DEVELOPMENT OF A HIGH RESOLUTION SENSING SYSTEM FOR AUTOMATED CRACK SEALING MACHINERY

Interim Report of

SHRP H-107A

Debra A. Krulewich

and

Steven A. Velinsky

Department of Mechanical, Aeronautical & Materials Engineering

University of California, Davis

July 31, 1992

Strategic Highway Research Program

National Research Council

Washington, D.C.

ABSTRACT

Current methods for highway maintenance are both labor and cost intensive. In particular, the sealing of cracks costs the state of California over $10 million annually. A crack sealing team consists of approximately eight individuals that seal one to two lane miles per day. This translates to approximately $1800 per mile. Furthermore, the highway maintenance workers are exposed to the dangers of moving traffic. Meanwhile, the traffic becomes congested since lane closing is necessary during crack sealing. In addition to these problems, the current maintenance procedures have resulted in a lack of efficiency and reliability.

Currently, research is underway at the Mechanical Engineering Department of the University of California, Davis to design and develop a crack sealing machine that will sense, prepare and seal cracks in pavement. By developing this machine, the goals are to minimize the exposure of workers to the dangers of traffic, considerably increase the speed of operation, and improve the quality and consistency of the resultant seal. Improving the quality of the seal will result in extended time between major road rehabilitation. Furthermore, increasing the operation speed will reduce highway congestion due to lane closure.

The purpose of this report is to develop the Local Sensing System that will detect cracks in pavement surfaces for the automated crack sealing machine. The Local Sensing System will locate crack position and measure crack width to an accuracy such that the crack preparation, sealant application and shaping of the seal can be performed through machine automation. A literature search was performed, which led to the development of the Local Sensing System. Recommendations and conclusions were made based on extensive testing of system performance. Based on the tests which were performed, it was concluded that the Local Sensing System met all requirements which are necessary to achieve a proper seal.

EXECUTIVE SUMMARY

Current methods for highway maintenance are both labor and cost intensive. In particular, the sealing of cracks costs the state of California over $10 million annually. A crack sealing team consists of approximately eight individuals that seal one to two lane miles per day. This translates to approximately $1800 per mile. Furthermore, the highway maintenance workers are exposed to the dangers of moving traffic. Meanwhile, the traffic becomes congested since lane closing is necessary during crack sealing. In addition to these problems, the current maintenance procedures have resulted in a lack of efficiency and reliability.

Pavement cracking occurs due to load, environment, pavement material problems, and problems with the existing subgrade. As the pavement is cyclically stressed due to heating and cooling and varying loads, the pavement undergoes tensile stresses which eventually result in failure in the form of cracking. Usually these cracks occur perpendicular to the road direction across the road width and in the center of the road in the direction of vehicle flow. Untreated, the cracks become larger and eventually moisture enters the subbase of the pavement. As traffic flows over these cracks, the water is forced out of the crack under the vehicle loads along with subbase material. With the loss of this subbase material, additional movement in the pavement near the crack is allowed which induces more cracking parallel to the original crack. Eventually a pothole will be created in the road surface which will require major repair and possibly rehabilitation. However, by identifying thermal cracks and sealing them at an early stage, it is possible to prevent major damage.

Currently, the crack sealing process is composed of first preparing the crack. This encompasses removing both loose material and moisture from within the crack. Furthermore, many states route the crack to achieve a properly dimensioned reservoir that optimally accepts the sealant. These procedures ensure a proper seal with a long life.

Following crack preparation, research has indicated that an optimal seal is applied when the road surface has been heated. Currently, heating of the road surface is accomplished using a hot air lance. However, some cooling occurs between heating and actual sealing. Through the use of an automated machine, it is possible to apply sealant without significant delay between heating and sealing, thereby producing a more reliable seal than that which is manually achieved with current methods. After the crack has been appropriately prepared and heated, the crack is then sealed.

Currently, research is underway at the Mechanical Engineering Department of the University of California, Davis to design and develop a crack sealing machine that will sense, prepare and seal cracks in pavement. By developing this machine, the goals are to minimize the exposure of workers to the dangers of traffic, considerably increase the speed of operation, and improve the quality and consistency of the resultant seal. Improving the quality of the seal will result in extended time between major road rehabilitation. Furthermore, increasing the operation speed will reduce highway congestion due to lane closure.

The purpose of this report is to develop the Local Sensing System that will detect cracks in pavement surfaces for the automated crack sealing machine. The Local Sensing System will locate crack position and measure crack width to an accuracy such that the crack preparation, sealant application and shaping of the seal can be performed through machine automation.

Initially, a literature review was performed to investigate existing methods of detecting cracks in pavement surfaces. Furthermore, the comparable task of seam tracking during automated welding was reviewed. The purpose of this literature review was to identify existing technologies which may be adapted to crack sensing during automated crack sealing.

Next, the sensing system was developed. Once the requirements of the sensing system were determined, a variety of sensing system technologies were investigated.

Based on the characteristics of the technologies considered, a sensor using the principle of structured light to measure depth was chosen. An initial technology feasibility test was performed which adequately demonstrated the ability of the technology to locate cracks in pavement.

Once the sensor technology was identified, a commercially available sensor was selected for prototype development of the crack sealing machine. Principles of operation have been described in detail. Next, the sensor performance was experimentally verified. Tests were performed which proved the sensing system reliability under a variety of environmental conditions.

Finally, conclusions and recommendations were made. Based on the tests which were performed, it was concluded that the Local Sensing System met all requirements which are necessary to achieve a proper seal. Recommendations for alterations in both the system hardware and software were suggested to optimize performance on the final crack sealing machine.

TABLE OF CONTENTS

List of Tables

List of Figures

CHAPTER 1 - INTRODUCTION

Current methods for highway maintenance are both labor and cost intensive. In particular, the sealing of cracks costs the state of California over $10 million annually. A crack sealing team consists of approximately eight individuals that seal one to two lane miles per day. This translates to approximately $1800 per mile. The highway maintenance workers are exposed to the dangers of moving traffic, and traffic becomes congested since lane closing is necessary. In addition to these problems, the current maintenance procedures have resulted in a lack of efficiency and reliability (Ravani and West, 1991; Jing et al., 1990; Skibniewski et al., 1990).

Pavement cracking occurs due to load, environment, pavement material problems, and problems with the existing subgrade (Jing et al. 1990; Peshkin et al., 1991). As the pavement is cyclically stressed due to heating and cooling and varying loads, the pavement undergoes tensile stresses which eventually result in failure in the form of cracking. Usually these cracks occur perpendicular to the road direction across the road width and in the center of the road in the direction of vehicle flow. Untreated, the cracks become larger and eventually moisture enters the subbase of the pavement. As traffic flows over these cracks, the water is forced out of the crack under the vehicle loads along with subbase material. With the loss of this subbase material, additional movement in the pavement near the crack is allowed which induces more cracking parallel to the original crack. Eventually a pothole will be created in the road surface which will require major repair and possibly rehabilitation. However, by identifying thermal cracks and sealing them at an early stage, it is possible to prevent major damage.

Currently, the crack sealing process is composed of first preparing the crack. This encompasses removing both loose material and moisture from within the crack. Furthermore, many states route the crack to achieve a properly dimensioned reservoir that optimally accepts the sealant. These procedures ensure a proper seal with a long life.

Following crack preparation, research has indicated that an optimal seal is applied when the road surface has been heated. Currently, heating of the road surface is accomplished using a hot air lance. However, some cooling occurs between heating and actual sealing. Through the use of an automated machine, it is possible to apply sealant without significant delay between heating and sealing, thereby producing a more reliable seal than that which is manually achieved with current methods. After the crack has been appropriately prepared and heated, the crack is then sealed.

The work of this thesis is part of a project to design and develop a crack sealing machine that will sense, prepare and seal cracks in pavement. By developing this machine, the goals are to:

- Minimize the exposure of workers to the dangers of traffic.
- Considerably increase the speed of operation.
- Improve the quality and consistency of the resultant seal.

Improving the quality of the seal will result in extended time between major road rehabilitation. Furthermore, increasing the operation speed will reduce highway congestion due to lane closure.

Design objectives of the automated crack sealing machine are to:

- Sense the location and size of cracks in both asphalt concrete (AC) and portland cement concrete (PCC) pavement which are between 3.175 to 25.4 mm in width.
- Prepare the crack for sealing by moisture and debris removal, routing, and preheating the road surface to ensure optimal seal.
- Dispense the sealant.
- Form the seal into its optimal configuration.
- Use as many commercially available components as possible.
- Ultimately perform the crack sealing operation at approximately 3.22 km/h (about ten times faster than the current rate).

Two component systems will be utilized to address the spectrum of commonly occurring cracks to be sealed. The first of which is a longitudinal crack sealing machine. The purpose of the longitudinal machine is to seal construction joints at the edge of the road. The second machine is a general crack sealing machine. The general machine will seal random cracks in the road surface.

The automated crack sealing machine architecture includes the following major sub-systems:

- Vision Sensing System (VSS)
- Local Sensing System (LSS)
- Applicator and Peripherals System (APS)
  - Heating/Cleaning/Debris Removal Subsystem
  - Router
  - Sealant Applicator
- Robot Positioning System (RPS)
  - General Machine Positioning System
  - Longitudinal Machine Positioning System
- Vehicle Orientation and Control (VOC)
- Integration and Control Unit (ICU)

The purpose of the VSS in conjunction with the LSS is to locate and describe pavement cracks. The APS includes the Heating/Cleaning/Debris Removal Subsystem, the Router, and the Sealant Applicator. The Heating/Cleaning/Debris Removal Subsystem will include all hardware and software necessary to heat and clean the road surface during crack preparation. The router will shape the crack to optimal sealing geometry. The Sealant Applicator is responsible for sealant dispensing and seal configuration. The Robot Positioning System is responsible for moving the applicator assembly along the crack during sealing. On the general machine, the General Machine Positioning System will consist of an intelligent robot arm and controller. On the longitudinal machine, the

Longitudinal Machine Positioning System will position the applicator assembly using a hydraulic controller. The Vehicle Orientation and Control System will monitor changes in vehicle position from when the crack is sensed to when it is actually sealed. Finally, the Integration and Control Unit will oversee the entire crack sealing procedure, by monitoring all peripherals to ensure proper operation and controlling communication between sub-systems.

The purpose of this thesis is to develop the Local Sensing System that will detect cracks in pavement surfaces for the general and longitudinal automated crack sealing machine. The purpose of the sensing system is to locate crack position and to measure crack width to an accuracy such that the crack preparation, sealant application and shaping of the seal can be performed through machine automation. No current machine vision system has been developed which meets the requirements of this project. Therefore, the sensing system will be unique. However, similar work has been researched and developed to sense cracks on a macroscopic level for highway assessments. Furthermore, the similar process of seam tracking during automated welding has been researched and technologies have been adapted to crack sensing tasks. A literature search on these methods has been performed and is reported in the following sections. Through this literature search, a proposed sensing system has been identified.

On the general crack sealing machine, the LSS will work in conjunction with the VSS to confirm the presence of a crack within a given area. The VSS will locate the approximate position of a possible crack using a video camera. The camera uses a line scan charged coupled device (CCD) as its sensing element. As the vehicle moves, lines across the lane width will be gathered to form an area view of the road surface. Through analyzing the patterns of gray levels which the camera senses, it is possible to determine the position of possible cracks. However, since the line scan camera only has two-dimensional measuring capabilities, it may mistake an oil spot, shadow, or previously filled crack for an actual crack. The purpose of the LSS on the general machine is to scan

the area around the potential crack location identified by the VSS and confirm or reject the presence of a crack. Furthermore, there are inherent inaccuracies in the VSS crack identification algorithm which gives it a resolution of approximately +/-25.4 mm. There are also errors associated with the VOC that will result in errors in the crack location identified by the VSS. Therefore, the local sensing will also provide more precise position information to the General Machine Positioning System. Local sensing will provide range information that can accurately sense the presence and position of a crack. However, local sensing alone would not be adequate because the local sensor requires a planned path to scan for random cracks. Given the operating speed of the vehicle, the update rate and field of view of the local sensor are not adequate to track random cracks without *a priori* knowledge of crack direction.

On the longitudinal crack sealing machine, the local sensor will provide all sensing information to the positioning system. Because the longitudinal cracks do not randomly vary in direction, it is possible to design a sensing system in which the LSS provides an error feedback signal to the Longitudinal Machine Positioning System. The start of the crack must initially be placed within the local sensor field of view, and then through real time control and feedback provided by the LSS, it will be possible for the positioning system to follow the longitudinal crack.

1.1 - Literature Review

A literature review was initially performed to investigate existing methods of detecting cracks in pavement surfaces. Furthermore, the comparable task of seam tracking during automated welding has been reviewed. The purpose of this literature review is to identify existing technologies which may be adapted to crack sensing during automated crack sealing.

1.1.1 - Pavement Distress Survey Systems

In order to maintain both highways and runways, it has become necessary to non-destructively evaluate pavement conditions. Pavement texture, roughness, and deflection measurements are important in evaluating the condition of the pavement for maintenance. Current methods of measurement are relatively slow and require lane closures. Due to the high cost of lane closures and traffic congestion caused by these lane closures, much work is currently underway to develop a rapid non-destructive test to measure pavement profiles. Unlike the Local Sensing System on the crack sealing machine, the purpose of this work is to locate the gross position of cracks in pavement; although, it may be possible to adapt these measurement techniques to local sensing on the crack sealing machine.

There is a definite need for a cost effective method for an automated pavement survey system. One method to extract pavement-surface-distress data is through video images. Through digital image processing algorithms, pertinent surface characteristics are identified (Ritchie, 1990; Wigan, 1992; Li et al., 1991). With the use of proper lighting, the presence of cracks can be determined by observing the shadows. A video system captures an image of the pavement. This analog image is then digitized by the use of a frame grabber. Next, the image is digitally filtered to remove unwanted noise. The cleaned up data is then processed by performing binary thresholding algorithms. A histogram is performed on the processed data, and the presence of cracks are determined by noting the variance of gray levels.

Two-dimensional data collected by vision systems has proven to be a reliable method of surveying pavement conditions on a macroscopic level. However, these techniques alone are not adequate to locate cracks in pavement. The third dimension contained in range data is necessary to reliably distinguish between oil spots, previously filled cracks, tire tracks, shadows, and actual cracks. Furthermore, the resolution and accuracy of these

camera systems are not adequate to locate the exact crack position relative to global coordinates.

Using stereo vision, it is possible for two cameras to measure three dimensions (Bruyelle and Postaire, 1991). One such application recognizing obstacles on the road in real time using dynamic stereo vision. Much like the human vision system, it is possible to reconstruct depth information by analyzing a pair of images acquired simultaneously at different angles. However, this method results in slow, complex algorithms which are not suitable for real time operation on small computers.

Another method used to survey pavement conditions measures range information (Elton and Harr, 1988). The pavement profile is measured using a laser range finding sensor. This sensor projects a small spot of laser light onto the road surface and the distance from the sensor to the road surface is measured. Four sensors are to be mounted on a moving vehicle and rapid measurements are made. This method was evaluated and thoroughly tested. Results showed a high correlation between measurements made with the laser sensors and the surveyed profile. One problem noted in this research was the sensitivity of sensor alignment. An accurate alignment procedure was necessary in order to collect good data. Furthermore, very slight misalignment caused data to quickly deviate from the surveyed profile. Even temperature changes effected the alignment of the four sensors.

Another automatic pavement distress survey system has been developed and is currently being applied to concrete bridges and tunnels for maintenance surveying (Fukuhara et al., 1990). This system measures without contact the cracking, rutting, and longitudinal profile of highway surfaces at speeds up to 60 km/h. The crack survey system scans a laser light in the lateral direction across the road surface. Then, a photomultiplier tube (PMT) mounted on the bumper detects the scattered light at an angle to reflected direction. When cracks are present, the amount of light detected by the PMT is reduced and the output from the PMT gives information about the presence of cracks.

A video camera is also mounted on the front bumper. If rutting occurs, the camera will observe a curved line projected from the scanned laser.

To measure the longitudinal profile, two laser range finding sensors are mounted in front and in back of the front wheel of the vehicle. Distance measurements from the sensor to the road surface are rapidly acquired to determine the longitudinal profile.

Currently, the Swedish Laser Road Surface Tester is actually being used for automated surface distress detection (Ritchie, 1990). This sensing system consists of eleven laser range finding sensors that can detect transverse cracks which cross their vision range. However, longitudinal cracks are not detected. Crack size, roughness, and transverse profiles are measured at full highway speed. The sensing system provides information about the incidence of cracks in pavement. No attempt is made to measure or locate individual cracks.

Recognizing the fact that data collected by a video camera alone is not adequate to locate cracks reliably, range information can be measured and combined with video data to determine crack location (Hendrickson et al. 1990, 1991). The video camera recognizes potential cracks. Objects which are not confirmed by the range sensor are considered to be previously filled cracks, oil spots, shadows, or other spatial noise. Range information is gathered by an infra-red laser range sensor which is swept across the pavement surface. This method of collecting range information without the vision system information is not adequate to locate the presence of cracks because data collection is too slow. Furthermore, an array of multiple range sensors would be too expensive. This proposed system has been laboratory tested on a stationary platform.

1.1.2 - Automated Welding Techniques

Because no research pertaining to the range finding task of the local sensing with the required resolution has been completed, the comparable task of seam tracking during automated welding was researched. During automated welding, the seam between two

adjoining pieces of metal is automatically detected by a sensor, and position information is provided to a controller which moves the welding torch. Detecting a seam is similar to detecting a crack, so it may be possible to adapt sensing technologies from automated welding to crack sealing. Much research using photoelectric, ultrasonic, and laser sensors for seam tracking during automated welding has been conducted.

Modulated infrared sensors have been used to determine joint location (Tan and Lucas, 1986). The infrared sensor was modulated and then swept across the seam. Range information was found by measuring the intensity of the reflected beam. By modulating the beam, it was possible to remove unwanted noise from the signal. This device was able to detect a seam in approximately 100 ms.

An ultrasonic sensor has also been used to measure range information (Tan and Lucas, 1986). The sensor projected an ultrasonic wave onto the surface to be measured, and the reflected beam was measured by a receiver. Range information was determined through the comparison of phase shift between the receiver and transmitter. Again, the sensor was swept across the seam using a DC motor. Results showed a longer time response than the infrared sensor of 500 ms, yet produced a resolution of 0.1 mm and was less sensitive to variances in the measured surface.

Difficulties in ultrasonic sensing such as air movement, temperature fluctuations and noise affect sensor performance (Siores et al., 1987). However, it was found that the added benefits of the wide frequency range and relatively slow velocities compensated for these problems. Furthermore, a continuously modulated frequency wave can be transmitted rather than the more conventional single frequency. Range was determined by measuring the frequency difference between transmitted and received signals rather than phase shift. This method is known as the echo principle. This provided a method of determining range without calibration. Secondly, the sensor became insensitive to varying surface conditions. Accuracies as high as 0.3 mm and time responses of 1 ms were achieved.

There are advantages associated with acoustic sensing versus computer vision using a structured light source and triangulation (Estochen et al., 1984). A problem with the computer vision approach is the physical constraints of the light source and vision system to the movement of the torch. Furthermore, acoustic sensing does not require the computational support and thereby delay that the vision system would require.

To evaluate the cross section of the seam, two approaches were considered. First, a vector of sensors spanning the entire cross section was considered. This method allows for the entire cross section to be scanned at the same time. Alternatively, the second approach considered a single sensor which swept the cross section and sequentially processed the data. The second approach was chosen due to its flexibility and ability to sample many surface features. The sweeping motion was accomplished by rotating the sensor about its midpoint in the direction of the scan. With this method, the range information heavily relied on the ability to accurately sense the angular position. In conclusion, it was noted that the system can operate in real time only if the seam does not significantly deviate from its path during scanning.

Another method of seam tracking is computer vision using structured light (Estochen et al., 1984). Needing to accurately measure the transit time of the beam during acoustic sensing is a disadvantage. Also, optical ranging requires more time to construct a 3-dimensional image since many points must be sampled to create the image. An alternative approach of structured light has neither of these drawbacks. This approach directs a thin line of laser light on a surface. The contour of the object is sensed with a camera mounted at a 45^{o} angle from the laser line, and a 3-dimensional image is constructed using the principle of triangulation. This image is constructed 200 times faster than other methods, at a rate of 240 points every 17 ms. The system can be configured to produce accuracies as small as 0.025 mm. The sensor itself is miniature, measuring 15.5 x 17.0 x 6.35 cm and weighing 1.589 kilograms, making it ideal to place on a robotic arm for guidance. This camera is currently being used by an American automobile manufacturer for seam

tracking. The seam can be tracked at 1.02 m/sec. It is also being used for inspection purposes at a pencil manufacturer.

Another method of using laser light to measure a surface transmits a single point of laser light onto a surface rather than a line (Brunet et al., 1986). Again, using triangulation, the 3-dimensional profile of the surface can be generated by scanning the single point laser range finder. One such system which has been researched and developed uses a small He-Ne laser and a linear CCD array. It was found that in order to obtain high accuracy, there must be a large angle between the laser and the CCD array, creating a larger unit. This problem was remedied by using a folding mirror. In this application, the sensor is mounted on the robot, and the robot scans the seam. Accuracies to 0.2 mm were achieved, and 50 cm/minute welding velocity was obtained... A second method uses a GaAlAs laser diode and a PIN-diode linear sensor chip rather than a CCD (Bamba et al., 1984). There are advantages of using the PIN-diode sensor chip rather than a CCD chip. The PIN-diode requires simpler image processing and electronic circuitry, and is less sensitive to external optical noise. It is believed that using the PIN-diode is cost effective and suitable for the welding environment.

In order to scan the cross section, the unit was rotated around the welding torch mechanically, unlike the previous sensor which was swept linearly by the robot arm. The points were then collected sequentially to create the 3-dimensional image. The laser range detecting approach determines distance regardless of the intensity of the beam, being dependent on the current flow through the PIN-diode. Furthermore, the laser beam is pulse-modulated to extract unwanted noise. During acquisition, the first step smoothes the data. The next step uses a line fitting procedure using least squares to determine the local gradient. From this local gradient, the edges of the scanned surface are located. This unit is compact, measuring 70 mm diameter cylinder. Scanning rates of 2 Hz and a resolution of 0.2 mm were achieved.

Rather than a single laser light source as the preceding two examples, a linear emitter array can be used as a sensing device (Ogilvie and Zemancheff, 1983). By using an array, there is no longer a need to scan the cross section. The light from each emitter is labeled differently through modulation so that it may be distinguished, requiring a separate computer to manage and process detector outputs.

A fifth use of the laser range finding sensor projects a narrow modulated laser beam onto the surface to be measured, thus allowing unwanted noise to be distinguished (Smati et al., 1983). A scanning motion is then accomplished using a stepper motor.

The first step in acquiring the data uses a simple weighted recursive filter of first order to smooth the data, thereby increasing the signal to noise ratio. Other smoothing techniques have been developed, but this method was chosen for its simplicity and ready implementation. Furthermore, it produced a good fit, with an error less than 5%.

The second step in acquiring data was edge detection. Two methods were considered. The first method calculates the rate of change of slope of the collected points, and an edge is defined to be a section where the change in slope has increased. This method is simple, requiring only addition and subtraction. The second method involves fitting the data into a straight line, and an edge is defined as the intersections of these lines. This method is more complicated to implement, requiring multiplication and division, thereby requiring more processing time. The first method was decided to provide adequate accuracy.

Scan rates of 2 Hz were achieved. Further work is being done to decrease the size of the unit and increase the scan rate. It is possible to achieve scan rates of 10 Hz using mirrors, however, the field of view becomes very limited and the sensor must be mounted very close to the surface.

Another seam sensor uses a laser light source and a photosensitive device as a receiver to determine range information via triangulation (Nayak et al., 1987). The data acquisition consists of the similar steps of collection, filtering erroneous data, processing to

determine the root of the seam, and coordinate transformations. This article does not detail how the scanning is performed. However, welding speeds of 152.4 cm per minute were achieved.

A laser range finding sensor using mirrors to create the scan has been developed and is commercially available (Appels, 1987). Again, a laser light source is transmitted and then received by a photosensitive device. Three-dimensional geometry is then determined by triangulation. This system is currently being used and achieves speeds close to 3 m/min.

Another scanning laser range finder using mirrors has been developed (Edling, 1986). This sensor achieves a resolution of 0.02 mm and a typical accuracy of +/- 0.4 mm. Tracking speed is as high as 20 mm/s.

There is an inherent problem in using mirrors to create a scanning laser range finder (Rioux, 1984). The problem is associated with the fact that a large angular separation is necessary for good resolution and field of view. However, this large separation not only creates a larger sensing unit, but creates shadow effects, thereby making certain points undetectable by the sensor. A solution to this problem is to synchronize the light spot projection with its detection. This is accomplished by using two mirror scanners rather than one mirror. The first mirror, as in the previously mentioned examples, causes the beam to scan the cross section. The second mirror is used so that the receiving sensor can follow the spot. In doing this, a more compact sensor can be achieved. Furthermore, by using synchronization, an increased resolution can be achieved without reducing the field of view.

