracks could be detected due to the roughness of the pavement surface. However, with the algorithms mentioned previously, these false cracks could be misinterpreted as cracks. In order to prevent the inclusion of false cracks, a new algorithm, called Partial-Cross-Correlation (PCC) algorithm, was developed in this research project. As mentioned previously, the system has two laser sensor heads. Through the dual-channels (each laser head uses one channel to connect to the lap top computer), two sets of data are obtained. Development of the PCC algorithm is based on the fact that the possibility of two false cracks occurring at the same position is almost zero. Mathematically, the profile data scanned by a laser sensor can be divided into two signals: $X(n)$ and $Ran(n)$, where $X(n)$ is a deterministic signal related to crack information and $Ran(n)$ is a random noise signal. The two sets of data from the two laser sensors can be expressed the following equation:

$$S_1(n) = X_1(n) + Ran_1(n)$$

$$S_2(n) = X_2(n) + Ran_2(n) \qquad (4\text{-}1)$$

where, $S_1(n)$ and $S_2(n)$ are data sequence from laser channels one and two, respectively. The cross-correlation analysis [25] between S_1 and S_2 can be performed as follows:

$$\begin{aligned} R(m) &= E\{[S_1(n)S_2(n+m)]\} \\ &= E\{[X_1(n) + R_1(n)][X_2(n+m) + R_2(n+m)]\} \\ &= E[X_1(n)X_2(n+m)] + E[X_1(n)R_2(n+m)] + E[X_2(n+m)R_1(n)] + E[R_1(n)R_2(n+m)] \end{aligned} \qquad (4\text{-}2)$$

where, $E(.)$ denotes the expectation of a random variable. It can be proven that the last three items of Eq. 4-2 are equal to zero because they are uncorrelated signals. Thus,

$$R(m) = E[X_1(n)X_2(n+m)] \tag{4-3}$$

From Eq. 4-3, it can be seen that the cross-correlation analysis removes the interference caused by the noise which is represented by the false cracks due to the roughness of the pavement surface. Thus, the algorithm can prevent the existence of false cracks.

To determine the exact location of a crack, each data set needs to be broken down into many small sub-sections. Then, the cross-correlation between the two respective sub-sections is performed one by one. The determination of the sub-section length should meet two criteria:

- The length of the sub-section should cover the entire crack width or most part of the crack width, and
- The length of the sub-section should not exceed the extent required by the X-direction crack detection accuracy.

Thus, the original data, $S_1(n)$ and $S_2(n)$ can be divided into K sub-sequences respectively, i.e. $ss_1(n)$ and $ss_2(n)$, and the length of each sub-sequence is M. The sub-section data can be described as follows:

$$ss_{1,2}(n) = S_{1,2}\left(n + \frac{k-1}{2}M\right) \qquad \begin{pmatrix} n = 0, 1, \cdots, M-1 \\ k = 1, 2, \cdots, K \end{pmatrix} \tag{4-4}$$

where, $ss_{1,2} = ss_1$ or $ss_{1,2} = ss_2$ and $S_{1,2} = S_1$ or $S_{1,2} = S_2$, depending on whether channel one or channel two is used. Finally, the PCC algorithm is described by:

$$\begin{aligned} P &= \underset{m=0}{\overset{M-1}{LMax}}[R(m)] \\ &= \underset{m=0}{\overset{M-1}{LMax}}[E[ss_1(n)ss_2(n+m)]\} \\ &= \underset{m=0}{\overset{M-1}{LMax}}\left\{E\left[S_1\left(n + \frac{k-1}{2}M\right)S_2\left(n + m + \frac{k-1}{2}M\right)\right]\right\} \quad \begin{pmatrix} n = 0, 1, \cdots, M-1 \\ k = 1, 2, \cdots, K \end{pmatrix} \end{aligned} \tag{4-5}$$

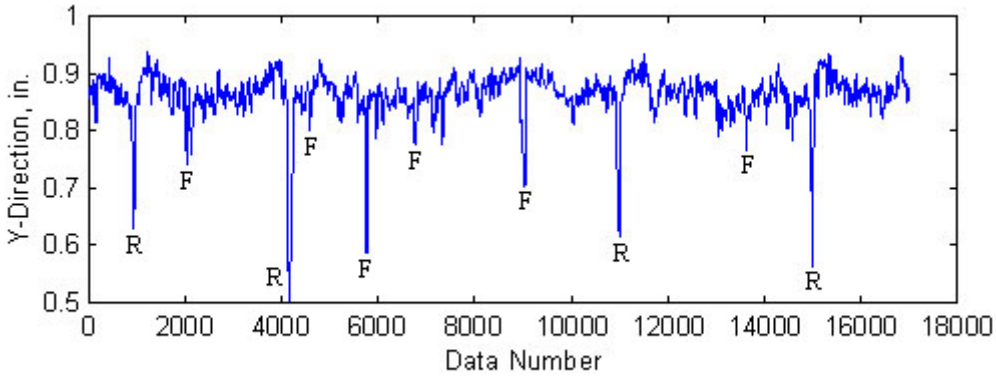
where, *LMax* denotes the local maximum among all the cross-correlation values for the respective sub-section. A software was developed to implement the PCC algorithm in the project. In the software, the maximum of *R(m)* in each sub-section is calculated. Then with the use of a preset threshold, the local maximums *P* are searched and validated to determine the position of a crack. This threshold is the single parameter in the algorithm to detect the cracks. It should be mentioned that using the convolution theorem and implementing it through the Fast Fourier Transform (FFT) could offer a numerically more effective way to calculate the correlation function. The algorithm developed here makes good use of the correlated and uncorrelated characteristics between the two sets of data to enhance the system power for crack detection.

FIELD TESTS OF THE PCC ALGORITHM

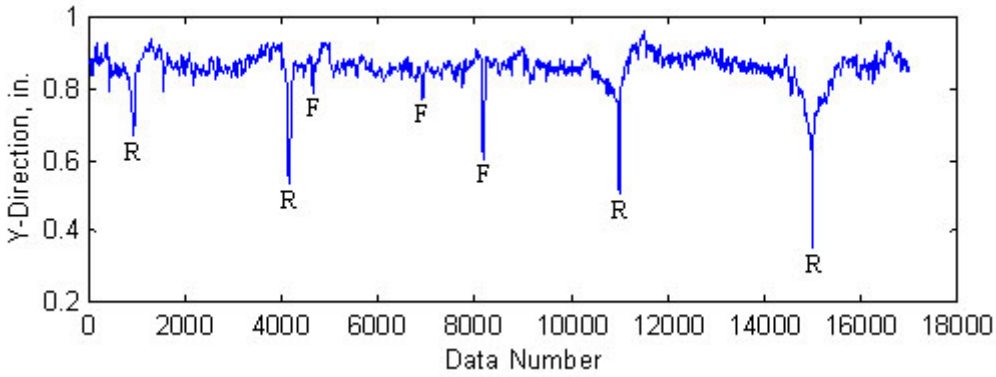
To evaluate the performance of the old algorithms used in past research studies and the PCC algorithm developed in this project, field tests were performed with the data collected from six different pavement sections. In each section, the data were obtained through sampling the pavement surface with the sampling interval of 0.25 mm (0.01 in.). Fig. 4-1 shows a part of the original data from the two laser channels. The real and false cracks were surveyed in the field. With the old crack detection algorithms, it is possible to identify some relatively small false cracks by means of carefully selected algorithm parameters. Nevertheless, if the size of a false crack is comparable with a real crack, it is impossible to correctly identify it.

Since it is difficult to prevent the existence of false cracks if only one channel is used, the PCC algorithm with two channels was used to combines the two sources of data to enhance the detection ability. To do the correlation analysis, the two original sets of data were filtered by a high-pass digital filter as the pre-processing. The filtered data are shown in Fig. 4-2. In this research, the PCC algorithm is based on the prerequisite that the two sets of data from the two channels are correlated. As shown in Fig. 4-3, the cross-correlation sequence values increase significantly as the lag (the variable "*m*" used in Eq. 4-3) is down to zero. This proves that the prerequisite is satisfied. After decomposing the data, the PCC algorithm results are shown in Fig. 4-4. From Fig. 4-4, it can be seen that

the four real cracks can be clearly detected with a preset threshold. The effect of the sub-section length to the PCC algorithm performance is shown in Fig. 4-5. As the length of each sub-section increases, the PCC values become bigger, resulting in better detection of a crack. However, the extension of the sub-section will decrease the X-direction accuracy for crack detection. Therefore, the final sub-section length was determined to be 50-point long or 12.5mm (0.5 in.) in length. The entire process of the PCC algorithm is performed by the software developed in the project.



