COMPUTER VISION FOR AUTOMATED SURFACE EVALUATION OF CONCRETE BRIDGE DECKS

BY PRATEEK PRASANNA

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Dr. Kristin J. Dana

and approved by

____

____

____

New Brunswick, New Jersey

May, 2013

ABSTRACT OF THE THESIS

Computer Vision for Automated Surface Evaluation of Concrete Bridge Decks

by Prateek Prasanna

Thesis Director: Dr. Kristin J. Dana

Structural health monitoring of concrete bridges requires accurate and efficient surface crack detection. Early detection of cracks helps prevent further damage. Safety inspection tests are conducted at regular intervals to assess deterioration. Traditional methods involve detection of cracks by human visual inspection. These methods are costly, inefficient and labor intensive, especially for long-span bridges. This thesis presents the use of computer vision and pattern recognition techniques in assessment of cracks on a concrete bridge surface. Bridge deck images are first collected using high-resolution cameras mounted on a robot. Statistical inference algorithms are then implemented to build an automated crack detection system. The proposed machine learning method reduces manual effort and enables automatic labeling over large bridge deck areas to quantify size and location for future reference or comparisons. A panoramic camera is used for the purpose of context localization. Additionally, we demonstrate image-stitching to obtain a coherent spatial mosaic of the bridge deck.

Acknowledgements

I would like to express my gratitude to my adviser Dr Kristin Dana for her constant support and encouragement. Her technical knowhow and advice helped me comprehend things and shape my research. I also thank her for bearing with my occasional indiscretions. I gratefully acknowledge the institutional support received through Federal Highway Administration's Long Term Bridge Performance Robotics program. A special note of thanks to Dr Ali Maher, Dr Nenad Gucunski, Dr Basily Basily, Dr Hung La, Ronny Lim, Parneet Kaur, Damoon Sima, Hooman Parvardeh, Francisco Romero and the entire robotics team. I have really enjoyed working in a team-environment towards making this project a success. I am also thankful to Kun Zhao for helping me with deck-mosaicing. It was a great experience sharing this laboratory with Siddarth Madan, Wenjia Yuan, Parneet Kaur and Chen Liang. I am obligated to Dr Sophocles Orfanidis and Dr Peter Meer for taking time out from their busy schedule to be my thesis committee members. My friends Shubham Jain and Sriram Shridhar have been a source of fun and motivation, especially when the chips were down.

Last but not the least, I am indebted to my parents and kid-sister, who were there for me and encouraged me at all times without even having the slightest idea of what I was doing.

Table of Contents

APPENDICES

List of Tables

List of Figures

Chapter 1

Introduction

Transportation accounts for about 11% of all the expenses in the US economy [1], the most important expense after housing, health care and food. A major part of these expenses goes into the construction and maintenance of bridge infrastructure.

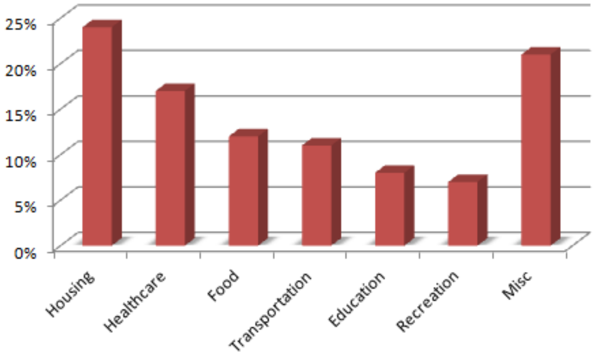


Figure 1.1: Transportation plays a major role in the nation's GDP. It accounts for more than one-tenth of the expenses in US economy [1].

With the rapidly developing infrastructure in transportation, crack detection is a problem of great interest since crack formation is one of the initial signs of degradation. Current method of site inspection is a drawn out process. First grid coordinates are marked on the bridge deck. These grids are 2ft × 2ft square regions. Skilled inspectors then go to the site and assess the deck condition, grid by grid, marking the corrosions and cracks on a chart, all under strict traffic control. Such methods heavily depend on the experience of the specialist. It can also tend to be inaccurate. Failure of detection of these initial cracks might lead to decrease in longevity of bridge, and sometimes collapse. Cracking in bridge decks accelerates the penetration of water, sulphates, chlorides and other corrosive agents through the concrete where they can damage the steel reinforcement, which leads to deterioration of the concrete structure as a whole, requiring costly maintenance and repair. Though cracking is a long-standing and notoriously difficult problem, in recent years more bridges are developing cracks almost immediately after being put into service rather than later as a result of traffic loads.

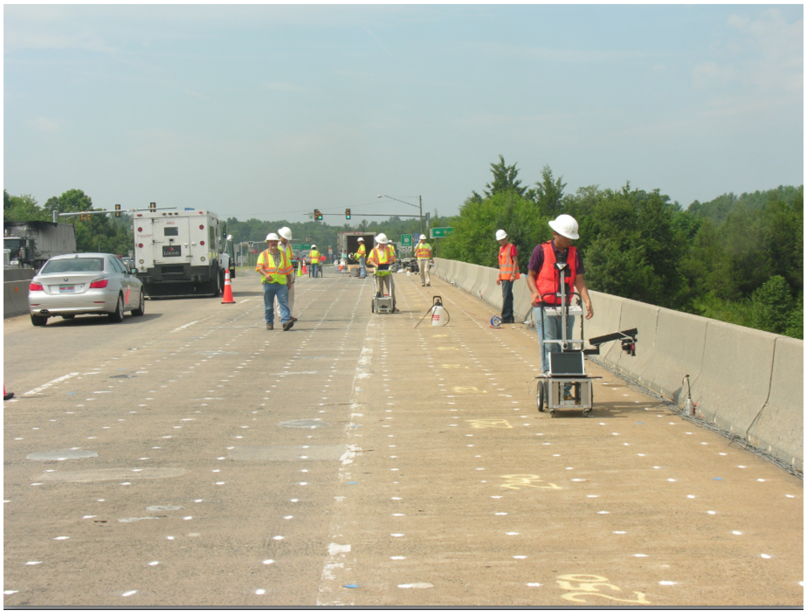


Figure 1.2: Various inspection processes being carried out on-field at the Virginia testing site. These include ground penetrating radar, resistivity, surface imaging and impact-echo tests. The white dots seen in the image are the grid-markings. *Courtesy*: Dr. Nenad Gucunski

Transportation plays a major role in the economic vitality and quality of life of society. Highways are the most important component amongst all transportation modes. Because of increased dependency on infrastructure, the engineered components of transportation infrastructure are deteriorating, and exhausting their capacity to meet the ever-expanding operational demands at a rapid pace. Ensuring operational continuity, safety and security requires a fundamental change in the way we plan, design, build, operate and maintain our transportation infrastructure. To effectively address this task, both political and public support is the need of the hour. This presents a challenge not only in financing and construction, but also involves meticulous planning, asset management, design and inspection. New and reliable methods must be identified to maintain and rebuild the infrastructure system at a war footing. Rapidly increasing traffic poses a major problem towards the renewal of our highway system. Bridge tests nowadays require closure of lanes to traffic for long durations at a stretch. Systems need to be developed that would help reduce the lane closure durations. In order for the safety and durability experiences of the past to be avoided, rapid and accurate inspection means should be tested and implemented. This is especially true for bridge decks, where the rate of deterioration is faster than that of other bridge components, and inspection and rehabilitation requires traffic interruption. New methods for their quick and accurate condition assessment and performance monitoring would result in:

- Reliable, fool-proof and standardized decision-making process
- Better allocation of financial resources and personnel to renew and rehabilitate bridge decks
- Reduction in manual labor and specialized training
- Reduced frequency and duration of traffic interruption caused by slow and ineffective inspection and monitoring procedures
- Increased efficiency as a result of reduction in inspection time

By making use of non-destructive evaluation (NDE) or non-destructive testing (NDT) methods, engineers are able to effectively determine irregularities in aging infrastructure, and take necessary measures. In addition to the ground-truth corroborated visually by specialists, data from NDE tests help in developing a comprehensive understanding of life-cycle and deterioration mechanisms of bridge decks. Repeated NDE tests at various stages of aging process ensures the longevity of bridge decks by preventing premature damage and failure.

The Long Term Bridge Performance (LTBP) program of Federal Highway Administration (FHWA) [2] focuses on detailed periodic inspections, monitoring, and evaluation of the population of bridges representing the national bridge inventory. Both NDE techniques and visual inspection methods are used for survey purposes. NDE techniques include ground-penetrating radar to detect flaws and corrosion inside the structure, sensor technologies that monitor traffic loading, cracks due to fatigue and corrosion, overloads and environmental conditions. Crack-detection, using image processing and pattern recognition techniques, is now being incorporated into their testing scheme. We present and test methods for automatic crack detection on real-world images of concrete bridge decks. The focus is on images where standard edge-finding algorithms fail due to background clutter. Several existing methods are combined into a unique approach. These methods include local curve fitting, intensity histogram classification, radon transforms, and histogram of gradient orientations classification. We achieve high accuracy on real data obtained with a robotic bridge inspection platform [4] shown in Figure 1.4. The methods take into account the irregularity and randomness of cracks and can be successfully integrated with a robot in order to reduce the currently pursued tedious process. We also implement image mosaicing on the concrete bridge images, wherein, images collected by the robot are stitched together to form a coherent spatial mosaic. Furthermore, the use of a 360 degree panoramic camera is demonstrated for the purpose of context localization.

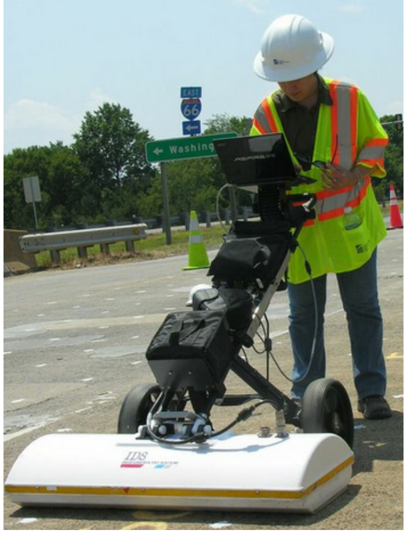


Figure 1.3: Ground Penetrating Radar used to detect subsurface corrosions [3]. *Courtesy*: Dr. Nenad Gucunski

The bridge-inspection robot, with the vision system mounted, was unveiled at a demonstration for the FHWA Administrator, Mr Victor Mendez. The robot featured

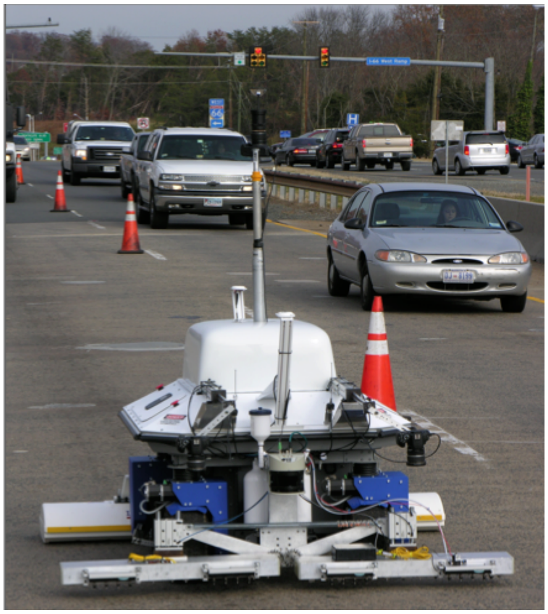


Figure 1.4: Bridge inspection robot with various systems mounted on it. It reduces manual labor by a considerable amount. *Courtesy*: Dr. Nenad Gucunski

featured in a recent news article which can be found at [5]. *The robot uses ground penetrating radar, ultrasonic waves and high definition imaging to see inside aging concrete bridges. It's like an MRI for bridges.* - NBC Washington.

1.1 Crack formation: factors and types

Before moving on to the detection algorithm, it is important to understand the causes that leads to the various types of crack formation. The following types of deterioration can generally be seen in concrete: scaling, spalling, cracking, abrasion damage, mortar flaking, alkali aggregate reactivity, delamination, freeze thaw and sulphate attack. More than 100000 bridges across US have exhibited early age bridge-deck cracking [6] . A lot of such bridges exhibit defects in early stages immediately after construction. There are reinforcement rods present in bridges, known as rebars. As cracks appear on the deck, paths are created for water and corrosive salts to reach these rods. The mixture of water and chloride ions are detrimental to the steel structures and cause corrosion. Crack types have been explored in prior work [7] and summarized as follows:

- Plastic shrinkage cracks: When the rate of evaporation of newly placed concrete exceeds the bleed rate, it results in the formation of such cracks. Extreme climate variations and high temperatures of concrete results in the increase in the evaporation rate.

- Flexural cracks: Concrete in its plastic stage can develop more cracks in the negative moment regions over the interior supports of continuous spans due to the dead weight of the girders and the newly placed concrete. Such cracks are called flexural cracks.

- Subsidence cracks: These cracks form over and parallel to the upper-most layer of reinforcement while the concrete dries when it settles around the bars.

- Temperature induced cracks: These cracks appear when unrestrained concrete undergoes volumetric changes due to variations in surface temperature.

- Abrasion Damage: Such cracks occur due to studded tires or blades of snow ploughs.

All the aforementioned cracks can be classified into three broad categories:

- Transverse cracks: These cracks, as shown in Figure 1.6, run perpendicular to the bridge girders. The width of such cracks range from 0.05 to 2 mm in width. Major cause of transverse cracking is restrained shrinkage [8] [9]. Figure 1.5 explains the formation of such cracks.

- Longitudinal cracks: Longitudinal cracks, as shown in Figure 1.7, run parallel to the bridge girders and generally form directly above the edges of the girders. This type of cracking is due to the presence of steel angles at these locations that are used to secure metal deck pans and cause a stress concentration. Longitudinal cracks might also occur due to obstructed settlement of the concrete during construction.

- Pattern cracks: Map cracking, also known as pattern cracking or alligator cracking, appears in random locations and directions. Such cracks occur due to poor construction practices such as drying of the concrete surface during placement and

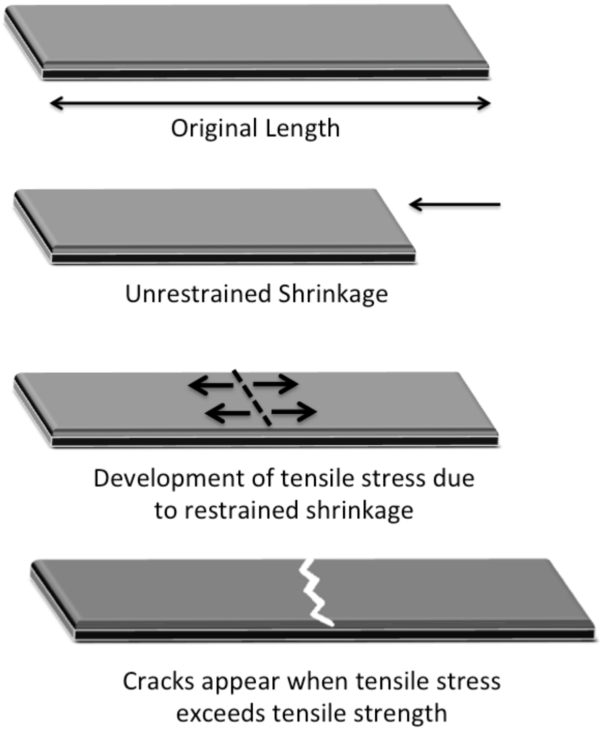

Original Length
Unrestrained Shrinkage
Development of tensile stress due to restrained shrinkage
Cracks appear when tensile stress exceeds tensile strength

Figure 1.5: Process of transverse cracking. Restrained shrinkage results in development of tensile stress. When this force exceeds the tensile strength, cracks begin to appear on the surface.

improper curing. They are generally narrow and shallow and are not considered highly problematic. Figure 1.8 shows pattern cracks on a surface.

There are three major factors that affect the cracks on concrete decks.