1.2 - Problem Statement and Objectives

There has been some previous work in distress survey systems where the overall condition of the road on a macroscopic scale has been sensed. However, this work is inadequate for purposes of the crack sealing machine due to the fact that the cracks must be located with a high spatial accuracy in order to achieve a proper seal. Thus, the objective of

this thesis is to select and develop a sensing system which can accurately and precisely locate cracks in pavement. This thesis contains detailed specifications and principles of operation along with experimental verification of the sensor performance.

CHAPTER 2 - SENSING SYSTEM

The previous chapter has demonstrated the need for an automated crack sealing machine. Furthermore, to develop such a machine, a Local Sensing System (LSS) is necessary to detect cracks in the pavement. There is no existing system which will fulfill all the requirements of the LSS. Therefore, a sensing technology which can accurately and precisely locate cracks in pavement must be selected and developed. The following chapter first develops the detailed requirements of the LSS. With these requirements in mind, different sensing technologies are investigated and considered for the task of local sensing. Based on the abilities of the considered sensors, a sensing technology is selected. An initial feasibility test is performed on the selected sensing technology. After the test has demonstrated that the sensing technology can detect cracks on pavement, detailed specifications of the sensing system are developed and the principles of operation are described.

2.1 - Requirements of the Sensing System

The objective of the crack sealing machine sensing system is to locate crack position and to measure crack width to an accuracy such that the crack preparation, sealant application and shaping of the seal can be performed by machine automation. The sensing system must identify cracks on both asphalt concrete (AC) and portland cement concrete (PCC) from 3.175 to 25.4 mm wide on a vehicle continuously traveling at a rate of 3.22 km/h.

As described in Chapter 1, on the general crack sealing machine, the LSS will work in conjunction with the VSS to confirm the presence of a crack within a given area. The VSS will locate the approximate position of a possible crack using a video camera. The purpose of the LSS on the general machine is to scan the area around the potential crack location identified by the VSS and confirm or reject the presence the a crack. Furthermore,

the LSS will also provide more precise position information to the General Machine Positioning System due to inherent inaccuracies in the VSS and Vehicle Orientation and Control System (VOC). In order for the local sensor to work in conjunction with the VSS, it is of primary importance that the sensor can distinguish actual cracks from oil spots, previously sealed cracks, shadows, etc. It must also provide adequate position information to compensate for the inaccuracies due to the VSS and VOC.

On the longitudinal crack sealing machine, the local sensor will provide all sensing information to the Longitudinal Machine Positioning System. Through real time control and feedback provided by the local sensor, it will be possible for the positioning system to follow the longitudinal crack.

The inherent resolution of the VSS algorithm is approximately +/- 25.4 mm. On the global crack sealing machine, the local sensor must view at least 50.8 mm of pavement, centered around the point which the VSS has identified to contain a crack. Further inaccuracies are added due to the VOC and RPS. Errors in the VOC and RPS are negligible in comparison to the errors in the VSS. Therefore, a 101.6 mm field of view for the local sensor should provide adequate viewing area to observe the crack identified by the VSS. The 101.6 mm field of view is also adequate for the longitudinal crack sealing machine.

The sensor will be mounted on the crack sealing vehicle such that it will be exposed to varying environmental and physical conditions. Therefore, the sensor must provide reliable results when exposed to dust and road debris. The sensor must also be physically strong enough to endure impacts from 9.5 mm aggregate and roadway debris not exceeding 0.25 kg. Furthermore, because the sensor will be operating on a moving vehicle, it must endure typical vehicle vibration and shock. Specifically, the sensor must withstand 3 g's peak vibration from 15 Hz to 100 Hz during operation.

To protect the sensor from road debris and aggregate, it is desired to mount the sensor at least 101.6 mm from the road surface. Therefore, the sensor must provide

reliable results when placed at least 101.6 mm vertically from the surface being measured.

Because the crack sealing machine must operate in as low as 0°C temperatures, the sensor performance should not be affected by temperature variations. It must provide reliable results when consistently exposed to temperatures as low as 0°C and as high as 71°C during operation. Furthermore, due to the fact that the pavement may be heated, temperature variations in the sensor path will likely exist and must not affect reliable results from the sensor.

The local sensor must also produce reliable results when exposed to varying lighting conditions and wind. The sensor must operate when exposed to direct sunlight and when exposed to shade. Windy conditions during sensor operation must not affect reliable results.

The sensing system must respond fast enough to track cracks on a vehicle moving 3.22 km/h. Typical random cracks in pavement do not vary by more than 45°. Without *a priori* information about the crack direction, in order for the sensor not to lose view of the crack, a cycle time of 18 Hz (0.056 ms between samples) must occur. Through the literature review, some sensors do exist with adequate response time. However, the cycle time requirement of 18 Hz includes both the response time of the sensor in addition to the processing time associated with crack identification.

On the longitudinal crack sealing machine, no pre-determined path plan exists since the LSS is the only sensing system. However, longitudinal crack directions do not significantly vary, so a 18 Hz update rate of the system will provide adequate information to the RPS.

Table 2.1 summarizes the sensor requirements. The sensor must identify cracks on both AC and PCC, varying in size from 3.175 to 25.4 mm in width. To detect cracks as small as 3.175 mm, according to the Nyquist sampling theorem, the sensor must have the ability to resolve a crack as small as 1.588 mm wide; so the resolution between points along the pavement surface must be 1.588 mm minimum. It must accurately sense range

Table 2.1 - Sensor Requirements

RESOLUTION ALONG SCAN	1.588 mm
VERTICAL RESOLUTION	1.588 mm
ACCURACY OF CRACK POSITION	3.175 mm
FIELD OF VIEW	101.6 mm
DISTANCE TO SURFACE	101.6 mm minimum
SYSTEM RESPONSE FREQUENCY	18 Hz
HUMIDITY	0 to 85%
VIBRATION	3 g peak from 15 Hz to 100 Hz
SHOCK	10 g
OPERATING TEMPERATURE	0.0 to 71.1° C
SERVICE LIFE	10 years
SENSOR MUST ENDURE	• wind and sunlight • dusty environment • surface color variations • moisture on pavement • debris in cracks • road surface height variations • temperature variations • electromagnetic interference
SENSOR MUST DISTINGUISH BETWEEN	• previously filled cracks • oil spots • shadows • actual cracks

data on both AC and PCC surfaces, which may vary in color from black to white and may contain reflective aggregate. Normal height variations in AC and PCC occur, and are approximately +/-1.588 mm. Height deviations greater than this amount must be recognized as a potential crack. Therefore, the sensor must have the ability to resolve vertical height measurements to 1.588 mm. Furthermore, the LSS must have the ability to locate crack position to within 3.175 mm accuracy.

2.2 - Sensing System Technologies

A variety of sensors technologies have been researched in order to select a sensing system which best meets the requirements established in Section 2.1. The Local Sensing System which is selected should be the most cost effective, off-the-shelf component which meets all the requirements. With this in mind, sensors which were considered for the task of local sensing are described in this section. At the end of this section, a sensing technology will be selected.

An extensive literature search has been conducted to gather background information in crack detection and tracking. Initial research considered all of the following sensors noted in Table 2.2 (Jing et al., 1990).

Sensing technologies were divided into two categories: tactile and non-tactile. Tactile sensors recognize a crack through direct physical contact with the pavement surface. Two types of tactile sensors were considered. The microswitch is a discrete device which senses either a switch closed or switch opened condition. The switch is composed of a rolling element which rides along a surface. This technology could be adapted to crack sensing by allowing the sensor to roll over pavement sections, where the switch is normally closed. When the switch rolls over a crack, a spring action will cause the switch to open, indicating a crack. To achieve the required resolution of 1.588 mm over a 10.16 cm field of view, 64 microswitches would be needed. Microswitches are readily available and very inexpensive.

Table 2.2 - Sensor Technologies

TACTILE	microswitches time domain reflectrometry
NON-TACTILE	spectral analysis capacitive inductive pneumatic far infrared temperature visible array acoustic ultrasonic optical • modulated light • light intensity • displacement

Time Domain Reflectrometry (TDR) measures the dielectric constant of a material between electrodes. Currently, Caltrans utilizes TDR to detect moisture under the pavement. It may be possible to detect the presence of a crack by separating two electrodes and riding the two electrodes along the pavement. When a crack is located between the two electrodes, both air and pavement would exist between the electrodes, thereby changing the dielectric constant. Each TDR electrical hardware and electrode pair costs approximately $7000. Again, 64 separate TDR sensors would be required to achieve the required resolution and field of view. This would total approximately $450,000, which is unreasonable for the task. Furthermore, this technology has never been applied when the electrodes ride along the surface being measured.

Tactile sensing has the further disadvantage of wear due to the physical contact. It is not believed that tactile sensing would meet the 10 year service life. Both types of tactile sensing were therefore eliminated as a possible sensor due to the high amounts of wear.

There is no physical contact between the sensing system and the pavement with non-tactile sensors. Non-tactile sensors have the ability to detect the presence of a crack remotely, and therefore have no inherent wear problems which would effect the service life. Because the longer service life, a non-tactile sensor is preferred for the task of local sensing.

Because pavement surfaces oxidize, it may be possible to detect the presence of a crack by observing the spectral analysis. Oxidized pavement exhibits less emissions than non-oxidized pavement. As a new crack is exposed to sunlight, it is thought that the oxidation rate of the crack would be greater than the surrounding pavement. This technology could be adapted to the task of detecting cracks by utilizing a discrete "spectrascope" which would indicate the level of oxidation of the surface being measured. Again, 64 separate sensors would be required to achieve a 1.588 mm resolution and a 101.6 mm field of view. No sensor of this type currently exists. Furthermore, there has been no work supporting the possibility that this technology is adaptable to sensing cracks in pavement. Therefore, spectral analysis was eliminated.

Capacitive sensors typically measure an air gap between the sensor and a metallic surface being measured. The sensor itself is one plate of the capacitor, while the surface being measured is the second plate. As the air gap varies, the resulting capacitance changes. This type of sensor is not applicable because pavement is non-conducting, and therefore would not act as the second plate of the capacitor.

An inductive proximity probe consists of a cylindrical jacket with a multi-turn coil. The probe is placed near a metallic surface. As the conductive surface moves closer to the probe, the air gap decreases and larger eddy currents are produced in the coil (Nachtigal et al., 1986). This current can be measured and processed to produce a measurement of the

air gap. Typically this transducer is utilized to measure displacements no greater than 2.5 mm; however, some commercially available units can measure up to 125 mm. This type of sensor is not applicable because pavement is non-conducting, and therefore would not create the eddy current changes.

Pneumatic sensors are very simple. A vacuum is created by surrounding a tube by an annular air flow. When an object is placed such that it alters the air flow around the tube, the resulting pressure changes in the tube. This sensing technology is typically used for sensing the presence or absence of large objects. Pneumatic sensors were initially tested (Jing et al., 1990) and produced unreliable and inconsistent results. Therefore, this sensing technology was omitted from consideration.

Far infrared temperature transducers were considered because the temperature of a crack was expected to be different from the temperature of the surrounding material (Jing et al., 1990). Infrared temperature transducers sense temperature by measuring the wavelength of the infrared which are emitted from the surface. These emitted wavelengths vary with temperature. Testing with an infrared temperature gun showed cracks in direct sunlight were at a higher temperature than the surrounding pavement, while cracks in the shade or with water intrusion were cooler than the surrounding pavement. Satisfactory results were achieved when the pavement was heated because the surface of the pavement cooled at a faster rate than the crack. The temperature gun was very slow, so an attempt was made to use infrared sensing elements with amplifiers. Response times of 3 Hz were achieved, which does not meet the requirements established in Section 2.1. Furthermore, it is not believed that a resolution of 1.588 mm could be achieved with this type of sensor. Therefore, far infrared temperature sensors were eliminated as a possible local sensor.

Visible array detectors are devices which measure visible light reflected from the surface. Particularly, the charge coupled device (CCD) was considered because it is lightweight, requires little power, and is very rugged. A CCD is fabricated on a single integrated circuit. As an image is produced on its surface, the charge at each pixel varies,

and the voltage values are serially scanned off (Jenkins, 1987). A CCD may be used to detect cracks by converting the data to gray levels (black to white levels of intensity) and then looking for darker regions, which indicate the presence of a crack. This method will not have the ability to distinguish between oil spots, shadows, previously sealed cracks, and actual cracks and therefore does not meet the requirements established in Section 2.1.

Acoustic sensing works by transmitting a sound wave onto a surface being measured, and then sensing the reflected sound wave. Range information is extracted by measuring the phase shift of the reflected and transmitted sound waves. Acoustic sensing was eliminated due to its inadequate resolution and time response.

Similarly, ultrasonic sensors determine distance by measuring the time lag between the source and reflected signals. This sensor is economical, commercially available, and commonly used for seam detection in automated welding. However, no sensors on the market were found to have small enough beam diameters to provide the required resolution. Furthermore, these sensors do not produce reliable results when the air temperature through the beam path varies, which will likely be the case if the pavement has been exposed to sunlight or has been heated for proper sealing.

Optical sensing techniques are generally non-perturbing to the measured system, have a very fast response, and have very good resolution (Nachtigal, 1986). All optical sensors rely on the detection of variances in the basic properties of one or more light beams. These properties are direction of propagation, intensity, wavelength, polarization, and relative phase. These properties are influenced by optical effects such as refraction, reflection, scattering, diffraction, and interference.

Optical sensors using a modulated light source are similar to the ultrasonic and acoustic sensors. A light source is pulsed, or modulated, at a known frequency. The light is focused onto a surface. The reflected light returns to a receiver. By measuring the phase shift between the transmitted and reflected light, the time of flight can be determined, and thus the distance can be measured. The speed of light through varying temperatures of air

does not change as does the speed of ultrasonic waves. Therefore, the fact that the temperature of air may vary in the beam path will not cause a problem. The literature review performed in Section 1.2 did not support the application of this technology to sensing cracks in pavement. Furthermore, to achieve the required field of view, either 64 separate sensors are needed, or the sensor must sweep the 101.6 mm field of view. It is believed that a sweeping motion at the required rate of 18 Hz would put undo physical strain and shock on the sensor. Furthermore, the use of 64 sensors would cost beyond the budget constraints of this project.

An optical sensor based on light intensity (photoelectric sensor) determines range information by transmitting a light source onto the surface being measured. The light diffusely reflects off the surface and is sensed by a receiver. By measuring the intensity of diffusely reflected light, the distance from the sensor to the surface can be determined. A laser is often used as the light source. It is economical and commercially available. Furthermore, photoelectric sensors are widely used in the comparable task of detecting seams for automated welding as previously described. However, the sensor performance would be significantly affected if the lens were to become dusty, thereby reducing the amount of reflected light being sensed. Furthermore, this type of sensor does not perform well on surfaces which vary in color or reflectivity, such as pavement.

The optical displacement transducer measures range information using the principle of triangulation. The principle of triangulation determines distance measurements by transmitting a focused laser light source onto an object and then imaging the diffusely reflected light onto a photosensitive device (Mundy et al., 1987; Case et al., 1987; Kanade et al., 1987). The photosensitive device (PSD) is an analog light sensor that is sensitive to the intensity and position of a light spot in its field of view. Knowing the position of the image on the PSD, the distance between the detector lens and light source and the projection angle of the source, the distance measurement can be geometrically determined. This type of sensor is more expensive than the previously mentioned sensors, yet is still

commercially available. It is also widely used for seam detection during automated welding.

Sensing systems based on triangulation are impervious to color variations, so a laser range finding sensor should work well on all pavements. Furthermore, since they measure the location where light is reflected rather than the amount of light being reflected, laser sensors are less sensitive to a dusty environment. Additionally, laser triangulation is insensitive to lighting conditions because the sensor provides its own lighting via the laser. Overall, laser triangulation is a proven reliable technique for extracting three-dimensional surface characteristics.

Based on the technology study, the most suitable sensing system is a laser range finding sensor based on the principle of triangulation. In the following section, this technology will be proven to perform on both AC and PCC.

2.3 - Technology Feasibility

In order to determine whether a laser range finder based on the principle of triangulation will detect cracks on both Portland Cement Concrete (PCC) and Asphalt Concrete (AC), a laser range finder has be thoroughly tested. To achieve the scan speed necessary per the requirements established in Section 2.1, a large amount of money would need to be invested. Rather then initially testing the actual system, a less expensive system based on the same technology was selected and thoroughly tested. In this way, the technology was proven before a large amount of money was invested for the actual system which would meet all requirements. The laser range finder tested had the specifications shown in Table 2.3.

Test apparatus was built such that the range finding sensor scanned a section of pavement perpendicular to the direction of motion while the sensor was moving forward. The test apparatus provided a physical constraint through gearing relating the x position

(forward motion) of the sensor and the y position (direction of scan) of the sensor. A photograph of the test setup is shown in Figure 2.1.

Table 2.3 - Laser Range Finder Specifications

MEASURABLE RANGE	60-140 mm
TIME RESPONSE (90%)	.7 ms
FREQUENCY RESPONSE	700 Hz
LINEARITY	1% full scale
RESOLUTION	180 μm
VIBRATION	10 - 55 Hz
SHOCK	10 G
TEMP FLUCTUATION	0.02% full scale/^{o}C
ILLUMINATION	4000 lux maximum
OPERATING TEMP	0 - 50^{o}C
HUMIDITY	35 - 85%
RECOMMENDED ANGLE	+/- 30^{o}
OUTPUT	+/- 4 V

For this experiment, the x and y positions were constrained such that

$$y = 2\sin(0.315\pi x) \tag{2.1}$$

where

x = forward motion of the sensor in mm, and

y = scan direction of sensor in mm.

From this relationship it is apparent that the sensor scanned in the y direction a total of 101.6 mm, and one cycle of sweep corresponded to 6.35 mm.

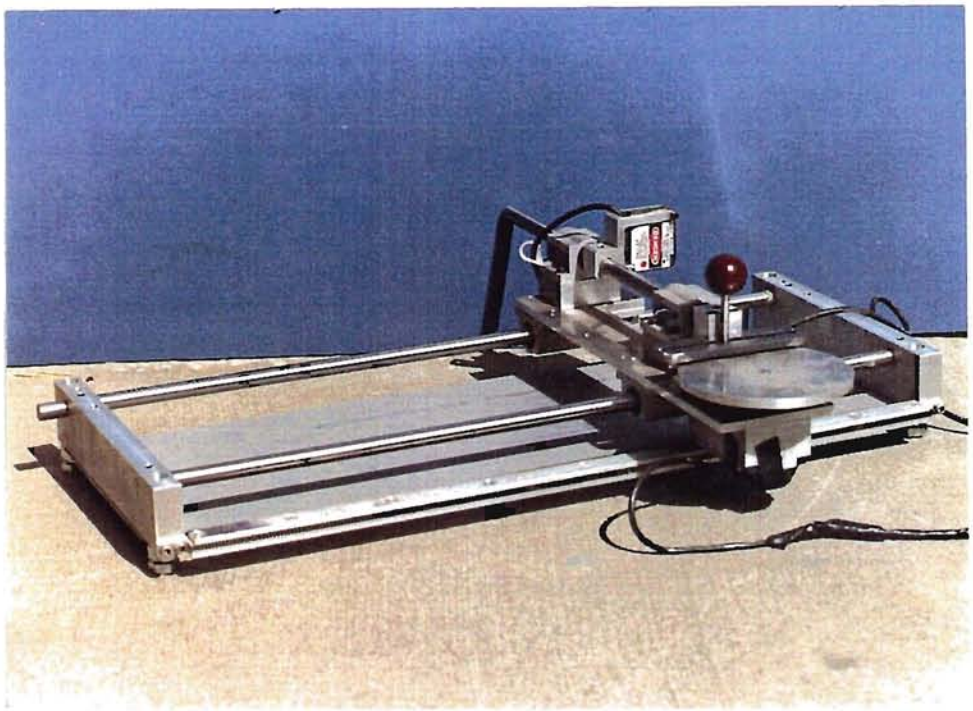


Figure 2.1 *Test Setup*

In order to determine crack coordinates, the x and y positions must be known. As described by Eqn. (2.1), the y position can be explicitly derived in software as a function of the x position. The x position must be measured. To measure the x position, a rotary encoder attached to a gear in the test apparatus was used. One revolution of the encoder shaft corresponded to one complete scan cycle. The pulse from the encoder was input into the counter of the I/O board. The x-position was therefore determined by reading the counter.

The following hardware was required for the experimental set-up:

1. 386-25 MHz microcomputer
2. I/O board with one A/D converter and one digital counter input (TTL level)
3. Laser range finder and signal conditioner

4. 28 VDC power supply for the laser range finder
5. Rotary encoder
6. 5 VDC power supply for the rotary encoder
7. Sensor motion controller (mechanically controlled through gearing)

The devices are connected as shown in Figure 2.2.

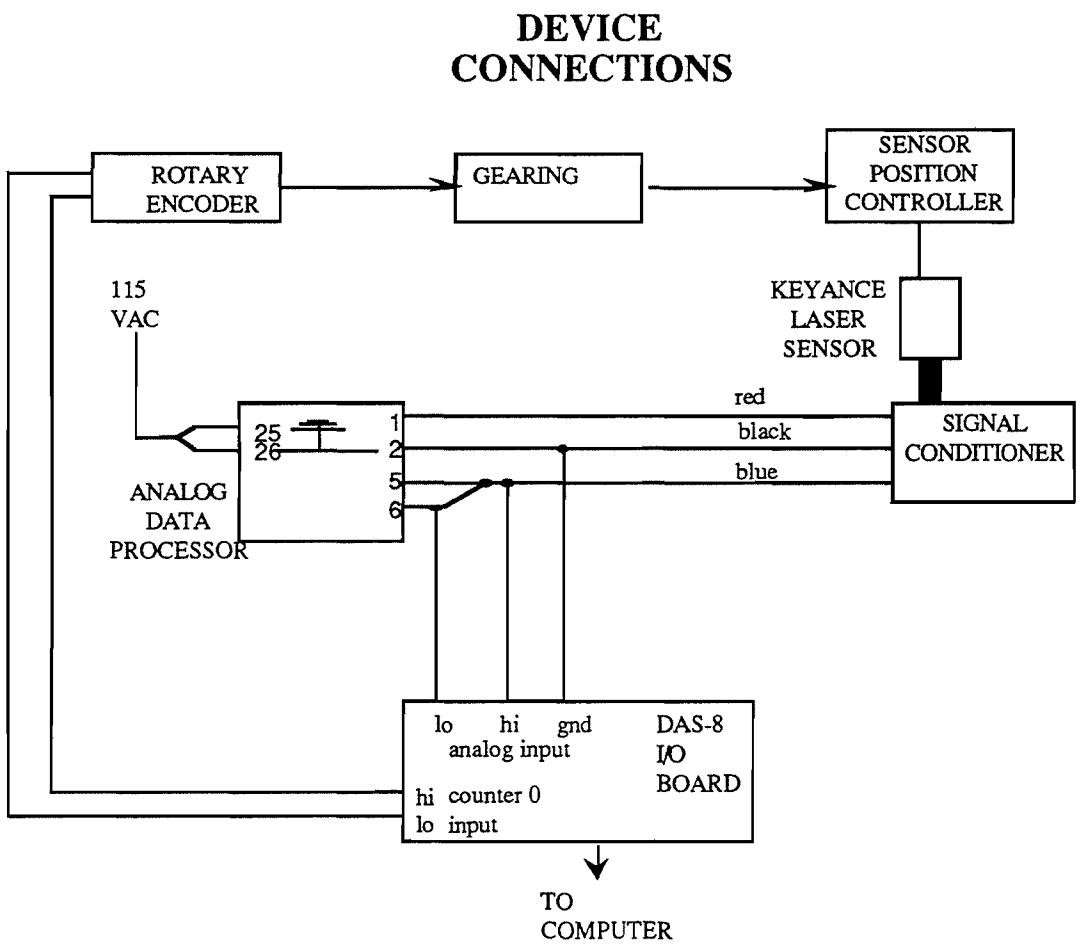


Figure 2.2 *Device Connections*

First, the signal from the laser range finder was input into the analog input of the I/O board. An analog to digital conversion on this signal was then performed. The input was appropriately scaled and converted to a distance measurement after analog to digital conversion.

Originally, the plan was to digitally filter signal. However, initial testing produced very clean output signals from the sensor which required no additional filtering. Some filtering was provided in that cracks detected which were less than some minimum width were ignored. This prevented unwanted spikes due to noise from appearing to be a crack. However, unwanted spikes did not seem to appear on the output signal.

Once data was input to the computer, the next step was to detect the edges of the crack. The currently sampled distance was compared to the average of the previously measured distances. It was necessary to compare this value to the average of the previously measured values to compensate for varying surface profiles and normal height deviations in pavement. If the current value varied from the averaged values by more than an acceptable tolerance, the sensor located the edge of a crack. The tolerance used was iteratively varied until the procedure consistently located crack positions.

If the program determined that the sensor detected the leading edge of a crack, the coordinates of the leading edge of the crack were stored. The program then began to search for the trailing edge of the crack. This was accomplished by comparing the current value to the averaged value. When the difference of these values was within the accepted tolerance, the trailing edge of the crack was considered to be found, and the coordinates of the trailing edge of the crack were stored.

The last step determined the crack coordinates. The leading edge and trailing edge coordinates were averaged, thereby finding the midpoint of the crack.