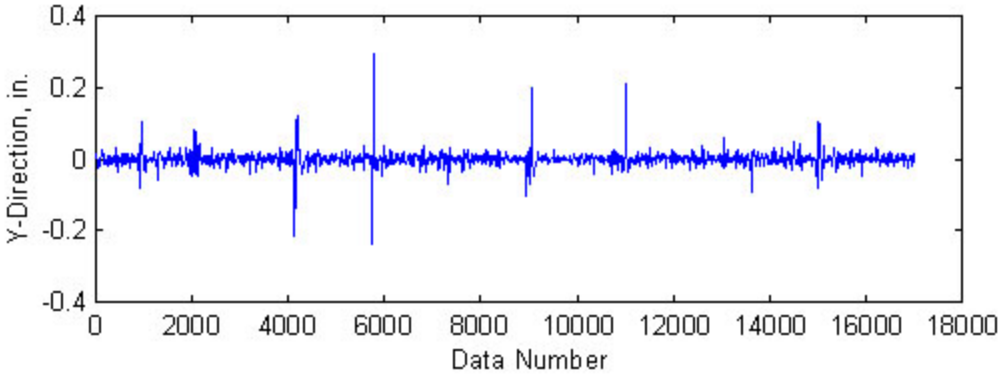
(a) Data from Channel One



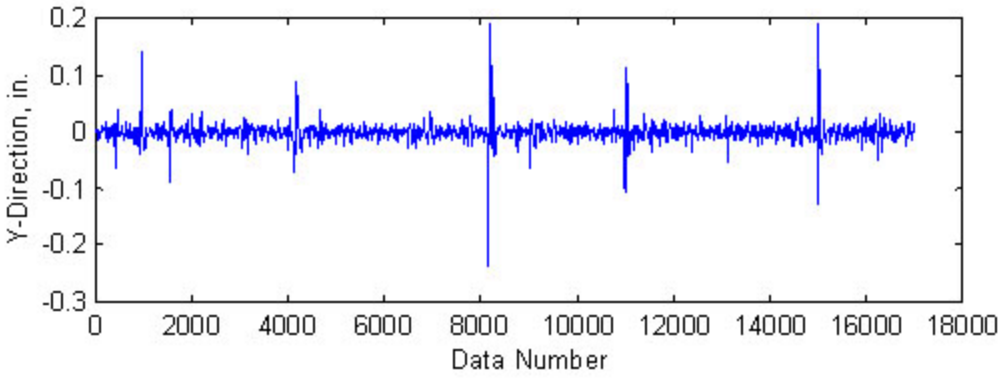
(b) Data from Channel Two

Note: R - Real Crack, F - False Crack

Figure 4-1. Original Field Data Sampled at Every 0.25 mm



(a) Filtered Data from Channel One



(b) Filtered Data from Channel Two

Figure 4-2. Data Filtered by a High-Pass Filter

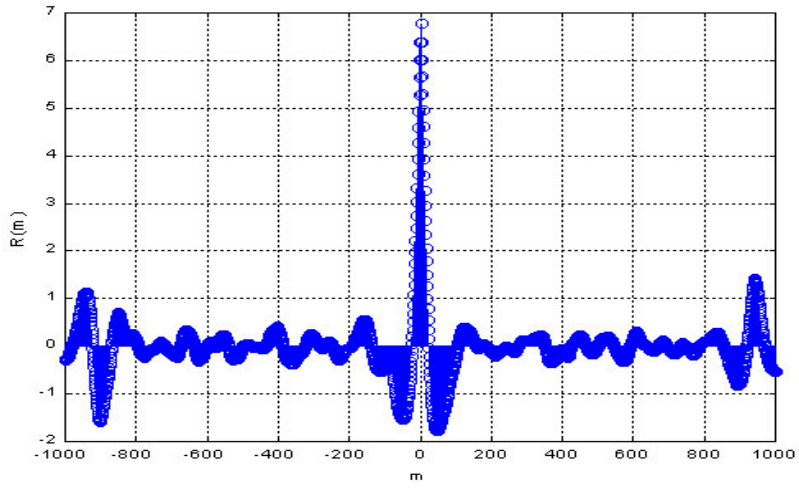


Figure 4-3. Integrate Cross-Correlation Analysis to Filtered Data

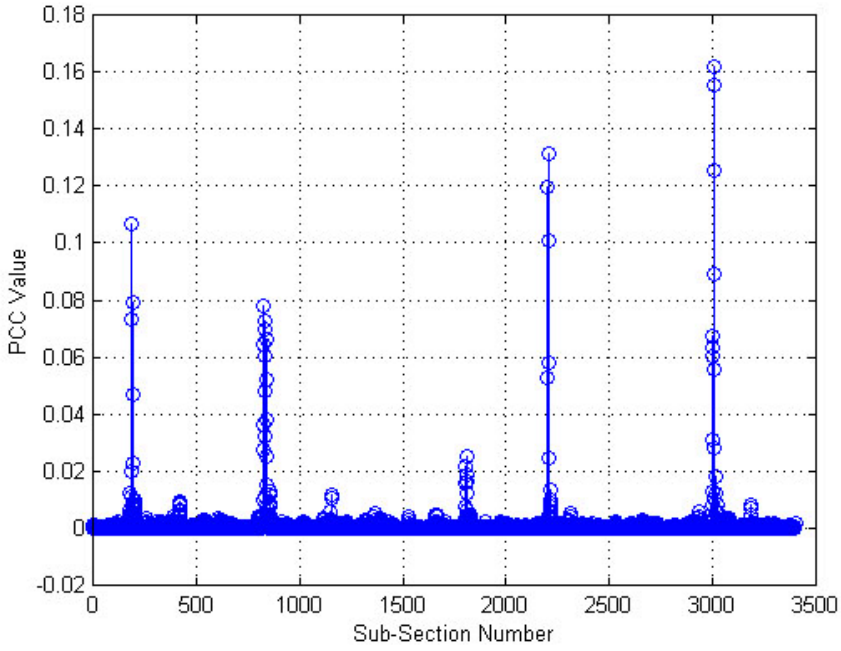


Figure 4-4. Partial Cross-Correlation Analysis Result

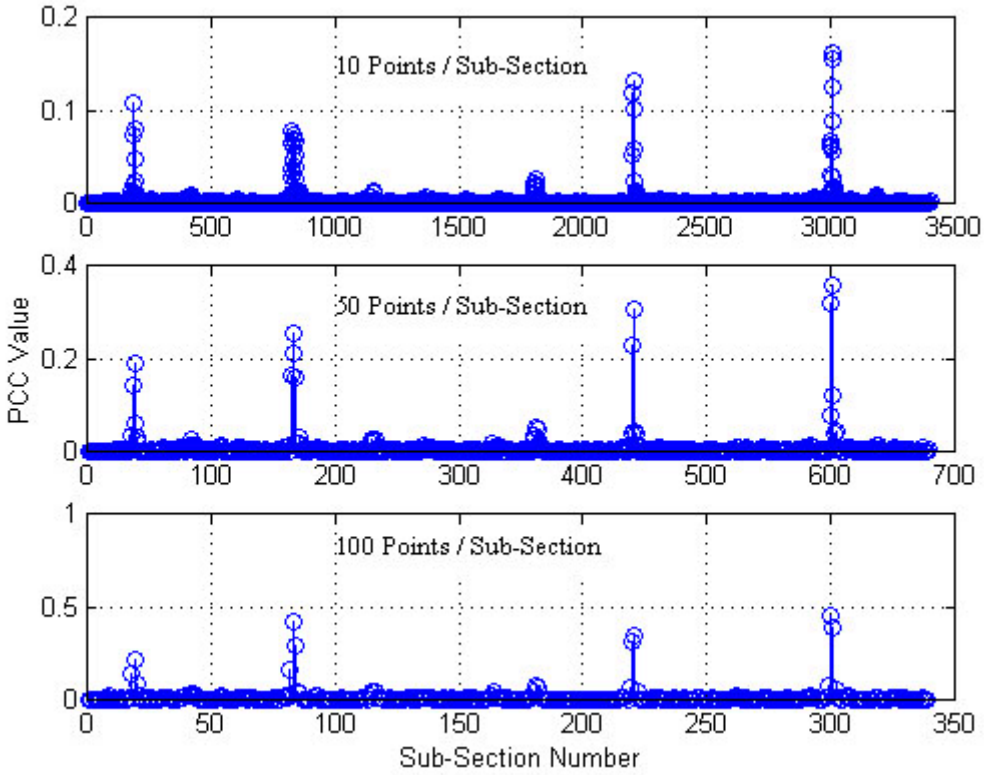


Figure 4-5. PCC Algorithm Evaluation Results with Different Sub-Section Length

To quantify the performance of the PCC algorithm, two performance indices (CDR and FDR) were used. The crack detection rate (CDR) is defined as the ratio of the number of cracks that are correctly detected to the total number of cracks in a selected field section, whereas the false crack detection rate (FDR) is defined as the ratio of the number of cracks that were incorrectly detected to the total number of cracks. Tab. 4-1 presents the performance comparison results between the Downup algorithm and the PCC algorithm. In the cement concrete pavement sections, both algorithms produced good results due to the relatively smooth pavement surface condition. As for asphalt concrete pavement sections, the FDR of Downup method was significantly higher than that of the PCC algorithm. Thus, it is indicated that the PCC algorithm based on the cross-correlation theory has better crack detection ability as compared to the Downup algorithm.

Table 4-1. Performance Comparison between Downup Method and PCC Algorithm

Section Number	Pavement Type	Number Of Cracks	Downup Algorithm		PCC Algorithm	
			CDR[a]	FDR[b]	CDR	FDR
1	Asphalt Concrete	41	95.1%	24.4%	92.7%	4.8%
2	Cement Concrete	29	96.6%	6.9%	96.6%	0.0%
3	Asphalt Concrete	32	90.6%	25.0%	96.9%	6.3%
4	Asphalt Concrete	37	92.3%	12.0%	87.5%	3.7%
5	Asphalt Concrete	45	91.0%	21.0%	92.3%	5.4%
6	Asphalt Concrete	40	97.26%	43.0%	95.7%	8.3%

[a] refers to Crack Detection Rate.

[b] refers to False Crack Detection Rate.

DISCUSSION

In pavement performance evaluation, monitoring of the longitudinal crack depth is even more important as compared to transverse cracks. The PCC algorithm developed in the research is not able to identify longitudinal cracks because the scanning direction of the laser sensors in the prototype is parallel to longitudinal cracks. Thus, the algorithm basically will filter out longitudinal cracks and only keep transverse cracks. The capability of filtering out longitudinal cracks and the impacts of longitudinal cracks to identification of transverse cracks were not assessed in the research. The test sites selected in the research did not present significant amount of longitudinal cracks. Such an assessment is recommended to be performed in the future research.

To automatic identify longitudinal cracks and estimate crack depth, the current sensor system should be improved. Scanning laser sensors with laser angle being dynamically changed should be used. This type of laser sensors are available in markets. The scanning direction of the laser sensors should be perpendicular to the direction of longitudinal cracks. The sensors should be able to obtain information such as crack width, crack slopes, measurable crack depth, etc. Similar to transverse cracks, estimating models should be developed to estimate the depth of longitudinal cracks. Practically, such a system will not aim at detecting all longitudinal cracks. This type of system will focus on the longitudinal cracks on wheel paths. The scanning laser sensors will only scan the cracks on wheel paths. It is possible that some longitudinal cracks on wheel paths will not be picked up by the laser sensors. However, statistically, the system will be able to provide longitudinal crack depth distribution on a specific pavement section.

CHAPTER 5. MODEL DEVELOMENT AND SYSTEM IMPLEMENTATION

INTRODUCTION

The automatic system has three basic subsystems: (1) distance measurement, (2) crack identification, and (3) crack depth estimation. The implementation of the system is to put all these three subsystems together in terms of software and hardware. In last two chapters, the first two subsystems were described and evaluation results were presented.

To estimate crack depth, a neural network model was developed based on training and testing results. Two sources of field data were used for training, testing, and validation purposes. The first set of data were collected by the two laser sensors and the speed sensor to scan the surface geometric of cracks, and the second set of data were obtained from a non-contact type of measuring system (impact echo) to statically measure crack depth. With the two data sets and pavement section related information, the neural network model was developed. The pavement section related information includes AADT, age, truck factor, etc. These pavement-related variables could be obtained directly from the FDOT pavement database. The output of the model is the estimated crack depth. This chapter summarizes the modeling procedure and presents the performance evaluation results of the model for crack depth estimation.

DATA COLLECTION

Field Data Collection

The major part of the data used for the model development was collected in the field. From the point of view of the neural network model development, the scope and extent of the data will directly affect the efficiency of the model. A critical issue in developing a neural network model is generalization, i.e. to have the outputs of the model approximate target values given inputs that are not in the training set. A neural network model that is not sufficiently complex can fail to fully detect the signal in a complicated data set, leading to under-fitting. However, if the model is so complex as to fit both the noise and the signal, it will cause over-fitting. Overfitting is especially dangerous because it can

easily lead to predictions that are far beyond the range of the training. Over-fitting can also produce wild predictions in multi-layer feed-forward network even with noise-free data [26, 27]. To avoid under-fitting and over-fitting, and, therefore, gain good generalization, the following conditions are necessary:

- The first necessary condition is that the inputs to the network contain sufficient information pertaining to the target, so that there exists a mathematical function relating correct outputs to inputs with the desired degree of accuracy.

- The second necessary condition is that the function that the network is trying to learn (that relates inputs to correct outputs) be, in some sense, smooth. In other words, a small change in the inputs should, most of the time, produce a small change in the outputs. For continuous inputs and targets, smoothness of the function implies continuity and restrictions on the first derivative over most of the input space.

- The third necessary condition for good generalization is that the training data set be a sufficiently large and representative subset of the set of all cases that the network is expected to generalize.

Actually, the best way to achieve satisfactory generalization is to use as much training data as possible. However, the amount of the data is limited by many factors in practice. In order to realize the necessary conditions mentioned above, during the data collection, the following rules are used to guide the selection of pavement sections:

- The pavement related variables should be evenly distributed over all the selected sections.

- The range of the crack depth should cover all the possible values of the actual crack depth.

In the project, 95 state road sections were selected from the FDOT pavement section database, which includes the pavement relevant variables such as AADT, Age, etc. All the sections distribute over the five counties in Florida (as listed in Tab. 5-1). To obtain the actual crack depth of the cracks, an Impact-Echo Test System by Physical Acoustics

Corporation was used. Fig. 5-1 shows the main components of the Impact-Echo Test System.