- Material Properties: A good quality concrete mix consists of various ingredients in appropriate proportions. A number of factors affect the shrinkage rate of concrete. These include, but are not limited to, cement type, cement content, water to cement ratio and addition of mixtures. Since higher strength concrete requires more cement paste, it experiences more shrinkage. Thermal contraction increases because of finer cements and higher cement content due to high temperatures from heat of hydration. Concrete mixtures, which can set faster than normal, shrink more than normal concrete because of the same reason. To counter this, shrinkage-compensating materials are added. A higher water to cement ratio increases concrete shrinkage. If the water quantity is more than required, it results in

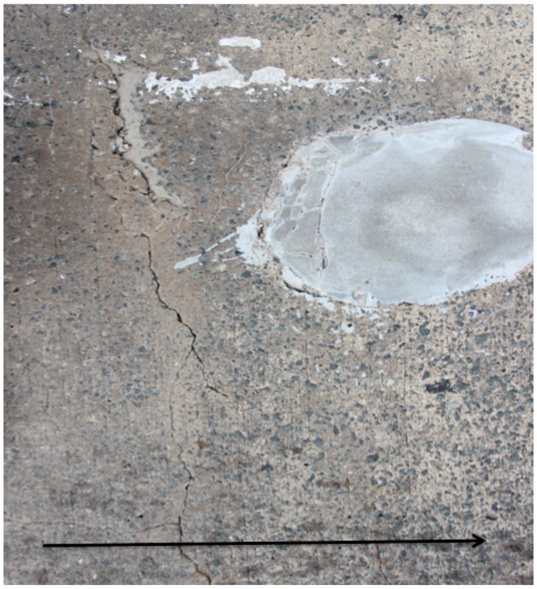


Figure 1.6: Transverse cracks on the surface. The black arrow indicates the direction of travel of the robot during image collection.

weaker concrete which is prone to more crack-formation. Shrinkage of the cement paste can be controlled by addition of aggregates. If the aggregate content is higher and the aggregate size is larger, concrete is less susceptible to shrinkage. Higher the modulus of elasticity, better is the shrinkage reduction. Well-graded aggregates also help minimize shrinking. Concrete admixtures help improve the cracking performance of concrete. The effect depends on the type of admixture applied. Water reducers and retarders are the types of mixtures that aid in reducing cracking tendency. Accelerators and silica fume tend to increase the heat of hydration, causing higher volumetric contraction. Air entrainment is useful to protect the deck from freeze-thaw cycles and for reducing the water content of concrete, without modifying its workability. Fly ash is another admixture that can reduce shrinkage by decreasing the amount of water in the mixture. It can also be used to reduce the amount of cement and reduce the heat of hydration.

- Design Properties: The width of cracks determine the amount of water and deicing salts that can penetrate into the deck and corrode the steel reinforcements. This highly affects the durability of bridge decks. Once cracks appear, their widths are

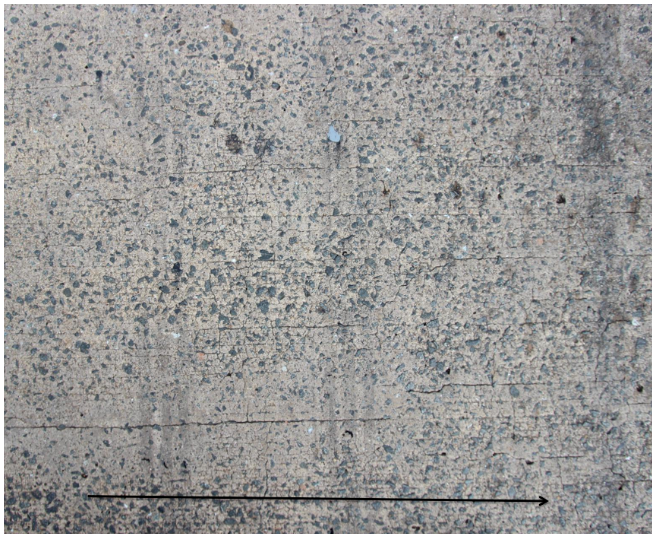


Figure 1.7: Longitudinal cracks on the surface. The black arrow indicates the direction of travel of the robot during image collection.

controlled and limited by the longitudinal deck reinforcement. Instead of a small number of wide cracks, a large number of narrow cracks appear because shrinkage strains are distributed along the reinforcement. If the deck is under-reinforced, the reinforcement will yield until shrinkage strains are distributed within the crack width, causing larger crack spacing and widths. Usually, deck cracks form halfway between other cracks or joints, where the tensile stresses of concrete are higher. There are standard provisions and recommendations specified to control crack widths caused by shrinkage and temperature.

- Construction Practices: Weather conditions play a significant role in affecting the rate of cracking on bridge decks especially during placement of concrete. Development of concrete strength is delayed when temperatures fall below 277 K during wet curing. Cracking tendency increases if concrete is placed during temperatures above 305 K, high wind speeds, and low relative humidity. Windbreaks and fogging equipment can be used to reduce moisture evaporation. Plastic shrinkage can be reduced and drying shrinkage can be delayed by using proper wet curing techniques. Curing results in increase of tensile strength with little or no shrinkage. Thus the concrete gains sufficient strength to resist stresses induced by restrained

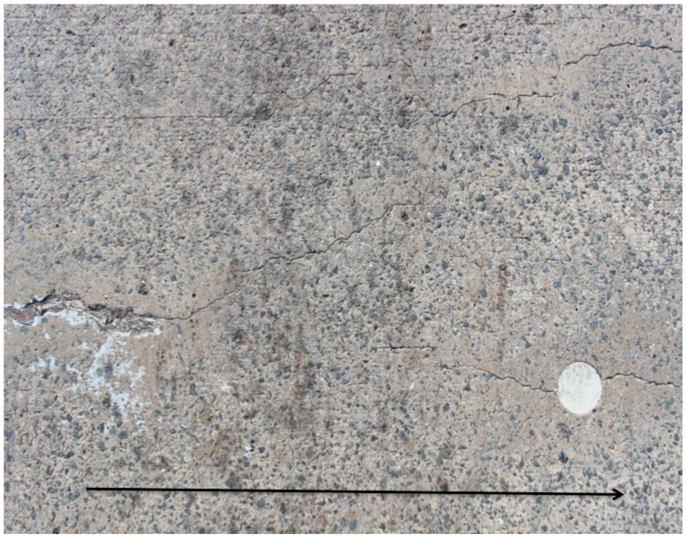


Figure 1.8: Cracks along various directions. Known as pattern or map cracks. The black arrow indicates the direction of travel of the robot during image collection.

shrinkage and prevent deck cracking. Use of sprinklers, covering the deck with wet burlap and plastic sheeting, and ponding water on the deck are some of the methods employed for wet curing.

1.2 Related Work

Research on automated crack detection has had varying degrees of success. Image processing-based approaches (wavelet and fourier transforms, canny filters, sobel filters [10]) and PCA-based techniques [11] are a few of the prior methods used for crack detection. Path planning of automated crack inspection systems has been proposed in [14] where Laplacian of Gaussian filters has been used.The output of the algorithm proposed in [14] is shown on one test image in Figure 1.10. Some crack detection schemes [15] rely on a set of geometric characteristics of segmented binary regions and make use of maximum aposteriori classifier. The percolation model [16, 17] requires certain parameter tuning and hence is not automated. In this method, a central pixel was evaluated using a cluster that was generated by the percolation model. This models considers the number of pixels, and their connectivity to identify cracks. The radon

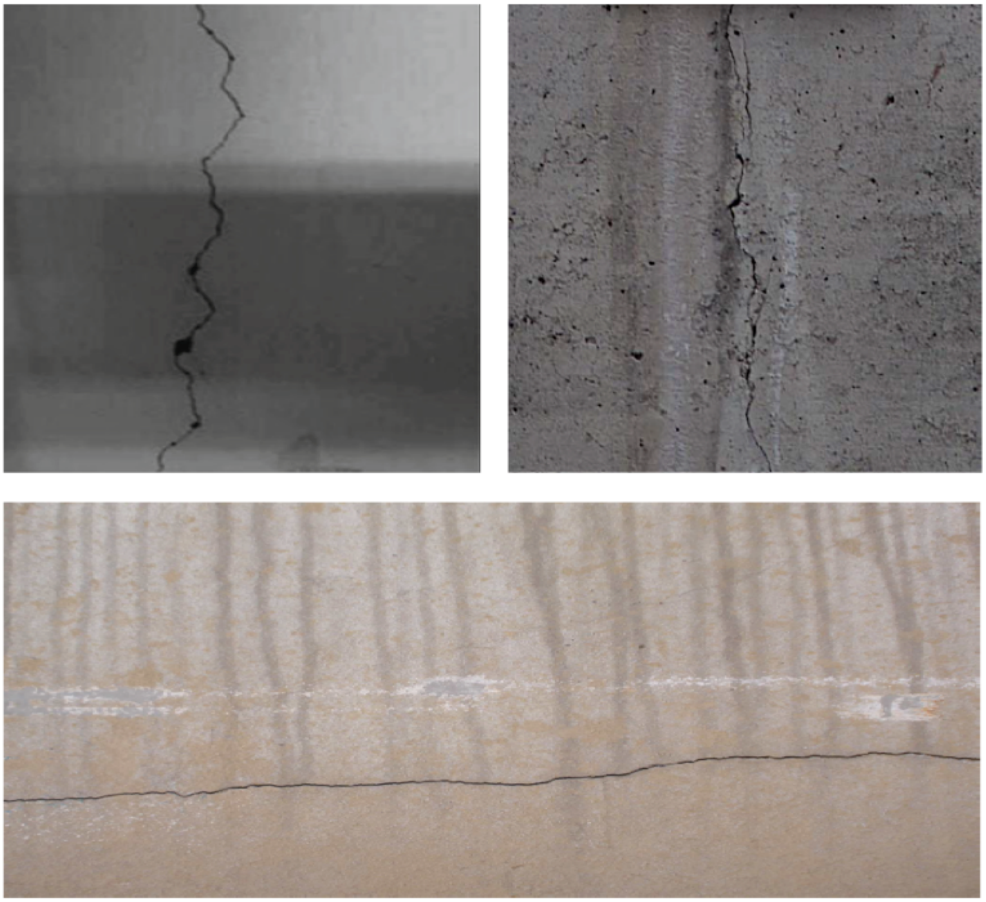


Figure 1.9: Previous works have dealt with images as shown in this figure [12] [13]. These concrete surface images have relatively less clutter than the images dealt with in this project. Edge detection techniques when applied to these images shown here yield good results. But such techniques fail when applied on images similar to the one shown in Figure 1.10(a) .

transform [18], edge detection [19] and morphological methods [20] yield excellent results but on images with very little noise and image clutter. Impact Echo test is another method used but measurement and analysis is time-consuming. Accuracy of detection depends on manual intervention and micro-adjustment of parameters in [21]. A NDT method for identifying cracks through changes in the reflection coefficient of a surface using wavelets was proposed in [22]. This fuzzy logic model used the reflection coefficient, operating frequency and the stand-off distance for width and depth estimation. A neural network method with five hidden layers has been described in [23]. Here the inputs are the major to minor axis ration and the area of all the objects. These features

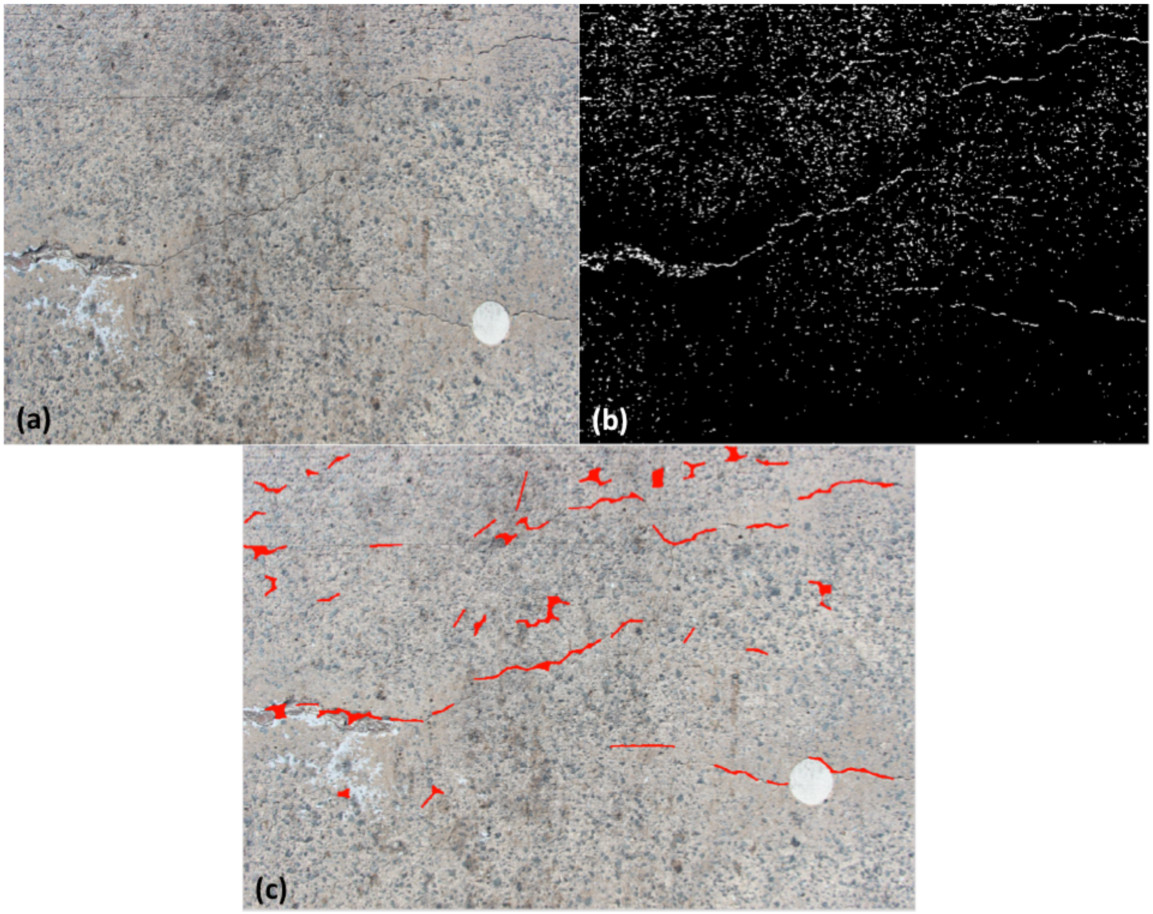


Figure 1.10: (a) Original image showing a surface with cracks. (b) Image showing the output of the algorithm applied on (a), as discussed in [14]. (c) Image showing the output of our algorithm.

are extracted after a sequence of subtraction processing, Gaussian filtering, thresholding and morphological closing operations. A preliminary study to identify and classify cracks using multi-light source photography and texture-based feature extraction has been discussed in [24]. Study of micro-crack propagation in compressed concrete using image correlations has been used in [25] [26]. Other bridge inspection systems using computer vision have been discussed in [20] [27]. Detection of faint curves in noisy images has been discussed in [28], where the algorithm efficiently searches for edges through a large set of curves by hierarchical construction of difference filters. The problem at hand is also similar do the detection of faint blood vessels in fundus images of the retina. Use of Gabor wavelets in detection of such vessels has been discussed in [29]. The noise surrounding the vessels , such as exudates, do not for an appreciable clutter as the ones seen in concrete crack images.

1.3 Organization of Thesis

The thesis is organized as follows. Chapter 2 describes the design of the vision system for the bridge deck inspection robot. This includes resolution tests and selection criteria for the cameras, mounting specifications, bandwidth and illumination requirements. Chapter 3 discusses the integration of the system with the robot. Chapter 4 illustrates the machine learning approach towards building the automated crack classification system. It concentrates on the identification of unique features and their subsequent use in classification. The results are presented in the form of confusion matrices and ROC curves. Methods and results for mosaicing are described in chapter 5. Chapter 6 discusses the panoramic image unwrapping system and crack density maps. We wind up the thesis by discussing the accomplishments and future work in Chapter 7. Miscellaneous information on graphical user interfaces and illumination requirement have been included in the appendix section.

Chapter 2

System Design

The bridge inspection robot needs to capture high quality images of concrete bridge decks. In order to facilitate proper coverage, the imaging system consists of two downward-facing and one 0-360 degree panoramic camera [30]. In addition to collection of still images, the downward facing cameras video-scan the entire bridge surface. The panoramic camera is used to obtain real-time contextual information. In this chapter, we summarize the criteria of selection of cameras with discussions on operation modes and resolution tests.