The sensor was tested on sample cracks in both AC and PCC. Figure 2.3 shows a plot of crack coordinates sensed on AC, while Figure 2.4 shows a plot of crack coordinates sensed on PCC. The solid lines represent the crack edges, while the dashed line represents the crack midpoint. The limiting factor on the accuracy of the test data was in the test apparatus rather than in the sensor. The encoder generated a pulse every three degrees. When this angle was converted to the y distance, a maximum error of 1.52 mm occurred. This error was observed by scanning a wooden board with a straight route. The y

coordinates showed a maximum variance of 1.52 mm, which indicates no additional errors in the sensing system. This is to be expected based on the manufacturer's specifications.

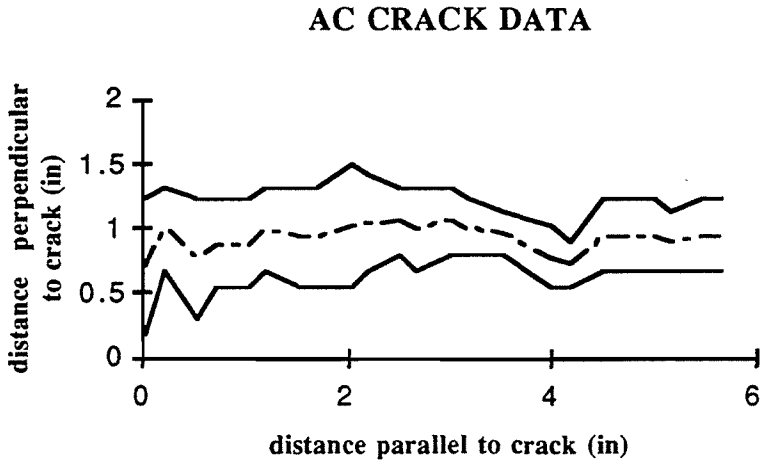


Figure 2.3 *AC Crack Position*

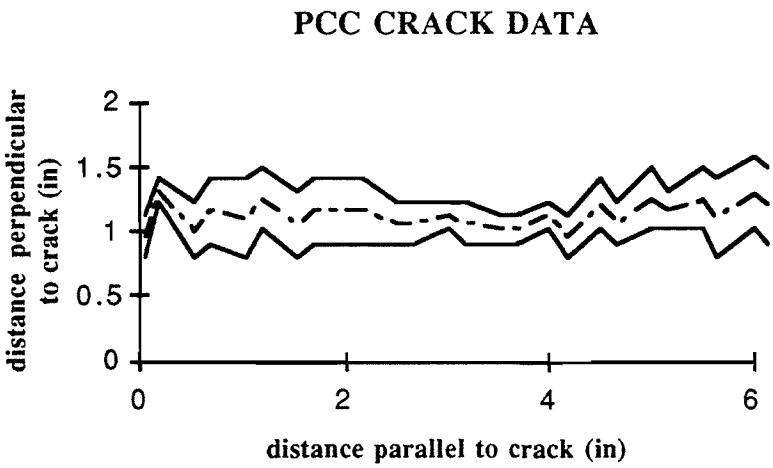


Figure 2.4 *PCC Crack Position*

This test has successfully proven that a laser range finder based on the principle of triangulation can accurately measure surface characteristics of both AC and PCC. Furthermore, from the surface profile data, the algorithm used was able to recognize cracks and determine crack coordinates.

2.4 - Detailed Component Description

A laser sensor using the principle of triangulation was thoroughly tested and the method of detection proved to be reliable. To employ the use of the laser range finding sensor, a scanning method to acquire data points is necessary. Furthermore, the scan rate must meet the requirements established in Section 2.1. There are four methods for acquiring these data points. The first and most simple consideration was to move the entire sensor using a motor. This method is considerably less expensive than other alternatives. If the entire sensor were to scan this area, the speed of the scan would be limited by the maximum allowable shock on the sensor. The speed must also be great enough to track the worst case crack which can turn 45^{o}. If one sensor is used to scan the 101.6 mm area, it is not possible to meet the two requirements; a speed great enough to track the worst case crack creates too much shock on the sensor. A second alternative would be to have multiple sensors scanning sections of the 101.6 mm area. If two sensors were used, the slowest possible scan rate would be 3.5 Hz. If three sensors were used, the slowest possible scan rate would be 2.3 Hz. It is believed that this scan rate would put excessive physical strain on the moving components, making this a likely failure point. Therefore, this method was considered to be inadequate for the task of local sensing.

Another possibility is the use of an array of discrete sensors across the 101.6 mm area. In order to achieve the required resolution of 1.588 mm, 64 sensors mounted 1.588 mm apart would be necessary. Because of the physical size constraints of the sensors available, it is not possible to mount the commercially available sensors in such close proximity. Furthermore, the cost of using 64 sensors is high. Therefore, this method was no longer considered for the task of local sensing.

A more complex method to achieve the needed scan rate can be accomplished by using rotating or scanning mirrors. Figure 2.5 illustrates an approach for incorporating the rotating mirrors into a working design. This method is considerably more expensive than the previously discussed scan method. Such a method is currently used in automated

welding to scan for a seam location (Appels, 1987; Nayak et al., 1987; Edling, 1986; Baranek et al., 1986). Based on the literature search, a scanning method employing synchronized scanning is desired for its compactness and increased field of view and resolution. Disadvantages in this scanning method are its high cost and number of moving parts. The performance of the sensor relies heavily on the synchronization of the two rotating mirrors, and the performance may degrade considerably due to misalignment of the mirrors.

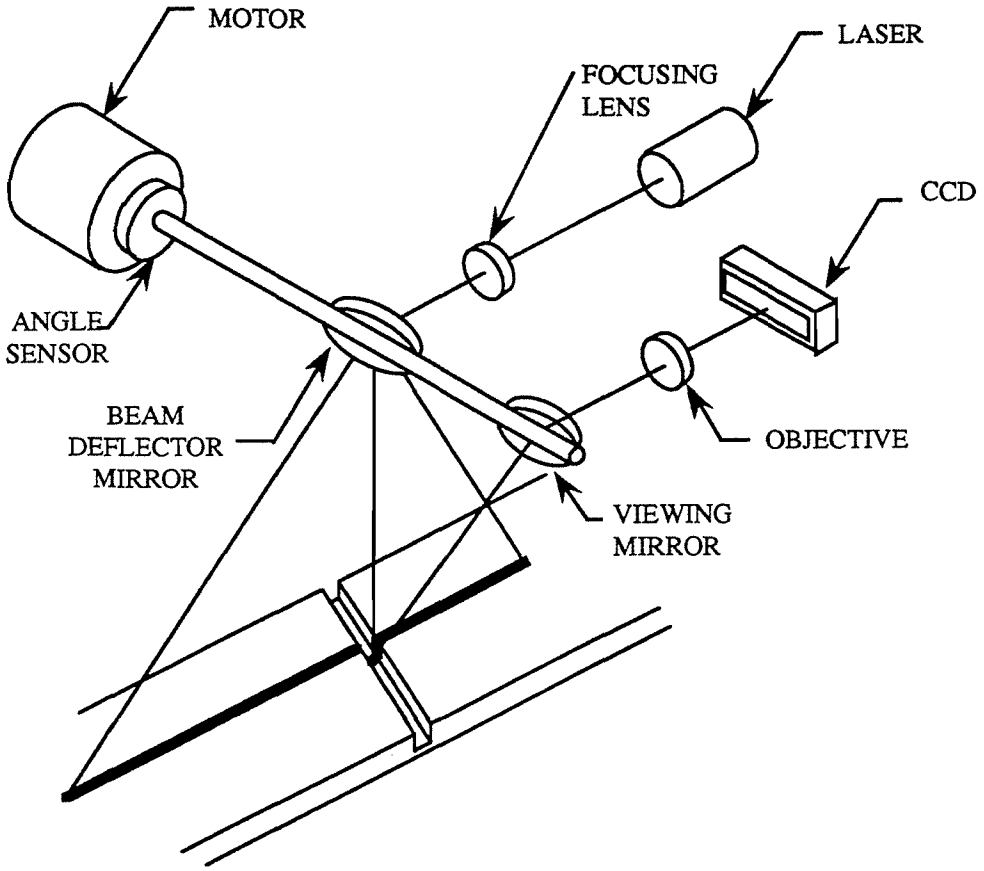


Figure 2.5 *Scanning Laser Range Finder With Rotating Mirrors*

Another method of using laser light to extract a 3-dimensional surface profile is known as structured light. Figure 2.6 illustrates a laser vision system based on structured light. A 3-dimensional surface profile is determined by projecting a laser pattern in a plane perpendicular to the surface being measured. The line of light is then observed by a CCD

camera at an angle, and, using triangulation, the surface features can be found. This technique is widely used in automated welding with much success. It is believed that structured light could provide all the information that is needed for local sensing. Additionally, laser range finders employing structured lighting techniques require no moving parts; therefore, they are mechanically simpler designs and are less prone to mechanical failures.

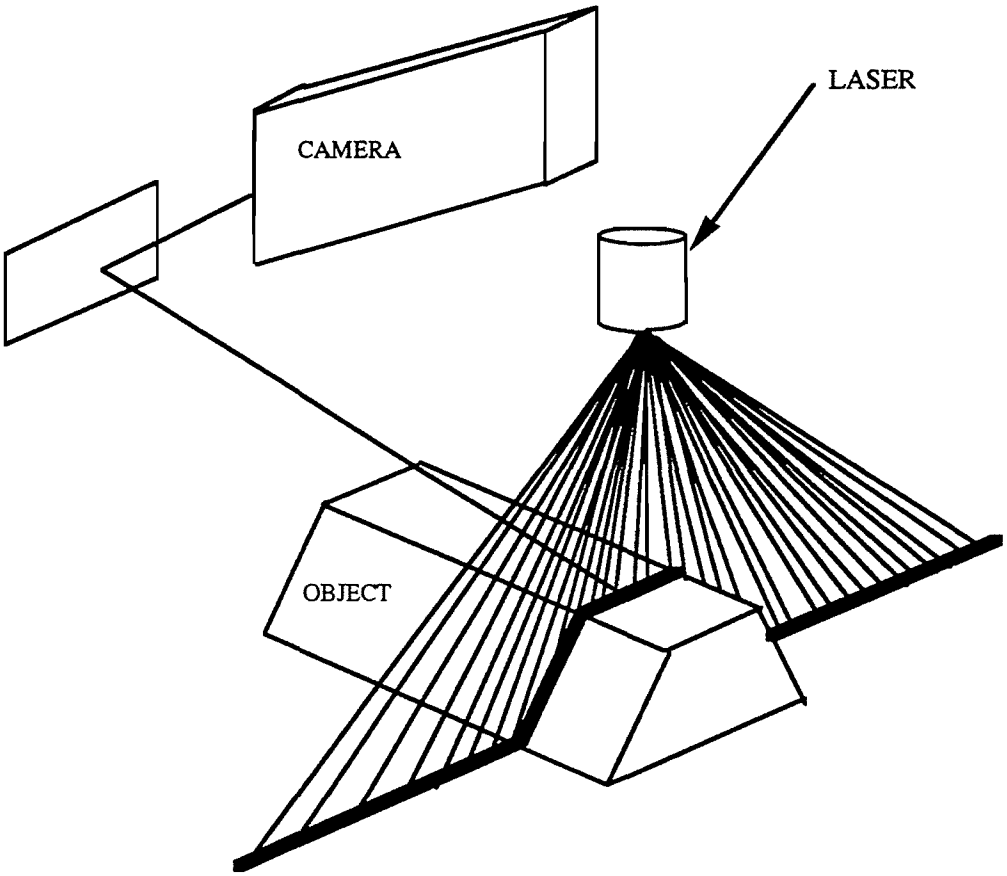


Figure 2.6 *Laser Range Finder Based on Principle of Structured Light*

Due to the sensor characteristics mentioned above and current applications of these sensors, it is believed that either a vision system using structured light or a synchronized scanning laser range finding sensor would suffice for local sensing. Both sensing methods have demonstrated ability to perform tasks similar to crack detection. Furthermore, as demonstrated by initial feasibility testing in Section 2.3, the laser triangulation sensing

method has proven to be reliable on pavement surfaces. To determine the optimum sensor, cost, reliability, accuracy, speed, size, weight and the number of moving parts (i.e., potential fail points) were considered. Initial estimates of the two types of sensors have shown the synchronized scanning laser range finder to be approximately double the cost of the alternative structured light technique. This is related to the complexity of the mechanical design of the rotating mirrors. The rotating mirrors introduces more moving parts and more possible failure points.

Both systems provide more accuracy than what is required for the crack sealing application. However, the structured lighting can measure up to 60 times per second while the scanning laser measures at 20 sweeps per second. As previously mentioned, the scanning requirement is 18 times per second for the sensor to track random cracks at 3.22 km/h.

One commercially available sensor using rotating mirrors weighs approximately 1.13 kg and measures 11.4 x 17.8 x 9.5 cm, while a commercially available structured light vision system weight 0.227 kg and measures 10.16 x 7.62 x 4.06 cm.

In conclusion, the laser vision system utilizing structured light is more economical. It has a simpler design with less moving parts. Three times faster scan rates are achieved with the structured light. Furthermore, the structured light sensor package is more compact and lighter. Therefore, it is determined that the laser vision system using structured light is the optimum sensor for crack detection in pavement at the current time.

A commercially available sensor was found which met the sensor requirements (MVS Modular Vision Systems Inc., 3195 De Miniac, Montreal, Canada, Model # MVS-30). The laser vision sensor is proven reliable, with no moving parts. It was specifically designed for tracking and inspection in robotics applications. Furthermore, this sensor is simple to use and rugged in harsh environments. The sensor has a built in heat exchanger for cooling and a cleaning mechanism which prevents dust build-up on the lens which

would distort the image. These attributes are necessary for proper performance of the sensor. A photograph of the sensor is shown in Figure 2.7.

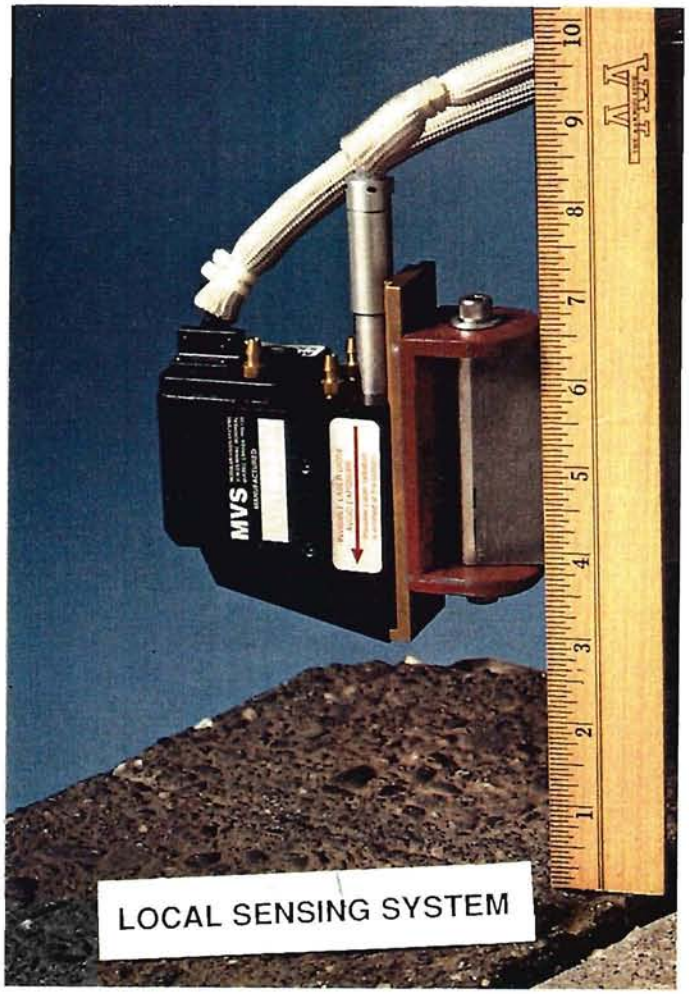


Figure 2.7 *Local Sensor*

The laser vision system consists of the hardware and software listed in Table 2.4. The system is connected as shown in Figure 2.8.

The sensor itself is a small package weighing 0.255 kg and measuring 10.16 x 7.62 x 4.06 cm. This package contains a laser light source, a CCD camera, and appropriate optics. The sensor will be mounted to the robot with a precision camera bracket. A vibration isolator will be placed between this bracket and the robot arm to protect the sensor from vibration and prevent image distortion. It should be noted that moderate vibration will not

destroy the reliability of the camera data because the camera captures a "blur" and the image is placed optimally within this blur to compensate for the vibration. Mounting will provide flexibility of vertical and lateral adjustment for initial calibration procedures.

Table 2.4 - LSS Hardware and Software

One Laser Vision Sensor
One 30 mW laser source
One sensor and laser power supply
One Laser Vision Image Processor
One co-processor board
One Microsoft compatible C5.1 library of driver routines
Menu Driven Program - Profile capture, store, segment, recall - Adjust laser intensity
Segmentation Program
386-25 MHz Microprocessor with ISA-Bus - VGA color monitor and graphics card - Four empty full sized slots - Two serial ports - 40 Mbyte hard drive - 4 Meg RAM

Power for both the laser and the camera is provided by a standard rack mounted power supply. Cabling will run between the sensor and this power supply as shown in Figure 2.8. Output from the camera will be input to the laser vision profile processing

board, which will be placed in an ISA Bus slot, where the camera data will be processed and the profile will be extracted. For increased performance, the profile data will be transferred to a coprocessor board, which also plugs into a standard ISA-Bus slot.

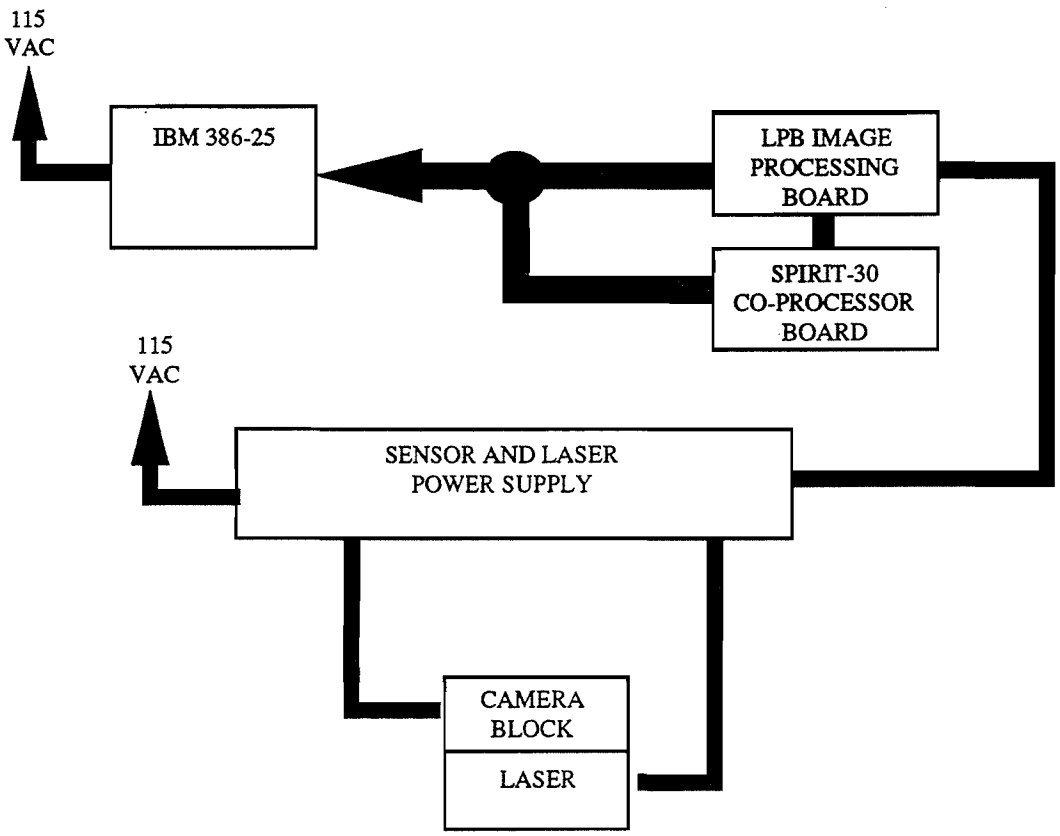


Figure 2.8 *Local Sensing System Device Connections*

A summary of the pertinent specifications for the laser vision system is shown in Table 2.5. For detailed specifications, see the manufacturers specifications in Appendix D.

Table 2.5 - Laser Vision System Specifications

	Horizontal		Vertical
Speed images/sec	60	30	
Resolution*	0.127 mm	0.0635 mm	0.152 mm
Accuracy position*	0.152 mm	0.0762 mm	0.203 mm
Accuracy mismatch*			0.0508 mm
Accuracy gap*	0.305 mm	0.152 mm	
Distance to surface	165.1 mm max.		
Field of View	110.24 mm max.		
Moisture	to 85%		
Vibration	typical vehicle vibration		
Temperature	-28.9 to 71.1°C		
Service Life	10 yrs.		
Speed	60 images per second - RS170 50 images per second - CCir standard		
Water cooling	0.946 liter per minute		
Air cooling	3.11 liter per minute		
Weight	250 g		
Processor	IBM-AT compatible, up to 8 MHz bus speed, requires 64K memory mapped space		

* Accuracy and resolution specifications are based on operating the sensor in the optimum area. The sensor will be operated outside this area to achieve a larger field of view. Effective resolution will be the field of view divided by 240. For a 101.6 mm field of view, this corresponds to a 0.4.24 mm resolution.

2.5 - Principles of Operation

The actual sensor being used on the crack sealing machine uses proprietary software and hardware to determine range information from camera data. However, there are developed generally used methods for extracting range information from camera data through the use of structured light. These methods have been researched and are presented in the following sections.

2.5.1 - Principle of Triangulation

As previously discussed, a laser vision system using the principle of triangulation was chosen to locally locate a crack in pavement. Figure 2.9 illustrates the principle of triangulation. The principle of triangulation determines distance measurements by transmitting a focused laser light source onto an object and then imaging the diffusely reflected light onto a photosensitive device (Mundy et al., 1987; Case et al., 1987; Kanade et al., 1987). The photosensitive device (PSD) is an analog light sensor that is sensitive to the intensity and position of a light spot in its field of view. Knowing the position of the image on the PSD, the distance between the detector lens and light source and the projection angle of the source, the distance measurement can be geometrically determined. The following expression relates between the range information and the location of light on the PSD (Doebelin, 1990).

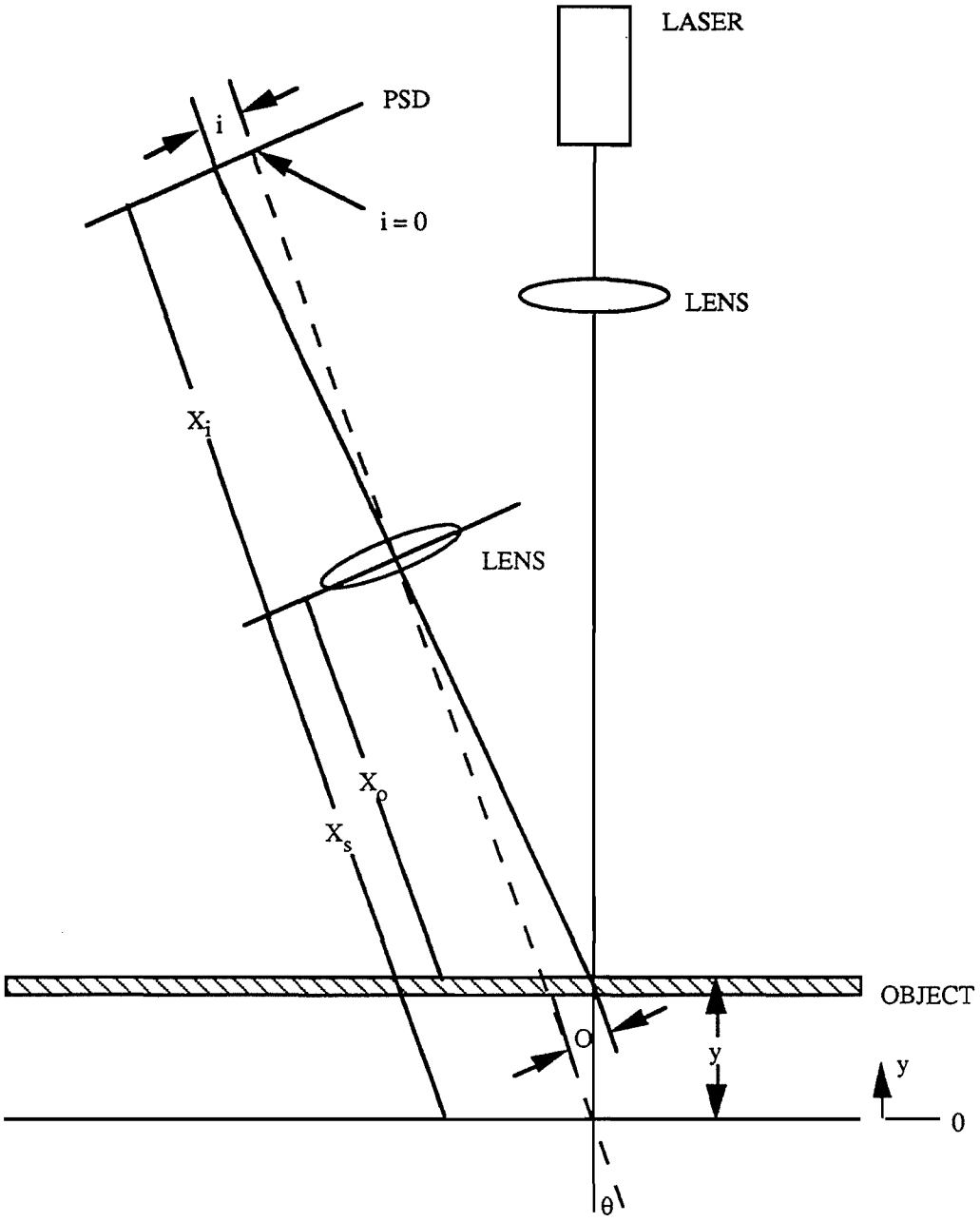


Figure 2.9 *Principle of Triangulation*

$$\frac{1}{f} = \frac{1}{X_i} + \frac{1}{X_s} \tag{2.2}$$

$$X_i = \frac{fX_s}{X_s - f} \tag{2.3}$$

where

f = focal length of lens,

X_i = fixed distance between lens and image, and

X_S = distance between lens and source.