Table 5-1. Distribution of Pavement Sections for Field Data Collection

County Name	Number of Pavement Sections
Hillsborough	18
Pasco	8
Manatee	13
Sarasota	9
Pinellas	27
Total	75

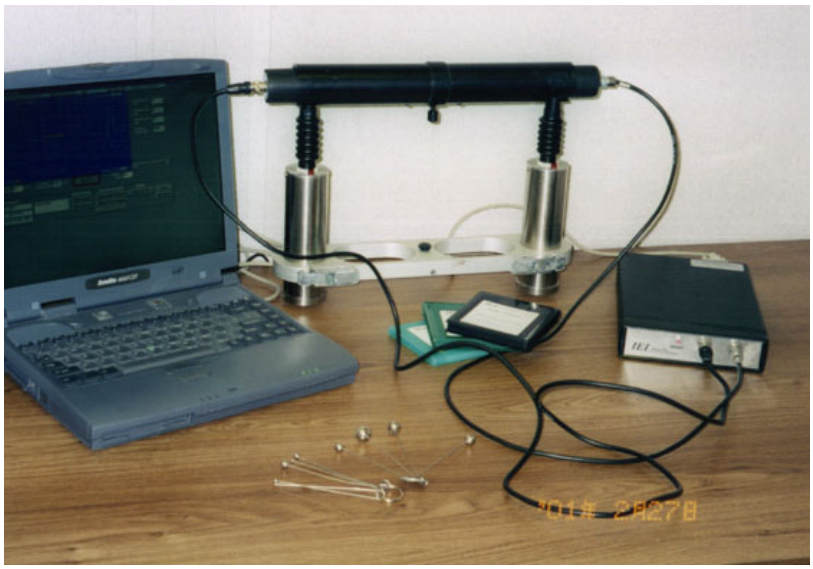


Figure 5-1. Impact Echo Test System

The crack depth measurement accuracy of Impact Echo Test System is within 4% ~ 10 % of the measurement range. The accuracy was verified by the company. The field data collection by the Impact Echo Test System includes four steps described as follows:

- Wave Speed Measurement: The travel speed of the impact wave in pavement surface was needed for the estimation of crack depth. Field test was conducted to estimate travel speed of the impact wave as shown (see Fig. 5-2).

Figure 5-2. Impact Echo Wave Speed Testing

- Determination of Measurement Points (see Fig. 5-3): This step was to determine the geometric relationships between the transducers, the impactor, and the crack.

Figure 5-3. Determination of Measurement Points for Impact Echo Testing

- Pavement Surface Polishing (see Fig. 5-4): A grinder was used to polish the measurement points to enhance the coupling condition between the pavement

surface and the transducers. This step is useful to increase the signal to noise ratio of the measurement units.

Figure 5-4. Pavement Surface Polish for Impact Echo Testing

- Crack Depth Measurement (see Fig. 5-5): The computer can automatically record the travel time of the impact wave and further estimate the crack depth according to the determined geometry relation.

Figure 5-5. Crack Depth Measurement Using Impact Echo Testing System

In the project, the entire procedure of measuring the crack depth in the field by the static method took about five months from November 2000 to March 2001. It was very time-

consuming to actually measure the crack depth by this static method. Traffic control and safety devices were used to close at least one traffic lane in the test site so that field engineers could be protected and traffic could be guided. During the field data collection, over 1500 cracks were measured from the 75 sections. In the mean time, the automatic crack depth measuring system developed in the project was also used to scan these same cracks to obtain crack opening geometric characteristics.

Core Sample Data Collection

In addition to field data, some core samples with cracks were obtained from FDOT. Crack depth of each core sample was measured with a measuring scale and the opening geometric characteristics were measured by the automatic crack depth measuring system. These core samples were used for the system validation to verify the system performance in terms of accuracy. Fig. 5-6 shows an example of the core sample. A total of 257 cracked core samples were collected in the project.

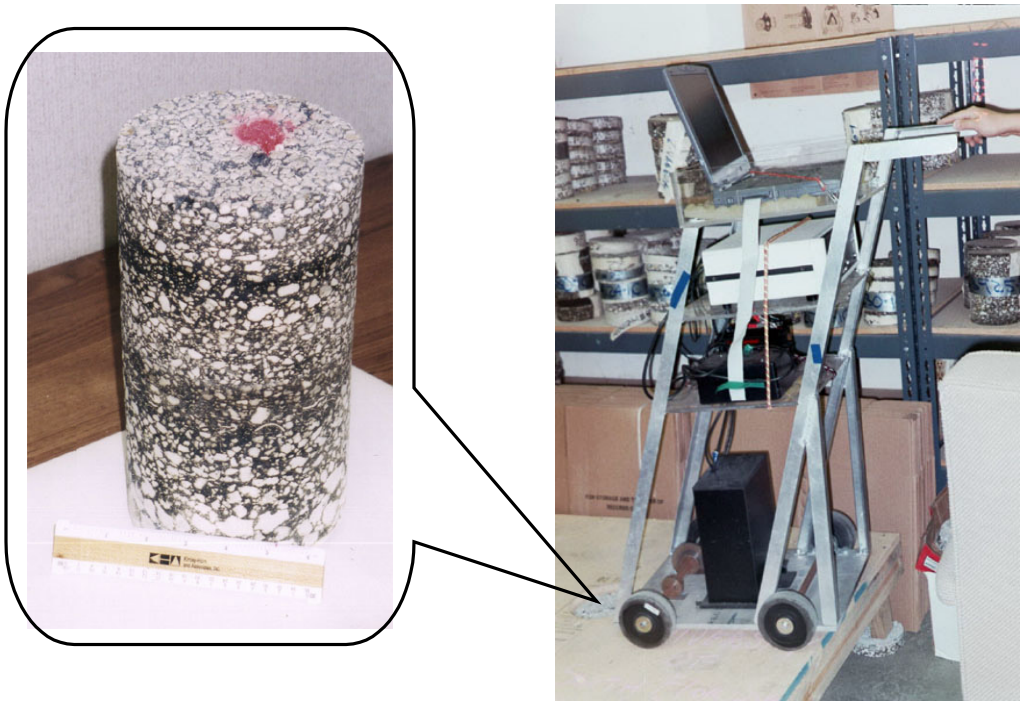


Figure 5-6. Data Collection from Pavement Core Samples

Development of a Database

With the data collected from field measurement and core samples, a pavement crack depth database was setup, which encompassed all of the variables used as the input and the output of the estimation model. The variables included in the database are listed in Tab. 5-2. The database also includes detailed information about the pavement sections, such as the RoadID, road name, starting position, and ending position. According to that information, the location of the pavement section can be easily determined. Distributions of the major variables are shown in Figs. 5-7 to 5-9.

Table 5-2. List of Variables Used in Model Development

	Variable Name		Unit
Input (Predictor)	Pavement Variables	AADT	vpd
		Truck Percentage	Percentage
		Age	Year
		Number of Lanes	N/A
	Measurement Variables	Crack Depth	Inch
		Crack Width	Inch
		Slop One	Radian
		Slop Two	Radian
Output (Target)	Actual Crack Depth		Inch

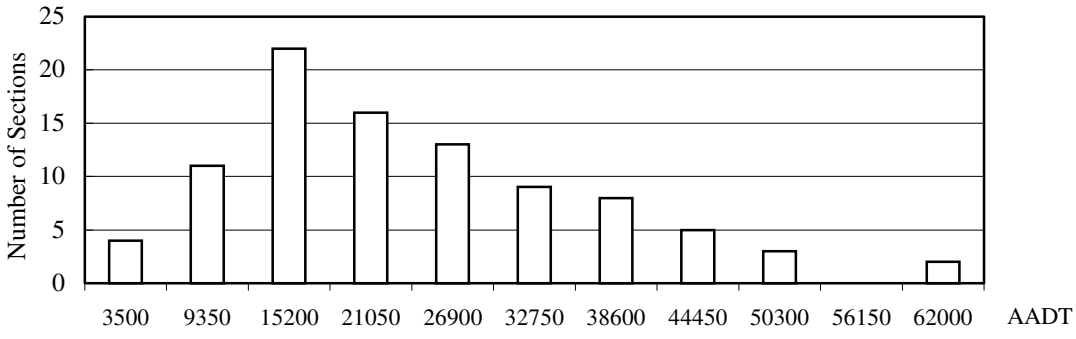


Figure 5-7. Distribution of AADT

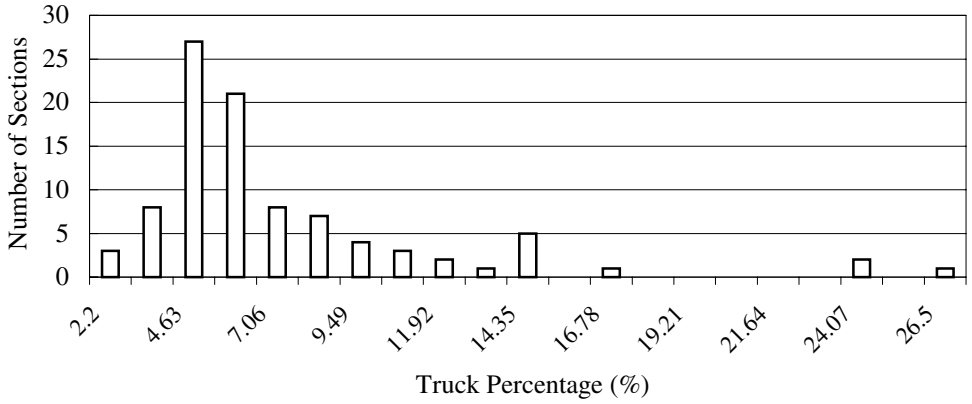


Figure 5-8. Distribution of Truck Percentage

Figure 5-9. Distribution of Pavement Age

MODEL DEVELOPMENT

Model development is a procedure in which the network tries to learn the inside relationship between the input and output vectors. The objective of the model development is to find a network with the best performance by means of adjusting the network architecture and tuning the inside connections of the network. In the research, to enhance the network training efficiency, the original data in the database were preprocessed. Furthermore, different training algorithms were compared. The flow chart of the neural network development is shown in Fig. 5-10.

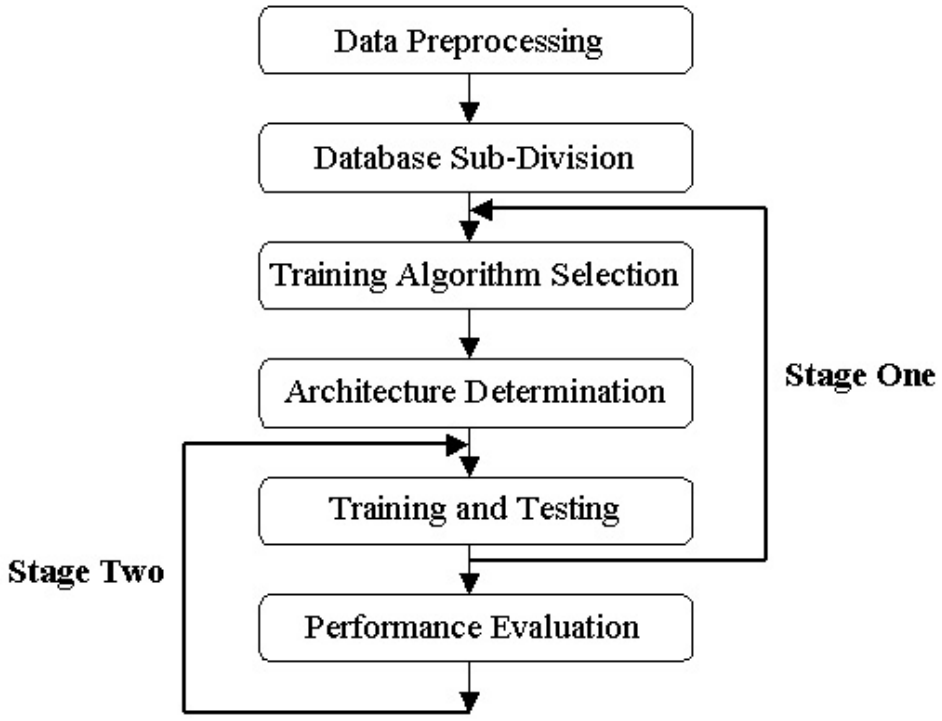


Figure 5-10. The Flow Chart for Neural Network Model Development

Data Preprocessing

As mentioned earlier, the model development is a nonlinear optimization problem in which the training algorithm keeps looking for the so-called global minimum in an error surface with many local minima. The inherent difficulty of the problem is aggravated by the typically very high dimension of the weight space. The original database gained from the data collection is comprised of a variety of quantities with different units and different value ranges. At the same time, some variables in the database are correlated. All of these will increase the dimension of the weight space and the irregularity of the error surface and further decrease the training efficiency. The data preprocessing and coding is useful to enhance the training efficiency. The procedure can be summarized as follows:

- Normalization: The normalization procedure normalizes the data (both inputs and targets) so that they will have zero mean and unity standard deviation. It will make the training process better behaved by improving the numerical condition.