2.1 Surface Imaging System

During the initial assessment, we had narrowed down to two cameras namely the Canon EOS Rebel T3i DSLR [31] and the Allied Vision Technologies (AVT) Manta G125 [32] video camera. This document provides a detailed description of the camera selection procedure with a detailed comparison of:

- Field of view (FOV)
- Working distance
- Resolution comparison

2.1.1 Mounting Height

From the FOV requirements, the mounting height (from lens to the ground) range is found out to be 2ft to 3ft for the Canon camera and 3.5ft to 5ft for the AVT Manta camera.

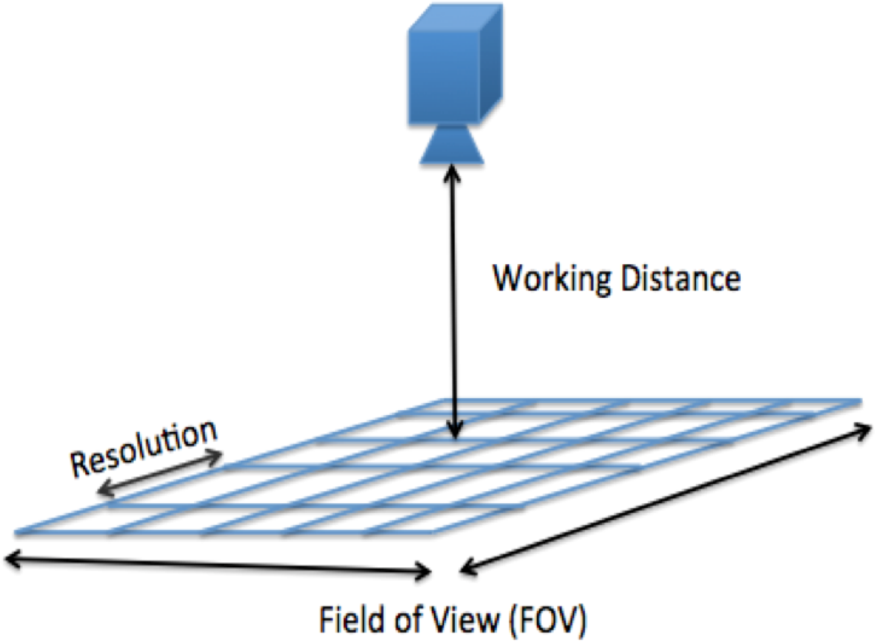


Figure 2.1: Representation of downward facing camera and mounting specifications.

Figure 2.2: Canon EOS Rebel T3i. Two such cameras are used for surface imaging of bridge decks.

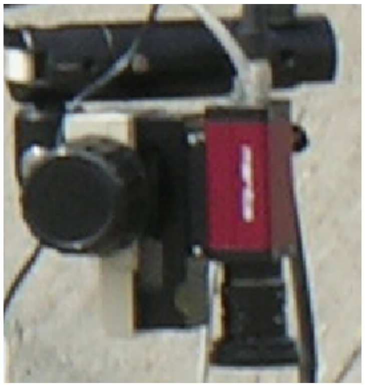


Figure 2.3: AVT Manta G125B with its mounting arrangement.

Working Distance	2ft 8.5in	2ft 3.5in
Field of View	3ft 9in×2ft 7in	3ft×2ft

Table 2.1: Canon EOS Rebel T3i.

2.1.2 Frame rate, image size and lens used

The frame rate of the Canon EOS Rebel T3i camera is 3.6 fps in still imaging mode and a maximum of 30 fps in the video mode. The AVT video camera has a frame rate of 30 fps. For an image size of 1920×1080 in the Canon camera, the storage size is 1.5Mb. For an image size of 1296×996 in the AVT camera, the storage size is 1.2Mb. A 18-55mm lens is used for the Canon camera while a 2.8/5mm lens is used for the AVT Manta camera.

2.1.3 Resolution comparison

Images of USAF 1951 resolution test chart [33] were captured using both the cameras. For efficient comparison, the chart was imaged under two conditions:
(i) Maintaining the same working distance for both the cameras.
(ii) Maintaining the same field of view for both the cameras.

Working Distance	3ft 4in	4ft 8in
Field of View	1ft 10in×1ft 6in	3ft 6in×2ft

Table 2.2: AVT Manta G125B.

The images are shown in Figure 2.4 and Figure 2.5. From Figure 2.4 and Figure 2.5,

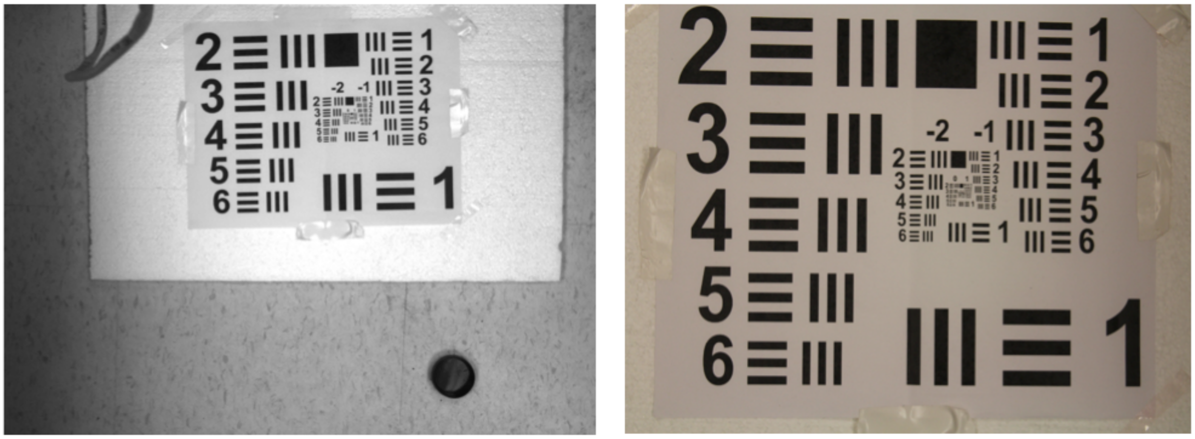


Figure 2.4: Images taken using the AVT Manta(left) and Canon(right) cameras respectively from a height of 2ft above ground level.

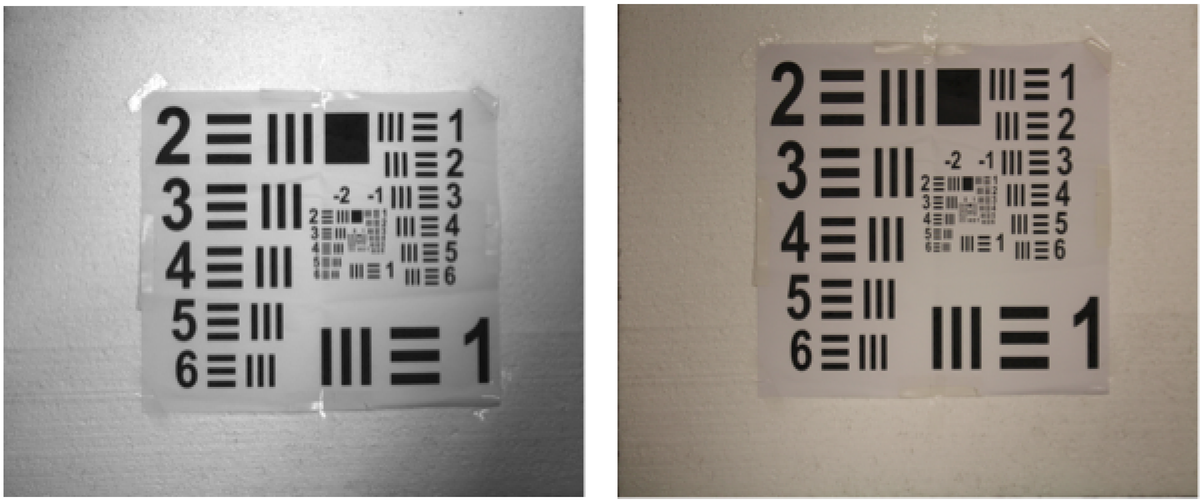


Figure 2.5: Images taken using the AVT Manta(left) and Canon(right) cameras respectively for the same field of view of 1ft 9in ×1ft 1.5in.

it can be observed that the resolving power of the Canon camera is significantly better than that of the AVT camera. The innermost block in the images of Figure 2.5 is magnified and shown in Figure 2.6.

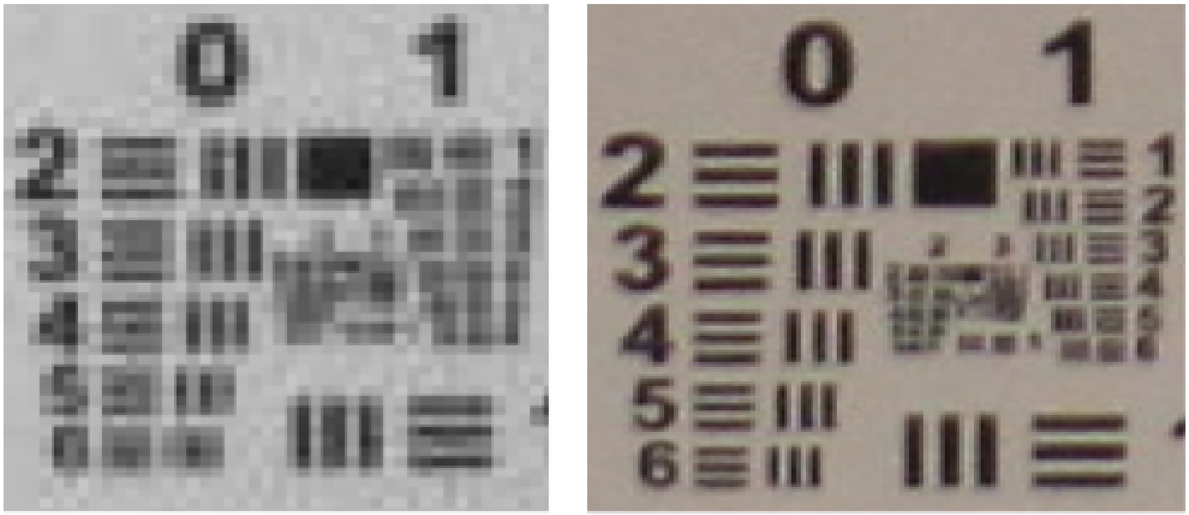


Figure 2.6: Innermost blocks in Figure 2.4 shown magnified. The Canon camera image(right) is found to have a better resolution as compared to the AVT camera image(left) .

2.1.4 Calculation of resolution

The diagonal resolution d_p is given by

$$d_p = \sqrt{w_p^2 + h_p^2}\,, \tag{2.1}$$

where w_p is the width resolution in pixels, h_p is the height resolution in pixels, d_p is the diagonal resolution in pixels and d_i is the diagonal size in inches. Pixels per Inch (PPI) is given by

$$PPI = \frac{d_p}{d_i}.$$

The resolution power of the Canon camera is 180.59 PPI (3 lines/mm or 76 lines/inch) while it is 70.08 PPI (1.5 lines/mm or 38 lines/inch) for the AVT camera. The images in Figure 2.6 were used to calculate the resolution. From the analysis presented above, it is concluded that the Canon DSLR would provide much better precision for the task at hand as the resolution is higher and the mounting height is within acceptable limits.

2.1.5 Timing parameters and Storage requirements

To collect and register images in memory, the robot has to halt at regular intervals along its path. For the Canon camera, when the working distance is 2ft 3.5in, the field of view (FOV) obtained is 3ft × 2ft.

The FOV in the direction of motion of the robot is 2ft. We have two downward facing cameras. We need to consider sufficient overlap between adjacent images in order to facilitate stitching. Taking a sideways overlap of 30% into account in each frame, that is 0.9ft, the total FOV for the two cameras combined is 5.1ft × 2ft.

In the direction of motion, assuming a 40% overlap between images, the robot should stop every d feet where

$$d = 2 - (0.4 \times 2) = 1\text{ft}\ \ 2.4\text{in}.$$

Similarly, when the working distance is 2ft 8.5in, the field of view (FOV) obtained is 3ft 9in × 2ft 7in.

The FOV in the direction of motion of the robot is 2ft 7in. Taking an overlap of 30% into account in each frame, the total FOV for the two cameras combined is 6.3ft × 2.58ft.

In the direction of motion, assuming a 40% overlap between images, the robot should stop at every d feet where

$$d = 2.58 - (0.4 \times 2.58) = 1\text{ft}\ \ 6.5\text{in}.$$

Field of View	2.58ft×6.3ft	2ft×5.1ft
Stopping Distance	1ft 6.5in	1ft 2.4in

Table 2.3: Stopping distance for still imaging.

To ensure proper coverage at a frame rate of 3.6 fps, deciding on the speed of the robot becomes critical. For the Canon camera, when the FOV in the direction of motion is 2ft, the coverage per second, c is given by

$$c = 3.6 \times 2 = 7.2\text{ft/sec}.$$

At a speed of 7.2ft/sec a 250ft long bridge (total strip size of 5.1ft × 250ft would be scanned in t seconds, where

$$t = \frac{250}{7.2} = 34.7\text{sec}.$$

Similarly, when the FOV in the direction of motion is 2.58ft,

$$c = 3.6 \times 2.58 = 9.3\text{ft/sec}.$$

Therefore, at a speed of 9.3ft/sec a 250ft long bridge (total strip size of 6.3ft × 250ft would be scanned in

$$t = \frac{250}{9.3} = 26.9\text{sec}.$$

Table 2.4 shows the maximum possible speed for the two FOVs.

The speeds calculated above conform to the requirements of the other equipment on the robot. Any speed exceeding the aforementioned ones would result in improper coverage and motion blur in extracted frames. Since each image captured by the Canon camera is 1.2Mb, the storage requirement would be high. Table 2.5 shows the minimum storage requirement for a single scan (250ft×6.3ft) and (250ft×5.1ft).

Field of View	2.58ft×6.3ft	2ft×5.1ft
Speed of robot	$<$ 9.3ft/sec	$<$ 7.2ft/sec
Time for a 250ft scan	27sec	35sec

Table 2.4: Speed requirement.

Field of View	2.5ft×6.3ft	2ft×5.1ft
No. of frames (for 250ft)	160×2	215×2
Size in Mb	384	516

Table 2.5: Storage requirement.

2.1.6 Power requirements

The robot has other systems like the GPR and resistivity probe. Apart from these systems, the movement of the robot and the retractable arms/masts consume a lot of power from the power source in the robot. The Canon EOS Rebel T3i works with a 7.2V 1120 mAh 8.1Wh Li-ion battery pack, and not on external power supply. This helps in saving power.

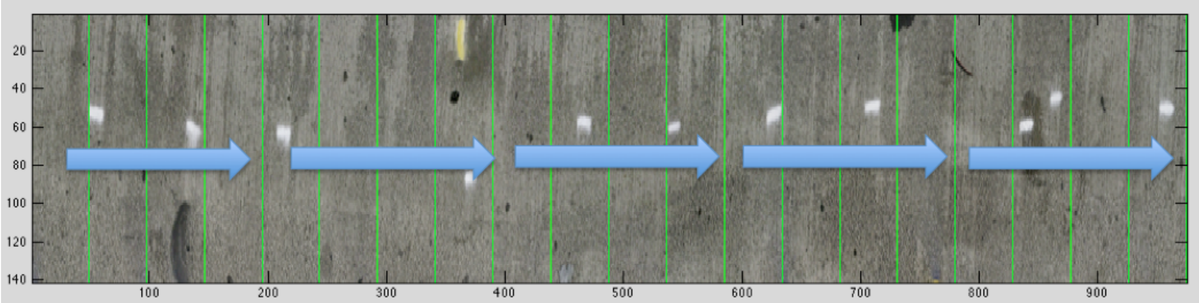


Figure 2.7: This figure shows a 24ft long and 3.47ft wide concrete surface. The corresponding image size is 975 × 141. Therefore each pixel corresponds to 0.2954 inches. For a FOV of 2ft × 5.1ft, the stopping distance is 14.4 inches which corresponds to 48.7 pixels. The blue arrows indicate the direction of motion of the robot and the green lines represent positions where the robot should stop and take images. A point (12ft, 1.73ft) in the world co-ordinate frame would transform to (487.5,70.5) in image co-ordinates.