By similar triangles,

$$i = \frac{OX_i}{X_o}, \tag{2.4}$$

where

O = distance along object from center to image,

X_O = object distance from the lens, and

i = distance along photosensitive device from center to image.

Further trigonometric relationships reveal

$$O = y\sin\theta, \text{ and} \tag{2.5}$$

$$\tan\theta = \frac{O}{X_s - X_o}, \tag{2.6}$$

where

X_S = distance between lens and source,

θ = angle between transmitter and receiver, and

y = standoff distance, where y = 0 is nominal.

Substituting Eqn. (2.5) into Eqn. (2.6) produces

$$\tan\theta = \frac{y\sin\theta}{X_s - X_o} \tag{2.7}$$

$$X_o = X_s - \frac{y \sin\theta}{\tan\theta} \tag{2.8}$$

Substituting Eqn. (2.3) into Eqn. (2.4) yields

$$i = \frac{OfX_s}{X_o(X_s - f)} \tag{2.9}$$

Additionally, substituting Eqn. (2.5) into Eqn. (2.6) yields

$$i = \frac{y \sin\theta f X_s}{X_o(X_s - f)} \tag{2.10}$$

and further substitutions of Eqn. (2.8) into Eqn. (2.10) gives

$$i = \frac{y \sin\theta f X_s}{(X_s - \frac{y \sin\theta}{\tan\theta})(X_s - f)} \tag{2.11}$$

$$i = \frac{fX_s}{X_s - f} \frac{y}{\frac{X_s}{\sin\theta} - \frac{y}{\tan\theta}}. \tag{2.12}$$

Because

$$\frac{y}{\tan\theta} << \frac{X_s}{\sin\theta}, \tag{2.13}$$

$$i \approx \frac{f}{X_s - f} y \sin\theta. \tag{2.14}$$

As seen by Eqn. (2.14), the depth measurement y is proportional to the measurement i on the PSD. However, without making the assumption which produced Eqn. (2.14),

nonlinearity can be taken into consideration by system static calibration and linearization by common hardware or software signal conditioning schemes.

2.5.2 - Structured Light

The laser vision sensor extracts three-dimensional surface profiles by using the principle of structured light. A laser light source is projected approximately perpendicular to the direction of the crack. The bright line produced by this laser source on the surface is then observed by a CCD camera at an angle (20^{o}-30^{o}). The analog camera data is then digitized, filtered, and processed using triangulation to determine depth information.

2.5.2.1 - Determining Range Information

Once the camera data has been digitized and filtered, the next step is to determine the location of the reflected light on the CCD array. The CCD data will be varying gray levels images at each pixel due to varying reflectivity of the scene. One method of determining the laser location would be a thresholding algorithm. By comparing the gray level of each pixel to some threshold level, each pixel will be converted to either black or white. The white pixels are considered to be where the laser light is located. However, this method does not address the fact that the intensities of different types of pavement may vary, and therefore the threshold value may need to vary for different types of pavement.

A simpler and more rugged method classifies each pixel along a row in the CCD as one of four classes (Mundy et al., 1987),

1. Positive going zero crossing,
2. Positive peak,
3. Negative going zero crossing, or
4. Negative peak.

The pixel location corresponding to the positive peak is considered to be the location of the laser light. The pixel location is then used to calculate range data using triangulation, as described in section 2.5.1.

Due to this peak detection, an important feature of the laser vision system is that the laser line must be as thin as possible because variances in surface reflectivity and surface roughness alter the intensity of the stripe. Therefore, the most intense point in the stripe image is not necessarily at the stripe center.

2.5.2.2 - Object Identification

The current algorithm to detect cracks in pavement relies solely on range information. However, structured light can provide much more information than that which is being used. In particular, objects can be recognized by a computer through the use of structured light. Range information gathered through structured light is condensed for purposes of object identification and manipulation.

After the image is filtered, it will appear to be a line containing bends and breaks (Davies, 1990); see Figure 2.10. Breaks consist of both jumps and discontinuities, where jumps are considered to be breaks in the line both horizontally and vertically and discontinuities are considered to be breaks in the line horizontally but not vertically. Bends are due to either convex or concave angle changes in which both faces are visible. When a concave edge is visible, both faces are always visible. However, when a convex edge is visible, sometimes one of the two side faces is hidden by the other, thus creating a break. Jumps and discontinuities in the laser line are due to actual edges and surfaces which are hidden from the laser source. The hidden sections of the line must be reconstructed in order for the object to be identified.

As described in section 2.5.2.1, the first step is determining where the laser reflection is located on the CCD array. Once these pixels are identified, meaningful data can be

extracted. This can be accomplished by mathematically manipulating the camera data into valuable information (Sugihara, 1987).

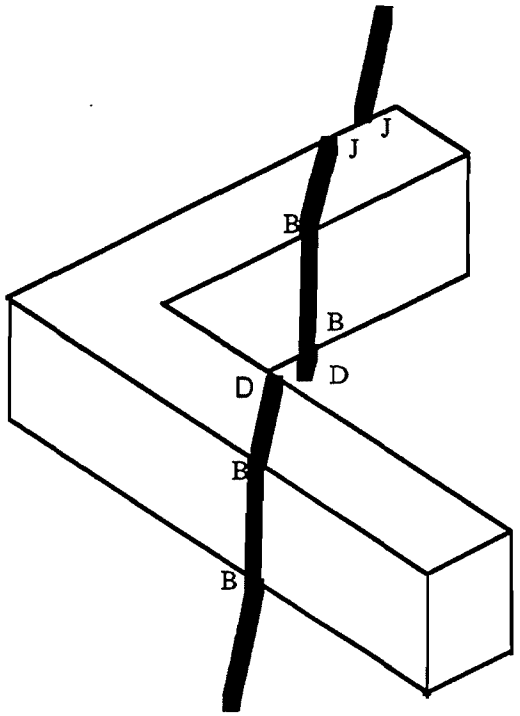


Figure 2.10 *Image Containing Bends (B), Jumps (J), and Discontinuities (D)*

Once the camera image has been determined, the next step in object recognition is edge detection. An edge is located by determining the location of bends, jumps and discontinuities. When the camera is to the right of the light source, a convex edge will cause the reflected light to bend upward, while a concave edge will cause the light to bend downward. When the camera moves across an obscuring edge, a discontinuity occurs; see Figure 2.11.

Therefore, the type of edge encountered can be determined by observing the direction which the light bends. To determine which direction the light bends, an operator can be derived. This operator can then be interpreted to indicate concave and convex bends, jumps and discontinuities. This operator has been defined to be

$$O(y) = \frac{1}{2\Delta x}(Y^+ + Y^- - 2Y) \tag{2.15}$$

where

x = horizontal direction along sensing element array,

y = vertical direction along sensing element array,

P = illuminated point on sensing element at coordinates X, Y,

Δx = distance between pixels along the x axis,

P^+ = illuminated point on sensing element positive Δx distance along x axis from P,

P^- = illuminated point on sensing element negative Δx distance along x axis from P,

Y = y value of)

Y^+ = y value of P^+, and

Y^- = y value of P^-

The operator O(y) indicates the following types of edges:

1. When O(y) moves across a planar surface, O(y) = 0.
2. When O(y) moves across a convex edge, O(y) > 0.
3. When O(y) moves across a concave edge, O(y) < 0.
4. When O(y) moves across an obscuring edge, O(y) goes from positive to negative, or from negative to positive.

Thus, by manipulating camera data into this operator, convex, concave, and discontinuous edges can be located.

To determine edge location from discontinuities and jumps, it is necessary to know whether the break is indicating an occluding edge or an occluded edge. An occluding edge is an actual edge in the surface, while an occluded edge is created by shadow regions and is not directly significant. To distinguish between the two types of edges, a general rule applies. If the line of laser light is projected from the left, then the left hand component of a

discontinuous edge will be the occluding edge and the right hand component will be the occluded edge.

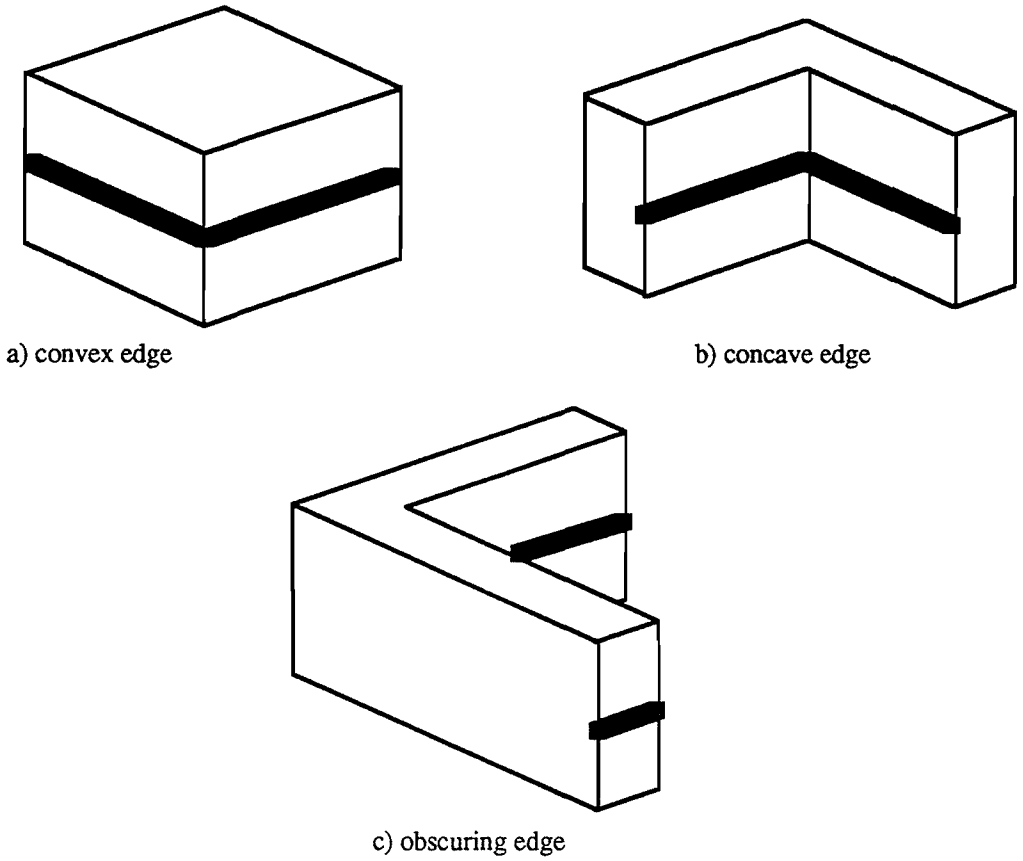


Figure 2.11 *Reflected Light Across Edges*

Next, it is necessary to determine the shape of the object in the hidden regions. This can be done using a reference junction dictionary. The reference dictionary contains all possible combinations of surfaces created by the intersection of different types of edges which have been identified. It is necessary to assume the shape types are limited so that the reference junction dictionary has a reasonable limit to its size. For instance, the type of objects being identified may be limited to tetrahedral shapes, or shapes which have vertices consisting of three intersecting faces. This significantly reduces the size of the junction dictionary and makes development time and computing time practical for applications.

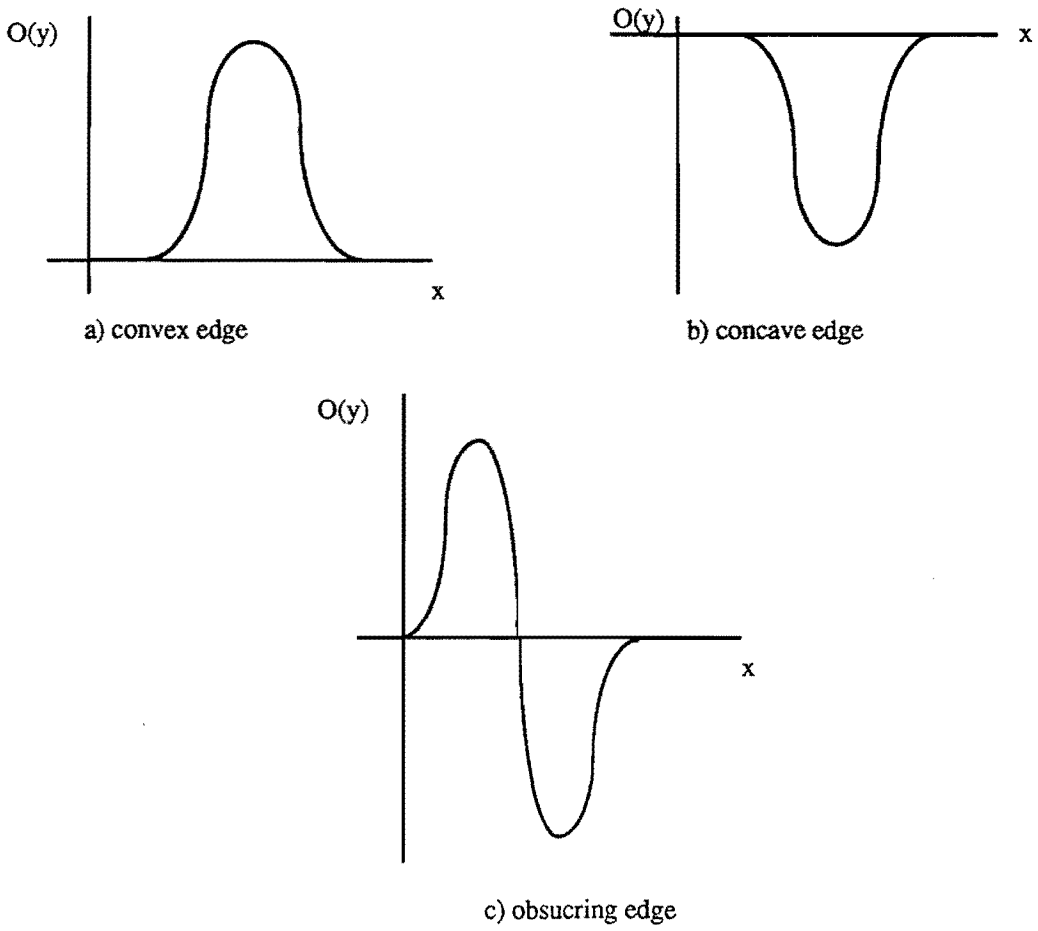


Figure 2.12 *Operator to Determine Direction of Light Bend*

2.6 - Crack Identification and Associated Signal Processing

In order to acquire and interpret the depth information gathered by the sensor described in Section 2.5, a customized software program was developed. This program acquires data, interprets the data and extracts information relating to the position of the crack, and communicates this information to the appropriate systems.

The calling program will operate in real time on a microcomputer with a 386-25 MHz microprocessor. The program will be compiled using Microsoft QuickC, which is compatible with the Microsoft compiler which has compiled the functions provided with

the laser vision sensor. The supplied functions are contained in a library which will be linked to the calling program during compilation.

A flowchart of the crack sensing algorithm is illustrated in Figure 2.13. As shown in this flowchart, the program consists of an initialization procedure followed by the main body of the program which senses the presence of cracks in pavement. In the main body, first a profile of the road surface is captured. Next, each individual data point along the profile is analyzed to determine if a crack is found. If a crack is located, the program sends position information to the Robot Positioning System and then loops back to the start of the main body and captures the next profile. If the entire profile is analyzed and no crack is located, a signal is sent to the ICU and then the program loops back the start of the main body and captures the next profile. Details of each task performed by the crack sensing program is described in the following sections.

Before the main body of the program can begin acquiring data and detecting crack location, an initialization procedure in software must be performed. During the initialization procedure, the laser vision system hardware will be initialized. This is done through calls to initialization procedures which were provided with the laser vision system software. The laser intensity will also be set to maximum intensity by calling a procedure provided with the laser vision system software. Maximum intensity has reliably given optimal results on pavement surfaces during testing procedures. Serial communication must also be initialized. Two serial ports will be used for communications (COM1 and COM2). The baud, number of stop bits, and parity (odd, even, or none) must be set for each serial port before communication can be successfully established. Using C function calls to the DOS operating system, the serial ports are initialized. Also, a profile will be extracted using laser vision system software and the typical roughness of the pavement surface being measured will be determined. This measurement is necessary in order to determine typical depth variations from the sensor to the pavement surface. From

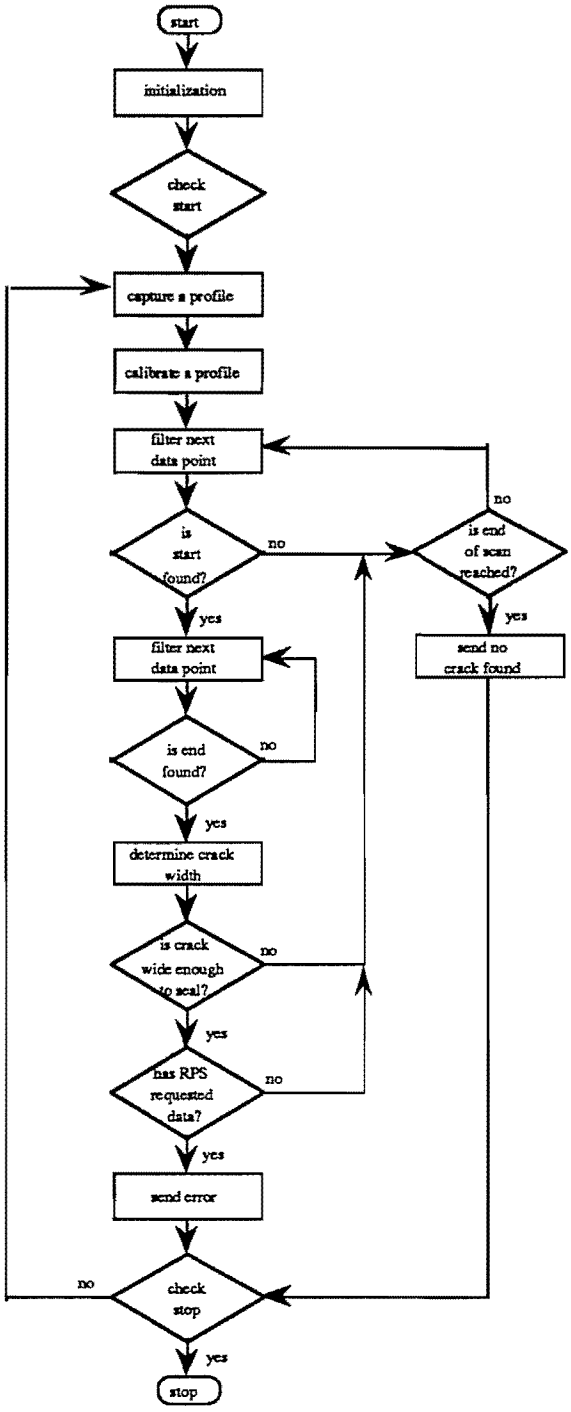


Figure 2.13 *Crack Locating Program Flowchart*

the typical variance, an acceptable tolerance will be set. Depth measurements exceeding the tolerance will be considered a crack. Last, the average depth to the pavement at the beginning of the profile must be determined This is accomplished by averaging the first twenty-five extracted data points over a typical section of pavement not containing a crack. This average will be used to initialize the digital filter.

To compensate for varying surface profiles and normal height deviations in pavement, a digital low pass filter will be used on the extracted data. A low pass filter was chosen to filter out normal high frequency variations in the depth measurements (Doebelin, 1990). The measurements of concern are low frequency components in the depth measurements due to the presence of a crack. The digital filter was modeled after a low pass second order filter. The filter constants were determined with the aid of a discrete Fourier transform (DFT) analysis. A detailed description of the filter design is found in Appendix A. Appendix B contains the computer program which performed the DFT. Figure 2.14 shows the profile of a crack measured by the local sensor before filtering, while Figure 2.15 shows the crack profile after filtering.

Before the program continues, it waits for a start signal from the ICU via the COM2 serial port. Once the signal to begin is received, the first task performed by the calling program is to determine if the sensor has located a crack. If the sensor is located over a crack, location will be determined and sent to the positioning system. If no crack is found, an indicating signal will be sent to the ICU. This will be performed by making a software function call to extract a profile and calibrate the data. The result will be x, y coordinates in mm, where x is the direction along the line of light, and y is the depth measurement from the sensor to the surface. The zero coordinate along the x direction is located in the center of the scan. The result is stored in an array accessible to the program. Each element in this array will be consecutively filtered and then compared to the previous value to determine if the starting edge of a crack has been located. If the current measurement varies by more

than the accepted tolerance determined in the initialization procedure, then the program assumes that the leading edge of the crack has been located.

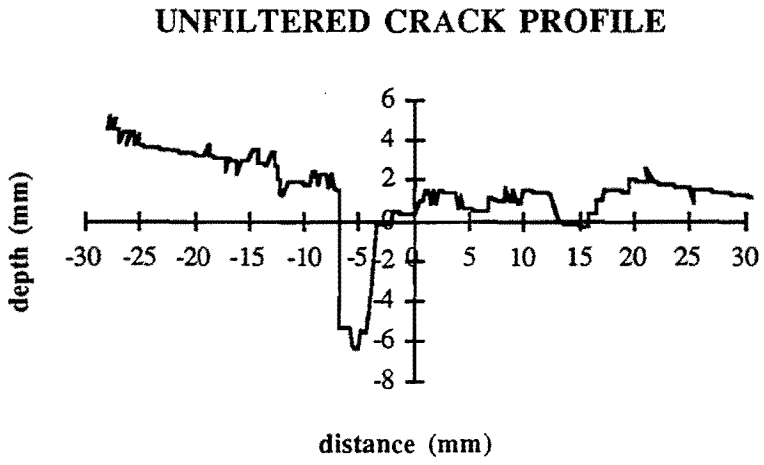


Figure 2.14 *Unfiltered crack profile*

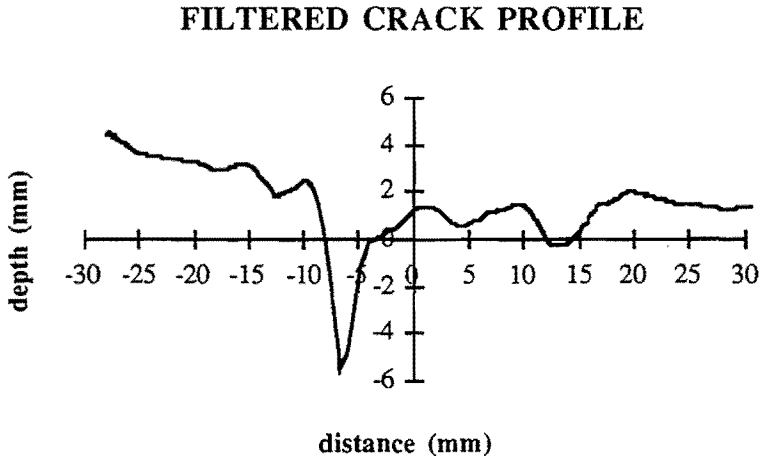


Figure 2.15 *Filtered crack profile*

A simpler routine would be to compare the current data to a setpoint value. However, this approach was not taken because the road surface profile may vary gradually within the sensor field of view by more than the accepted tolerance without a crack being

present. By comparing the current value to the previous value, gradual changes in profile are allowed for, and will not erroneously indicate the presence of a crack.

A problem arises in that a previous value is needed in both the filtering routine and the crack detection routine. Without an initial value, it would be impossible to detect a crack if it was located at the first data point. To solve this problem, the average value determined during the initialization routine will be used.