- Principal Component Analysis: This step orthogonalizes the components of the input vectors (so that they are uncorrelated with each other). It orders the resulting orthogonal components (principal components) so that those with the largest variation come first, and it eliminates those components which contribute the least to the variation in the data set.

Even though there are many applications in which the importance of the inputs has been evaluated, the question of which features in the training set are used by a particular feed-forward network can be excruciatingly difficult to answer [28]. The main point is that there is no single measure of importance that is appropriate for all the applications. A common measure of the importance of an input is the change in the error function when the input is removed from the network. Unfortunately, it is not practical and may even be impossible to try all the possible combinations of the input vectors to evaluate the error function. On the other hand, it is important to retrain the network after removing each input, which can be time-consuming. Otherwise, the outputs of the network are likely to be meaningless. Nevertheless, there is no developed rule to guide the selection of an optimal subset of inputs according to the measure of importance. More importantly, removal of some inputs may affect the importance of the others in the model. Since the number of the input variables in this study is under the control of the neural network model, all the available variables listed in Tab. 5-2 are used in this project.

Usage of the Database

In this project, the preprocessed database was divided into the three sub-sets as shown in Fig. 5-11. Each set had different purpose as described as follows:

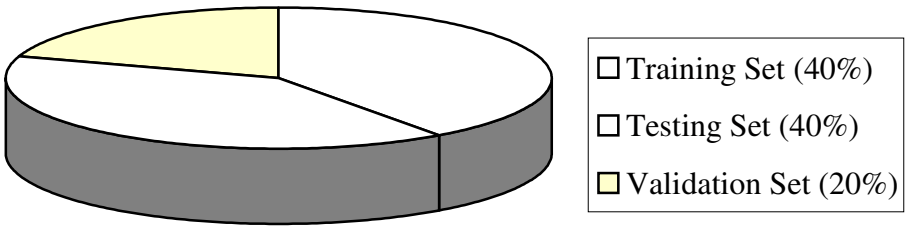


Figure 5-11. Usage of Database

- Training set: A set of data used for learning or fitting the parameters [i.e., weights] of the neural network.

- Testing set: A set of data used to tune the parameters [i.e., architecture, not the weights] and assess the performance of the network.

- Validation set: A set of data used to confirm the performance [generalization] of a fully specified network and control the training procedure.

As shown in Fig. 5-10, the procedure for neural network model development comprises two stages. In the first stage, with the use of the training and testing data set, the training algorithm and network architecture will be determined. In the second stage, with the determined network model, validation set is used to control the tuning of the network weights, whereas, the testing data set will be used to gain an unbiased estimation of the generalization error.

Network Architecture

Mathematically, a network is represented by a weighted and directed graph, in which a collection of nodes are connected by directed or oriented links that carry associated weights as shown in Fig. 2-7. As to the multi-layer feed-forward network, there are two major indices for network topology description, i.e. the number of layers and the number of units in each layer. Furthermore, in this project, the hidden layer and hidden units in each hidden layer are of more concern since the input and the output layer are fixed. The network topology determines the function implemented by the network. Generally, one hidden layer with an arbitrarily large number of units suffices to achieve universal approximation problems [19]. Nevertheless, the need to construct a two-hidden layer network arises in some special problems, such as image and speech recognition. The multiple hidden layer architectures are motivated by attempts to incorporate spatially or temporally localized features. Unfortunately, using more than two hidden layers exacerbates the problem of local minima, and it is important to use many random initializations or other methods for global optimization. Therefore, in this project, only

one and two-hidden layer architectures are investigated. As to the number of the hidden units in each layer, in most situations, there is no way to determine the best number of hidden units without training several networks and estimating the generalization error of each. Too few hidden units will get high training error and high generalization error due to under-fitting and high statistical bias, whereas too many hidden units may get low training error but still have high generalization error due to over-fitting and high variance. In practice, the best number depends in a complex way on:

- Numbers of input and output units,
- Number of training cases,
- Amount of noise in the targets,
- Complexity of the function or classification to be learned,
- Network architecture,
- Type of hidden unit activation function,
- Training algorithm, and
- Regularization.

Even though some books and articles offer rules of thumb for choosing architecture, the best number of hidden units may be determined through network training experiments. In addition, as mentioned previously, the activation functions for the hidden units are needed to introduce non-linearity into the network. With non-linearity, hidden units would make the network more powerful. Almost any nonlinear function does the job. In this study, the activation function used by the backpropagation learning must be differentiable and bounded. The *sigmoidal* functions such as logistic and *tanh* and the *Gaussian* function are the most common choices. Nevertheless, functions such as *tanh* and *tansig* that produce both positive and negative values tend to yield faster training than functions that produce only positive values such as *logistic*, because of better numerical conditioning. Finally, the *tansig* activation function is applied due to its faster Matlab implementation. Its algorithm is given in the following equation:

$$y = \frac{2}{1 + exp(-2 * x)} - 1 \tag{5-1}$$

Selection of Network Training Algorithm

Methods for network training have been studied for hundreds of years, and there is a huge literature based on the subject fields such as numerical analysis, operations research, and statistical computing. Actually, there is no single best method for solving a given problem. The method selection is based on the characteristics of the problem to be solved. For feed-forward networks with the most popular differentiable activation functions and error functions, three general types of algorithms have been found to be effective for most practical purposes:

- For a small number of weights, stabilized Newton and Gauss-Newton algorithms, including various Levenberg-Marquardt (LM) and trust-region algorithms, are efficient. The memory required by these algorithms is proportional to the square of the number of weights.

- For a moderate number of weights, quasi-Newton algorithms are efficient. The memory required is proportional to the square of the number of weights.

- For a large number of weights, various conjugate-gradient algorithms are efficient. The memory required by these algorithms is proportional to the number of weights.

Another important consideration in the choice of the algorithms is that neural networks are often ill-conditioned, especially when there are many hidden units. Algorithms that use only first-order information, such as steepest descent and standard Backpropagation, are notoriously slow for ill-conditioned problems. Generally speaking, an algorithm using more second-order information may result in better behave under ill-conditions [29]. Compared with other algorithms, the quasi-Newton methods and the LM algorithm [30] use more second-order information. In the Newton method, the network connections (weights and bias) are updated by the following equation:

$$w_{k+1} = w_k - A_k^{-1} g_k \tag{5-2}$$

where, A is the Hessian matrix (second derivatives) of the performance index at the current values of the weights and biases.

Newton's method often converges faster than other methods. Unfortunately, it is complex and expensive to compute the Hessian matrix for feed-forward neural networks. The LM algorithm, like the Newton's method, is designed to approach second-order training speed while it uses Jacobian matrix to approximate the Hessian matrix. The Jacobian matrix can be computed through a standard backpropagation technique that is much less complex than computing the Hessian matrix. Thus, the basic step of the LM algorithm has the following equation:

$$w_{k+1} = w_k - \left[J^T J + \mu I\right]^{-1} g_k \tag{5-3}$$

and the gradient can be computed by:

$$g_k = J^T e_k \tag{5-4}$$

where, J is the Jacobian matrix which contains first derivatives of the network errors with respect to the weights and biases, and e_k is a vector of network errors. When the scalar μ is zero, this is just the Newton's method. When μ is large, this becomes gradient descent with a small step size. The aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm. Therefore, in this research, the LM algorithm was finally selected to train the neural network.

Neural Network Training and Testing

The main goal of training and testing is to determine the network with the best performance. In this project, various networks with different architectures were trained by minimizing an appropriate error function defined with respect to a training data set. The training software, called MATLAB, was used as the major tool for network development. The Neural Network Toolbox in MATLAB is a powerful collection of functions for design, training, and simulation of neural networks, and it supports a wide range of network architectures with an unlimited number of processing elements and interconnections (up to operating system constraints). MATLAB supports various

architectures and training methods. It includes several variations on backpropagation (including the fast Levenberg-Marquardt algorithm). The Toolbox is delivered as MATLAB M-files, enabling users to see the algorithms and implementations, as well as to make changes or create new functions to address a specific application.

Overall, the training algorithm can be implemented in two different ways: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode, all of the inputs are applied to the network before the weights are updated. The incremental learning is often used for on-line, constructive, or sequential learning [31]. Since the objective of the modeling was to develop a network that could be used as a component in the final system for crack depth prediction, there was no need for on-line learning. Thus, the batch mode was used, i.e. the weights and biases of the network were updated only after the entire training set had been applied to the network.

In the standard backpropagation algorithm, it is difficult to decide the exact value for the learning rate. A low learning rate makes the network learn slowly, and a high learning rate will make the weights and objective function diverge. Figs. 5-12 and 5-13 show the training procedures with low and high learning rates, respectively.

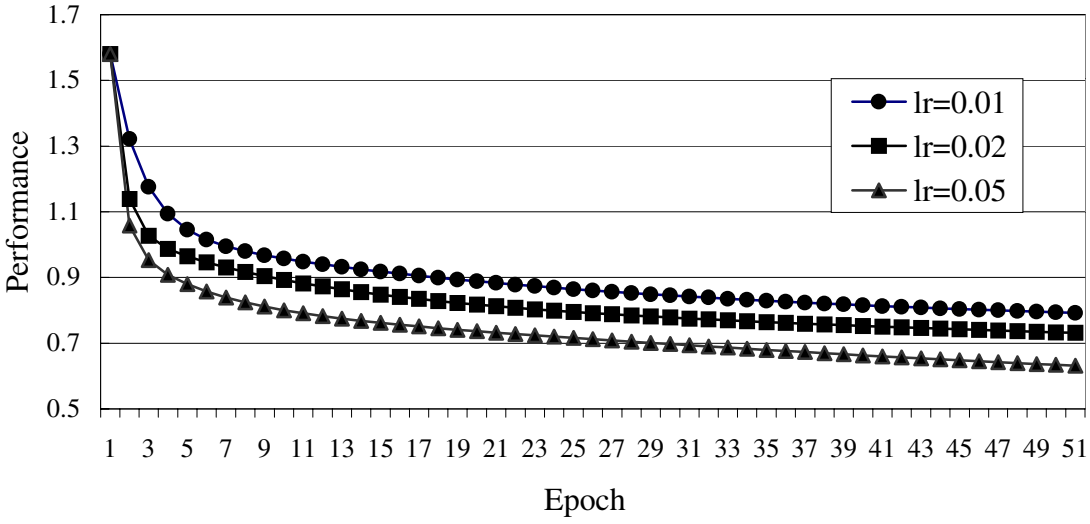


Figure 5-12. Network Training with Low Learning Rates (lr)