2.2 0-360 degree panoramic camera

The panoramic camera is fixed to a retractable mast located at the center of the robot. It is pneumatic-operated and can be raised to a maximum height of 15ft. Normally, a height of 7ft is optimum to get a proper view of the surroundings.

2.2.1 Requirements

There are two major uses of a panoramic imaging system on the robot:

- Still images for context localization.
- Video recording and real-time unwrapping to monitor path of the robot.

0-360 Panoramic Optic is attached onto the camera/camcorder lens. For the video unwrapping, we make use of software provided by GoPano (Mac OSX) [34] or 0-360 (Windows 7). These software work on live video feed obtained as a FireWire or Thunderbolt input. Because of this constraint, the Canon EOS Rebel T3i cannot be used directly. Instead, we need a system that would provide a compatible video-feed to the software for unwrapping purpose.

2.2.2 Mac OS and wired streaming

The following components are required for live streaming:

- 0-360 degree Optic
- Canon EOS Rebel T3i
- Grass Valley ADVC110 Bi-Directional Analog / Digital Converter
- Laptop (Macbook Pro)

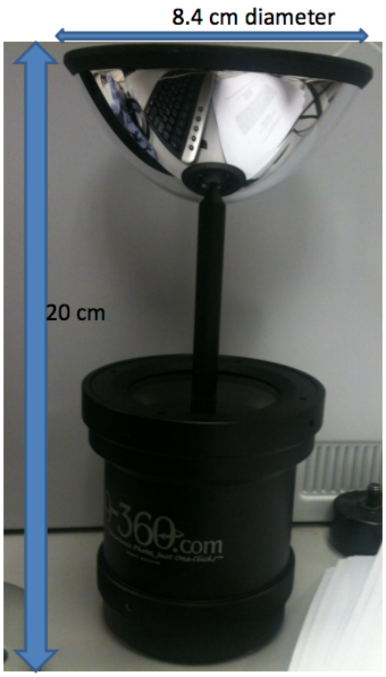


Figure 2.8: 360 degree mirror for panoramic imaging.

The bi-directional Analog/Digital Converter acts as an interface between the camera

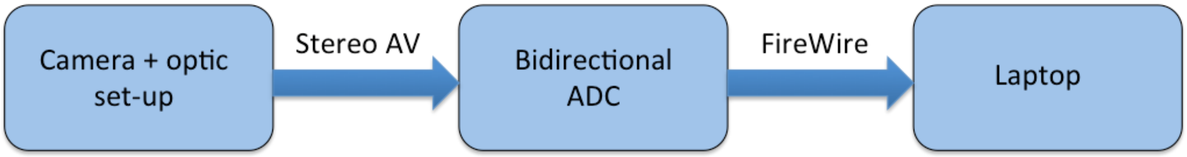


Figure 2.9: Live streaming process using Canon EOS Rebel T3i.

and the laptop. It provides the video unwrapping software a live video feed. It is connected to the laptop via a firewire cable.

- Power Source for the converter: IEEE 1394 bus powered or DC5V from EIAJ2 DC jack.
- Weight of the converter: 2.2 lbs.

Since the primary task of the overhead camera system is video-imaging, a camcorder that would directly enable live video streaming onto the computer is more preferable. Apart from lowering the power requirements, it reduces the overall extra weight on the system, arising due to the converter. The only camcorder that provides such functionality is the Canon VIXIA HV40 High Definition Camcorder. The camcorder has a FireWire-out port that provides the live feed to the unwrapper software. Figure 2.10 shows the schematic of the live streaming process using the Canon camcorder.

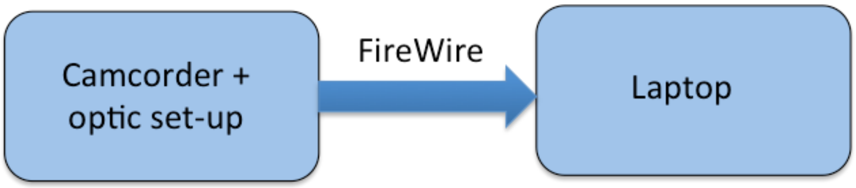


Figure 2.10: Live streaming process using Canon VIXIA HV40.

2.2.3 Video Unwrapping Software (MacOS)

VideoWarp Director from GoPano is used along with the VideoWarp player for unwrapping and viewing the 360 degree video respectively. The player interactively plays panoramic videos published by VideoWarp. Users can click and drag to navigate through panoramic videos with complete control. The unwrapped panorama can be viewed in four different modes:

- Perspective view: Interactively shows a portion of the unwrapped video.

- Cylindrical View: Entire 360 degree image viewed in one frame.
- Spherical view : Navigation in a spherical fashion (as seen in the mirror).
- Perspective + Cylindrical view: Any portion in the unwrapped frame can be selected and viewed.

Apart from this, the user can pan through the frame, zoom and tilt as and when required in the playback mode. The unwrapping modes are shown in Figure 2.11.

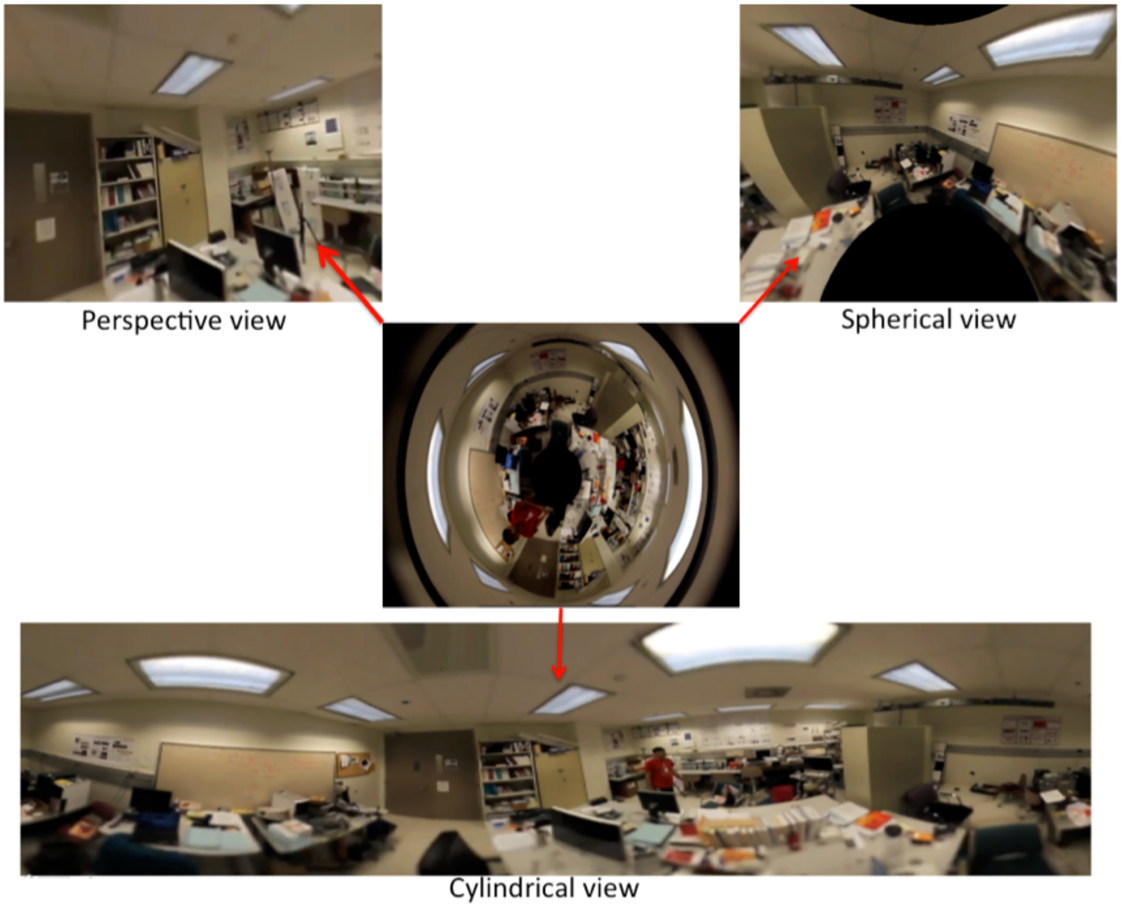


Figure 2.11: Frames of the unwrapped video in four different viewing modes.

Spherical projection: In this type of projection, points on a sphere is projected onto a plane. Other than at the projection point, the projection is defined on the entire sphere. It does not preserve distances or areas of objects but preserves angles. If cartesian co-ordinates are considered, the co-ordinates on the sphere (x_{sph},y_{sph},z_{sph})

and on the plane (x_{pla},y_{pla}) are related by

$$(x_{pla}, y_{pla}) = (\frac{x_{sph}}{1 - z_{sph}}, \frac{y_{sph}}{1 - z_{sph}}) \tag{2.2}$$

and

$$(x_{sph}, y_{sph}, z_{sph}) = (\frac{2x_{pla}}{1 + x_{pla}^2 + y_{pla}^2}, \frac{2y_{pla}}{1 + x_{pla}^2 + y_{pla}^2}, \frac{-1 + x_{pla}^2 + y_{pla}^2}{1 + x_{pla}^2 + y_{pla}^2}). \tag{2.3}$$

Cylindrical projection: In this type of projection, lines along the latitude when projected, become equi-spaced parallel lines whereas the lines along the longitude become parallel but not equi-spaced.

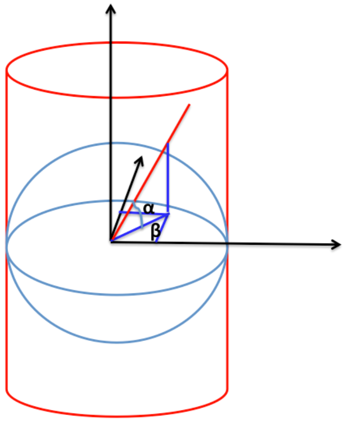


Figure 2.12: Cylindrical projection.

The planar co-ordinates (x_{pla},y_{pla}) are related to the angles α and β given by

$$x_{pla} = p\beta \tag{2.4}$$

and

$$y_{pla} = q * tan(\alpha), \tag{2.5}$$

where p and q are constants.

- Frame rate of the camcorder: 24fps.

- Mounting height: $>$ 6 ft.

- Power requirements: Canon BP-2L14 Battery Pack - 1450mAh, 7.2V.

With continuous operation, the battery life of a large capacity battery is 2.5 hours. The camcorder has 43mm threads and the optic has 58mm threads. A step-up ring adapter (43-58) is required to attach the lens mirror to the camcorder.

2.2.4 Windows OS and wireless streaming

The Canon Vixia HV40 is connected to the PC via Firewire. The unwrapping software used here is different and is provided by 0-360. To get a live video feed, the following sequence of operations is carried out:

- Set up a local server (which would be the computer on the robot). This is accomplished using Flash Media Server 4.5 [35].
- Collect video and assign this feed an IP address. This is accomplished using Flash Media Live Encoder.
- View it remotely on another computer. Real Time Messaging (RTM) protocol is used.
- Unwrap the video on remote computer.

2.3 Bandwidth requirement

The bandwidth requirement of the entire imaging system would be maximum when all the cameras work in HD video mode. This would have a direct bearing on the wireless router chosen for the transmission. For the Canon camera,

- At a resolution of 1920 × 1080 (1080p Full HD), the bandwidth required is 5.4 Mbps to 6 Mbps.
- At a resolution of 1280 × 720 (720p HD), the bandwidth required is 3.7 Mbps to 6 Mbps.

For the Canon camcorder,

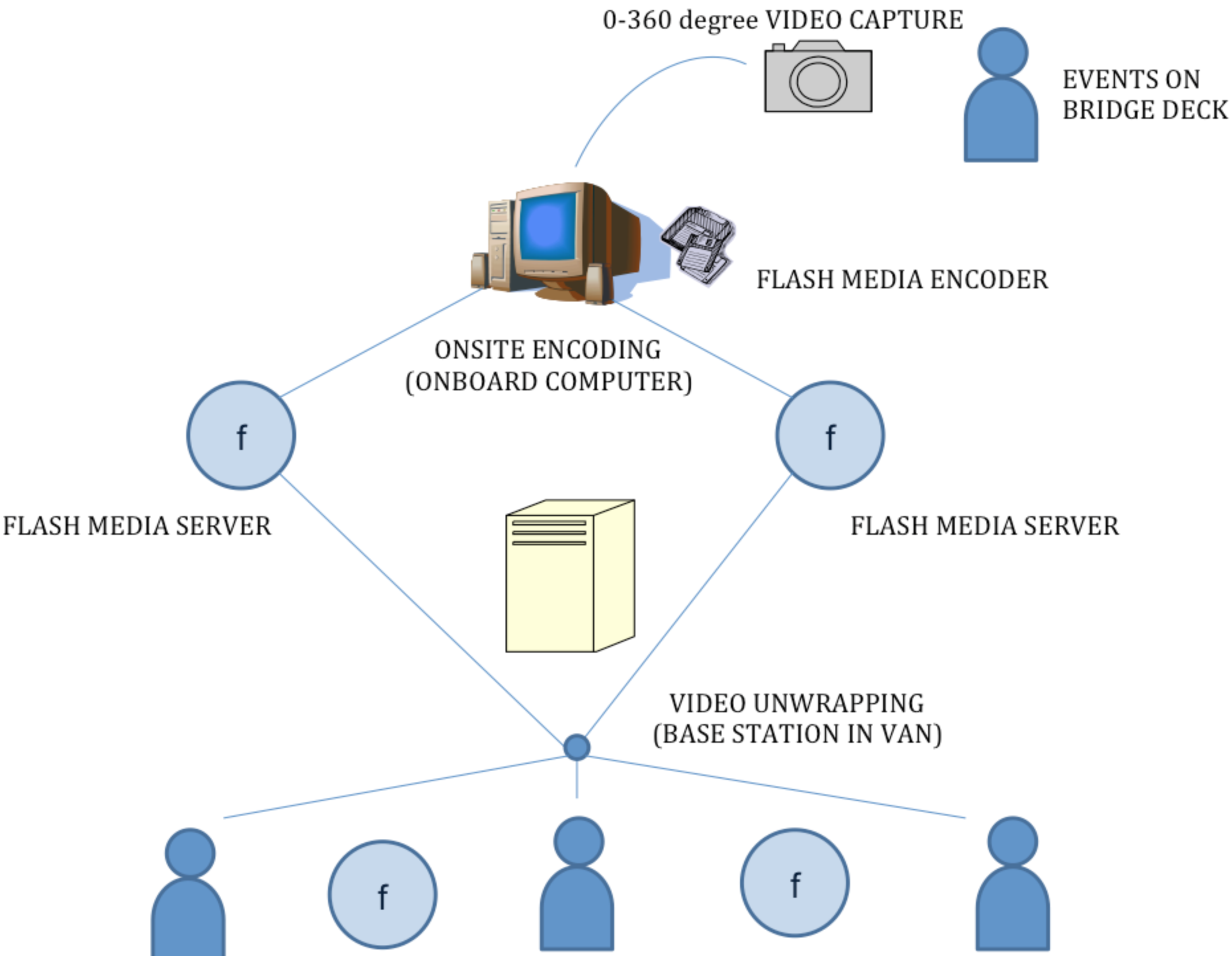

0-360 degree VIDEO CAPTURE
EVENTS ON BRIDGE DECK
FLASH MEDIA ENCODER
ONSITE ENCODING (ONBOARD COMPUTER)
f
f
FLASH MEDIA SERVER
FLASH MEDIA SERVER
VIDEO UNWRAPPING (BASE STATION IN VAN)
f
f

Figure 2.13: Illustration of the technical process of producing and broadcasting a live event.

- At a resolution of 1440 × 1080 (1080p Full HD), the bandwidth required is 3 Mbps.

The total bandwidth required for the two downward-facing cameras and the camcorder would be 15 Mbps (all in video mode).

2.4 Summary

The components of the vision system are listed in Section 2.4.1 through Section 2.4.3.