Once the program has determined that the sensor has detected the leading edge of a crack, a flag will be set, and the x coordinate of the leading edge of the crack will be stored. This flag will remain set until the trailing edge of the crack is located. The trailing edge of the crack will next be located by again comparing the current depth measurement to the last measurement taken. When the depth variance exceeds the tolerance, it is assumed that the crack wall has been located. The trailing edge of the crack has been located when the difference between the current and previous depth measurements return to within the accepted tolerance after the call of the crack has been detected. The x coordinate of the trailing edge of the crack is then stored.

The next software task is to determine the width of the crack. This is simply determined by subtracting the leading edge coordinate from the trailing edge coordinate. Cracks less than 3.175 mm in width will not be sealed[1]. Therefore, if the crack width is calculated to be less than 3.175 mm, the crack is ignored (same as no crack located). The effect of this is a threshold filtering which will ignore unwanted spikes due to noise in the data. In the case where a crack of less than 3.175 mm in width is located, the program will continue to analyze the remainder of data points in the scan line for a crack in the same manner as described above.

---

[1]During machine development a minimum crack size of 1/8 inch for sealing has been assumed. Should specifications be modified, the Local Sensing System has adequate resolution to locate cracks as small as 0.025 inches.

Once the crack has been located and the width has been determined to be greater than 3.175 mm, the midpoint of the crack is calculated by averaging the leading edge and trailing edge coordinates. This location is the error feedback signal required by the positioning system to follow the crack.

Once the error signal is determined, the program checks to see if the RPS has requested a new piece of information. When the RPS is ready for a signal, it writes a byte to the LSS serial port. Through DOS commands, the crack detecting program checks to see if a byte appears in its serial port. If no byte appears, the RPS is not ready for data, and the program continues. If the port is not empty, it is cleared, and the error signal is sent to the RPS. The feedback signal will be directly provided to the robot positioning system via a serial port. Using C function calls to the DOS operating system, data can be sent one byte at a time per the RS-232 standard protocol.

The program will then loop back to the beginning, where a profile is once again extracted and analyzed. The remainder of data points extracted during the last measurement are not further analyzed because the current machine operation does not address situations when two cracks have been located within the local sensor field of view. Because the other sub-systems do not currently support the situation of multiple cracks, it would be a waste of processing time to further analyze the data. However, only minor modifications to the program are necessary to alter the operation so that error signals are sent to the positioning system each time the sensor locates a crack regardless of the number of cracks which have been located.

If each element of the extracted profile is analyzed and no crack location is found, an indicating signal is sent to the ICU. This signal is also sent over a second serial port (COM2). Under normal conditions, the laser vision sensor will be located over a crack and no signal will be sent to the ICU. Then program loops back to the starting point, where once again a profile is extracted and analyzed.

CHAPTER 3 - EXPERIMENTAL VERIFICATION

In Chapter 2, the Local Sensing System (LSS) requirements were established and a technology was selected. An initial feasibility test was performed which proved that the sensing system technology can successfully identify cracks in both AC and PCC. The selected sensor is a laser vision system based on the principle of structured light. Its corresponding software has the ability to locate a crack in pavement and send position information to the Robot Positioning System. In Chapter 3, the operation of this sensing system will be experimentally verified.

The objective of the experimental verification is to test the instrument response of the LSS. The instrument response is considered to be output produced for a given input. The input into the system is the absolute location of a crack in the surface being measured. The output of the system is the crack location generated by the LSS hardware and software. Additionally, the LSS must output a signal when no crack is located in the surface being measured. The LSS must produce viable output for all expected input under a variety of different operating conditions. This output must meet the requirements established in section 2.1.

Before output characteristics of an instrument can be analyzed, the instrument must be calibrated. Calibration procedure consists of determining output for a given input over the range of inputs which the instrument is to be used. This will be accomplished by comparing the response of the LSS to known input. A problem arises in that the known input is considered to be the true input, while in reality it is affected by the response of the measuring instrument which is determining the input. Therefore, to produce a meaningful calibration results, the instrument used to measure the input must have better response than the system being calibrated.

From the calibration data, a calibration curve, or a plot of the instrument output versus the input over the usable range, will be generated. A best fit line will be generated

through the linear region of the plot using the least squares method (Doebelin 1990). From this curve, the sensitivity, linearity, bias, and accuracy of the LSS will be determined. Sensitivity (also referred to as gain) is the slope of the calibration curve in the linear region. Linearity is the departure of the calibration curve from a straight line relationship. Bias, or systematic error, is a common offset to all readings, corresponding to the y intercept of the calibration curve. Finally, accuracy is the ability of the instrument to produce an output which corresponds to the true input. To quantitatively measure the accuracy, a percent error will be calculated. The percent error is defined as the difference between a given data point and the calibration curve. The value will be expressed as a percent of the instruments full scale reading, using the point which strays most from the calibration curve.

Once the calibration curve has been generated, the precision, or repeatability, of the Local Sensing System will be determined. Precision is defined as the ability of an instrument to produce consistent output for a given input. This will be accomplished by measuring crack location using the LSS. Enough points will be measured so that a normal curve can be generated. From the data, the sampled standard deviation of the sample will be calculated (Doebelin 1990). The sampled standard deviation is defined by

$$s = \sqrt{\frac{\sum_{i=1}^{N}(X_i - \overline{X})^2}{N - 1}} \tag{3.1}$$

where

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N} \tag{3.2}$$

and

X_i = individual reading,
N = total number of readings,
s = sample standard deviation, and
$\overline{X}$ = sample mean average.

For a perfect Gaussian distribution, the following relationship arises:

68% of the readings lie within +/-1 standard deviation of the average,

95% of the readings lie within +/-2 standard deviations of the average, and

99.7% of the readings lie within +/-3 standard deviations of the average.

Typically, a 95% confidence level is used as a measure of the precision of the instrument.

Once the static instrument response has been determined, a variety of tests will be performed. The objective of these tests is to ensure proper performance of the Local Sensing System under different operating conditions. Specifically, the sensor must have the ability to distinguish between oil spots, shadows, previously sealed cracks, and actual cracks under different lighting conditions. The sensor must produce reliable and accurate results which meet the requirements established in section 2.1.

3.1 - Test Set-up

In order to perform the experimental verification, the hardware and software shown in Table 3.1 are necessary. In order to perform the static calibration procedure, the sensor will be mounted on a robot arm. Using a teach pendant and the robot controller, it is possible to very accurately position the sensor. The surface being measured will be a piece of plywood board containing a 22.2 mm routed crack. The precision measurement procedure will also be performed while the sensor is mounted on the robot arm. Again, the surface being measured will be the same piece of plywood. A photograph of the sensor mounted on the robot arm is shown in Figure 3.1.

To accomplish the performance tests, the sensor will be mounted on a test apparatus. The test apparatus allows the sensor to be placed at any desired position over the surface being measured. In this case, the sensor will be mounted 95.25 mm from the surface being measured, indicative of its relative position to the road on the actual crack sealing machine. A photograph of the test apparatus for the performance tests is shown in Figure 3.2.

Table 3.1 - Hardware Requirements for Experimental Verification

One Laser Vision Sensor
One 30mW laser source
One sensor and laser power supply
One Laser Vision Image Processor
One co-processor board
Crack identification program
PCC sample containing a crack between 3.175 mm to 25.4 mm wide
AC sample containing a crack between 3.175 mm to 25.4 mm wide
PCC sample containing a sealed crack
AC sample containing a sealed crack
Robot, robot controller, and teach pendant
Laser Vision Sensor mounting bracket
Plywood board containing a 22.2 mm route
386-25 MHz microcomputer
Test apparatus

The system connections are shown in Figure 3.3. The power supply requires an external 110 VAC source, which will be provided by a standard outlet. A cable will run between the power supply and the laser vision sensor, which will provide power to the camera and laser, and will provide a synchronization pulse to the camera.

The image processing board and co-processor will both plug into a standard ISA-bus slot in the microcomputer. There is a cable connection between the image processing board and the co-processor. A second cable runs from the sensor into the computer to the

image processing board. The data collected by the sensor enters the computer via this cable. The computer also requires an external 110 VAC power source.

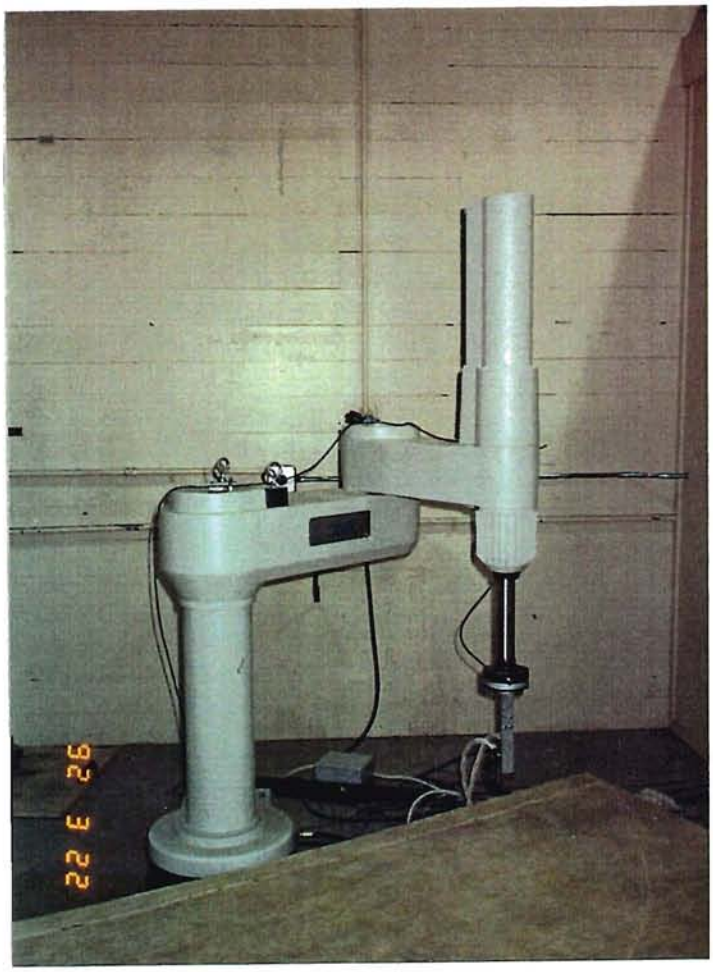


Figure 3.1 *Sensor Mounted on Robot Arm*

Figure 3.2 *Performance Test Apparatus*

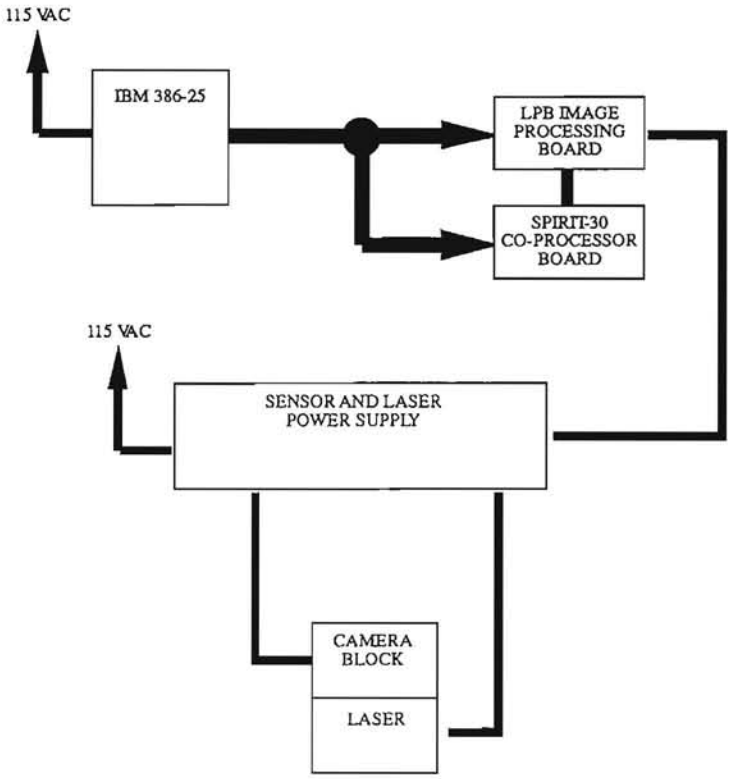


Figure 3.3 *Local Sensing System Device Connections*

3.2 - Procedure

The following procedure will be performed to confirm sensor performance. First, static instrument response of the Local Sensing System will be determined. Data will be collected to produce a calibration curve and thereby determine the LSS sensitivity, linearity, bias, and accuracy. Next, the precision of the LSS will be determined by measuring the many output points for the same input. Statistics will be performed on the data to determine the sample standard deviation and the precision of the instrument. Finally, the performance of the local sensor will be determined in a variety of operation conditions. The procedure will be performed over four pavement samples under two different lighting conditions, corresponding to eight experimental runs. The four pavement samples include a PCC pavement section containing a crack, an AC pavement section containing a crack, a PCC pavement section containing a sealed crack, and an AC pavement section containing a sealed crack. The experiment will be run when the pavement sample is exposed to direct sunlight and when the pavement sample is located in the shade. Profile data and crack location will be collected and stored for analysis. Each test run will last 300 seconds.

3.2.1 - Static Calibration Procedure

The calibration procedure consists of the following steps:

1. Connect the hardware as shown in Figure 3.3.
2. Mount the sensor on the robot arm as shown in Figure 3.1.
3. Place the sensor over the center of the routed crack so that the direction of laser light is perpendicular to the direction of the routed crack.
4. Using the teach pendant, adjust the height of the sensor so that the bottom edge of the sensor is 95.25 mm above the surface being measured.
5. Record a measurement corresponding to crack location.
6. Move the sensor 2 mm perpendicular to the crack direction using the teach pendant.
7. Record another measurement corresponding to crack location.

8. Continue taking measurements every 2 mm until the crack is no longer in the field of view of the sensor.
9. Return sensor to the start point over the center of the crack.
10. Record a measurement corresponding to crack position.
11. Move the sensor 2 mm in the opposite direction as previously performed in step 5.
12. Record another measurement corresponding to crack location.
13. Continue taking measurements every 2 mm until the crack is no longer in the field of view of the sensor.
14. Repeat procedure for a second set of calibration points
15. Perform least squares curve fit on linear portion of data to produce curve fit.
16. Plot data points and calibration curve.
17. Determine sensitivity, linearity, bias, and accuracy.

3.2.2 - Precision Measurement Procedure

Next, the precision of the LSS will be determined. In order to do so, the following steps will be performed.

1. Connect the hardware as shown in Figure 3.3.
2. Mount the sensor on the robot arm as shown in Figure 3.1.
3. Place the sensor approximately over the center of the routed crack so that the direction of laser light is perpendicular to the direction of the routed crack.
4. Using the teach pendant, adjust the height of the sensor so that the bottom edge of the sensor is 95.25 mm above the surface being measured.
5. Record 300 measurements of crack location at this location.
6. Move the sensor over approximately 10 mm perpendicular to the direction of the crack using the teach pendant.
7. Record 300 measurements of crack location at this location.

8. Move the sensor 20 mm in the opposite direction, so that the sensor is -10 mm from the center of the crack.
9. Record 300 measurements of crack location at this location.
10. Determine sampled standard deviation for the three sets of sampled data.

3.2.3 - Performance Testing Procedure

Finally, the performance of the LSS under different operating conditions will be determined. To measure sensor performance, the following procedure will be performed on both AC and PCC pavement samples:

1. Connect hardware as shown in Figure 3.3.
2. Mount the sensor on the test apparatus 95.25 mm from the surface being measured as shown in Figure 3.2.
3. Place test apparatus with sensor attached such that pavement sample is exposed to direct sunlight.
5. Place sensor over a crack in the pavement sample.
4. Record measurements of crack location for 60 seconds.
5. Move the test apparatus with sensor attached such that pavement sample is in the shade. Do not move the sensor.
6. Record measurements of crack location for 60 seconds.
7. Return test apparatus and sensor so that pavement sample is exposed to direct sunlight.
8. Place sensor over a section of pavement not containing a crack.
9. Record measurements of crack location for 60 seconds. If the sensor is operating properly, no data should be collected since no crack is present.
10. Move the test apparatus with the sensor attached such that pavement sample is in the shade. Do not move the sensor.

11. Record measurements of crack location for 60 seconds. If the sensor is operating properly, no data should be collected since no crack is present.
12. Place the sensor over the sealed crack in the pavement sample.
13. Record measurements of crack location for 60 seconds. If the sensor is operating properly, no data should be collected since no crack is present.
14. Move the test apparatus with the sensor attached such that pavement sample is in the shade. Do not move the sensor.
15. Record measurements of crack location for 60 seconds. If the sensor is operating properly, no data should be collected since no crack is present.

3.3 - Data

Data was acquired per the procedures listed in Section 3.2. The data collected is contained in this section.

3.3.1 - Static Calibration Data

The data contained in Table 3.2 comprises the points which generate the calibration curve. This data is the crack position in mm sensed by the LSS as the local sensor was moved 2 mm perpendicular to the crack for each point.

3.3.2 - Precision Measurement Data

Three sets of 300 measurements of crack location have been collected to determine the statistical precision of the LSS. This data is plotted in Figure 3.4. The x axis corresponds to the sample number, while the y axis corresponds to the crack location in millimeters. Because the sensor was not moved during each of the three tests, the crack position determined by the LSS should not vary for each test.

Table 3.2 - Static Calibration Data

KNOWN INPUT (mm)	OUTPUT 1 (mm)	OUTPUT 2 (mm)
38	32.7	
36	32.7	32.3
34	31.0	31.9
32	30.9	30.0
30	30.0	30.0
28	28.2	27.9
26	24.0	24.0
24	23.8	23.9
22	21.9	21.8
20	20.1	18.2
18	17.9	16.0
16	14.0	14.0
14	12.2	14.1
12	10.1	10.3
10	8.2	8.0
8	6.1	8.0
6	4.1	6.1
4	3.8	1.9
2	1.9	0.0
0	-0.1	-0.2
0	0.0	-0.1
-2	-2.1	0.0
-4	-4.2	-4.4
-6	-4.2	-6.2
-8	-8.3	-8.2
-10	-8.2	-10.0
-12	-12.2	-12.2
-14	-14.3	-12.0
-16	-14.0	-16.0
-18	-15.8	-18.1
-20	-21.7	-19.9
-22	-23.3	-19.6
-24	-23.1	-23.6
-26	-26.7	-25.3
-28	-26.5	-25.0
-30	-27.9	-26.4
-32	-29.7	-30.1
-34	-31.1	-31.1
-36	-32.8	-32.1
-38		-33.0

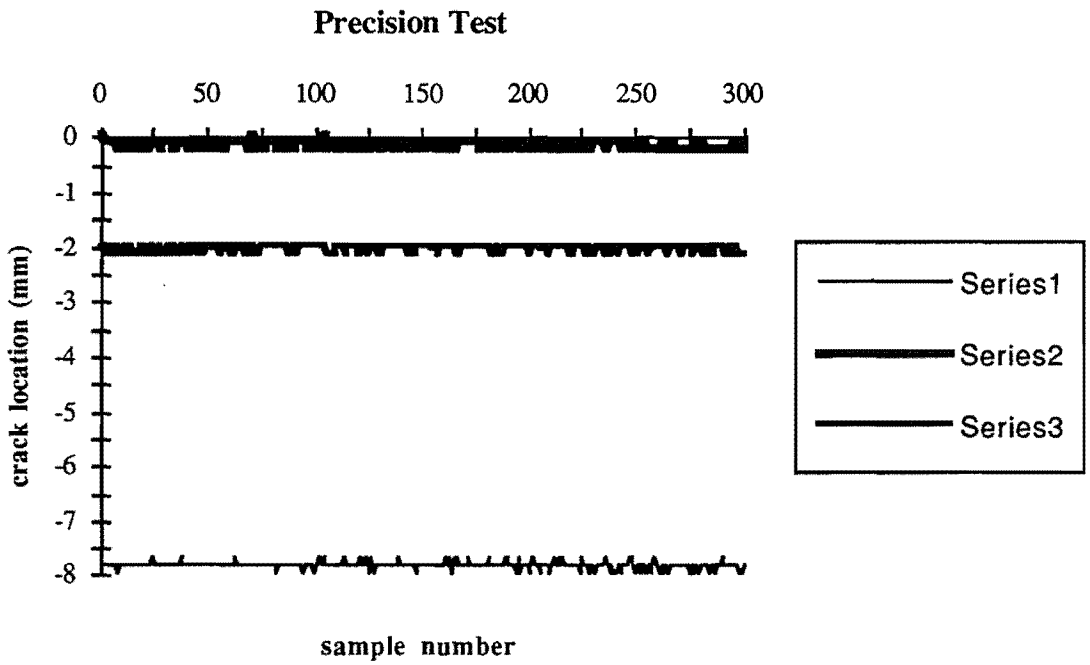


Figure 3.4 *Precision Measurement Data*

3.3.3 - Performance Test Data

The LSS was placed over a crack and crack location was recorded for 60 seconds in both sunlight and shade. This test was performed over PCC and AC pavement samples. Over 1600 pieces of data for each test were collected in a 60 second period. The data are plotted in Figure 3.5, Figure 3.6, Figure 3.7, and Figure 3.8. The x axis corresponds to the sample number, and the y axis corresponds to the crack position determined by the LSS in millimeters.