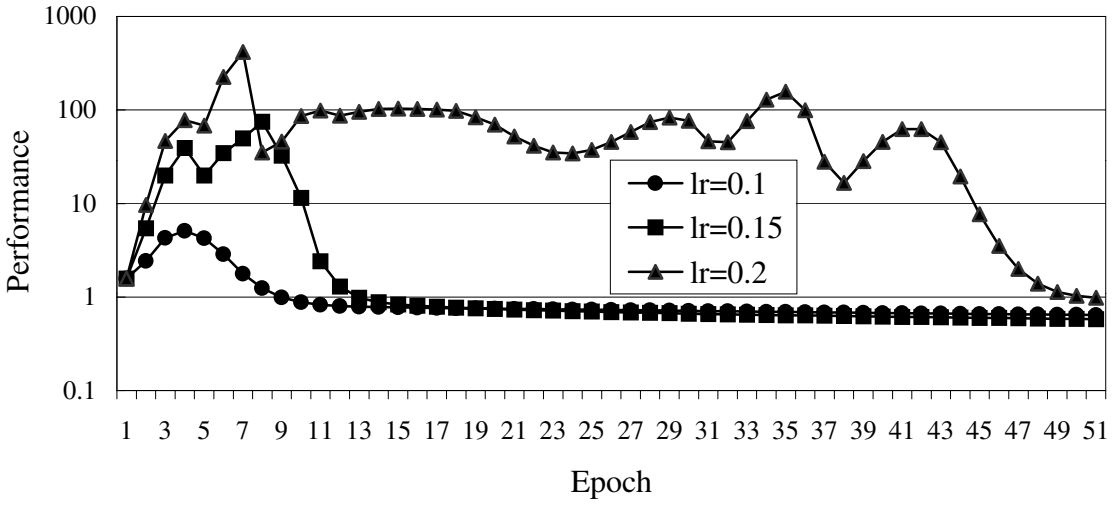


Figure 5-13. Network Training with High Learning Rates (lr)

In this research, the training performance, mse, was calculated by the following equation:

$$mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \tag{5-5}$$

where t represents the network targets, a represents the network outputs, and N represents the number of outputs. Actually, as to the typical feed-forward neural network with hidden units, the objective function has many local and global optima [31]. Hence, the optimal learning rate often changes dramatically during the training process. Thus, there is no need to use a constant learning rate like the standard backpropagation method. As mentioned earlier, there are more efficient, reliable, and convenient methods, in which the learning rate is adjusted automatically according to the training process. A brief introduction of each algorithm is provided in Tab. 5-3. With the different algorithms, the training processes were compared and the comparison results are presented in Fig. 5-14. In the figure, it is indicated that the function, *trainlm* which is a LM algorithm, yields the fastest training. It is further proved that the algorithm selection was reasonable to this application.

To determine the network architecture, i.e. the number of hidden layers and the number of hidden units in each layer, a practical way is to train several networks and estimate the

training error and the generalization error [26]. The network prediction error was used as the major measure of the generalization error in the project. There are many combinations of network architectures. Figs. 5-15 and 5-16 present part of the experimental results for one-hidden layer and two-hidden layer architectures, respectively.

Table 5-3. Summary of Different Training Algorithms

Function Name in MATLAB	Description
traingd	Basic gradient descent, slow response, can be used in incremental mode training.
traingdx	Adaptive learning rate, faster training than *traingd*, but can only be used in batch mode training.
trainrp	Resilient backpropagation, simple batch mode training algorithm with fast convergence and minimal storage requirements.
traincgp	Polak-Ribiére conjugate gradient algorithm, slightly larger storage requirements than *traincgf*, faster convergence on some problems.
trainbfg	BFGS quasi-Newton method, requires storage of approximate Hessian matrix and has more computation in each iteration than conjugate gradient algorithms, but usually converges in fewer iterations.
trainlm	Levenberg-Marquardt algorithm, fastest training algorithm for networks of moderate size, has memory reduction feature for use when the training set is large.

In Fig. 5-15, one-hidden layer was used for network training, in which the number of hidden units were changed from 5 to 15 (Testing_5 – Testing_15). In Fig. 5-16, two-hidden layer network was used. However, the number of hidden units in the first hidden layer was fixed at 10 and the number of hidden units in the second hidden layer was changed from 1 to 10 (Testing_10 to Testing_10_10).

From Figs. 5-15 and 5-16, it can be seen that the network training error keeps decreasing while the number of the hidden layer and the number of hidden units in each layer increases. However, the testing errors, a measure of the network generalization error, begin to increase after the number of hidden units increase to a certain number.

According to the training experimental results shown in the Figs. 5-15 and 5-16, it is indicated that a network with ten hidden units in the first hidden layer and five hidden units in the second hidden layer yields the best performance.

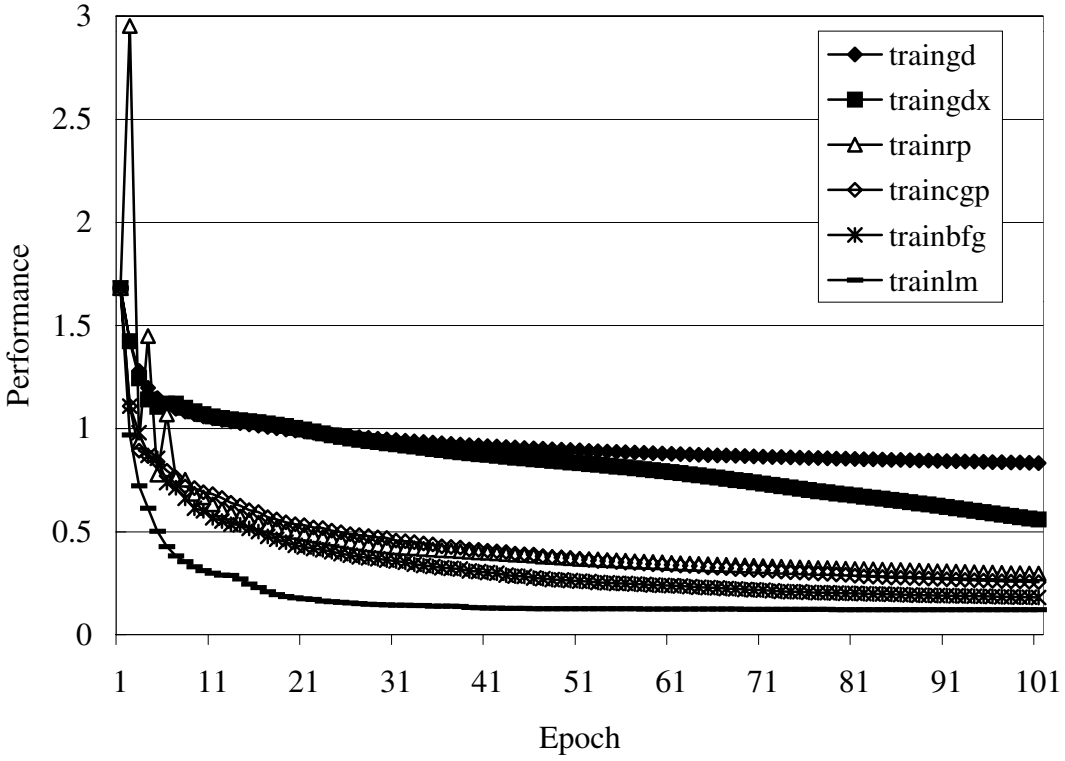


Figure 5-14. Network Training with Different Training Algorithms

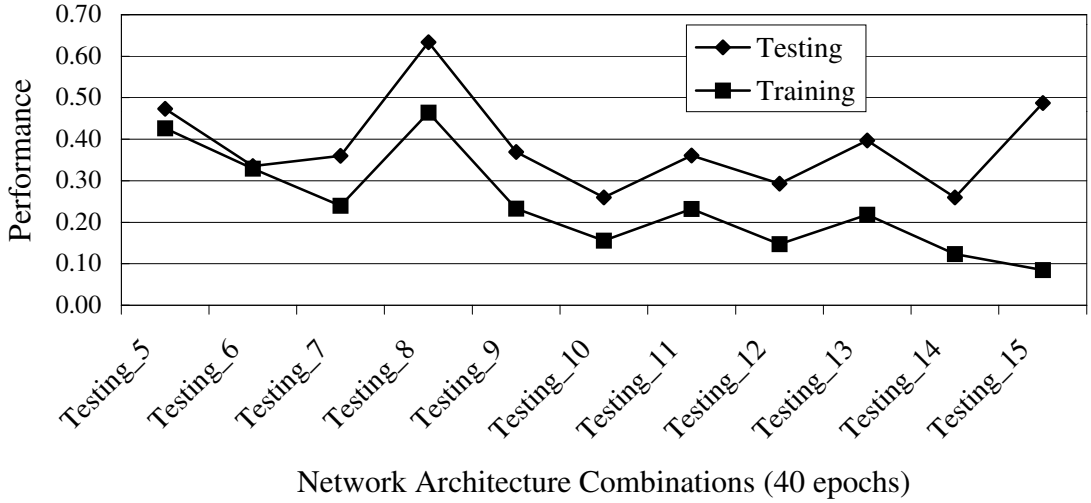


Figure 5-15. Training with Different Network Architectures (One-Hidden Layer)

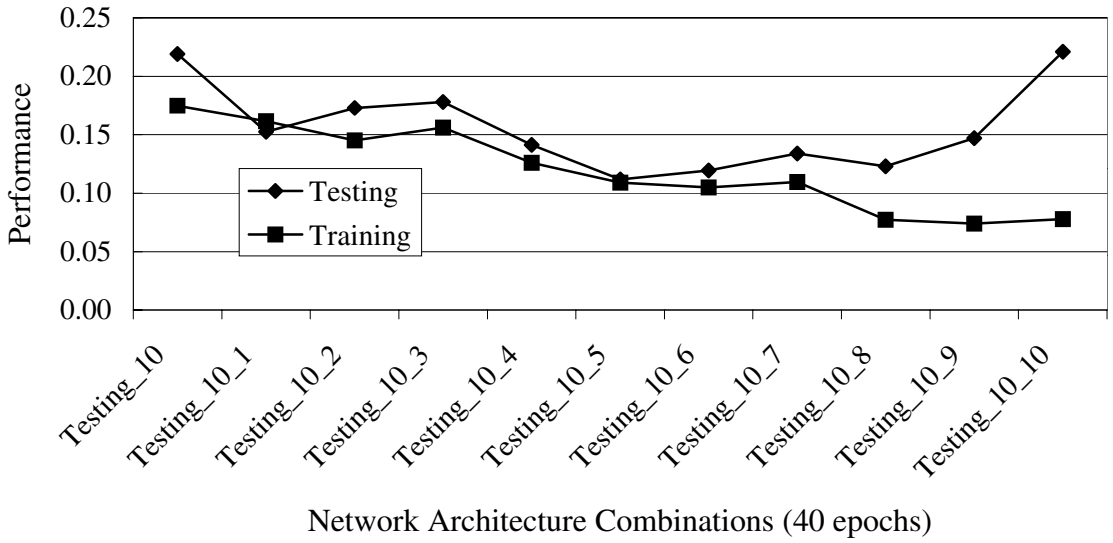


Figure 5-16. Training with Different Network Architectures (Two-Hidden Layer)

Based on the results from a series of training experiments, it was found that adding more hidden layers did not bring any improvement in network performance. In some cases, as compared with the two-hidden-layer architectures, adding more hidden layers even deteriorated the network. In fact, the network generalization power may become unstable as the network complexity keeps increasing. Experimental results showed that use of more hidden layers exacerbated the problem of local minima and further reduced the generalization ability of the network. Thus, a two-hidden-layer network with *10* hidden units in the first hidden layer and 5 hidden units in the second hidden layer was finally selected as the architecture for the neural network model for crack depth estimation. As indicated by the experimental results, the network with such an architecture may result in the best performance in terms of the estimation accuracy.

MODEL PERFORMANCE ANALYSIS

The final objective of training and testing a network is to find the network having the best performance on new data instead of the known data. Thus, the error function of the existing data should not be used as the single criterion to evaluate the performance of the network. The simplest approach to compare different networks is to evaluate the error

function using data which is independent of that used for training. This approach is called the hold out method [32]. Since this procedure can itself lead to some over-fitting to the testing data set, the performance of the selected network should be confirmed by measuring its performance on a second independent data set - validation set.