2.4.1 Components of the imaging system

- Two Canon EOS Rebel T3i DSLRs with Canon EF-S 18-55mm IS II Lens.

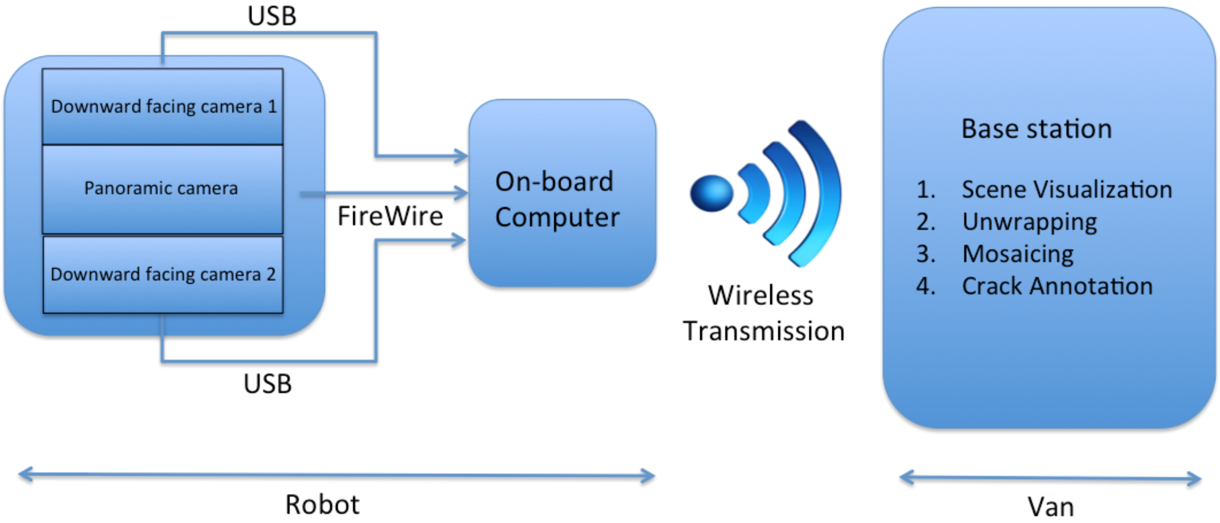


Figure 2.14: Diagram showing the data flow from the robot to the base-station in the van.

- Canon VIXIA HV40 Camcorder.
- 0-360 Panoramic Optic.
- 43-58mm step-up ring adapter.
- Two LED Cowboy Studio CN-160 video light sources
- Grass Valley ADVC110 Bi-Directional Analog / Digital Converter (optional)

2.4.2 Power Requirement

- Camera: Two 7.2V 1120 mAh , 8.1Wh Li-ion battery packs.
- Camcorder: Canon BP-2L14 Battery Pack - 1450mAh, 7.2V.
- ADC: IEEE 1394 bus powered or DC5V from EIAJ2 DC jack.
- LED light source: AA batteries (12).

2.4.3 Bandwidth Requirement

- Camera: 12 Mbps

- Camcorder: 3 Mbps

Total BW = 15 Mbps

Chapter 3

Integration of System with Robot

3.1 Calculation of Mounting Specifications

As a part of the ongoing LTBP robotics project at CAIT (Center for Advanced Infrastructure and Transportation) [36], Rutgers University, the individual inspection systems are mounted on the Seeker Robot. These include the following systems:

- Ground Penetrating Radar
- Surface Imaging Cameras
- Panoramic Camera
- Impact-Echo system
- Resistivity system

The system for surface and panoramic imaging was designed by the computer vision group. A 25-30% overlap between successive surface images is required. The lens used for the Canon camera is a 10-22mm f3.5/4.5 USM wide-angle lens. The field of view from a height of 31 inches is 34 inches×45 inches. Before building the mounting system on the robot, extensive tests were performed to find out the required specifications. As a first step, we designed a mobile cart to simulate the vision-system on the robot. This aluminum-frame cart had adjustable arms for the cameras, in addition to a platform for a laptop. This cart was initially taken to bridge decks for preliminary image collection. Apart from the requirements for image stitching, the mounting system design has the following constraints:

- The arms can have a maximum height of 2ft 4in. A higher height for the cameras would require the use of aluminum supports (or plates).

- The distance between the arms has to be 2ft 8in, as shown in Figure 3.1.

- The coverage of the GPR and the impact-echo test for each swath is 6ft wide. So, precise measurements need to be made so that there is sufficient overlap between consecutive swaths.

The final design of the cart conformed to the following mounting specifications.

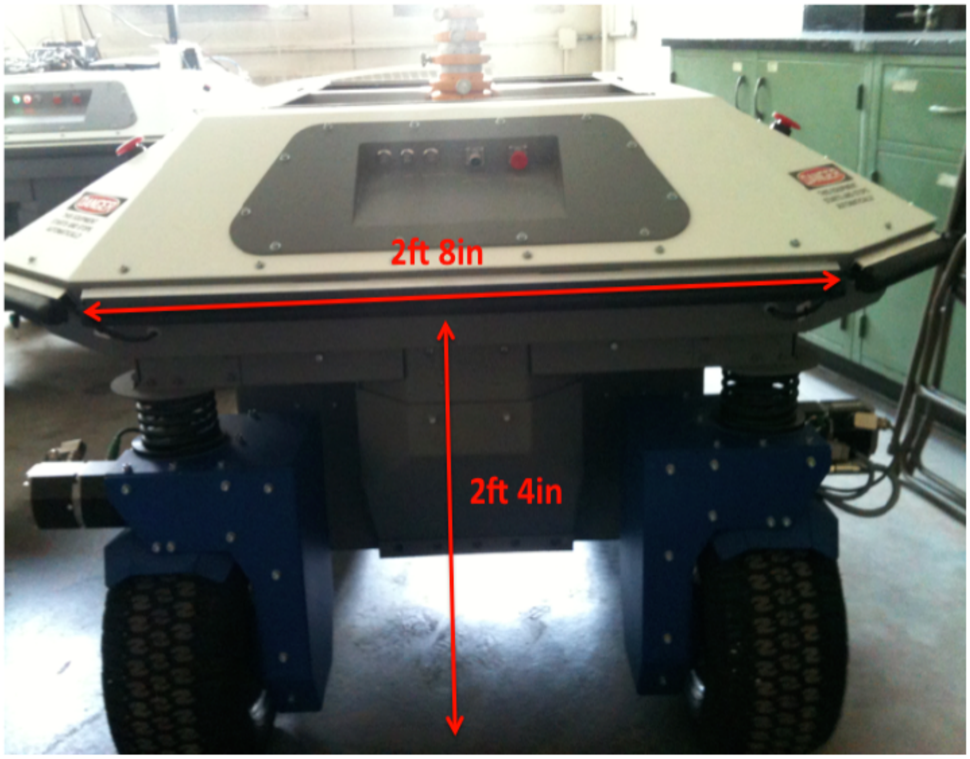


Figure 3.1: Maximum separation between the arms and maximum height of the mounting arms from the ground.

- Distance of ground from the screws = 31 inches.

- Distance between cameras = 36 inches.

- Distance between cameras and cart edge = 19 inches.

The above mounting specifications result in a field of view of 34 inches × 52 inches The ideal overlap required would be 30% of 52 inches (16 inches perpendicular to direction of motion) and 30% of 34 inches (10.5 inches along the direction of motion). What we actually get is 16 inches and 9.5 inches respectively. Figure 3.2 shows the design outline for the mobile cart system. Figure 3.3 shows the image-overlap measurements. Figure 3.4 and Figure 3.5 show the top and front views of the mounting system.

3.2 Automated Image Collection System

Canon provides software development kit (SDK) for its DSLR cameras. Using this SDK, an acquisition system was built that has two modes of operation:

- Image collection only when the robot stops at every 2ft interval.

- Image collection every t seconds when the robot is in motion.

This is a fully automated image-collection system. The base-station is located in a van, which can be located as far as 150ft from the robot. The snapshots of the GUI built for this system are shown in Figure 3.8.

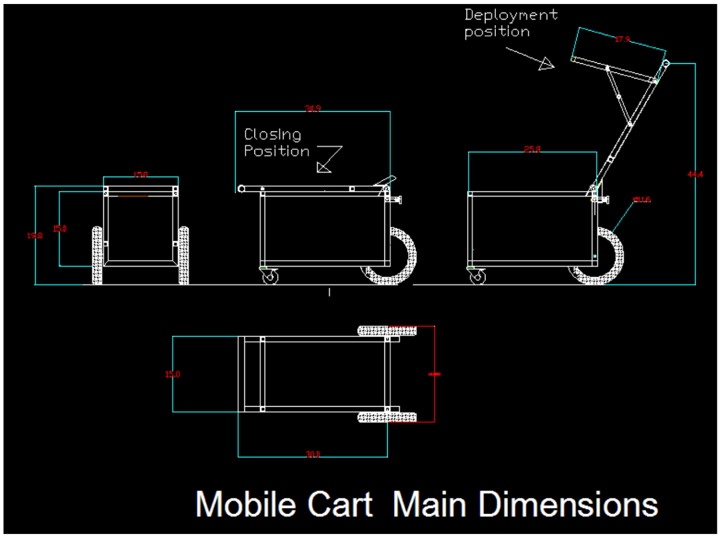


Figure 3.2: Autocad generated schematic showing main dimensions of the mobile cart. *Courtesy*: Dr. Basily Basily

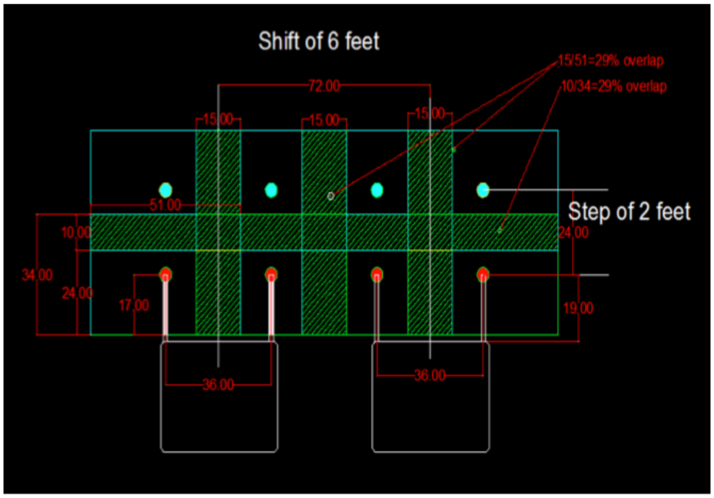


Figure 3.3: Illustration of image-overlap measurements for the mounting specifications calculated earlier. *Courtesy*: Dr. Basily Basily

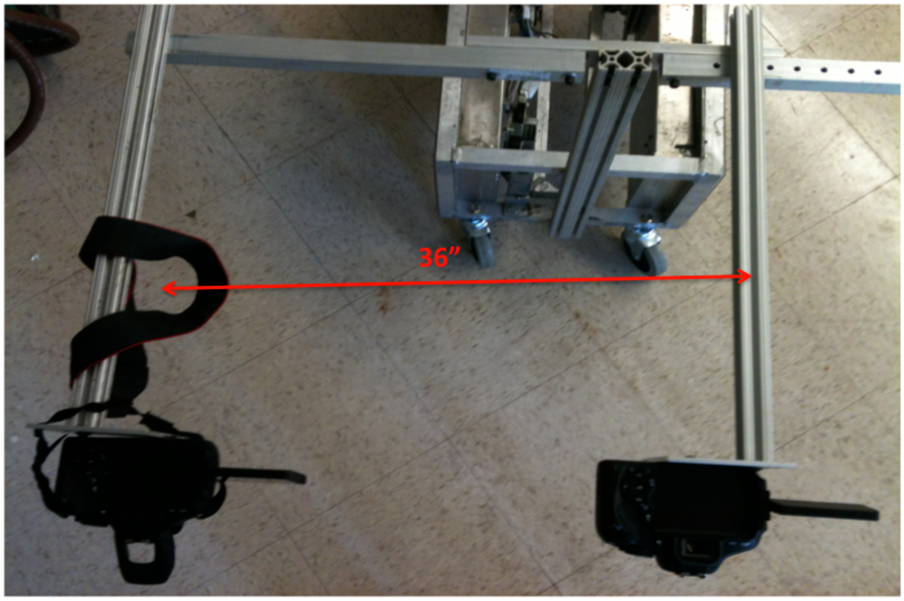


Figure 3.4: Top view of the mounting system showing the separation between the arms.

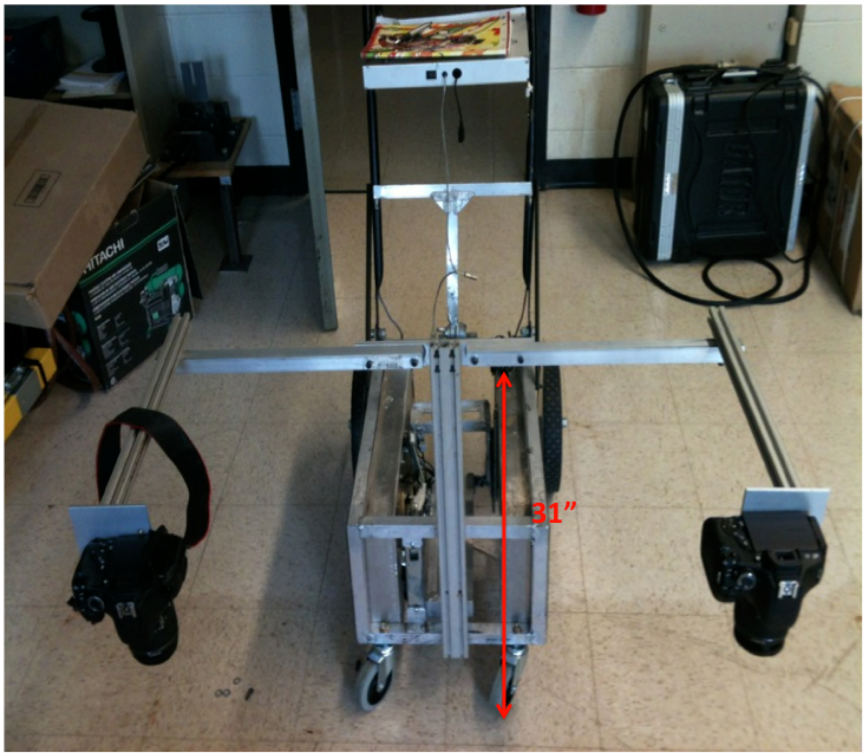


Figure 3.5: Front view of the mounting system showing the showing the distance between the ground and the arms.

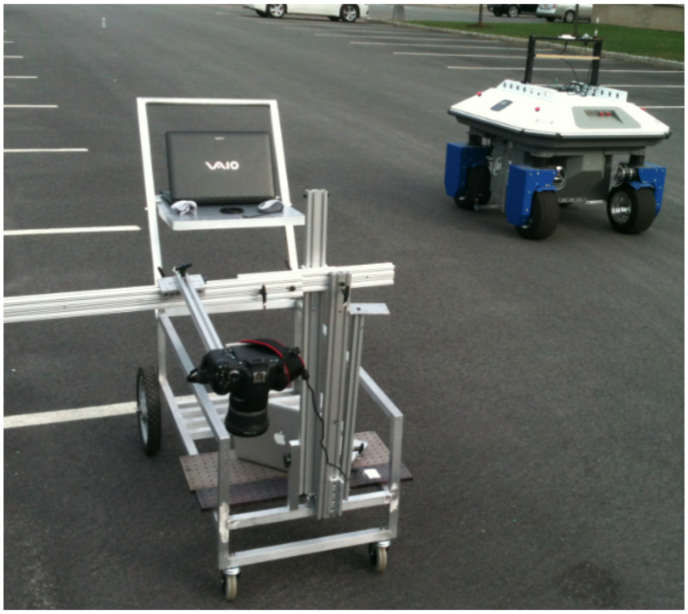


Figure 3.6: Both the cart and the robot: The mounting system on the cart has been transferred on to the robot as shown in Figure 3.7.