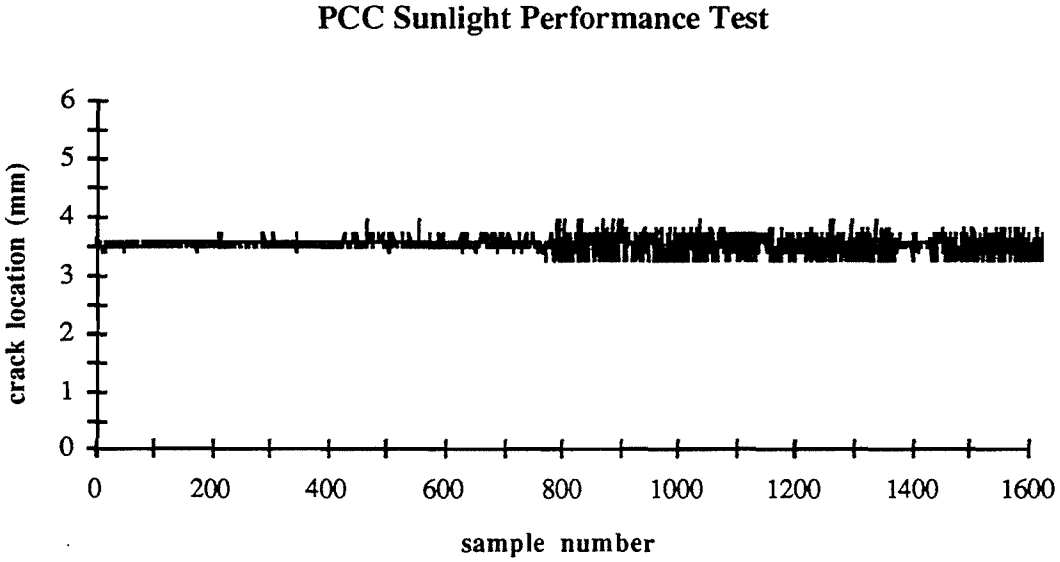


Figure 3.5 *PCC Sunlight Performance Test Data*

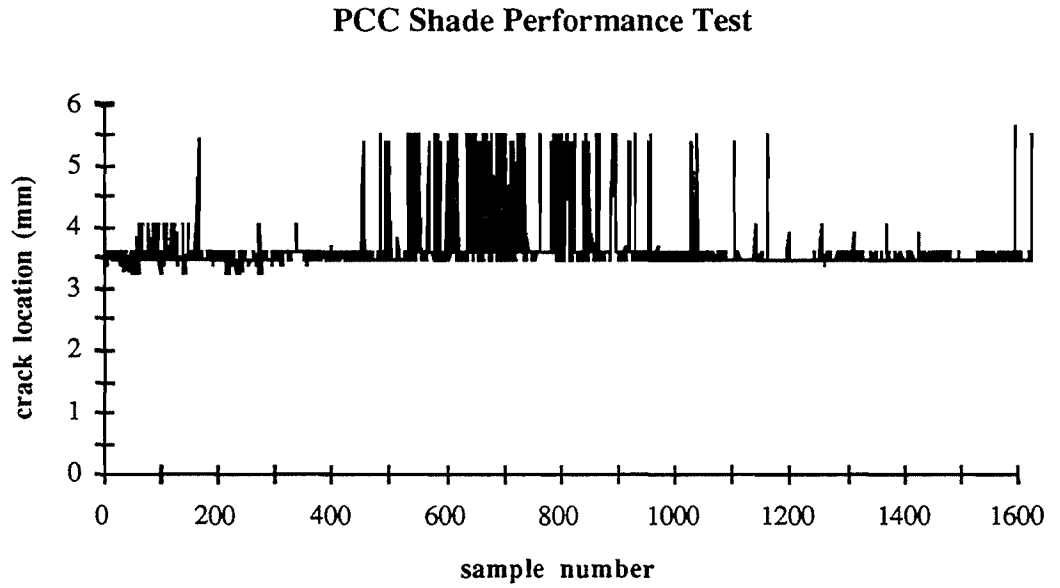


Figure 3.6 *PCC Shade Performance Test Data*

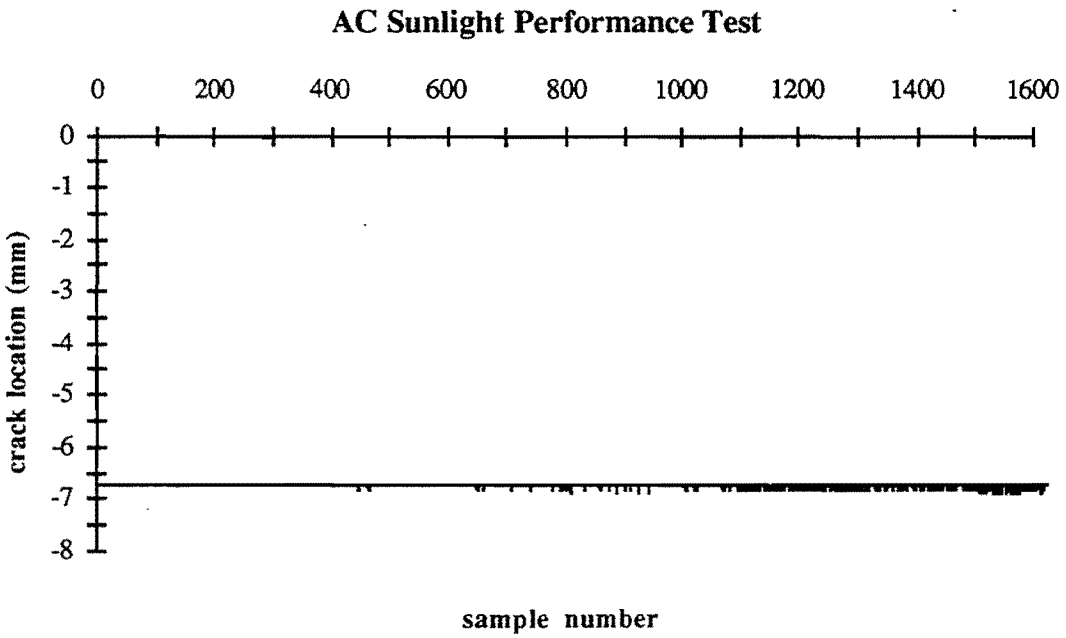


Figure 3.7 *AC Sunlight Performance Test Data*

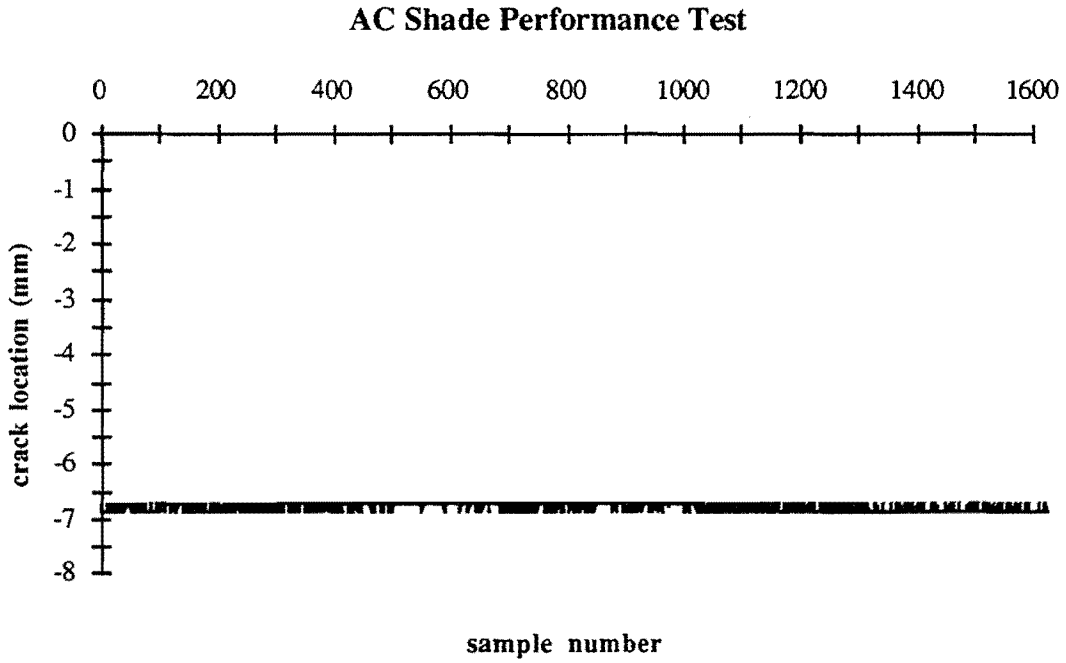


Figure 3.8 *AC Shade Performance Test Data*

3.4 - Results

3.4.1 - Time Response and Resolution

The time response of the sensor was over 27 samples/s for all tests. The requirements established in Section 2.1 specify a response time of 18 samples/s. Therefore, the LSS exceeds the requirements.

Furthermore, the resolution of the sensor is determined by the field of view. For each scan, 240 points of depth measurements are gathered. When the field of view is set to 76.2 mm, the result is 0.3175 mm. The required resolution established in Section 2.1 is 1.5875 mm, so again the performance of the sensor exceeds the requirements.

3.4.2 - Static Calibration Results

Calibration procedure consists of determining output for a given input over the range of inputs which the instrument is to be used. This has been accomplished by comparing the response of the LSS to known input. From the calibration data, a calibration curve, or a plot of the instrument output versus the input over the usable range, has been generated. A best fit line has been fit to the data through the linear region of the plot using least squares method. The calibration curve is shown in Figure 3.9.

From this curve, the sensitivity, linearity, bias, and accuracy of the LSS has been determined. Sensitivity (also referred to as gain) or the slope of the calibration curve in the linear region, was found to equal 0.95. The linear range of the sensor was found to be between -28 millimeters to +28 mm. In this region, the sensor output followed a straight line relationship to the input. It should be noted that the linear region is dependent on the field of view of the sensor and the width of the crack being sensed. See Section 3.5.1 for a discussion of the linearity of the sensor output. Bias, or systematic error, has been determined to be -0.2 mm. Finally, accuracy is the ability of the instrument to produce an output which corresponds to the true input. To quantitatively measure the accuracy, a percent error has been calculated. The value is expressed as a percent of the instruments

full scale reading, using the point which strays most from the calibration curve. Percent error was found to be 4.29%. These results are summarized in Table 3.3.

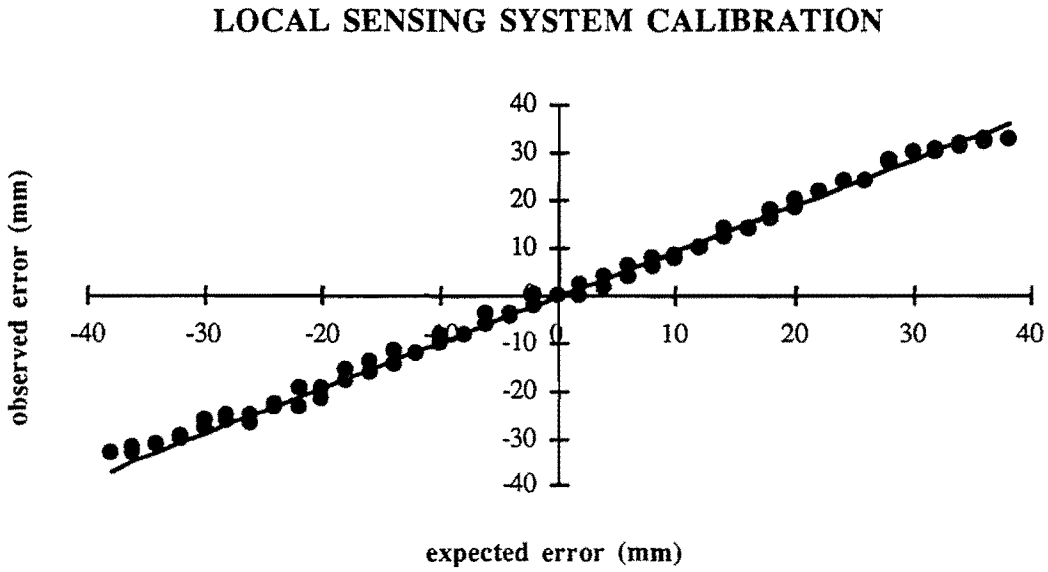


Figure 3.9 *Calibration Curve*

Table 3.3 - Static Calibration Results

SENSITIVITY	0.95920
LINEAR RANGE	-28 mm to +28 mm
BIAS	-0.24725 mm
ACCURACY (% error)	4.29% of full scale

3.4.3 - Precision Measurement Results

Precision measurement results are summarized in Table 3.4. Precision is defined as the ability of an instrument to produce consistent output for a given input. This has been determined by measuring crack location using the LSS and then statistically calculating the sampled standard deviation of the sample. A confidence level of 95% represents the

probability that 95% of the measurements will lie within a given range. The 95% confidence range corresponds to +/- 1.96 times the sampled standard deviation. The 95% confidence level has been determined to represent the precision of the LSS. Average accuracy between the three tests was found to be +/- 0.114 millimeters.

Table 3.4 - Precision Measurement Test Results

TEST NUMBER	AVERAGE	STANDARD DEVIATION	PRECISION (95% confidence)
1	-0.100873	0.059887	+/- 0.117 mm
2	-2.030466	0.055967	+/- 0.110 mm
3	-7.772348	0.059401	+/- 0.116 mm

3.4.4 - Performance Test Results

Performance test results are summarized in Table 3.5. The sensor must have the ability to distinguish between oil spots, shadows, previously sealed cracks, and actual cracks under different lighting conditions on both AC and PCC pavement. To ensure proper performance under these conditions, the test procedure in Section 3.2.3 has been performed. To determine the accuracy of the LSS under these different conditions, a 95% confidence level has been calculated by determining the sampled standard deviation, as explained in the introduction to Chapter 3. Results are summarized in Table 3.4.

In addition to these results, the sensor consistently found no crack when the crack was located over a sealed crack in both shade and sunlight on AC and PCC pavement. Furthermore, the sensor never detected a crack when no crack was present in both AC and PCC pavement in both sunlight and shade.

Table 3.5 - Performance Test Results

	AVERAGE	STANDARD DEVIATION	PRECISION (95% confidence)
AC SUN	-6.727851	0.042584	+/- 0.083 mm
AC SHADE	-6.774620	0.070319	+/- 0.138 mm
PCC SUN	3.673418	0.506216	+/- 0.992 mm
PCC SHADE	3.543648	0.133550	+/- 0.262 mm

Because the sensor location was not moved between the sunlight and shade tests for each pavement sample, the average crack location should ideally be the same for each test. For the test performed over AC pavement, the average crack location varied by 0.047 mm while the crack location varied by 0.13 mm over PCC pavement.

3.5 - Discussion

The objective of the experimental verification was to test the instrument response of the LSS. The instrument response is considered to be output produced for a given input. The input into the system is the absolute location of a crack in the surface being measured. The output of the system is the crack location generated by the LSS hardware and software. Additionally, the LSS must output a signal when no crack is located in the surface being measured. The LSS must produce viable output for all expected input under a variety of different operating conditions. This output must meet the requirements established in section 2.1.

The first step in determining the system response was to perform a static calibration on the LSS. This was accomplished by comparing the response of the LSS to known input. From the calibration data, a calibration curve, or a plot of the instrument output

versus the input over the usable range, was generated. A best fit line was fit through the linear region of the plot using least squares method. This curve is plotted in Figure 3.9 From this curve, the sensitivity, linearity, bias, and accuracy of the LSS has been determined. A problem arose in that the known input is considered to be the true input, while in reality it is affected by the response of the measuring instrument which is determining the input. In the calibration test, an Adept robot arm was used to position the sensor over the known location of the crack. The accuracy of the robot arm position is +/- 1^{o} for each joint, and the corresponding arm lengths are 55.88 cm and 50.8 cm (Adept Technologies Incorporated, 1991). This corresponds to a worst case error of +/-25.4 mm when the joint angles are at 90^{o}. The percent error determined from the calibration procedure of 4.29% is considered to be primarily from the inaccuracies of the robot positioning system rather than the LSS. This conclusion is supported by the results of the precision measurement found in Section 3.4.2. The precision measurement results indicated that 95% of the crack location measurements should fall within +/-0.223 mm of the actual input. As seen by the calibration data in Table 3.2, the calibration data varied by a maximum of 2.3 mm. However, a 2.3 mm inaccuracy is still acceptable for the configuration of the Sealant Applicator System. As seen in Table 2.1, the sensor requirements specify a 3.175 mm allowable inaccuracy, so the LSS accuracy easily falls within the requirements established to achieve a proper seal.

Further error is caused by the fact that it is not physically possible to place the sensor exactly over the center of the crack at the start of the calibration procedure. This accounts for the bias error determined during the calibration procedure.

Ideally, the sensitivity of the LSS should be 1. However, the calibration curve revealed a sensitivity of 0.96. If a crack appeared at the end of the field of view of the sensor, a 1 to 1 relationship between the input and output would result in a crack location of 50.8 mm from the sensor (assuming a 101.6 mm field of view). A sensitivity of 0.96

rather than 1 would produce a crack location of 48.77 mm from the sensor. The 2.03 mm offset is negligible.

The results of the calibration curve showed a linear region of the input/output relationship to fall between +28 mm and -28 mm. The reason for the non-linear region is in the software. As the crack approaches the end of the field of view of the sensor, eventually the sensor will see the beginning of the crack but not the end of the crack. In software, the sensing system assumes that the end of the crack occurs at the end of the field of view, and calculates the width of the crack with this assumption. When less than 3.175 mm of the crack is contained in the field of view of the sensor, the LSS considers the crack to be too small to be sealed. Disadvantages in this algorithm are the inherent non-linearity which is produced at the ends of the field of view, and the fact that if a crack which is too large to seal only partially appears in the field of view of the sensor, the sensing system will not detect its entire width and will allow this crack to be sealed. Alternatively, when a crack only partially appears in the field of view of the sensor, the sensing system could ignore this crack because it cannot determine accurate crack location information. The effect of this would be to significantly reduce the usable portion of the field of view. For a 25.4 mm crack, the first and last 25.4 mm of the field of view would be ineffective. It has been decided that the disadvantages associated with a reduced field of view are more detrimental than the disadvantages associated with non-linearities and inability of locating cracks which are too large.

During the calibration procedure, a crack which was 11.11 mm wide was being measured. When 3.175 mm of the crack remained in the field of view of the sensor, the LSS determined the crack location to be 1.588 mm from the edge of the field of view within the field of view. However, the center of the crack was actually 2.38 mm outside of the field of view. This corresponds to a 3.97 mm error, which is apparent in the data gathered during the calibration procedure.

Maximum error due to the non-linearity would occur for the widest crack of 25.4 mm. When 3.175 mm of the crack remains in the field of view of the sensor, the crack position determined by the LSS will be 11.11 mm off from the actual position. This error is still within the accepted accuracy of the LSS to achieve a proper seal.

During the performance tests, the crack location was sensed in both sunlight and shade over both PCC and AC pavement samples. The accuracies produced for the four tests, found in Table 3.4, all lie within acceptable tolerances of the system.

Furthermore, the LSS never mistook a sealed crack or no crack to be a crack in the pavement. This was true over both AC and PCC pavement samples in both sunlight and shade.

Both the resolution and time response of the sensor far exceeded the requirements established in Section 2.1. Through initial integration with the Robot Positioning System, it became apparent that further time delays were associated with the communications between the two systems. Originally, the RPS requested the LSS to send a new error signal each time a new error was calculated. The baud of the serial port was set to 4800 . Since 8 bits of data, 1 parity bit, and 1 stop bit are sent for each error signal, this corresponds to 10 bits of data for each error signal transmission. Therefore, the delay caused by serial transmission was 1/480 of a second, or 2 ms. This is negligible. However, the LSS was operating at a faster rate than the RPS, and eventually the RPS buffers became filled and the RPS was no longer reading the most current error signal. Once the buffers were filled, new information sent by the LSS was discarded. This created many problems. Therefore, a more complicated protocol was developed in which the RPS would send a byte over the serial line to the LSS when it was ready for a new error signal. When the LSS determined a new error signal, it first checked to see if the RPS had requested information. If it had requested information, the information was sent. If the RPS was not ready for new information, the LSS would loop through another scan, and would check again when the next error signal was calculated. In this case, there are 2

pieces of information being sent over the serial line (one from the RPS to the LSS and one from the LSS to the RPS), which causes more delay. Furthermore, in the worst case situation, immediately after the LSS checks to see if the RPS has requested information, the RPS sends a request bit. In this situation, the LSS will perform another loop before it checks again for the request signal, causing another 1/27 second delay. At this point, the resulting response time of the system no longer meets the 18 Hz requirement. A suggestion for the solution of this problem is contained in Section 4.2.

CHAPTER 4 - CONCLUSIONS AND RECOMMENDATIONS

4.1 - Conclusions

The Local Sensing System has consistently produced reliable and accurate results under a variety of operating conditions which fall well within the specifications established in Section 2.1. During the static calibration procedure, the accuracy, bias, sensitivity, and linearity of the LSS were determined.

The accuracy of the LSS was found to be 4.42% of full scale. It is believed that the majority of the inaccuracy is due to the robot positioning system which determined the input to the system. This known input had a possible error of up to +/-25.4 mm. Still, the 4.42% error easily lies within the +/-50.8 mm accuracy required to achieve a proper seal.

The bias of the LSS was found to be -0.25 mm. It is believed that this error is due to the fact that the sensor was not accurately placed over the center of the crack at the beginning of the procedure. This would account for an offset which is constant to all data collected. However, this bias is negligible compared with the system requirements established in Section 2.1.

The sensitivity of the sensing system was found to be 0.96, rather than a 1 to 1 relationship between the input and output. This causes a maximum error of 2.03 mm at the end of the field of view, which is also negligible compared with the system requirements established in Section 2.1.

Finally, the maximum error due to non-linearity in the LSS was found to be 11.18 mm. This would occur when a crack of 25.4 mm in width appears in the edge of the field of view of the sensor. This error is still well within the +/-50.8 mm accuracy necessary to achieve a proper seal.

The precision of the LSS was determined by taking a sample of data and determining the sample standard deviation. 95% of the data is contained within 1.96 times the sampled standard deviation. This 95% confidence was taken to be an indication of the precision of

the LSS. The precision measurement results indicated that 95% of the crack location measurements should fall within +/-0.223 mm of the actual input. These results fall within the requirements established in Section 2.1.

The performance tests proved that the sensor produces reliable and accurate results under a variety of testing configurations which are expected to occur on the actual machine. The LSS never found a crack when no crack was present. Furthermore, the precision of the output over AC and PCC in both sunlight and shade fell within the requirements established in Section 2.1

4.2 - Recommendations

The sensor which was selected for the crack sealing machine was chosen because it was the only commercially available sensor which met the requirements established in Section 2.1. However, a sensor better fit for this application could be custom developed. This would require a substantial amount of money and time, and would not be appropriate unless multiple machines were to be built.

The custom sensor would have a larger field of view. Through testing of the sensor, optimal performance was achieved when the field of view was 76.2 mm or less. To increase the operating speed of the crack sealing vehicle, a larger field of view would be necessary so that the sensor did not loose sight of the crack. Furthermore, the purchased sensor has many characteristics which are not necessary for this application which the custom sensor would not require. For example, the resolution and accuracy of the purchased sensor far exceed the requirements. By removing unnecessary functions of the sensor, it may be possible to increase the response time of the sensor.

Some cracks contain vegetation, and may go unnoticed by the current crack detection algorithm. Two possibilities of recognizing cracks filled with vegetation are recommended for investigation. The first possible method of detecting cracks filled with vegetation would be to perform a histogram on the depth measurements gathered in one scan. By

performing this histogram, it should be possible to immediately classify the scanned pavement as containing no crack, containing a clean crack, or containing a crack filled with vegetation. The second method of detecting the presence of a crack filled with vegetation would be to perform a histogram on the intensity of light reflected back into the sensor. It is expected that the intensity of reflected light over vegetation would be significantly different than the intensity of reflected light when no vegetation was present. By performing the intensity histogram, it should also be possible to classify the scanned pavement as containing no crack, containing a clean crack, or containing a crack filled with vegetation.

Once the type of crack (clean crack or filled crack) has been identified, the algorithm would jump to the appropriate function which would find the crack. The method of finding the location of a clean crack would remain the same as the existing program which has been developed and tested. The method of finding the location of a filled crack could be performed by either analyzing the depth measurements or the intensity of light at each measurement, and again noting that the intensity of reflected light off vegetation should be significantly different than other reflected light.

The current program is written in such a manner that this modification would be minor. The program first jumps to a function which determines crack type. In the current configuration, the function always sets the crack type to be clean. In the recommended alterations would contain appropriate code to distinguish between clean and filled cracks, and this function would return the crack type. Once the crack type has been determined, the program would jump to the appropriate function to locate the crack position. Currently, there is a function called find_clean_crack and a function called find_filled_crack. The program always jumps to the find_clean_crack function, and the find_filled_crack function has not been developed. The last recommendation is associated with the communication between the RPS and LSS. As discussed in Section 3.5, communication between the two systems significantly slowed the effective rate at which the RPS was receiving new

information. The reason behind this is the fact that the operating time of the RPS is slower than that of the LSS. In the current design, a protocol was developed in which the RPS sends a byte over the serial line to the LSS when it was ready for a new error signal. When the LSS determines a new error signal, it first checks to see if the RPS has requested information. If it has requested information, the information is sent. If the RPS is not ready for new information, the LSS loops through another scan, and does not poll the RPS again until it finishes the loop and determines a new error signal. In this case, there are 2 pieces of information being sent over the serial line (one from the RPS to the LSS and one from the LSS to the RPS), which causes more delay. Furthermore, in the worst case situation, immediately after the LSS checks to see if the RPS has requested information, the RPS sends a request bit. In this situation, the LSS will perform another loop before it checks again for the request signal, causing another 1/27 second delay. At this point, the resulting response time of the system no longer meets the 18 Hz requirement. One simple solution would be to have the LSS poll the RPS periodically rather than only one time each loop.

However, a better solution would be to allow the RPS to create a hardware interrupt in the LSS when it requests information. In this case, the RPS would send one bit of data to the LSS. This bit would be connected to the interrupt line of the microcomputer. When a hardware interrupt occurs, the current program pointer gets stored on the stack, and the program is halted and the interrupt is serviced. Once the interrupt is serviced, the program will resume execution at the place in which it was halted. To create the interrupt service routine, the interrupt vector table in the microcomputer would need to be modified to include the pointer to the interrupt service routine. The interrupt service routine would send the most recent error signal calculated by the crack locating algorithm to the RPS.

Further increases in response time can be made by developing driver routines to communicate serially with the RPS. Currently, DOS functions are being used to perform serial communication. Reading information from the serial port with the DOS functions is

limited to 300 baud. This rate was unacceptable. To increase this rate, a serial communication package is initially installed on the LSS computer before the LSS program begins. This package causes signals sent over the serial port to be stored in memory. By using the DOS functions, the commands now access this memory and it is possible to increase transmission to an acceptable rate. A better solution to this problem would be to develop or buy software functions which directly communicate with the serial communications board. This modification would be minor to the current program. Currently, the program includes a function called send_offset which sends the error signal to the RPS using the DOS functions. This function would need to be altered so that it calls the new functions rather than the DOS functions. Also the function which initializes the serial port would need similar modification. The current program was written in a modular style so that these alterations could be easily accomplished.

REFERENCES

Adept Technologies Incorporated (1991) *V/V+ Reference Guide*, Version 10.1.

Agapakis, J.E., and Katz, J. (1989) "Vision-Guided System Tracks Welding Seams", *Robotics World*, Vol.7, No.2, p. 23-5.

Appels, J.A.C. (1987) "Application and Economic Aspects of a 3-Dimensional HeNe Laser Sensor", *Automated and Robotic Welding*, paper 47, p. 223-231.

Bamba, T., Muruyama, H., Kodaira, N., and Tsuda, E. (1984) "A Visual Seam Tracking System for Arc-Welding Robots", *Proceedings of the 14th International Symposium on Industrial Robots*, p. 365-74.

Beranek, B., Boillot, J.-P., and Ferrie, F.P. (1986) "Laser Sensor for Adaptive Welding.", *Proc. SPIE Int. Soc. Opt. Eng.*, Vol.665, p. 195-9.

Browne, A., and Norton-Wayne, L. (1986) *Vision and Information Processing for Automation*, Plenum Press, New York, p. 212-215, 242-243.

Brunet, R., Turon, P., and Lacoste, F. (1986) "Laser Dimensional Measurement Sensor Applied to Robot Welding", *Proceedings of the 6th International Conference on Robot Vision and Sensory Controls: RoViSeC 6*, p. 229-33.

Case, S., Jalkio, J., and Kim, R. (1987), "3-D Vision System Analysis and Design", *Three-Dimensional Machine Vision*, Kluwer Academic Publishers, Boston, p. 63-95.

Davies, E.R. (1990), *Machine Vision: Theory, Algorithms, Practicalities*, Academic Press, Harcourt Brace Jovanovich, Publishers, Boston, p. 387-389.

Doebelin, E.O. (1990), *Measurement Systems: Application and Design*, Fourth Edition, McGraw-Hill Publishing Company, New York, p. 38-93, 268-286, 777-781.