With the use of the training data set and testing data set on the network model, the process of network training is shown in Fig. 5-17.

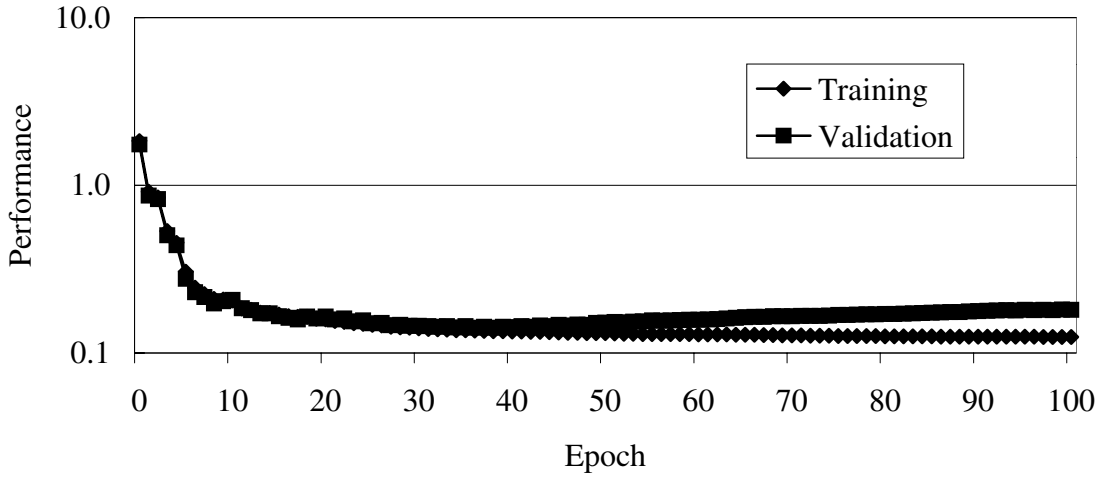


Figure 5-17. Network Training Process Using the Final Model Selected

It can be seen from Fig. 5-17 that the validation error decreases during the initial phase of training, so does the training set error. However, when the network begins to over-fit the data, the error on the validation set begins to rise. With continual training, the over-fitting becomes rock-ribbed. To avoid over-fitting efficiently, a typical recommendation is that the number of weights should be no more than 1/30 of the number of training cases [33]. Unfortunately, this was not feasible in this application. Thus, a method called early stopping [34] was adapted in this project. The basic idea behind the early stopping is to compute the validation error periodically during the training and stop the training when the validation error starts to go up. Early stopping can be applied successfully to networks in which the number of weights approximate the sample size and even far exceeds the sample size. Furthermore, it requires only one major decision to be made, i.e. what proportion of validation cases to use. A simulation study [35] suggested a trend in which a more complex function requires a smaller validation percentage. Hence, in the

project, the validation set proportion was set to 20% and both the training and testing set consisted of 40% of the whole data set. Fig. 5-18 shows the training process in which the early stopping was applied in the project. With the application of early stopping, an unbiased estimation of network generalization error could be gained according to the network performance on the testing data set. The results are shown in Fig. 5-19, in which linear regression was performed to illuminate the network performance.

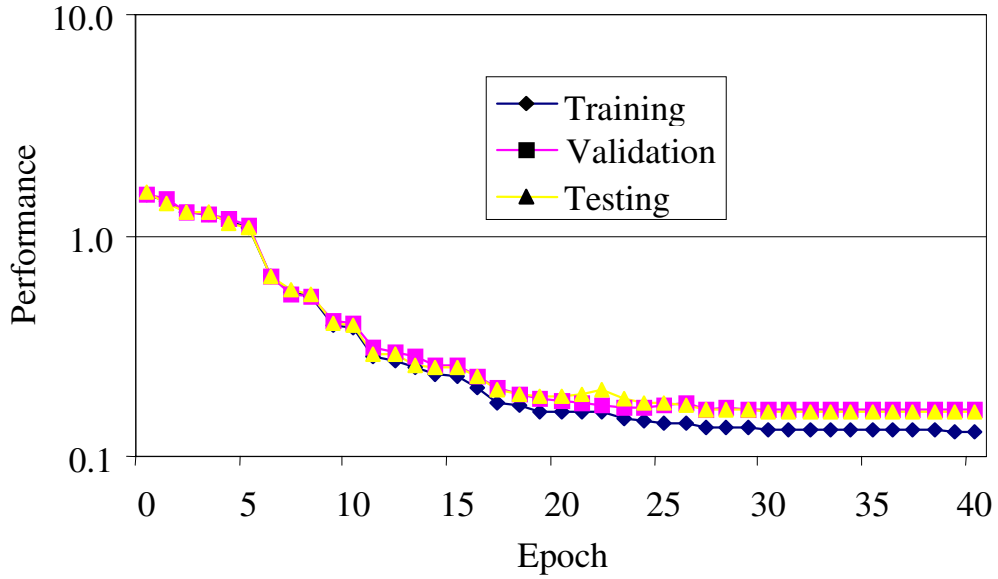


Figure 5-18. Network Training Process With Early Stopping

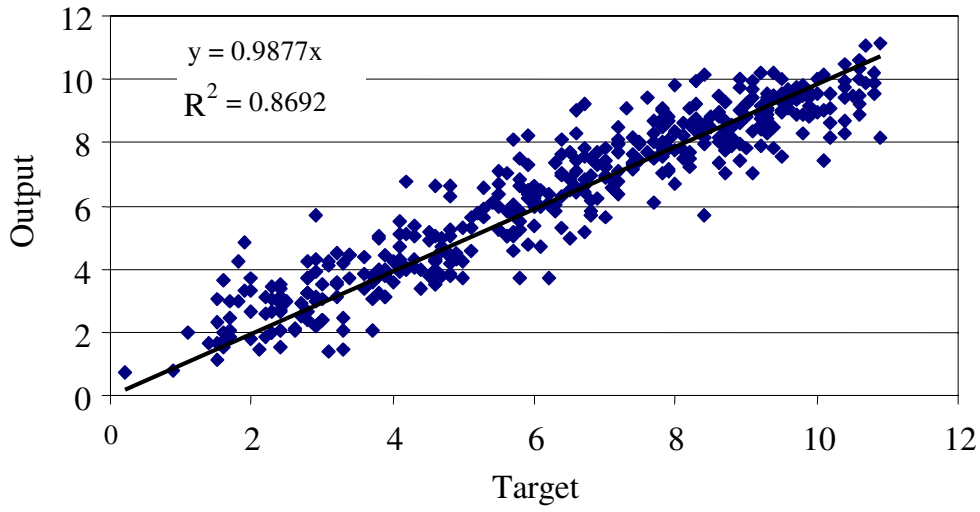


Figure 5-19. Network Performance Based on Testing Data Set

The performance analysis results based on the entire data set (including training, testing, and validation) are presented in Fig. 5-20. Figs. 5-21 and 5-22 present the network prediction-error range and its distribution, respectively. Performance analysis results based on different data sets (such as training, testing, and validation data sets) can also be compared. The estimation errors based on different data sets are presented in Fig. 5-23. The performance results presented in these figures show that the network outputs tracked the target reasonably well. According to the performance analysis results, it can be concluded that the neural network model developed in the research can be used to estimate the actual crack depth with reasonable accuracy.