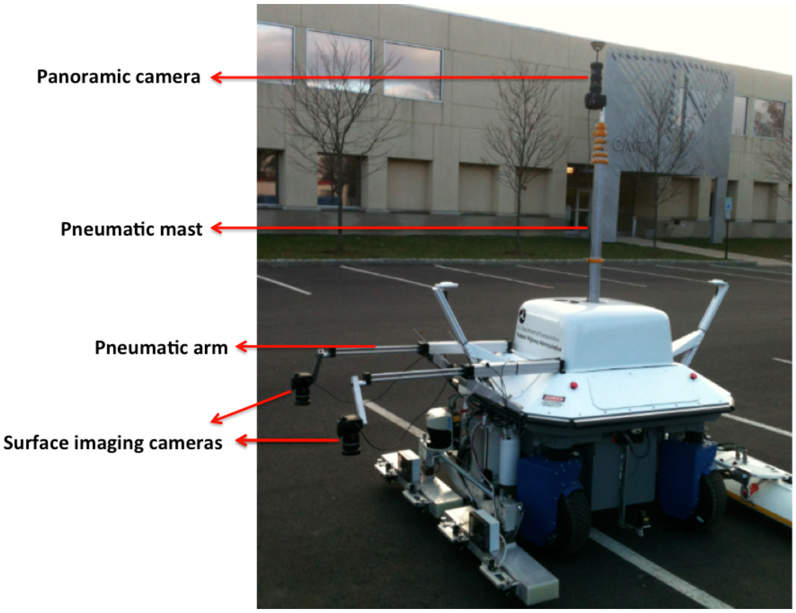


Figure 3.7: Image shows the camera mounting system on the robot. The surface-imaging cameras are mounted on retractable pneumatic arms and the central panoramic camera is mounted on a retractable tall pneumatic mast.

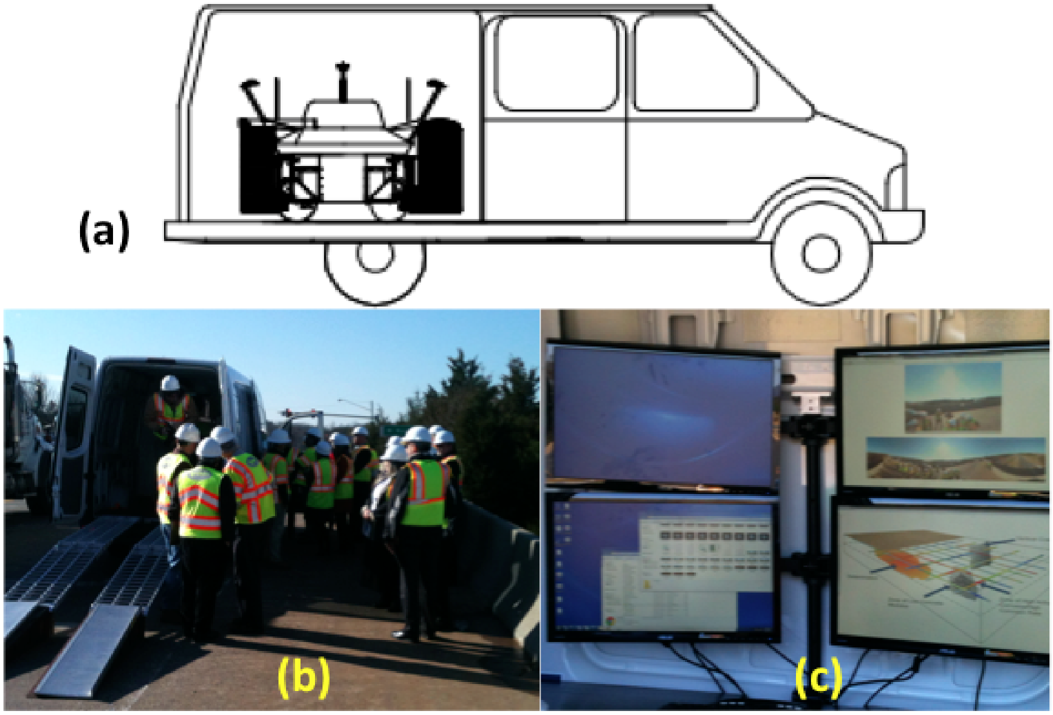


Figure 3.8: (a) Schematic of the van and the robot inside it (*Courtesy:* Dr. Hung La). (b) The van for the robot. This picture was taken during the Virginia demonstration. (c) PC screens at the base-station.

Chapter 4

Crack Detection: Methodology and Results

There is a major difference between the appearance of cracks on concrete and asphalt surfaces. In asphalt cracks, the background is very distinct from the cracks, whereas, in concrete cracks, there are distractors present in the form of blebs and stains all over the image. This problem is illustrated in Figure 4.1. With the cracks on asphalt surface, simple edge detection methods can almost perfectly detect the cracks. But when it comes to concrete surfaces, simple edge detection gives erroneous results because of the presence of the additional image clutter around the cracks.

In this chapter, we discuss in detail different features investigated for the purpose of building a crack-detection algorithm. Features in Section 4.3 and Section 4.4 were extensively tested on both the bridge datasets. Features in Section 4.5, Section 4.6 and Section 4.7 were tested on the California bridge dataset.

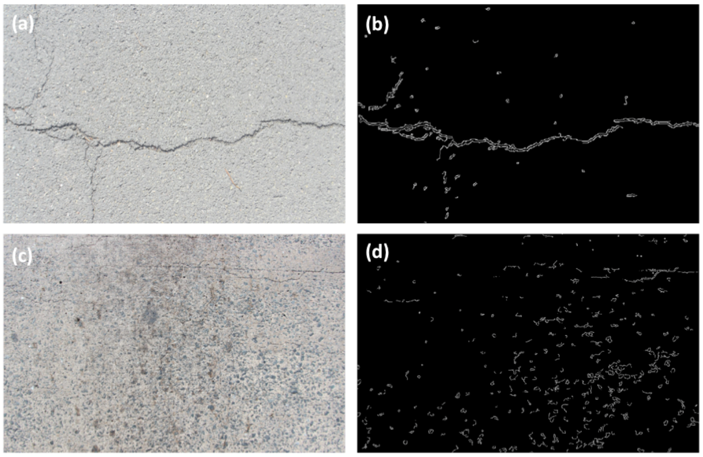


Figure 4.1: (a) Crack on an asphalt surface. The crack is very distinct from the background and can easily be segmented as shown in (b) using a simple canny edge detector. (c) Cracks on a concrete surface. These cracks are not very distinct from the background and are surrounded by distractors, which makes segmentation and detection difficult as shown in (d). Hence a robust machine learning approach is necessary.

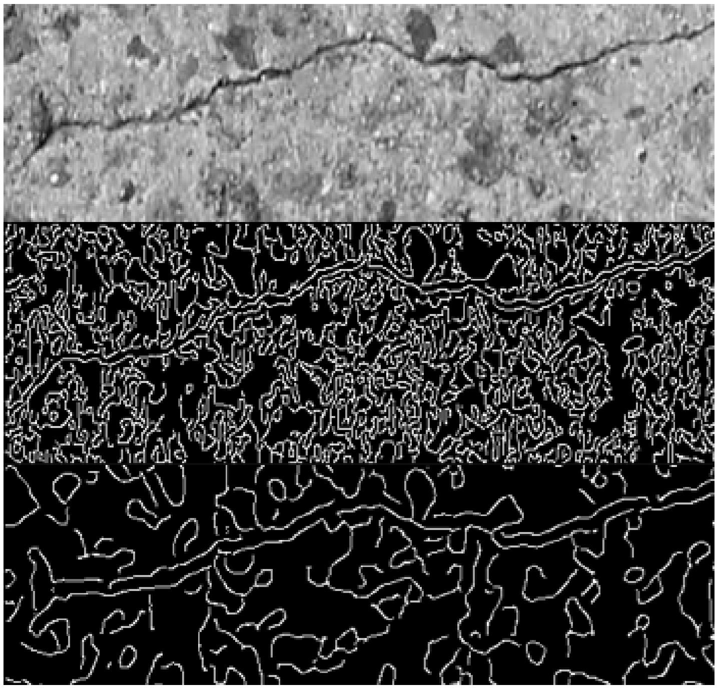


Figure 4.2: Original Image and two outputs of canny edge detector [37] with the values of σ as 1.414 and 4 respectively. As expected, the approach was not sufficient because of the presence of many spurious edge-like structures. There is no clear demarcation between cracks and normal regions based on the detected edges resulting in improper classification. A more robust machine learning based approach was found necessary.

4.1 Collection and labeling of data

As part of this project, bridges at Gainesville, Virginia and Sacramento, California were surveyed (Courtesy: CAIT, Rutgers University). High quality surface images of the bridge deck were collected. Since labeled data was unavailable, the labeling for cracked and uncracked regions was done by visual inspection. During the initial stages of testing, the cart was taken to the bridge in Virginia (Figure 4.3). On the first day of survey, still images of the deck were captured for the entire 280 feet span of the bridge, for the right lane. Apart from this, 360 degree images were taken at various positions of the bridge so as to enable the location of a particular feature of interest with respect to the global scene. On the second day, the entire left lane of the bridge was videotaped. A few of the sample images and one of the 360 degree image captured on the bridge is shown in Figure 4.4 and Figure 4.5 respectively.

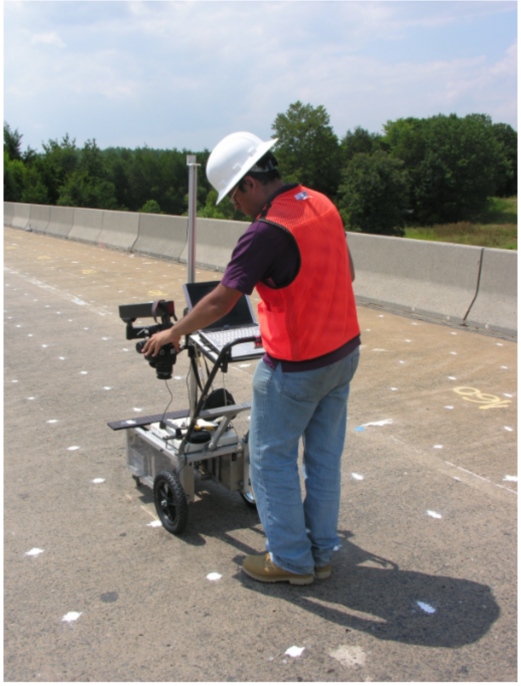


Figure 4.3: Initial stages of test, where the cart was used for image collection.

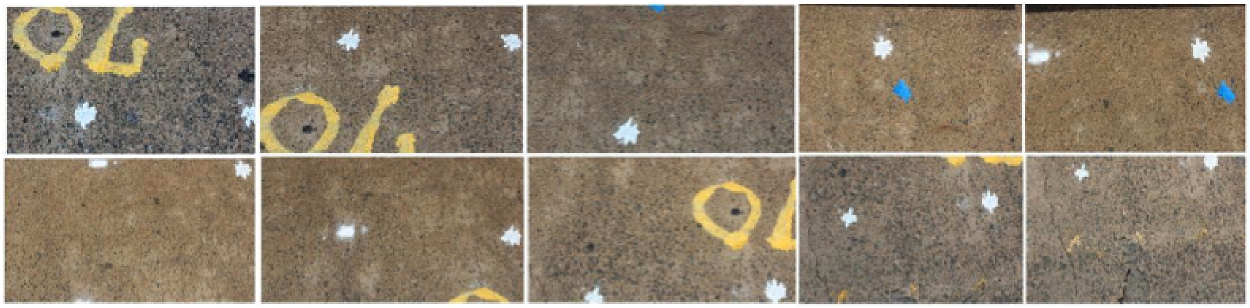


Figure 4.4: Sample deck images of Virginia Bridge.

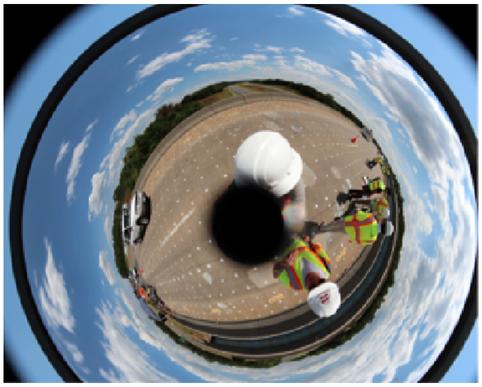


Figure 4.5: Image captured using the panoramic camera on the Virginia bridge deck.

4.2 Classification Experiments

The primary task for classification is to distinguish between blocks that have cracks and blocks that don't. The size of blocks we consider vary depending on the feature vector that we choose. After extracting feature vectors, we train our supervised learners to classify patches into one of the two classes. It is therefore a binary classification problem with the classes being *0* for *cracks* or *1* for *no-cracks*. A few of the cracks and no-crack blocks are shown in Figure 4.6.

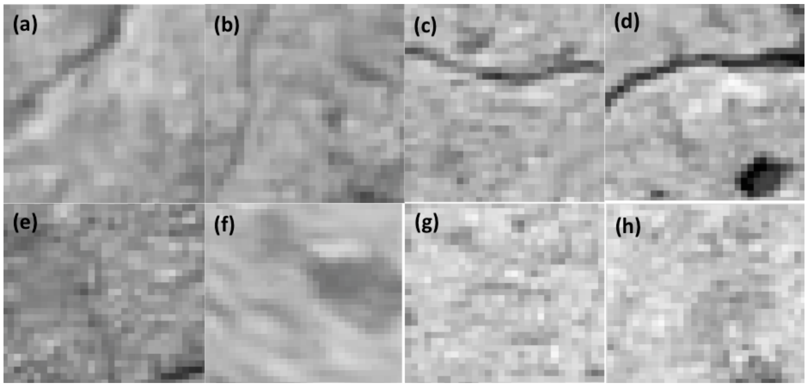


Figure 4.6: Positive and negative training samples. (a)-(d) show patches with cracks. (e)-(h) show patches without cracks. For our training and validation purposes, we construct a dataset of 1000 samples having equal number of positive and negative instances.

A training and validation set is constructed from such positive and negative samples. The following criteria are used for evaluating the performance of our classifier:

- Confusion Matrix
- Accuracy, Sensitivity and Specificity
- F-score
- Area under ROC curve

4.2.1 Performance Metrics

Equations 4.1 through 4.6 give the formulae for the performance metrics used in our classification results analysis.

		Predicted Label	
		No-crack	Crack
True Label	No-crack	True Negative (TN)	False Positive (FP)
	Crack	False Negative (FN)	True Positive (TP)

Table 4.1: Confusion Matrix

Accuracy A is defined as the fraction of instances (both positives and negatives) correctly classified from the total given set, and given by

$$A = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4.1}$$

Sensitivity *Sens* is also known as *Recall* or *True Positive Rate.* It is the fraction of correctly classified true instances from the total number of true instances in the set, and given by

$$Sens = \frac{TP}{TP + FN}. \tag{4.2}$$

Specificity *Spec* is also known as *True Negative Rate.* It is the fraction of negative instances correctly classified as such, given by

$$Spec = \frac{TN}{TN + FP}. \tag{4.3}$$

A perfect predictor would be described as 100% sensitive and 100% specific. Theoretically, however, any predictor will possess a minimum error bound known as the Bayes error rate.

Precision *Prec*, also known as positive predictive value, is the proportion of positive test instances that are true positives, given by

$$Prec = \frac{TP}{TP + FP}. \tag{4.4}$$

The harmonic (weighted) average of precision and recall is known as the F-score *FS*, given by

$$FS = \frac{2 * Prec * Sens}{Prec + Sens}. \tag{4.5}$$

Precision is thus a measure of a classification test's accuracy.