Edling, G. (1986) "New Generation of Seam Tracking Systems (Arc Welding).", *Changing Face of Manufacturing, Proceedings of the 8th Annual British Robot Association Conference*, p. 5-10.

Egharevba, F, Fenn, R., and Siores, E. (1987) "Adaptive Control in Arc Welding Utilising Ultrasonic Sensors", *Automated and Robotic Welding*, paper 8, p. 115-24.

Estochen, E. L., Neuman, C. P., and Prinz, F. B. (1984) "Application of Acoustic Sensors to Robotic Seam Tracking", *IEEE Trans. Industrial Electronics*, Vol. IE-31, No. 3, p. 219-224.

Fukuhara, T. Terada, K, Nagao, M. Kasahara, A, and Ichihashi, S. (1990), "Automatic Pavement-Distress-Survey System", *Journal of Transportation Engineering*, Vol. 116, No. 2, p. 280-286.

Haas, C. and Hendrickson, C. (1990) "A Model of Pavement Surfaces", Dept. of Civil Engineering, Carnegie Mellon University, 106 p.

Hendrickson, C., McNeil, S., Bullock, D., Haas, C., Peters, D., Grove, D., Kenneally, K., and Wichman, S., "Perception and Control for Automated Pavement Crack Sealing", *Applications of Advanced Technologies in Transportation Engineering*, p. 66-70.

Jenkins, T.E. (1987), *Optical Sensing Techniques and Signal Processing*, Prentice Hall International, New Jersey, p. 74-87.

Jing, L, Kirschke, K., Quinlan, T., Schultheis, E., Smith, M., Velinsky, S. (1990) "On The Automatic Sealing of Cracks in Pavement", Manufacturing Automation and Productivity Program Technical Report #90-03-01 University of California, Davis.

Kanade, T., and Fuhrman, M. (1987), "A Noncontact Optical Proximity Sensor for Measuring Surface Shape", *Three-Dimensional Machine Vision*, Kluwer Academic Publishers, Boston, p. 151-192.

Kirschke, K. R. and Velinsky, S. A. (1992) "A Histogram Based Approach for Automated Pavement Crack Sensing", *Journal of Transportation Engineering*, Vol. 118, in press.

Mundy, J.L., and Porter III, G.B. (1987), "A Three-Dimensional Sensor Based on Structured Light", *Three-Dimensional Machine Vision*, Kluwer Academic Publishers, Boston, p. 3-61.

Nachtigal, C.L., and Stevenson, W.H. (1986), *Modeling and Instrumentation for Physical Systems, ME 385 Course Notes,* Purdue University, West Lafayette, IN.

Nayak, N., Ray, A., and Vavreck, A.N. (1987) "An Adaptive Real-Time Intelligent Seam Tracking System.", *Journal of Manufacturing Systems*, Vol.6, No.3, p. 241-524.

Ogilvie, G.J., and Zemancheff, P.A. (1983) "An optical sensing system.", *Australian Welding Institute 31st Annual Conference WELCOM 83*, p. 1-5 supl. of 205+62 p.

Peshkin, D., Smith, K., and Johnson, K. (1991), "Toward the Development of Functional and Operational Characteristics of Automated Pavement Sealing Equipment", ERES Consultants, Inc., 8 p.

Phillips, C.L., Nagle, H.T. (1990), *Digital Control System Analysis and Design*, Prentice Hall, New Jersey, p. 25-70, 400-456.

Ravani, B. and West, T.H. (1991), "Applications of Robotics and Automation in Highway Maintenance Operations", *Proceedings of the 2nd ASCE International Conference on Applications of Advanced Technologies in Transportation Engineering*, p. 61-65.

Rioux, Marc (1984), "Laser Range Finder Based on Synchronized Scanners", *Applied Optics*, Vol. 23 No. 23, p. 3837-3844.

Ritchie, Stephen G. (1990), "Digital Imaging Concepts and Applications in Pavement Management", *Journal of Transportation Engineering*, Vol. 116, No. 2, p. 287-298.

Skibniewski, M. and Hendrickson, C. (1990), "Automation and Robotics for Road Construction and Maintenance", *Journal of Transportation Engineering*, Vol. 116, No. 2, p. 261-271.

Sugihara, K. (1987), "Use of Vertex-Type Knowledge for Range Data Analysis", *Three-Dimensional Machine Vision*, Kluwer Academic Publishers, Boston, p. 267-298.

Smati, Z., Smith, C.J., and Yapp, D. (1983), "An industrial robot using sensory feedback for an automatic multipass welding system", *Proceedings of the 6th British Robot Association Annual Conference*, p. 91-100.

Tan, C. and Lucas, J. (1986), "Low Cost Sensors for Seam Tracking in Arc Welding", *Computer Technology in Welding*, paper 42, p 69-76.

Thomas, B.S. (1986), "Optical seam trackers: tough requirements for design and laser safety.", *Robotics Eng.*, Vol.8, No.6, p. 8-11.

Velinsky, S.A. and Kirschke, K.R. (1991) "Design Considerations for Automated Pavement Crack Sealing Machinery", *Proceedings of the 2nd ASCE International Conference on Applications of Advanced Technologies in Transportation Engineering*, p. 76-80.

Wigan, M.R. (1992), "Image-Processing Techniques Applied to Road Problems", *Journal of Transportation Engineering*, Vol. 118, No. 1, p. 62-83.

APPENDIX A - FILTER DESIGN

In order to improve the performance of the crack locating program, the data collected by the local sensor is put through a digital low pass filter. Data consists of depth information creating a cross-sectional profile of the surface being measured. Normal pavement will contain high frequency measurements corresponding to normal variances in pavement surface and low frequency components corresponding to the presence of a crack. Through filtering the data, variances in the pavement will be removed, while crack information will remain.

The digital filter design was modeled after a second order low pass Butterworth filter. A second order filter was selected for its sharp cutoff (-40 dB/decade). A Butterworth filter has a quality factor Q of 0.707, which corresponds to a gain of -3 dB at the cutoff frequency (Doebelin 1990). This type of filter was selected for this characteristic. An overshoot at the cutoff frequency is undesirable because positive gain around the cutoff frequency could cause erroneous results. A gain of -3 dB will produce reliable output from the filter.

A Butterworth filter has the Laplace transform of

$$H(s) = \frac{\omega_o^2}{s^2 + 2\zeta\omega_o s + \omega_o^2} \qquad (A.1)$$

where

ω_o = cutoff frequency,

ζ = damping ratio, and

H(s) = transfer function of filter.

For a Butterworth filter, the damping ratio is set by the quality factor. The damping ratio is determined by

$$Q = 0.707 = \frac{1}{2\zeta} \tag{A.2}$$

$$\zeta = 0.707 \tag{A.3}$$

where

Q = quality factor.

In order to realize this filter into the digital domain, a numerical integration method was selected (Phillips et al. 1990). Furthermore, a trapezoidal integration approximation was selected. To determine the z transform of the filter, a mapping from s to z was performed as follows:

$$H(z) = H(s)\Big|_{s=\frac{2(z-1)}{t(z+1)}} \tag{A.4}$$

$$H(z) = \frac{\omega_o^2}{\frac{4(z-1)^2}{T^2(z+1)^2} + \frac{4\zeta\omega_o(z-1)}{T(z+1)} + \omega_o^2} \tag{A.5}$$

where

T = sampling interval.

Standard format for the z transform is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_1 + b_2 z^{-1} + b_3 z^{-2}}{1 + a_2 z^{-1} + a_3 z^{-2}} \tag{A.6}$$

where

Y(z) = filtered output,

X(z) = unfiltered input,

b_1, b_2, b_3 = filter constants, and

a_2, a_3 = filter constants.

Converting this transfer function to the digital time domain reveals

$$Y(z)[1 + a_2 z^{-1} + a_3 z^{-2}] = X(z)[b_1 + b_2 z^{-1} + b_3 z^{-2}] \quad \text{(A.7)}$$

$$y_k = b_1 x_k + b_2 x_{k-1} + b_3 x_{k-2} - a_2 y_{k-1} + a_3 y_{k-2} \quad \text{(A.8)}$$

where

$y_i = i^{th}$ filtered output,

$x_i = i^{th}$ unfiltered input.

Rearranging H(z), the filter constants are found to be

$$b_1 = \frac{T^2\omega_o^2}{4 + 4T\zeta\omega_o + T^2\omega_o^2} \quad \text{(A.9)}$$

$$b_2 = \frac{2T^2\omega_o^2}{4 + 4T\zeta\omega_o + T^2\omega_o^2} \quad \text{(A.10)}$$

$$b_3 = \frac{T^2\omega_o^2}{4 + 4T\zeta\omega_o + T^2\omega_o^2} \quad \text{(A.11)}$$

$$a_2 = \frac{2T^2\omega_o^2 - 8}{4 + 4T\zeta\omega_o + T^2\omega_o^2} \quad \text{(A.12)}$$

$$a_3 = \frac{4 - 4T\zeta\omega_o + T^2\omega_o^2}{4 + 4T\zeta\omega_o + T^2\omega_o^2} \quad \text{(A.13)}$$

In order to determine the cutoff frequency, a discrete Fourier transform (DFT) was performed on profile data collected by the local sensor. The DFT algorithm used is found in Appendix B. A plot of the DFT output is shown in Figure A.1. From this plot, it is apparent that there are low frequency components due to the presence of a crack and higher

frequency components due to normal variances in the pavement. The pertinent information exists below 0.24 mm^{-1} (1.51 rad/mm), so this value was selected as the cutoff frequency.

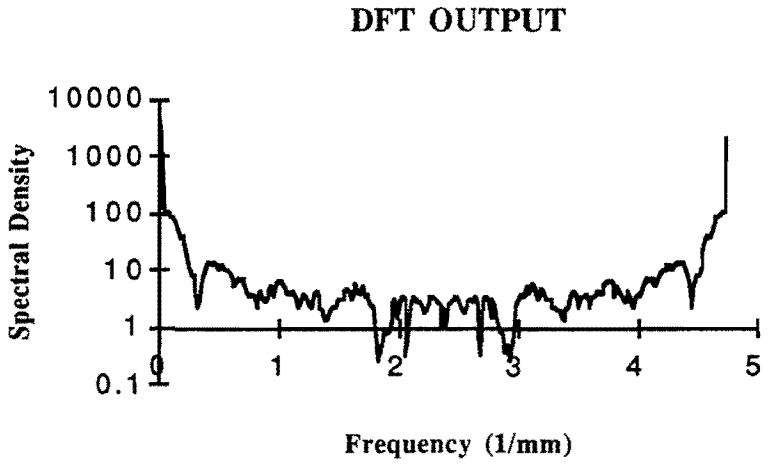


Figure A.1 *DFT Output*

A typical sampling of data from the local sensor has a period of approximately 0.21 mm. Using T = 0.21 and $\omega_0 = 0.24$, the filter constants are

$b_1 = 0.0201$,

$b_2 = 0.0402$,

$b_3 = 0.0201$,

$a_2 = -1.5610$, and

$a_3 = 0.6414$.

An unfiltered profile produced by data from the local sensor is shown in Figure A.2. This same data was input to the filter described above, and produced the profile shown in Figure A.3.

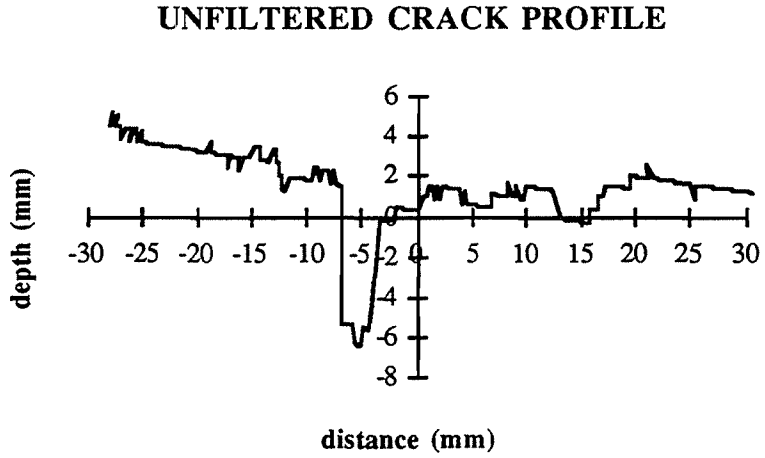


Figure A.2 *Unfiltered crack profile*

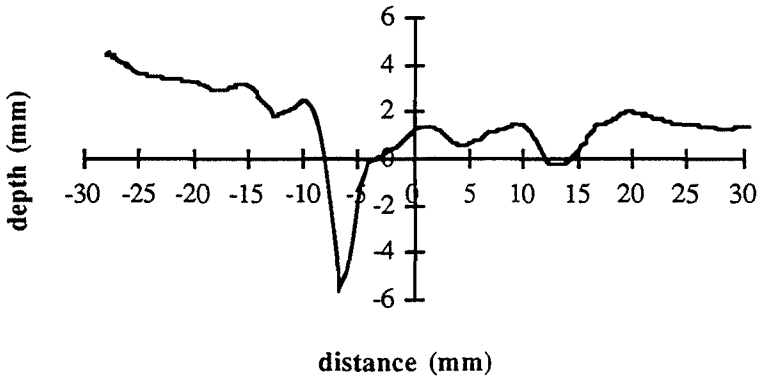


Figure A.3 *Filtered crack profile*

A problem associated with this filtering routine is the fact that there are no previous measurements for the filtering routine when determining the first points. To remedy this problem, the filter values are initialized to determine the first points. The previous points needed in the filtering routine are initially set to the average height from the sensor to the surface being measured. This value is determined during the initialization procedure in the crack locating program.

APPENDIX B - DFT ALGORITHM

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>

/*****************************************************************/
/*      Debbie Krulewich                                         */
/*March 9, 1992                                                  */
/*      dft.c                                                    */
/*      This program reads in the data output from the local sensing system and  */
/*      then then analyzes the frequency content of the input signal by discrete */
/*      Fourier transform computed by a fast Fourier transform algorithm.        */
/*****************************************************************/

#define REC_LEN 1024            /* record length        */
#define TIME 0                  /* place holder in wave array storing time     */
#define VOLTAGE 1               /* place holder in wave array storing voltage  */
#define MAX 50                  /* maximum size of character arrays  */
#define M 8                 /* index 2**M = number of samples  */
#define PI 3.14159265

/*      structure declarations                                   */
struct cmplx                /* stores imaginary and real parts of complex #
        */
{
        double real;
        double imag;
};
struct dft                  /* stores mag and freq of discrete Fourier transform  */
{
        float mag;
        float freq;
};

struct dat
{
        float dist;
        float time;
};

/*      function declarations                                    */
void get_file_name(char data_file[]);
void read_data(char data_file[], struct dat meas[]);
void store_wave(struct dat meas[], struct cmplx wave[]);
float convert_time(struct dat meas[]);
void store(struct dft output[]);
void fft(struct cmplx wave[], int m, int n);
```

```
struct cmplx mult(struct cmplx a, struct cmplx b);
struct cmplx add(struct cmplx a, struct cmplx b);
struct cmplx subt(struct cmplx a, struct cmplx b);
void spectrum(struct cmplx wave[], int n, struct dft out_data[], float t);
void window(int N, struct cmplx wave[]);

void main(void)
{
        static struct cmplx wave[REC_LEN];          /* stores voltage of input wave
                */
        static struct dft out_data[REC_LEN];        /* output from dft
        */
        char data_file[MAX];                /* stores name of input file          */
        static struct dat meas[REC_LEN];            /* stores measurements from file
                */
        float T;                                    /* time period between samples
                */

        get_file_name(data_file);           /* gets input file name
        */

/*      reads data from input file                                                 */
        read_data(data_file, meas);

/*    stores input data in wave*/
        printf("\nStoring wave");
        store_wave(meas, wave);

/*      determines corresponding time of each voltage measurement
        */
        printf("\nFinding T");
        T = convert_time(meas);

/*    perform windowing*/
        printf("\nWindowing");
        window(REC_LEN, wave);

/*    perform fft*/
        printf("\nPerforming fft");
        fft(wave, M, REC_LEN);

/*      determine spectrum frequency and magnitude
        */
        printf("\nDetermining spectrum");
        spectrum(wave, REC_LEN, out_data, T);

/*      stores spectrum data to disk                                               */
        printf("\nStoring data to disk");
        store(out_data);
```

```
}

void get_file_name(char data_file[])
/* Prompts the user for the input file name and stores the name in data_file  */
{
        printf("Enter the name of the waveform text file>>\n\t");
        scanf("%s", data_file);
}

void read_data(char data_file[], struct dat meas[])
/* Opens the input file and reads data from file.
        */
{
        FILE *data_ptr;
        int i;
        char string1[MAX];
        char string2[MAX];

        data_ptr = fopen(data_file, "r");       /* sets pointer to input file
        */

/*      if fopen returns a NULL then there was an error opening the
        */
/*      file and the program is exited                                          */
        if (data_ptr == (FILE *)NULL)
        {
                printf("ERROR OPENING %s", data_file);
                exit(-1);
        }

/*    gets measurements from input file */
        for (i = 0; i < REC_LEN; i++)
        {
                fscanf(data_ptr, "%s%s", string1, string2);
                meas[i].time = atof(string1);
                meas[i].dist = atof(string2);
                printf("\ni=%d\tdist=%f\ttime=%f",i,meas[i].dist,meas[i].time);
        }
}

void store_wave(struct dat meas[], struct cmplx wave[])
/* calculates voltage from measurement data in input file
        */
{
        int i;

        for (i = 0; i < REC_LEN; i++)
        {
                wave[i].real = meas[i].dist;
```

```
                wave[i].imag = 0;
        }
}

float convert_time(struct dat meas[])
/* determines time period between each sample
        */
{
        float T;

        T = abs(meas[0].time - meas[1].time);
        return T;
}

void store(struct dft out_data[])
/* stores magnitude and frequency from dft to disk                              */
{
        FILE *store_ptr;
        char store_file[MAX];
        int i;

/*      prompt the user for the name of the output file
        */
        printf("Enter the name of the storage file>>\n\t");
        scanf("%s", store_file);

/*      set pointer to output file
        */
        store_ptr = fopen(store_file, "w");

/*      if fopen returns a NULL then there was an error opening the
        */
/*      file and the program is exited                                          */
        if (store_ptr == (FILE *)NULL)
        {
                printf("ERROR OPENING %s", store_file);
                exit(-1);
        }
/*      write wave data to output file                                          */
        for (i = 0; i < REC_LEN; i++)
                fprintf(store_ptr, "%f\t%f\n", out_data[i].mag, out_data[i].freq);
}


void fft(struct cmplx A[], int m, int N)
/* FFT subroutine, an efficient numerical technique for calculating the DFT of a sequence
        */
{
        struct cmplx U, W, T;
```

```
        int j = 1;
        int i, k, L, LE, LE1, ip;

        for (i = 1; i < N; i++)
        {
                if (i < j)
                {
                        T = A[j-1];
                        A[j-1] = A[i-1];
                        A[i-1] = T;
                }
                k = N/2;
                while (k < j)
                {
                        j -= k;
                        k = k/2;
                }
                j += k;
        }
        for (L = 1; L <= m; L++)
        {
                LE = pow((double)2, (double)L);
                LE1 = LE/2;
                U.real = 1.0;
                U.imag = 0.0;
                W.real = cos((double)(PI/LE1));
                W.imag = sin((double)(PI/LE1));
                for (j = 1; j <= LE1; j++)
                {
                        for (i = j; i <= N; i+=LE)
                        {
                                ip = i + LE1;
                                T = mult(A[ip-1], U);
                                A[ip-1] = subt(A[i-1], T);
                                A[i-1] = add(A[i-1], T);
                        }
                        U = mult(U, W);
                }
        }
}

struct cmplx mult(struct cmplx a, struct cmplx b)
/* Multiplies two complex numbers together                                    */
{
        struct cmplx result;                        /* stores result complex number
        */
        result.real = (a.real * b.real) - (a.imag * b.imag);
        result.imag = (a.real * b.imag) + (a.imag * b.real);
        return result;
```

```
}

struct cmplx add(struct cmplx a, struct cmplx b)
/* Adds two complex numbers together
        */
{
        struct cmplx result;                    /* stores result complex number
        */
        result.real = a.real + b.real;
        result.imag = a.imag + b.imag;
        return result;
}

struct cmplx subt(struct cmplx a, struct cmplx b)
/* Subtracts two complex numbers                                        */
{
        struct cmplx result;                    /* stores result complex number
        */
        result.real = a.real - b.real;
        result.imag = a.imag - b.imag;
        return result;
}

void spectrum(struct cmplx wave[], int N, struct dft out_data[], float T)
/* Calculates the magnitude at each frequency sample and stores the frequency and */
/* spectrum magnitude in an output data file                            */
{
        int k;
        float temp;
        for (k = 0; k < N; k++)
        {
                temp = (pow(wave[k].real, (double)2)) + (pow(wave[k].imag, (double)2));
                out_data[k].mag = sqrt((double)temp);
                out_data[k].freq = (float)k/((float)N * T);
        }
}

void window(int N, struct cmplx wave[])
/* "windows" the sampled data so that the distortion due to abrupt transitions at the start
        */
/* and end of the sampled data are minimized.  The effect of the windowing is to cause the
        */
/* tails of the sequence to gradually taper off.  A Hamming window if used here.
        */
{
        int n;
        for (n = 0; n < N; n++)
                wave[n].real = wave[n].real * (0.54 - 0.46 * cos((double)((2*PI*n)/(N-
1))));
```