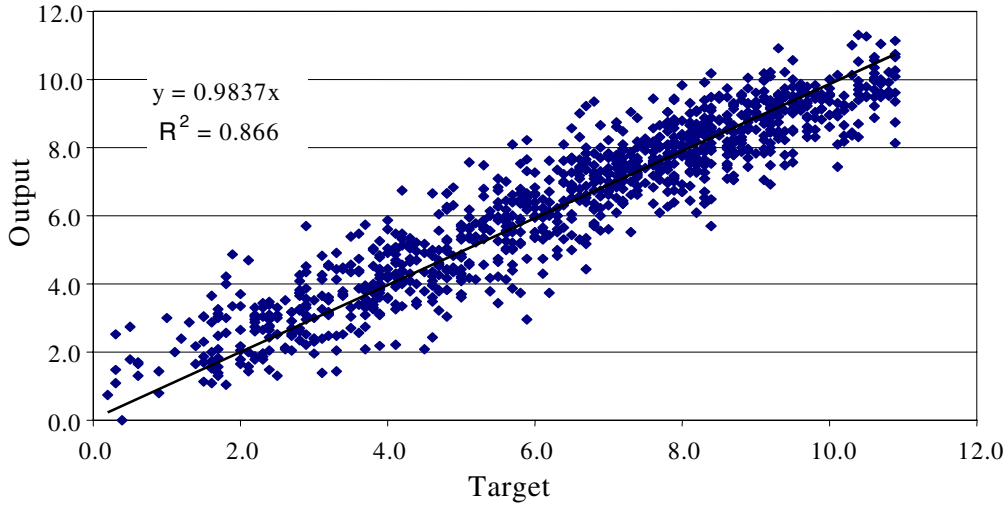


Figure 5-20. Network Performance Based on the Entire Data Set

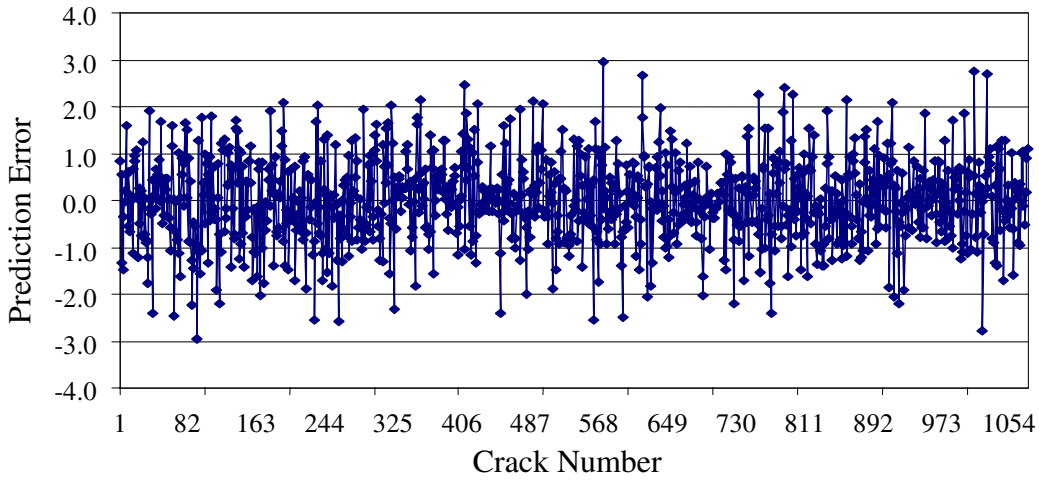


Figure 5-21. Network Estimation Error

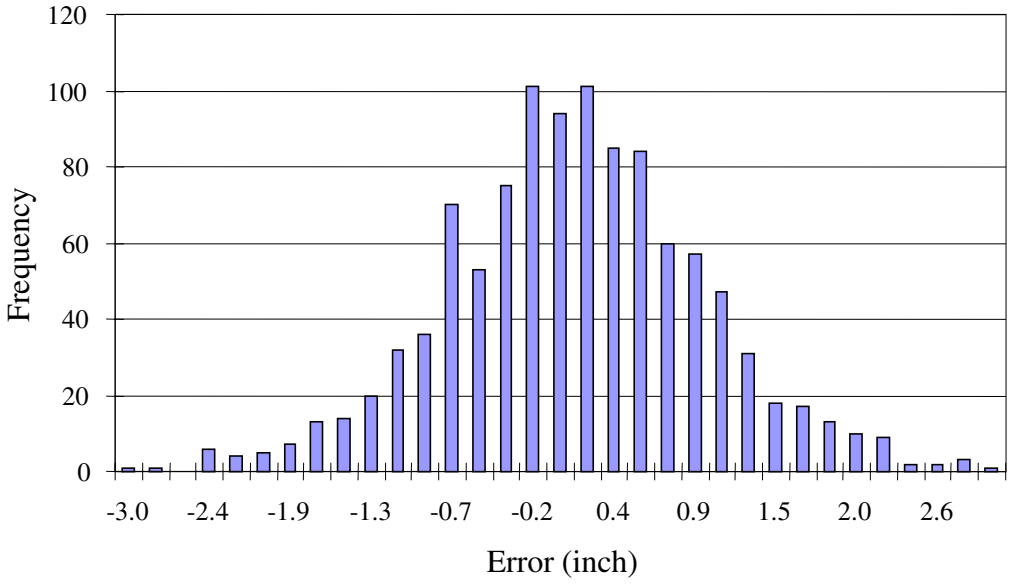


Figure 5-22. Network Estimation Error Distribution

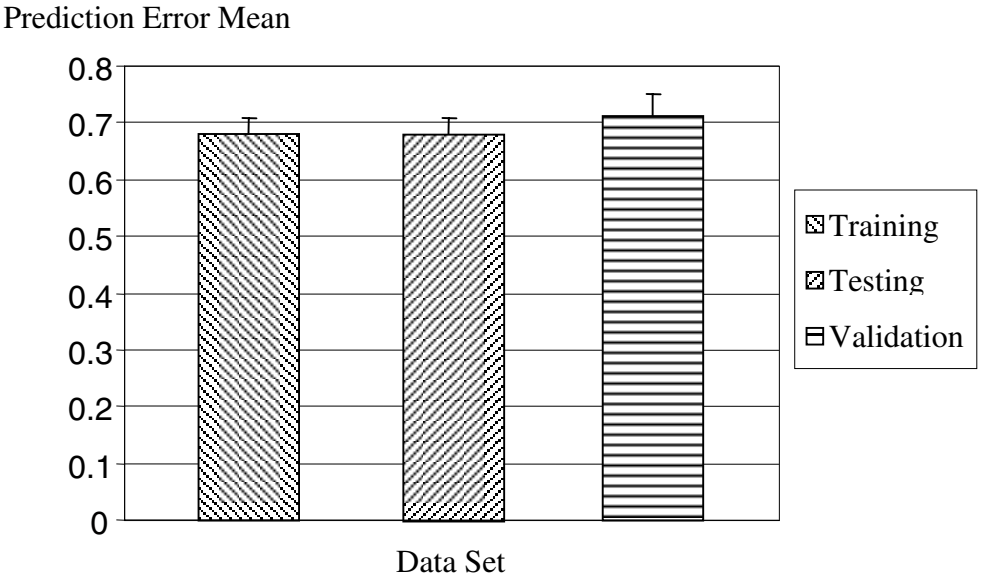


Figure 5-23. Estimation Errors Corresponding to Different Data Sets

MODEL VALIDATION BY DATA FROM CORE SAMPLES

To further verify the accuracy of the model developed in the project, model validation based on data from core samples was performed. The basic idea was to measure the crack on each core sample by the automatic system and use the model to estimate the depth of crack appeared on each sample with other pavement related information provided by

FDOT Materials Office. Then, the estimated crack depth was compared with the real crack depth manually measured in the lab. However, the information on crack types (transverse crack or longitudinal crack) was not available. Thus, the bias caused by crack type factor could be introduced in the validation. The validation results are shown in Figs. 5-24 to 5-26. Fig. 5-24 shows the goodness of fit of the estimated crack depth data with the actually crack depth data measured from core samples. The estimated crack depth data were obtained by the automatic crack depth measuring system developed through the project. Fig. 5-25 presents the estimation error between the real crack data and the estimated crack depth data. Fig. 5-26 depict the error distribution. From these figures, it is indicated that the validation analysis showed that the system including the hardware and software can produce reasonable estimation results. However, as compared with the validation results from the data collected by the impact echo system, the validation results from core samples did not show as good accuracy as the validation results from the impact echo data. Part of the main reasons for this could be due to the fact that no information on crack types was available for the crack data from the core samples.

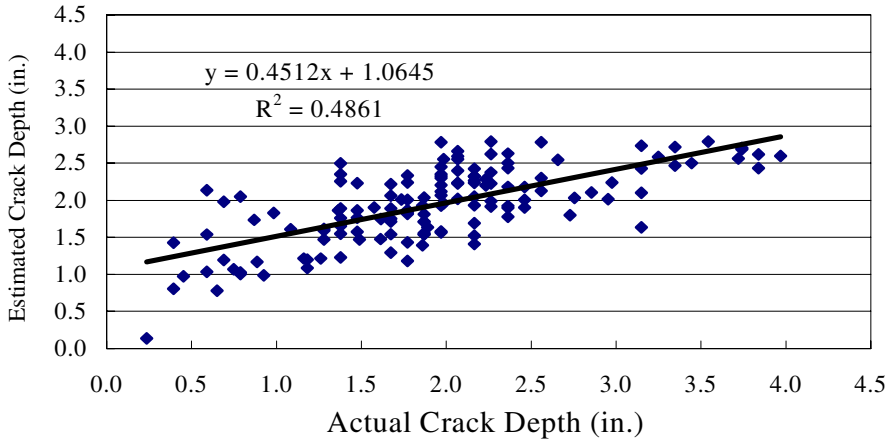


Figure 5-24. Goodness of Fit from Validation Analysis

CONCLUSION

The model developed in the research and modeling results presented in this chapter indicated that neural network models can be used to estimate crack depth. The model inputs should include crack opening geometric characteristics and pavement section

related information. However, since the real cracks measured in the field by the system and the impact echo were only transverse and block cracks (no sufficient field support to measure longitudinal cracks.), the model development (called modeling) was based on transverse and block cracks only. To make the model applicable to longitudinal cracks, more field data collection efforts should be conducted to get more longitudinal cracks.

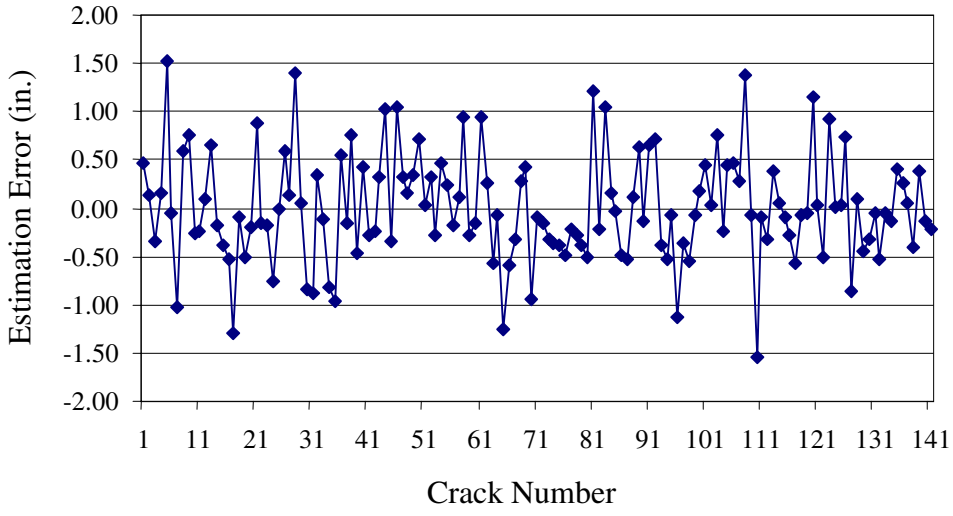


Figure 5-25. Estimation Error from Validation Analysis

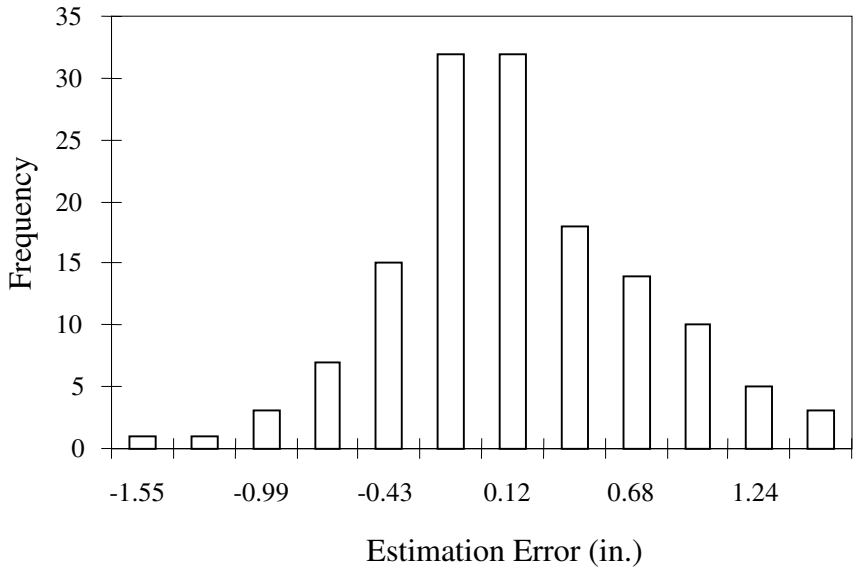


Figure 5-26. Validation Error Distribution

As stated previously, the system mainly consists of three major sub-systems. The first sub-system is to automatically measure the distance traversed by the system; the second sub-system is to automatically identify cracks; and the third sub-system is to estimate crack depth with the data obtained from the first two sub-systems and the pavement section related information. The main element in the third sub-system is the neural network model which can estimate crack depth based on crack opening geometric characteristics and pavement section related information. With the combination of all three sub-systems, the system can automatically measure cracks and estimated crack depth while moving along the pavement section.

CHAPTER 6. SUMMARIES, CONCLUSIONS, AND RECOMMENDATIONS

SUMMARIES

Cracking is a predominant form of pavement distress manifestation and, hence, crack evaluation is an important task to pavement management. Rapid and reliable methods for estimating crack depth will indeed facilitate more effective pavement maintenance and rehabilitation decision-making.

In recent years, research studies have been performed to develop automatic and non-destructive techniques to evaluate pavement surface cracking. However, based on the literature review, it was found that most of the existing systems only focus on identifying crack location, crack length, and crack width. Up to now, core sample method is the only way used by departments of transportation in US to collect crack depth data for pavement management. As a destructive method, core sampling is both time and resource consuming.

To realize automatic detection of pavement surface crack depth on Florida roadways in a non-destructive way, a research project was performed in last two years by the Department of Civil and Environmental Engineering at the University of South Florida and sponsored by Florida Department of Transportation. In the project, the research team reviewed existing documents and information databases to search available technologies applicable to the application. Based on the literature search and review, no technologies available to be used in fields to dynamically estimate pavement surface crack depth. Some preliminary experiments were performed to test which types of sensors can be used for the application. Based on the preliminary experimental results, it was concluded that laser sensors with high resolution, high accuracy, and high sampling rate could be used to indirectly measure pavement crack depth. However, a model was needed to estimate crack depth with the support of information on crack opening geometric characteristics and pavement section characteristics. This concept to indirectly estimate crack depth was presented at a project meeting attended by pavement and materials engineers from Florida Department of Transportation and was approved in the meeting.

In the project, a prototype system was developed to automatically and dynamically measure pavement cracks and estimate the crack depth. The prototype system can automatically detect the existence of cracks and their exact positions on the roadways. Then, crack depth can be estimated by the neural network model installed in the system. The system mainly consists of three sub-systems, including distance measuring, crack identifying, and crack depth estimation. The main efforts in the research were to develop these sub-systems and calibrate the models in these sub-systems by collect field data or core samples.