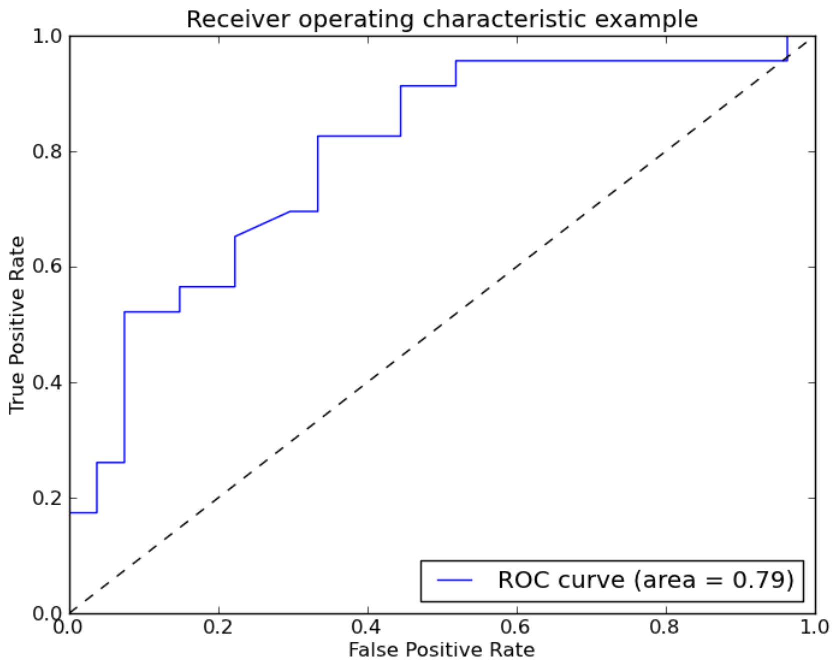


Figure 4.7: A sample ROC curve

The area under the curve (AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming *positive* ranks higher than *negative*) [38] . Accuracy (in the tables that follow) is measured by the area under the ROC curve. An area of *1* represents a perfect test. An area of *0.5* means *just guessing.*

After building our classifier, we first conduct a 10-fold cross-validation to evaluate the performance on the training set data [39]. We then repeat the same for the validation set before moving on to the test-set, which is unlabeled.

From the bridge data sets, we take 1000 patches. We use 500 as our training set images and the rest 500 as our validation set images. These images consist of an equal number of positive and negative samples.

4.3 Classification using intensity histogram-based features

4.3.1 Fitting local curves to lowest intensity points

In 30×30 neighborhoods in grayscale imagespace, curves were fitted to pixels whose intensity lied a fixed percentage below the average block intensity. The curve fitting was done using both RANdom SAmple Consensus (RANSAC) [40] and least squares estimation, but in most cases, least squares estimation was found sufficient. In Section 4.3.4, we describe how RANSAC gives better results than least squares estimation. The blue lines in Figure 4.8 are the local curves. This local curve-fitting process forms an intrinsic part of our crack detection process and we investigate for optimum features along and around the fitted curves. The process gives us the advantage of investigating edges in a small window rather that in the image as a whole.

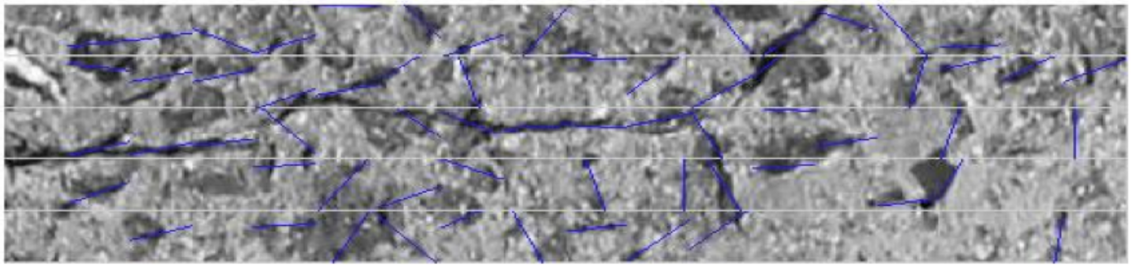


Figure 4.8: Blue lines show the detected curves in the local curve-fitting process. For the initial set of experiments, curves were fit to minimum intensity points using least squares estimation. But eventually points having minimum Laplace pyramid values were used for fitting curves. Also, instead of least squares method, we used RANSAC as the accuracy of the curvefitting process greatly increased.

As seen from the Figure 4.8, many crack regions are correctly detected, as evident from the blue local curves. Some of the detected local curves appear along uncracked regions. The classification algorithm improves upon this incorrect curve-fitting by classifying the detected curves as *cracks* or *no-cracks*.

The first set of features we consider include raw intensity histogram values along local curves and in local regions (Section 4.3.2). Better classifier accuracy was obtained with features discussed in Section 4.3.4.

4.3.2 Histogram-based features in regions and along local curves

Feature vectors are computed as input to both training and testing phases of the classifier. In order to identify feature vectors, histogram plots are then obtained for the following:

1. In 30×30 neighborhoods containing the detected local curves

2. Along the detected local curves. The rationale behind selecting these feature vectors is illustrated in Figure 4.8 and 4.9 respectively.

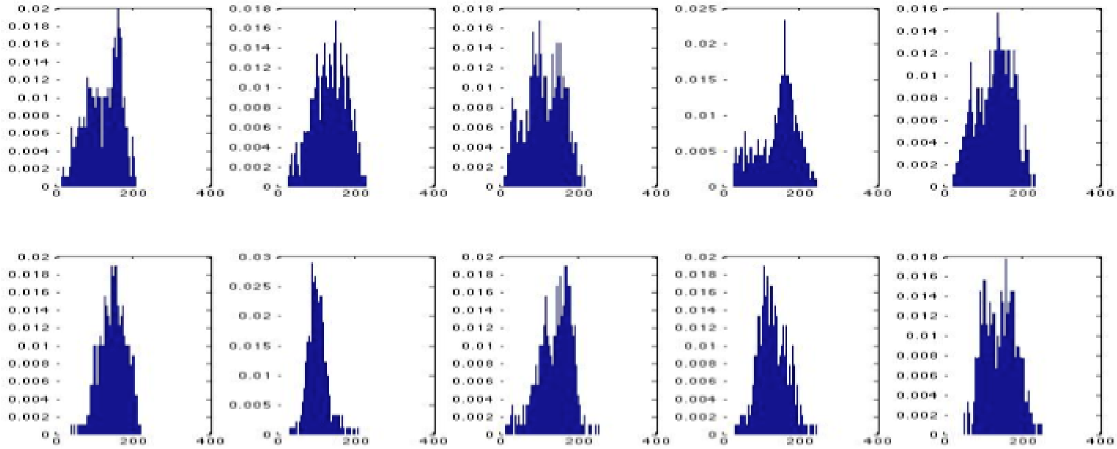


Figure 4.9: The first row shows histogram plots of neighborhoods having cracks and the second row comprises the neighborhood histograms without cracks. It is observed that the histograms having cracked regions have a greater width because of greater number of pixels closer to 0.

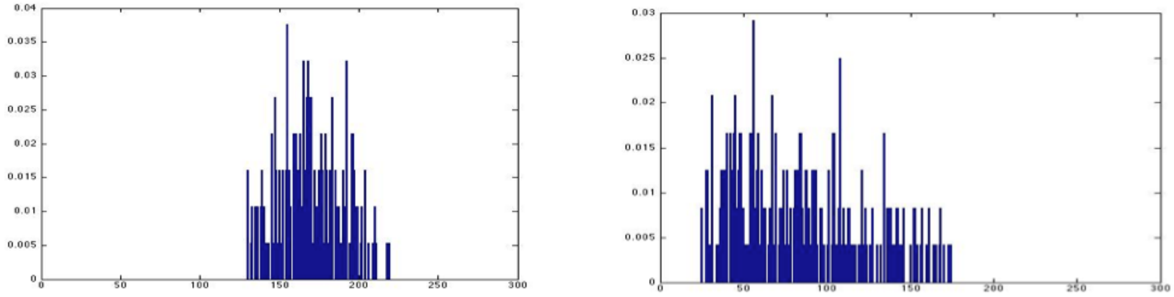


Figure 4.10: Histograms along local curves fitted to an uncracked and a cracked region. The second histogram along the cracked region is shifted to the left as compared to the one along the uncracked one because of the greater density of the darker pixels.

4.3.3 Classification results using raw intensity histogram values.

The SVM algorithm with a linear kernel function is then used for the classification purpose. Results are presented as confusion matrices. The confusion matrices are further interpreted by displaying the results on the images. The results on a test image is shown in Figure 4.11. Decisions are made per block basis as shown in Figure 4.11. The false positive rate and false negative rate were found to be 13.5% and 10.2% respectively. The classification accuracy was computed to be 76.3%.

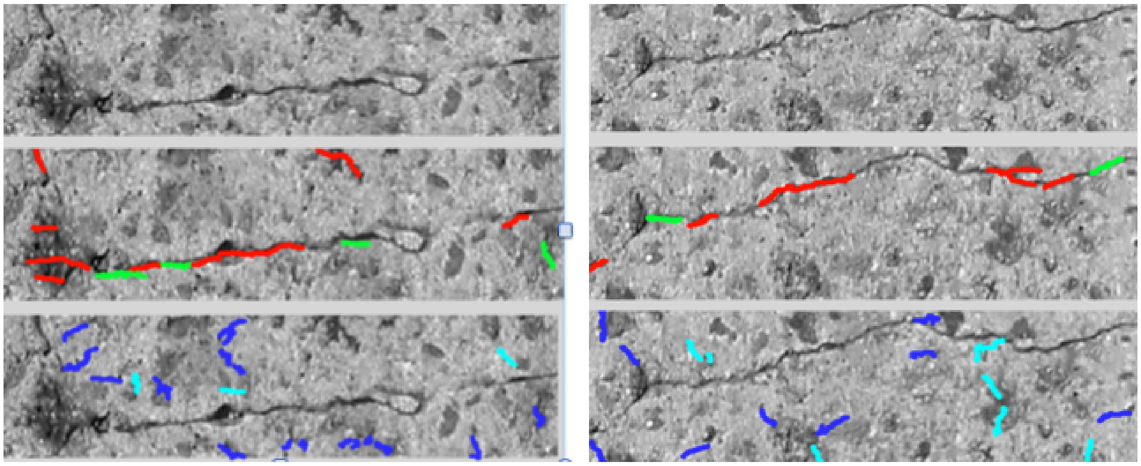


Figure 4.11: Original images along with classification results. Red lines show correctly detected cracks and blue lines show the correctly detected non-crack regions. Green lines show undetected cracks and the lines in cyan show the non-crack regions classified as cracks.

4.3.4 New feature vectors

Because of design and coverage-area constraints, a wide angle lens (10-22mm) was subsequently used instead of the 18-55mm lens. Change in camera lens resulted in a major change in its field of view (FOV). The new FOV was approximately 1.67 times the original. This resulted in poor classifier performance. So, we had to look for more robust features in the images. The following features are considered as inputs to the classifier:

- Mean (μ_{hist}) of intensity histogram along local curves (F1)
- Standard deviation (σ_{hist}) of intensity histogram along local curves (F2)

- Mean (μ_{gm}) of gradient magnitudes along local curves (F3)
- Standard deviation (σ_{gm}) of gradient magnitudes along local curves (F4)
- Ratio of the mean of intensity histogram along a local curve to the mean intensity in the block (F5)

The mean of intensity histogram, mean and standard deviation of gradient magnitudes and the ratio were found to be the best performing features. The basis of selection of these feature vectors is illustrated in Figure 4.12.

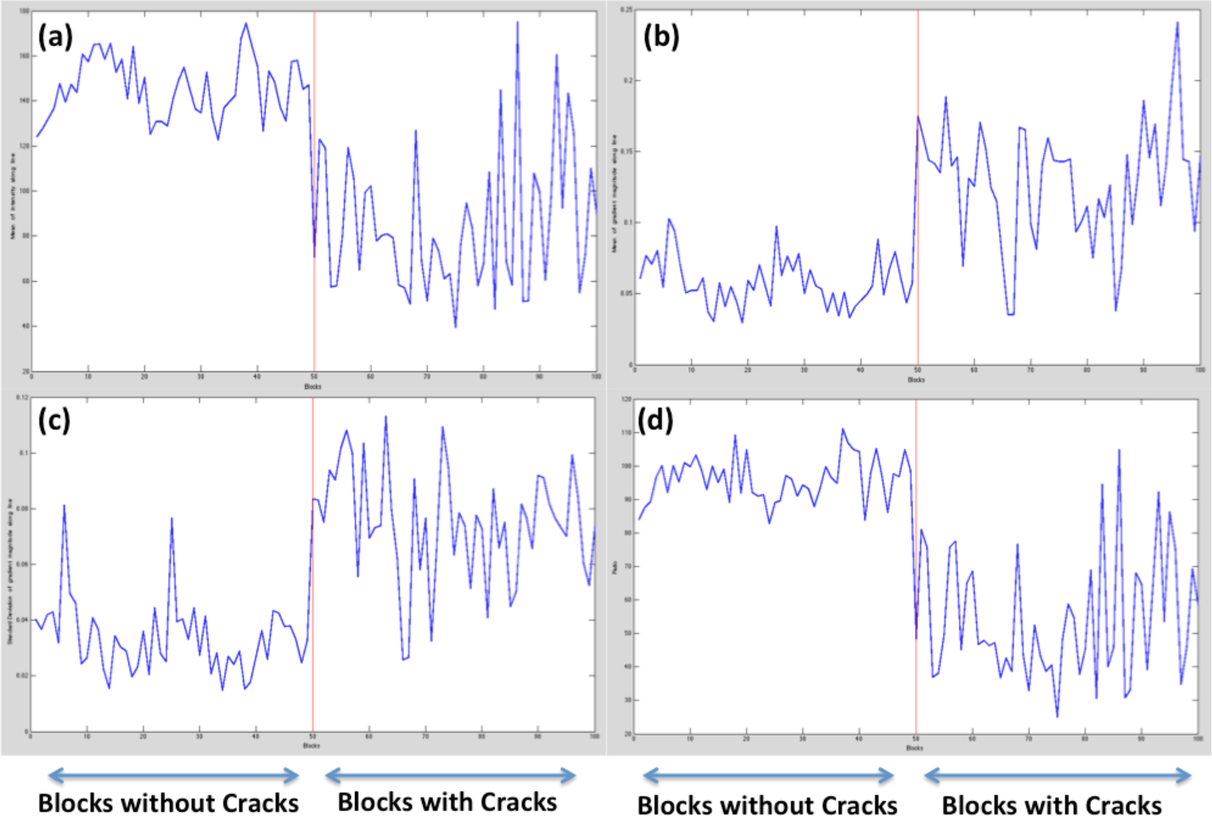


Figure 4.12: This figure shows the variation in behavior of features in blocks with and without cracks. (a) Mean of intensity along curves that lie on cracks is less than that of those which don't lie on cracks. (b) Mean of gradient magnitudes along curves that lie on cracks is greater than that of those which don't lie on cracks. (c) Standard deviation of gradient magnitudes along curves that lie on cracks is greater than that of those which don't lie on cracks. (d) Ratio of the mean of intensity histogram along a local curve is lesser in a crack region as compared to a no-crack region.

4.3.5 Application of RANSAC

We carried out experiments with curves fitted to regions using both RANSAC and least squares method. Upon analyzing the classifier performances, it was concluded that RANSAC performed much better than the least squares method. Figure 4.13 provides a visual illustration of the same. From Table 4.2 and Table 4.3, the accuracy for least squares method was found to be 76%, whereas it was 82.8% for RANSAC method. It is also found that the true positive rate increases if we consider 15×15 blocks instead of 30×30. This is because in a 15×15 block, a crack can be approximated as a straight line, more than that in a 30×30 block (where it is usually a curve).

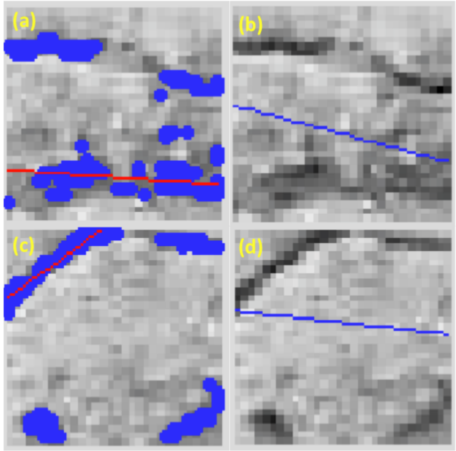


Figure 4.13: Comparison between RANSAC and least square fit of curves. Red lines in (a) and (c) shows the curves fitted to the minimum intensity points using RANSAC. Blue lines in (b) and (d) shows the curves fitted to the minimum intensity points using least squares method. It is clearly seen that the second method misses the cracks in the regions shown.