```
}
```

APPENDIX C - CRACK IDENTIFICATION PROGRAM

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <signal.h>
#include <time.h>
#include <graph.h>
#include <bios.h>
#include <dos.h>

#include "calib8.h"
#include "profile.h"

#define TOO_SMALL       1.0     /* width of crack in mm which is too small to seal */
#define TOO_BIG         50.0    /* width of crack in mm which is too large to seal */
#define N               25
#define NOT_FOUND       0
#define FOUND           1
#define CLEAN           0
#define VEGETATION      1
#define YES         1
#define NO              0
#define INTENSITY       9       /* intensity of laser (1-9) */
#define START_PROG      1
#define END_PROG        0
#define FS_MM           101.6
#define NUM_PTS         nb_line_field   /* number of points in each scan line */
#define FS_BITS         254.0           /* full scale number of bits */
#define SEND_DATA       26
#define NO_CRACK        1000.0

/* define serial communication constants */
#define WORD_LENGTH     _COM_CHR8           /* 8 bits per character */
                                            /* _COM_CHR7 for 7 bits per
character */
#define STOP_BITS       _COM_STOP1          /* 1 stop bit */
                                            /* _COM_STOP2 for 2 stop bits */
#define PARITY          _COM_ODDPARITY /* odd parity */
                                            /* _COM_EVENPARITY for even
                                            parity */
                                            /* _NO_PARITY for no parity */
#define BAUD_RATE       _COM_4800  /*4800 baud */
                        /*_COM_110 for 110 baud */
                        /*_COM_150 for 150 baud */
                        /*_COM_300 for 300 baud */
                        /*_COM_600 for 600 baud */
                        /*_COM_1200 for 1200 baud */
                        /*_COM_2400 for 2400 baud */
```

```
                        /*_COM_4800 for 4800 baud */
                        /*_COM_9600 for 9600 baud */
                        /* _COM_XXX for XXX baud */
#define COM1            0                       /* com1 port assignment */
#define COM2            1                       /* com2 port assignment */

/* set filter constants */
#define A1              1:0000
#define A2              -1.5610
#define A3              .6414
#define B1              .0201
#define B2              .0402
#define B3              .0201

extern void   p_init_all(void);
extern void    wait_for_profile(void);
extern void    s_cam_to_user(F_COOR *ptC, CALIBRATION8 near *ptCal);
void           send_offset(float offset);
void           init_serial_port(void);
void           init(float *tolerance, float *avg);
void           set_pavement_type(float *tolerance, float *avg);
void           menu1(void);
void           menu2(void);
void           filter_init(float x[], float y[], float avg);
void           set_video(void);
void           restore_video(void);
int            crack_type(void);
void           find_clean_crack(float tolerance, float avg, float *offset);
void           find_filled_crack(float tolerance, float avg, float *offset);
float          filter(float x[], float y[]);
void           emergency_out(int sig);
void           last_call(void);
void           check_start(int *run_status);
void           check_stop(int *run_status);
void           send_not_found(void);

void main(void)
{
        float tolerance;
        float avg;
        int type_of_crack;
        int count = 0;
        float offset;
        int i;
        int run_status = END_PROG;

        do
        {
                check_start(&run_status);
```

```
        }while (run_status = END_PROG);

        init(&tolerance, &avg);
        init_serial_port();

        do
        {
                wait_for_profile();     /* reads a profile */

                type_of_crack = crack_type();

                switch (type_of_crack)
                {
                        case CLEAN:
                                find_clean_crack(tolerance, avg, &offset);
                                break;
                        case VEGETATION:
                                find_filled_crack(tolerance, avg, &offset);
                                break;
                        default:
                                break;
                }
                check_stop(&run_status);
        }while (run_status != END_PROG);
}

void check_start(int *run_status)
{
        *run_status = START_PROG;
}

void check_stop(int *run_status)
{
        if (kbhit())
                *run_status = END_PROG;
        else
                *run_status = START_PROG;
}

void init_serial_port(void)
{
        unsigned data;

        data = (WORD_LENGTH | STOP_BITS | PARITY | BAUD_RATE);

        _bios_serialcom(_COM_INIT, COM2, data);
}

void send_offset(float offset)
```

```
{
        unsigned status;
        unsigned scaled_offset;
        static int i = 0;
        int data;
        int send = NO;
        char *error;
        char out_error[11];
        int dec, sign;
        int count = 6;

        _settextposition(15,35);

        strcpy(error, fcvt((double)offset, count, &dec, &sign));

        out_error[1] = '\0';
        if (dec <= 0)
        {
                strncat(out_error, " ", 1);
                dec = 0;
        }
        else
        {
                strncat(out_error, error, 1);
        }
        strncat(out_error, ".", 1);
        strncat(out_error, (error + dec), 3);

        if (sign == 0)
                out_error[0] = ' ';
        else
                out_error[0] = '-';

        strncat(out_error, " mm", 3);

        _outtext(out_error);



/*      check to see if "DATA READY" flag is set        */
          status = 0x100 & _bios_serialcom(_COM_STATUS, COM2, 0);

        if (status == 0x100)
        {
                while (status == 0x100)
                {
/*               get data from serial port              */
                        data = 0xff & _bios_serialcom(_COM_RECEIVE, COM2, 0);
                        if (data == SEND_DATA)
```

```
                        send = YES;
                    status = 0x100 & _bios_serialcom(_COM_STATUS, COM2, 0);
                }
                if (send)
                {
/*              send error signal to RPS            */
                    scaled_offset = (unsigned)((offset + FS_MM/2) * FS_BITS /
FS_MM);
                    if (scaled_offset > (FS_BITS+1))
                        scaled_offset = (unsigned)(FS_BITS + 1);        // no crack
found

                    /* wait until transmit holding register empty flag is set */
                    do
                    {
                        status = 0x2000 & _bios_serialcom(_COM_STATUS,
COM2, 0);
                    } while (status != 0x2000);

                    status = _bios_serialcom(_COM_SEND, COM2, scaled_offset);

                    if ((status & 0x8000) == 0x8000)
                    {
                        printf("\nError sending offset over serial port!!\n");
                    }
                }
        }
}

void init(float *tolerance, float *avg)
{
        int i;

        if (signal(SIGINT,emergency_out) == SIG_ERR)    /*trap ^C*/
        {
            perror("signal failed");
            exit(0);
        }
        atexit(last_call); /* the system will call last_call when the program terminates */

        p_trap_kb();      /*save actual key board fct so on return we restore it*/
        u_set_dir();      /*See II.3 Definition of proper DOS environment*/
        set_LPB_board(0); /*read if present descriptions of the LPB board*/
        p_int_reset();    /* resets interrupts 1-6 enable bit in CTRL1 rgister */
        p_int_init();     /*must be done once only*/
        global_init();    /*read camera parameters*/
        p_init_all();     /* initialize board */

        set_laser(INTENSITY);        /* set laser intensity */
```

```
        set_video();

        _setbkcolor(_BLUE);
        _clearscreen(_GCLEARSCREEN);
        _settextcolor((short)_WHITE);

        menu1();

        set_pavement_type(tolerance, avg);

        menu2();
}

void menu1(void)
{
        char dummy;

        _outtext("*************************************************\n");
        _outtext("*                                               *\n");
        _outtext("*        LASER VISION SYSTEM                    *\n");
        _outtext("*                                               *\n");
        _outtext("*************************************************\n");
        _outtext("\n\nBEFORE BEGINNING:\n");
        _outtext("You must sample the type of pavement which is being scanned.\n");
        _outtext("Every time the type of pavement is changed, the program must\n");
        _outtext("be restarted and the type is re-initialized.\n");
        _outtext("\nHIT ANY KEY AND ENTER WHEN THE SENSOR IS PLACED
OVER A SAMPLE\n");
        _outtext("OF PAVEMENT CONTAINING NO CRACK\n");
        scanf("%s", &dummy);
}

void set_pavement_type(float *tolerance, float *avg)
{
        int i;
        float total = 0;
        float min_v = 1000.0, max_v = -1000.0;
        F_COOR f_coor;
        float x[3], y[2];
        float vo;

        wait_for_profile();    /* reads a profile */

        /* initialize pavement type */
        for (i = 0; i < N; i++)
        {
                f_coor.line = (float)i;
                f_coor.pixel = (float)address[i];
```

```
                s_cam_to_user(&f_coor, calib8);
                total += f_coor.v;
        }
        *avg = total/N;

        filter_init(x, y, *avg);

        for (i = 0; i < N; i++)
        {
                f_coor.line = (float)i;
                f_coor.pixel = (float)address[i];
                s_cam_to_user(&f_coor, calib8);

                x[0] = x[1];
                x[1] = x[2];
                x[2] = f_coor.v;

                vo = filter(x, y);

                y[0] = y[1];
                y[1] = vo;

                if (vo > max_v)
                        max_v = vo;
                if (vo < min_v)
                        min_v = vo;
        }
        *tolerance = (max_v - min_v);

}

void filter_init(float x[], float y[], float avg)
{
        int i;

        for (i = 0; i < 3; i++)
                x[i] = avg;
        for (i = 0; i < 2; i++)
                y[i] = avg;
}


void menu2(void)
{
        char dummy;

        _clearscreen(_GCLEARSCREEN);
        _outtext("*************************************************\n");
        _outtext("*                                               *\n");
```

```
        _outtext("*        LASER VISION SYSTEM              *\n");
        _outtext("*                                        *\n");
        _outtext("**********************************************\n");


        _outtext("\nInitialization complete.  Ready to begin sampling.\n");
        _outtext("Place sensor over section to scanned.\n");
        _outtext("Hit any key and enter to continue.>>");
        scanf("%s", &dummy);
}

int crack_type(void)
{
        return CLEAN;
}

float filter(float x[], float y[])
/* discrete realization for a recursive high pass filter */
{
        float vo;

        vo = B1 * x[2] + B2 * x[1] + B3 * x[0] - A2 * y[1] - A3 * y[0];

        return vo;
}

void find_clean_crack(float tolerance, float avg, float *offset)
{
        F_COOR f_coor;
        int i, j;
        int crack_start, crack_end, rise;
        float u_start, u_end;
        float width;
        float vo;
        float x[3], y[2];

        filter_init(x, y, avg);

        crack_start = NOT_FOUND;
        crack_end = NOT_FOUND;
        rise = NOT_FOUND;
        for (i = 0; ((i < nb_line_field) && (!crack_end)); i++)
        {
                f_coor.line = (float)i;
                f_coor.pixel = (float)address[i];
                s_cam_to_user(&f_coor, calib8);

                x[0] = x[1];
                x[1] = x[2];
```

```
x[2] = f_coor.v;

vo = filter(x, y);

y[0] = y[1];
y[1] = vo;

if (crack_start == NOT_FOUND)
{
        if (y[1] > (y[0] - tolerance))        /* crack not found yet */
        {

        }
        else                        /* start of crack found */
        {
                crack_start = FOUND;
                u_start = f_coor.u;
        }
}
else if (rise == NOT_FOUND)           /* sensor is over crack */
{
        if (y[1] > (y[0] + tolerance))        /* rise of crack found */
        {
                rise = FOUND;
        }
}
else if (crack_end == NOT_FOUND)
{
        if (y[1] < (y[0] + tolerance))
        {
                crack_end = FOUND;
                u_end = f_coor.u;
                width = fabs(u_start - u_end);
                if (width < TOO_SMALL)
                {
                        printf("Crack too small\n");
                        crack_start = NOT_FOUND;
                        crack_end = NOT_FOUND;
                        rise = NOT_FOUND;
                }
                else if (width > TOO_BIG)
                {
                        printf ("Crack too big\n");
                        crack_start = NOT_FOUND;
                        crack_end = NOT_FOUND;
                        rise = NOT_FOUND;
                }
                else
                {
```

```
                                        *offset = (u_start + u_end)/2;
                                        send_offset(*offset);
                                }
                        }
                }
        }
        if (!crack_end)       /* crack end not found */
        {
                if (crack_start)
                {
                        u_end = f_coor.u;
                        width = fabs(u_start - u_end);
                        if (width < TOO_SMALL)
                        {
                                printf("Crack too small\n");
                                send_not_found();

                        }
                        else if (width > TOO_BIG)
                        {
                                printf ("Crack too big\n");
                                send_not_found();
                        }
                        else
                        {
                                *offset = (u_start + u_end)/2;
                                send_offset(*offset);
                        }
                }
                else
                {
                        send_not_found();
                }
        }
}

void find_filled_crack(float tolerance, float avg, float *offset)
{
        printf("\nProgram incomplete for filled cracks");
}

void send_not_found(void)
{
        printf("\nNo crack found!!\n");
        send_offset(NO_CRACK);
}

void last_call()
{
```

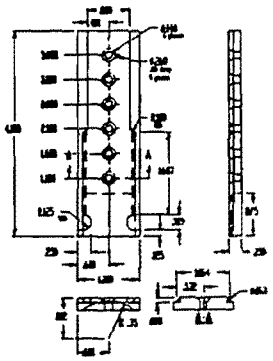
```
    p_restore_kb();          //must be done, key board ISR has been changed
    p_int_reset();
    restore_vectors();
    u_reset_dir();        //return to previous directory
    set_mode(3);
    restore_video();
}

void emergency_out(int sig)
{
    exit(0);
}

void set_video(void)
{
        struct videoconfig video_info;
        short videomode = _HRESBW;

        _getvideoconfig(&video_info);  /* Call to find adapter */
        switch (video_info.adapter)
        {
                case _MDPA:
                        printf("This program needs a graphics adapter.\n");
                        exit(0);
                case _CGA:
                        videomode = _HRESBW;    /* 2 color 640x200 CGA mode */
                        break;
                case _EGA:
                        videomode = _ERESCOLOR; /* 16 color 640x350 EGA mode */
                        break;
                case _VGA:
                        videomode = _VRES16COLOR; /* 16 color 640x480 VGA mode
*/
                        break;
        }
        /* Set adapter to selected mode */
        _setvideomode(videomode);
        /* Call _getvideoconfig again to find resolution and colors */
        _getvideoconfig(&video_info);
}
void restore_video(void)
{
        _setvideomode(_DEFAULTMODE);
}
```

APPENDIX D - MANUFACTURER'S SPECIFICATIONS

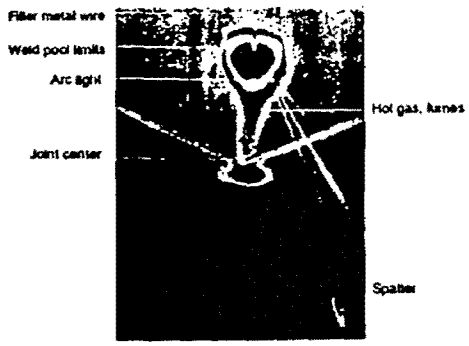
Resolution and Accuracy

The Laser Vision camera is calibrated to work in the above mentioned optimum area. All accuracy and resolution specifications are specified for this area.

	Horizontal		Vertical
Speed images/sec	60	30	
Resolution:	0.005" 0.125mm	0.0025" 0.064mm	0.006" 0.15mm
Accuracy position:	0.006" 0.15mm	0.003" 0.076mm	0.008" 0.2mm
Accuracy mismatch:			0.002" 0.05mm
Accuracy gap:	0.012" 0.3mm	0.006" 0.15mm	



Camera Bracket



MVS-30 Weld pool

Mounting

The LaserVision sensor is mounted on the torch using the camera bracket supplied. This **precision machined** part should be installed without any warping on a custom machined and **insulated** bracket mounted on the welding torch. Mounting should ensure flexibility of vertical or lateral adjustment. A 5° sensor tilt towards the torch tip is recommended. The distance to torch tip should be as short as possible, but at least 0.5" longer than the longest expected tack weld.

Applications

The **MVS-30** LaserVision sensor is a **medium resolution** sensor specifically designed for both **tracking and inspection robotic** applications. The elongated field of view helps in the initial part location, as well as the **weld pool** observation. It is best used for **V-grooved butt** joints, **large lap** joints and **fillet joints**. The maximum lap joint height is about 1" or 25mm. MVS-30 sensor is designed for **MIG, subarc, plasma** and **fluxcore** with welding currents up to **900A.**

Specifications

Speed: 60 images per second - RS170
50 images per second - CCIR standard

Cooling: liquid 1/4 US gallon (1l) per minute (air cooling for subarc and currents up to 50 A)

Air: 0.11 CFM (3l per minute)

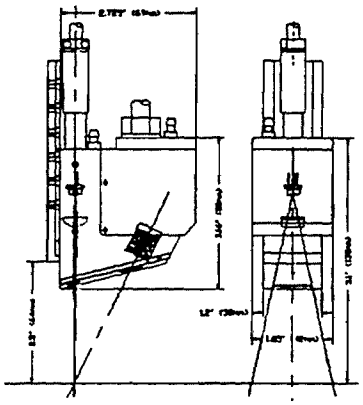
Weight: 9oz (250g)

MVS Modular Vision Systems inc., 3195 De Miniac, Montreal, Canada, H4S-1S9, (514)-333-0140, FAX (514)-333-8636

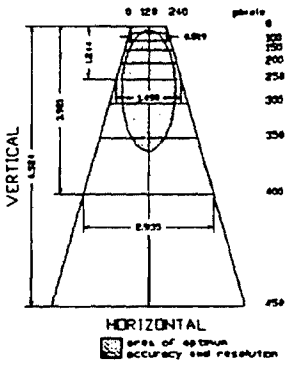
LaserVision Sensor MVS-30 Specifications

General

LaserVision is a new generation of highly ***reliable*** laser range (profile) sensors with ***no moving parts***, specifically designed for welding and sealant dispensing applications. It is the first really ***affordable*** vision based sensor, providing ***high processing speed*** and reliable tracking with more than adequate information for statistical process control and improved parameter control. At the same time, LaserVision is ***simple to use*** and ***rugged*** enough to provide trouble free service in any welding or other hostile industrial environment. A unique patented[1] design allows for ***over 200 hours of maintenance free operation*** under extreme spatter conditions (900A fluxcore). The output from the sensor is a common TV signal, allowing the images to be recorded for the Quality Assurance on a standard VCR.



MVS-30 Outline and Optics



MVS-30 Field of View

Principle of Operation

The LaserVision sensor uses a laser light projected in a plane approximately perpendicular to the observed joint. The cross section of the laser plane of light and the part produces a bright line. When this line is observed by a CCD camera at an angle (20° to 30°) it shows the surface features.

A dedicated vision processor board LPB-200 extracts the surface profile of 60 times per second – even under extreme arc light and spatter conditions. The relative distance of the surface points under the sensor is then calculated (by triangulation) and features of the profile, such as joint position and geometry, are extracted and measured.

Field of View

The field of view is trapezoidal in shape (see drawing) due to the angle of observation of the laser plane. An important feature of this approach is that a straight line remains a straight line, but angles are not preserved. This geometry allows for all the tracking algorithms to be performed in the camera space. A simple set of equations, with eight coefficients obtained by the LaserVision camera calibration procedure, describes the camera space and all the range points can be easily calibrated. The shaded area is the optimum working area for the LaserVision camera where resolution is highest, focus for both the laser line and the camera is optimal and distortion of the optics is minimal.

[1] US patent #4,859,829, August 22, 1989. Canada, Western Europe and Japan applied for.

3 LASERVISION SENSOR AND PROCESSOR

3.1 Introduction

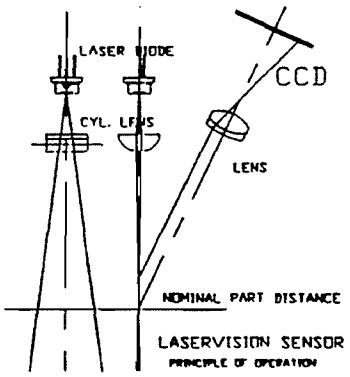


Figure 3.1.1 LaserVision Sensor Optics and Principle of Operation

The LaserVision sensor is a **range** type sensor. It uses a laser light projected in a plane approximately perpendicular to the observed joint. The cross section of the laser plane of light and the part produces a bright line. When this line is observed by a CCD camera at an angle (20° to 30°) it shows the surface features. A dedicated vision processor board LPB-200 extracts this ***profile*** of the surface 60 times per second even under extreme arc light and spatter conditions. A relative distance of the surface points under the sensor is then calculated (by triangulation) and features of the profile, as joint position and geometry, are calculated by an array processor SKY-320 and the AT-PC compatible computer. Each calculated joint position point is further verified, filtered and stored into the memory as a ***trajectory queue***. This point is then output at appropriate moment to the positioning subsystem when the torch arrives at the position where this particular joint position is measured (see the Motion Control section of this manual).

3.2 The LaserVision Sensor

Warning: Please read LASER SAFETY INFORMATION! A serious eye injury can result if the laser safety is not respected.

The LaserVision sensor consists of a CCD camera (a solid state TV camera) and a semiconductor laser. A pinhole, lens and filter combination serves as the objective for the CCD camera. A cylindrical lens is used to focus the laser beam into a plane of light. The beam is further restricted by the slot on the sliding protective plate. Small glass windows are used behind the slots and in front of the lens in order to further protect the lenses from spatter and metal fumes.

The entire camera is pressurized to prevent welding fumes from entering. Pressure is relieved through both the laser slot and the pinhole. In order not to disturb a gas shield around the torch the direction of the blown gas is away from the weld pool and the amount of the gas used is minimal (3.5 litres, or slightly less than one US gallon per minute). A clean pressurized air or inert gas should be used. If shop compressed air is used a reliable water, oil and dust filter should be installed in the air line. 1/8" barbed connectors are used for air and cooling water connection, suitable for 1/8" PVC tubing. The connector is rated for 150 psi of pressure when proper tubing is used.

Water cooling is mandatory for open arc currents of more than 50A. Air cooling can also be used for applications of less than 50A on open arc or a subarc system with currents up to 200-300A .

The interference filters are:

- 30nm bandwidth for TIG arc up to 100A and wire feeder in front; up 50A TIG without wire feeder in front and subarc applications.
- 10nm bandwidth for higher current TIG, plasma, MIG and flux core wire.
- 5nm bandwidth for some very bright arcs, usually plasma, with use of a fibre optic laser.

In case of 10nm filter bandwidth chilled (and heated in case of low operating temperatures) water is required to maintain a precise operating temperature for the laser (±3°C).

For more information consult the LaserVision sensor data sheets.

3.3 LaserVision Sensor Control and Processing

The camera video and synchronization signals are fed via the camera power supply to the LaserVision Processing Board LPB-200 (200-SYS-01). The camera power supply is factory adjusted. If required, please refer to the CCD camera and power supply information included.

The laser intensity is also controlled by the same processor board via signal IPUL (ILIN in earlier versions). A Laser Filter Board LFB-265 (265-SYS-02) provides optical isolation for the laser intensity control signal and a dedicated *floating* laser power supply connections. The laser control signal is a pulse with modulated signal with 60Hz base frequency. Maximum intensity and linearity of the control is adjusted by the potentiometer P1.

The laser power supply +5.25V and -12.0V is switched by the relay R1 by the interconnection board IB-240. The IB-240 board enables the laser only if:

- the EMERGENCY STOP is not pressed,
- there is no ALARM condition (watch-dog timer) and
- the LASER push button is engaged.

The signal received by the LPB-200 board as well as the processed profiles and tracking cursors can be observed on a profile monitor fed by the LPB-200.

For more information on the LaserVision Processing Board see the LaserVision Profile Processing Board - Technical Description

For the maintenance consult the LaserVision Camera Maintenance section.

4 LASERVISION PROFILE PROCESSING BOARD LPB-200

4.1 Introduction

MVS LaserVision Profile Board (LPB) is an image processing board specifically designed for extracting profiles of objects using a structured light and CCD camera. These profiles are generated by projecting a laser line on an object and observing it at an angle with a standard CCD video camera. Digital filtering techniques are used in order to ensure reliable operation in a high noise environment (i.e. arc welding) and suppress reflection artifacts.

LPB plugs into a single slot of an IBM-AT compatible computer.

4.2 LPB Main Features

The LPB can be used either alone as the only vision module in the system or with additional modules for increased performance, such as the DSP board with the Texas Instrument DSP processor TMS32010. A separate output port is provided for the transfer of profile data to the DSP board.

The principal features of the LPB module:

* Camera input, digitized at 8 bit per pixel.
* High resolution of 512 pixels per line standard.
* Highly stable digital phase locked loop synchronization of the internal pixel clock to the horizontal sync signal. A non cumulative jitter is less than +/- 12% of the pixel clock period, allowing for sub-pixel measurement accuracy. A reliable operation is achieved even with the standard VCR.
* Two groups of 2Kx8 (8Kx8 optional) bit input Look-up Tables, one group for processing and other for histogram.
* Monitor output with 8Kx8 (32kx8 optional) bit output Look-up Tables.
* Histogram circuit for 256 possible levels operating either on entire frame or area of interest window.
* Real time digital filter for the accurate feature extraction (profile)
* Profile extractor stores x, y coordinates and intensities of the most probable line points into 2Kx16 (8Kx16 optional) profile memory capable to contain 4 (16) profile vectors of 240 coordinates. This memory is accessible either to the AT host or a separate DSP processor board via a DSP output port.
* Eight selectable Area of Interest windows, easily movable around the picture area by specifying only the X-Y offset coordinates. Both histogram and profile extractor can be set to work only within this area of interest window.

- Flexible RAM based video clock and cursor generation allows for easy synchronization with wide range of standard and nonstandard video inputs.
- Capability to display "raw" profile or other intermediate results by simply loading pixel coordinate for each line into FIFO circuit.
- Laser intensity control output at 8 bit resolution.

4.3 Functional Overview

Main functions of the LaserVision Profile Board are shown on the Block Diagram (LPB.DWG).

The LaserVision camera is connected directly to the LPB via the provided cable. External synchronization is normally used, but there is a provision to use internal sync extraction from the video signal (VCR use) with some sacrifice in vertical positioning accuracy. Generation of internal synchronization signals is RAM based and allows for nonstandard video signals.

Initialization program supplied with the LPB loads necessary values for the RS-170 standard (North American B&W video) or CCIR standard (European) depending on type of camera. Same memory also serves for generation of two independent cursors.

The video signal coming from the camera is first conditioned then digitized to 8 bit accuracy. Two sets of look-up tables are provided, one for the digital filter and other for the histogram circuit. This allows entirely independent operation of the histogram circuit.

The digital filter is optimized for both noise suppression and laser line signal extraction. The laser line signals are enhanced and all other noise signals as ambient light are attenuated.

The digital filter circuit is followed by the profile extraction circuit. This circuit selects the peak of the laser line signal for each active video line and stores the result into Profile Memory during the horizontal blanking interval.

Results stored in the Profile Memory are accessible for further processing either by the host computer or via the DSP Output Port by the DSP board.

In order to further improve the noise immunity of the processing the LPB features the Area of Interest Window. Up to 8 different windows can be stored into window memory. Windows are selected through the control registers and they can easily be moved around the active video frame via X-Y offset registers. Both Histogram and Peak detector circuits can be set to operate only within the window and to ignore areas outside the window.

Typically the window shape is selected to closely match the expected joint profile. Once the joint profile is recognized and tracked, the window is set to closely follow the joint profile. Thus any noise outside the window of interest is automatically rejected. The described windowing technique also improves rejection of reflection artifacts.

The calculated profile or results of other intermediate calculations can be displayed on the monitor via the FIFO (first in first out) circuit. The FIFO has depth of 512 9 bit words, and it can be accessed via a single port. Total of 512 accesses fills the FIFO memory. When activated, the content of the FIFO memory is read synchronously with every active video line starting from the "zero" location. A single dot is output to the screen for every of 480 active lines at the pixel position equal to the address value stored into the corresponding FIFO location. The same output is automatically replayed every video frame without further program intervention.

The Output Look-Up Table circuit assigns gray levels to the digital filter results and the filtering operation can be observed in real time on the monitor. Windows, profile (FIFO) and cursors are displayed as bright overlay.

4.4 Specifications

Camera Input:	Video: 1Vpp
	Sync: TTL compatible, Standard RS170 or CCIR, other standards can be programmed.
Monitor Output:	RS170 or CCIR Composite sync.
Digitization:	Rate: 9.8304 Mhz standard
	Resolution: 8 bits.
	Jitter: less than +/- 12% of pixel width, Non cumulative.
Processor:	IBM-AT compatible, up to 8 Mhz, bus speed, requires 64K memory mapped space.
Output Port:	16 bit data, TTL compatible handshake control, up to 10 Mhz transfer rate, SKY320 compatible.
Measurements:	Width resolution: 240 points at 60 images per second, 480 points at 30 images per second, RS170 standard, or 256 points at 50 images per second, 512 points at 25 images per second, CCIR standard.
	Height resolution: 512 points.
	Nonlinear field of view due to triangulation technique used; however straight lines in actual space remain straight lines in transformed (camera) space. Calibration can be applied on final results only (i.e. after segmentation).