The system hardware includes two Microtrak 7000 laser measurement units and one M12x1-180ASN-500 analog speed measuring unit. A NI-DAQCard-1200 is used for data acquisition and digitalization. A TOSHIBA laptop computer with a Pentium II processor performs the system central control and signal processing. The measurement units installed in the prototype system can accurately record the microscopic pavement profile, including detailed vertical texture changes. With a user-friendly interface, the system software achieves the signal processing and data management. In the system signal processing, a Scan-Rate-Effect-Canceling (SEC) model developed in the project is used to calibrate the distance measurement. The SEC model further enhances the system measurement accuracy. In the project, a new algorithm, called Partial Cross Correlation (PCC) algorithm, was developed for crack identification. The PCC algorithm can effectively reduce the possibility of false crack detection and significantly enhances the accuracy for crack identification. To estimate crack depth, a neural network model was developed to map the relationship between the system measures and the actual crack depth. The main inputs to the neural network model are crack opening geometric characteristics and pavement section related information.

In the prototype system, the laser sensors are installed with the face close to pavement surface. For practical use, the space between sensor surface and pavement surface should be large enough so that the sensors will not be damaged when the vehicle is moving at high speed. Thus, new laser sensors will be searched to ensure the space between the sensor surface and pavement surface is large enough.

To develop and calibrate the neural network model, field data were collected from five counties in Florida with the use of an impact echo system and the prototype system. The impact echo system was used to statically measure actual crack depth and the prototype system was used to measure the crack opening geometric characteristics. In addition, many core samples were collected from Florida Department of Transportation for the purpose of model validation. The data collected from fields by the impact echo system and the prototype system were used for the training and testing of the neural network model. Based on a series of training and testing experiments, different network training algorithms and network architectures were investigated and compared. With the determined architecture and algorithm, early stopping was introduced into the final network training and testing. Early stopping technique efficiently avoids the over-fitting to the given data. According to the final testing and validation results, it was proved that the developed estimation model could produce satisfactory generalization and reasonable accuracy for crack depth estimation.

CONCLUSIONS

Based on literature review and preliminary experiments performed in the research, it was concluded that no existing technologies available to directly measure highway pavement surface crack depth, and laser sensors plus a model could be used to indirectly and dynamically estimate pavement crack depth. Current available technologies can only statically estimate crack depth for the structures with uniform materials such as concrete and metal materials. Such technologies cannot be practically applied to highway pavement situation. Therefore, it was decided to use laser sensors plus estimating models for the application.

Three models were developed in the research to measure distance traversed by the system, to identify cracks, and to estimate crack depth, respectively. Development and calibration of the three models were based on field data collected during the research. Based on field experimental results, the following conclusions can be made:

- The prototype system can accurately measure the distance with an error of less than 0.5%.

- The prototype can identify cracks and locate crack locations with reasonably good accuracy as shown in Table 4-1. The algorithm developed in the project to identify cracks performed better than other algorithms developed in past years, in terms of crack detection rate and false crack detection rate.

- Based on training and testing results, a two-hidden-layer neural network model with 10 hidden units in the first hidden layer and 5 hidden units in the second hidden layer was finally selected as the architecture for the neural network model for crack depth estimation. Performance evaluation results indicated that the prototype system can estimate crack depth with a reasonably good accuracy as shown in Figs. 5-19 and 5-20. The correlation coefficient (R^2 value) between the crack depth data estimated by the prototype system and the crack depth data obtained from field data collection by the impact echo system was larger than 0.866. However, based on the validation results, the correlation coefficient between the crack depth data estimated by the prototype system and the core crack measurements was 0.4861 (Figure 5-24). It should be noted that there were only 257 core samples available for validation purpose. The insufficient number of core samples could result in poor correlation.

The prototype system can automatically identify cracks and estimate crack depth. The main technologies used in the system are sensor technology, computer interface technology, signal processing technology, and modeling technology. With the use of the automatic system, the current process for crack depth estimation can be greatly improved, in terms of time, cost, and efficiency.

The crack data collected from fields were all transverse or block cracks. Thus, the model for crack depth estimation may be applicable to transverse or block cracks only. More data from other types of cracks will be needed to develop models to estimate crack depth of all types.

The prototype system uses a laptop computer to sample data with very high sampling density (very small sampling interval). Also, the prototype displays the pavement microscopic texture profile graphics in real time format, which definitely costs significant

computer time and greatly limits the moving speed of the prototype with given sampling interval. Thus, the system with current format cannot be moved at high speed due to the limitation of sampling rate. To practically use the system for real crack depth estimation at normal traffic speed, at least three steps should be taken: (1) an industrial computer with higher sampling frequency should be used; (2) a lower sampling density (interval) should be adopted; and (3) the system should not display real-time profile graphics, but only display data information. With such improvements, the system will be able to move at normal traffic speed.

Overall, the methodologies developed in the project are applicable to the application. The results obtained from the project show promising to develop a real system to dynamically and automatically estimate pavement crack depth.

RECOMMENDATIONS

The current format of the prototype can only be used as an experimental tool. It cannot be practically used in real application. A much larger scale research is needed to implement the research finding obtained in the research. The proposed research to be performed in the future will be to develop a system that can be operated at normal traffic speed to dynamically and automatically measure cracks (including transverse and longitudinal cracks) and estimate crack depth with statistically good accuracy. The system will be actually used in real application.

To fully develop the system, the proposed research will have three phases. The first phase will focus on designing the system and putting together all necessary parts including a vehicle, sensors and transducers, an industrial type computer. The system to be developed through this phase can actually measure cracks, but cannot estimate crack depth. In the second phase, a much larger scale field data collection activity will be performed for the model development. In this phase, longitudinal and transverse cracks will be measured to obtain crack depth data. Based on the data, a better neural network model will be developed. In the third phase, the system developed in phases one and two will be tested in fields and modifications to the system will be made based on field tests.

REFERENCES

1. Washer, G. A., Developments for the Non-Destructive Evaluation of Highway Bridges in the USA. NDT & E International, Vol. 4, Aug. 1998, pp. 245-249.

2. David R. Luhr., A Proposed Methodology to Quantify and Verify Automated Crack Survey Measurements. Paper Submitted for Presentation and Publication at the 1999 Annual Transportation Research Board Meeting, 1999.

3. Payne, D. and Walker, R. S. The Use of Lasers for pavement Crack Detection. Texas Sate Department of Highways and Public Transportation Research Report 1141-1. University of Texas at Arlington, Dec. 1998.

4. Liviu Bursanescu and François Blais, Automated Pavement Distress Data Collection and Analysis: a 3-D Approach. http://www.gietech.com/ie/english/.

5. Krautkramer, 1997 "Determination of Crack Depth Using Ultrasonics - An overlook" Utonline Application Workshop in May '97.

6. Lin Y. and Su, W.C., 1996, The Use of Stress Waves for Determining the Depth of Surface-Opening Cracks in Concrete Structures, Materials Journal of the American Concrete Institute, Vol. 93, No. 5.

7. Sansalone, M., and Carino, N. J., 1988a, "Detecting Honeycombing, the Depth of Surface-opening Cracks, and Ungrouted Ducts Using the Impact-Echo Method," Concrete International, April, pp. 38-46.

8. Sansalone, M., Lin, J.M., and Streett, W.B., "Determining the Depth of Surface-Opening Cracks Using Impact-generated Stress Waves and Time-of-Flight Techniques", Materials Journal of the American Concrete Institute, 1997.

9. Roger, S., Walker and Robert L. Harris. Non-contact Pavement Crack Detection System. Transportation Research Record 1311, TRB, National Research Council, Washington, D.C., 1991, pp. 149-157.

10. Udaya B. Halabe, Hung-Liang Chen, 1997 "Detectionof Sub-Surface Anomalies in Concrete Bridge Decks Using Ground Penetrating Radar" ACI Materials Journal Sep.-Oct. 1997 pp 369-408.

11. Chun Lok Lau, Tom Scullion, and Paul Chan, "Modeling of Ground-Penetrating Radar Wave Propagation in Pavement Systems" TRR 1355, pp 99-107, 1994.

12. Timo Saarenketo and Tom Scullion, Road Evaluation with Ground Penetrating Radar. Journal of Applied Geophysics 43, 119–138, 2000.

13. Eric Landis, Michael Peterson, Scott Selleck, Surendra Shah, Zongjin li, 1994 "Developments in NDE of Concrete" BIRL Industrial Research Laboratory.

14. Craig A. Roberts. A Comparative Analysis of Two Artificial Neural Networks Using Pavement Performance Prediction. Computer-Aided Civil and Infrastructure Engineering, 13 (1998) 339-348.

15. Sam Owusu-Ababio. Effect of Neural Network Topology on Flexible Pavement Cracking Prediction. Computer-Aided Civil and Infrastructure Engineering, 13 (1998) 349-355.

16. Nii O. Attoh-Okine. Analysis of Learning Rate and Momentum Term in Backpropagation Neural Network Algorithm Trained to Predict Pavement Performance. Advances in Engineering Software, 30 (1999) 291-302.

17. Kornelija Zgonc and Jan D. Achenbach. A Neural Network for Crack Sizing Trained by Finite Element Calculations. NDT&E International, Vol. 29, No. 3, pp. 147-155, 1996.

18. Jae-Youn Kang and Ji-Ho Song. Neural Network Applications in Determining the Fatigue Crack Opening Load. Int. J. Fatigue Vol. 20, No. 1, pp. 57-69, 1998.

19. Haykin, S., Neural Networks: A Comprehensive Foundation. Macmillan College Publishing Company, New York, 1994.

20. Terrence L. Fine, Feedforward Neural Network Methodology, Springer - Verlag, New York, Inc., ISBN 0-387-98745-2, 1999.

21. Deco, G. and Obradovic, D., An Information-Theoretic Approach to Neural Computing, NY: Springer-Verlag, 1996.

22. Bishop, C.M., Neural Networks for Pattern Recognition, Oxford: Oxford University Press, 1995.

23. Ripley, B.D., Pattern Recognition and Neural Networks, Cambridge: Cambridge University Press, 1996.

24. Judd, J.S. (1990), Neural Network Design and the Complexity of Learning, Cambridge, MA: The MIT Press.

25. Oppenheim, A. V. and Schafer, R. W. Discrete Time Signal Processing. Prentice Hall Inc. Englewood Cliffs, 1989.

26. Geman, S., Bienenstock, E. and Doursat, R., "Neural Networks and the Bias/Variance Dilemma", Neural Computation, 4, 1-58, 1992.

27. Smith, M., Neural Networks for Statistical Modeling, Boston: International Thomson Computer Press, ISBN 1-850-32842-0, 1996.

28. Masters, T., Practical Neural Network Recipes in C++, San Diego: Academic Press, 1994.

29. Saarinen, S., Bramley, R., and Cybenko, G., Ill-conditioning in neural network training problems. Siam J. of Scientific Computing, 14, 693-714, 1993.

30. More, J.J., The Levenberg-Marquardt Algorithm: Implementation and Theory. Watson, G.A., ed., Numerical Analysis, Lecture Notes in Mathematics 630, Springer-Verlag, Heidelberg, 105-116, 1977.

31. Bertsekas, D. P. and Tsitsiklis, J. N., Neuro-Dynamic Programming, Belmont, MA: Athena Scientific, ISBN 1-886529-10-8, 1996.

32. Weiss, S.M. & Kulikowski, C.A., Computer Systems That Learn, Morgan Kaufmann, 1991.

33. Boger, Z., and Guterman, H., "Knowledge Extraction from Artificial Neural Network Models," IEEE Systems, Man, and Cybernetics Conference, Orlando, FL, 1997.

34. Nelson, M.C. and Illingworth, W.T., A Practical Guide to Neural Nets, Reading, MA: Addison-Wesley, 1991.

35. Sarle, W.S., "Stopped Training and Other Remedies for Overfitting," Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics, 352-360, 1995.