Label	**0**	**1**
0	199	51
1	69	181

Table 4.2: Confusion matrix for least-squares fit.

Label	**0**	**1**
0	225	25
1	61	189

Table 4.3: Confusion matrix for RANSAC fit.

4.3.6 Classification Results using local curve features

The following classifiers were used for evaluating the performance of the feature vectors:

- Support Vector Machines [41]
- Adaboost.M1 [42]
- Random Forests Classifier [43]

We use the features individually and in groups to evaluate the performance of each of the three classifiers mentioned above. The classifier performances in terms of the performance metrics is thoroughly detailed in Table 4.22 through Table 4.24 [44]. We first evaluate the performance on our training set. A 10-fold cross validation is used for the same. Thereafter, the classifiers are tested on the validation set of data before moving on to the test phase.

Label	**0**	**1**
0	217	33
1	43	207

Table 4.4: Confusion matrix for (F1,F2) + SVM

Label	**0**	**1**
0	175	75
1	76	174

Table 4.5: Confusion matrix for F3 + SVM

Label	**0**	**1**
0	186	64
1	84	166

Table 4.6: Confusion matrix for F4 + SVM

Label	**0**	**1**
0	180	70
1	75	175

Table 4.7: Confusion matrix for (F3,F4) + SVM

Label	**0**	**1**
0	229	21
1	52	198

Table 4.8: Confusion matrix for F5 + SVM

Label	**0**	**1**
0	228	22
1	44	206

Table 4.9: Confusion matrix for (F1,F3,F4,F5) + SVM

We then present the graphs showing the variation of true positive rates and classifier accuracies as the number of training samples is increased keeping the number of samples in the validation set intact.

Label	0	1
0	226	24
1	55	195

Table 4.10: Confusion matrix for (F1,F2) + Adaboost

Label	0	1
0	210	40
1	102	148

Table 4.11: Confusion matrix for F3 + Adaboost

Label	0	1
0	143	107
1	46	204

Table 4.12: Confusion matrix for F4 + Adaboost

Label	0	1
0	207	43
1	101	149

Table 4.13: Confusion matrix for (F3,F4) + Adaboost

Label	0	1
0	241	9
1	60	190

Table 4.14: Confusion matrix for F5 + Adaboost

Label	0	1
0	241	9
1	60	190

Table 4.15: Confusion matrix for (F1,F3,F4,F5)+Adaboost

Label	0	1
0	214	36
1	55	195

Table 4.16: Confusion matrix for (F1,F2) + RF

Label	0	1
0	167	83
1	90	160

Table 4.17: Confusion matrix for F3 + RF

Label	0	1
0	164	86
1	93	157

Table 4.18: Confusion matrix for F4 + RF

Label	0	1
0	180	70
1	94	156

Table 4.19: Confusion matrix for (F3,F4) + RF

Label	0	1
0	200	50
0	53	197

Table 4.20: Confusion matrix for F5 + RF

Label	0	1
0	220	30
1	51	199

Table 4.21: Confusion matrix for (F1,F3,F4,F5) + RF

Feature Vectors	Accuracy (%)	Sensitivity	Specificity	Precision	F-score
F1,F2	84.8	0.828	0.868	0.863	0.845
F3	69.8	0.696	0.7	0.699	0.697
F4	70.4	0.664	0.744	0.722	0.692
F3,F4	71	0.7	0.72	0.714	0.707
F5	85.4	0.792	0.916	0.904	0.844
F1,F3,F4,F5	86.8	0.824	0.912	0.904	0.862

Table 4.22: Comparison of SVM classifier performances with different feature vectors for training set data.

Feature Vectors	Accuracy (%)	Sensitivity	Specificity	Precision	F-score
F1,F2	84	0.78	0.904	0.89	0.832
F3	69.8	0.592	0.84	0.787	0.676
F4	72.7	0.816	0.572	0.656	0.727
F3,F4	85.4	0.596	0.828	0.776	0.674
F5	88.8	0.76	0.964	0.955	0.846
F1,F3,F4,F5	89.7	0.76	0.964	0.955	0.846

Table 4.23: Comparison of Adaboost classifier performances with different feature vectors for training set data.

Feature Vectors	Accuracy (%)	Sensitivity	Specificity	Precision	F-score
F1,F2	87.2	0.78	0.856	0.844	0.811
F3	69	0.64	0.668	.658	0.649
F4	68.4	0.628	0.656	0.646	0.637
F3,F4	75.6	0.624	0.72	0.69	0.655
F5	84.7	0.788	0.8	0.796	0.788
F1,F3,F4,F5	89.5	0.796	0.88	0.869	0.831

Table 4.24: Comparison of Random Forest classifier performances with different feature vectors for training set data.

The following conclusions are drawn from the experiments:

- σ_{hist} was found to be the poorest performing feature vector.
- Ratio, μ_{hist}, σ_{gm} and μ_{gm} were the best ones in order.
- RANSAC fit performed better than corresponding least squares fit.
- Random Forest classifier had better performance than SVM and Adaboost.

- 15×15 blocks instead of 30×30 resulted in improved performance.

4.3.7 Validation set evaluation

We use the 4×1 feature vector consisting of F1, F3, F4 and F5 as the four features and then evaluate our classifier on the validation set data.

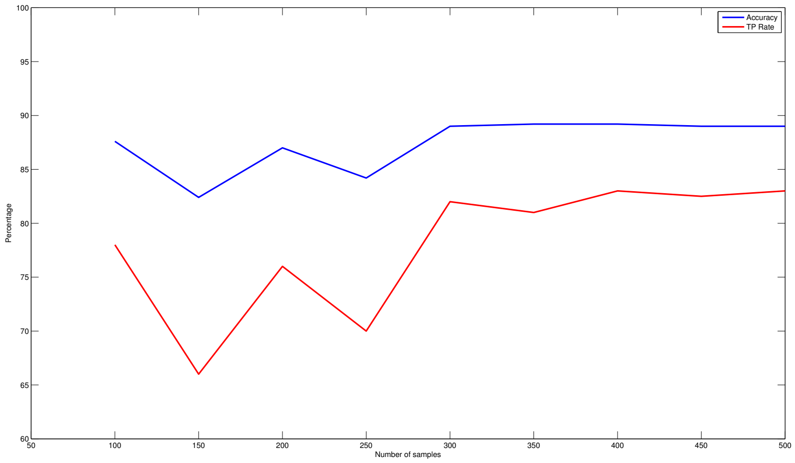


Figure 4.14: Accuracy and true positive rate of validation set vs. number of samples in training set for SVM classifier.

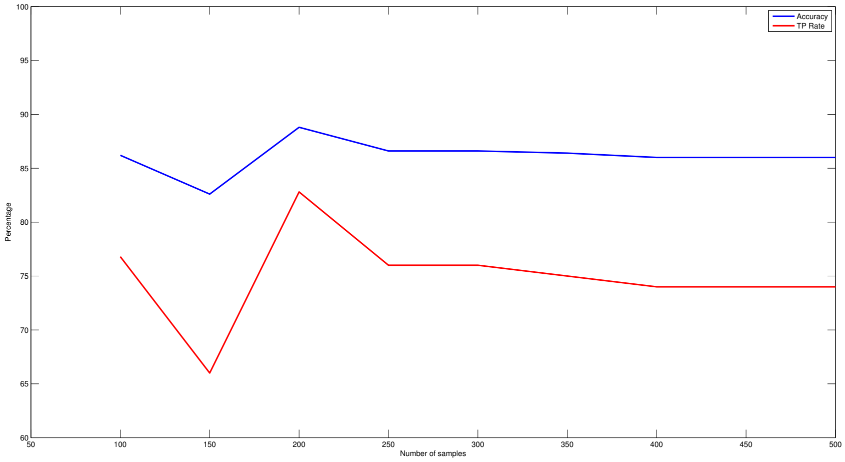


Figure 4.15: Accuracy and true positive rate of validation set vs. number of samples in training set for Adaboost classifier.

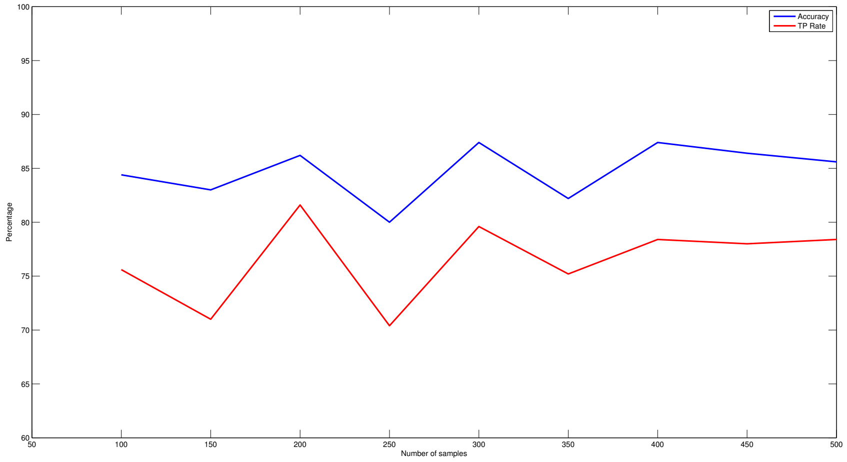


Figure 4.16: Accuracy and true positive rate of validation set vs. number of samples in training set for Random Forest classifier.

Label	**0**	**1**
0	237	13
1	42	208

Table 4.25: Confusion matrix for SVM classifier

Label	**0**	**1**
0	245	5
1	65	185

Table 4.26: Confusion matrix for Adaboost classifier

Label	**0**	**1**
0	232	18
1	54	196

Table 4.27: Confusion matrix for Random Forest classifier

Classifier	Accuracy (%)	Sensitivity	Specificity	Precision	F-score
SVM	89	0.832	0.948	0.849	0.896
Adaboost	91.2	0.74	0.98	0.974	0.841
Random Forest	92	0.784	0.928	0.916	0.845

Table 4.28: Classifier performance on the validation set data.

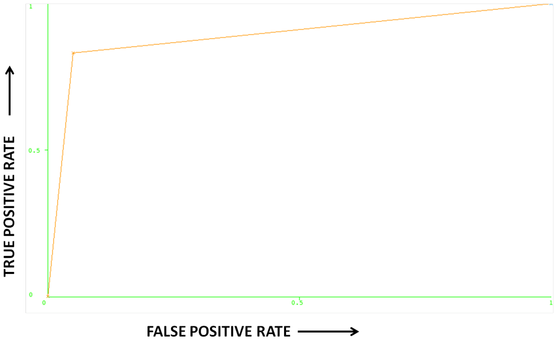


Figure 4.17: ROC curve for SVM classifier on validation set.

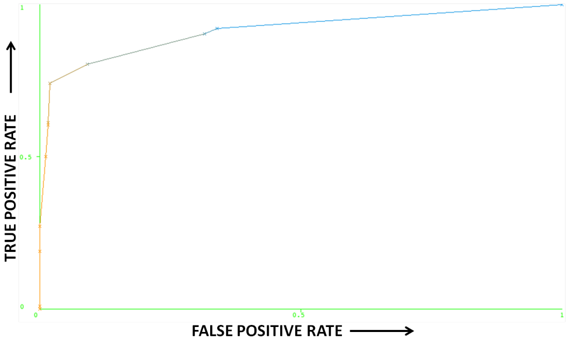


Figure 4.18: ROC curve for Adaboost classifier on validation set.

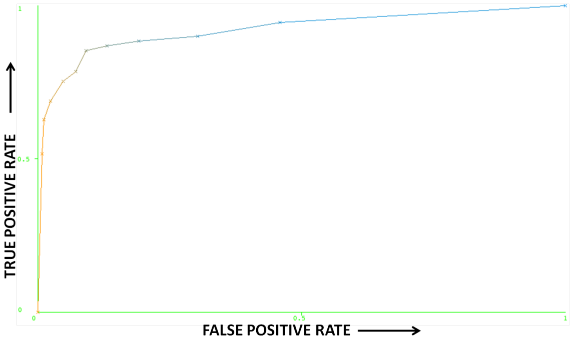


Figure 4.19: ROC curve for Random Forest classifier on validation set.

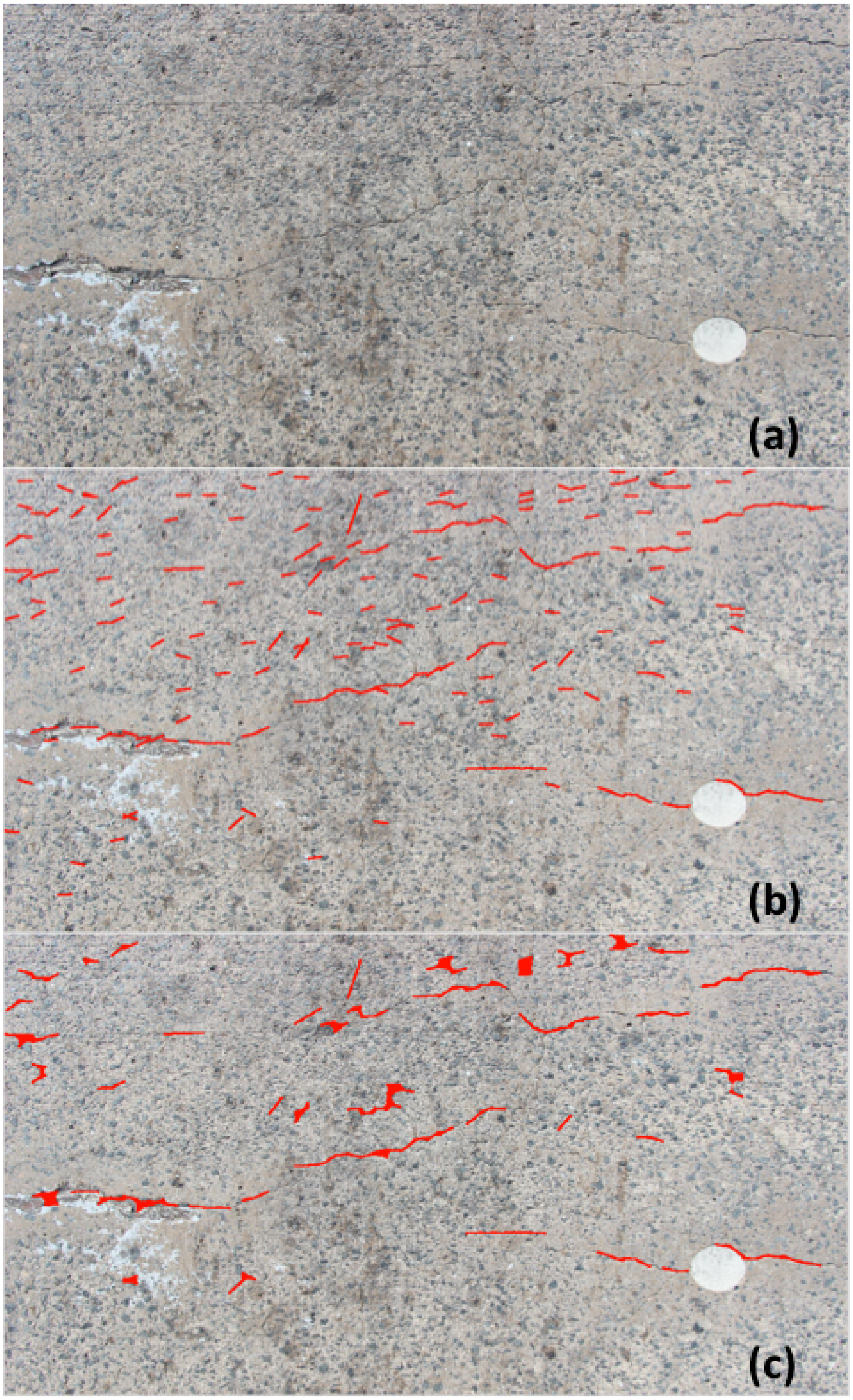


Figure 4.20: (a) Raw image from the Virginia bridge deck (b) Image showing the detected cracks (c) Most of the false positives are removed by morphological operations. In this set of operations, (b) is first converted to a binary image. The cracks are now in white against a black background. A closing operation is first performed. The holes in the resulting binary image are then filled. The curves with total pixels less than a threshold are removed. The image corresponds to a 4 ft × 3 ft section on the bridge.

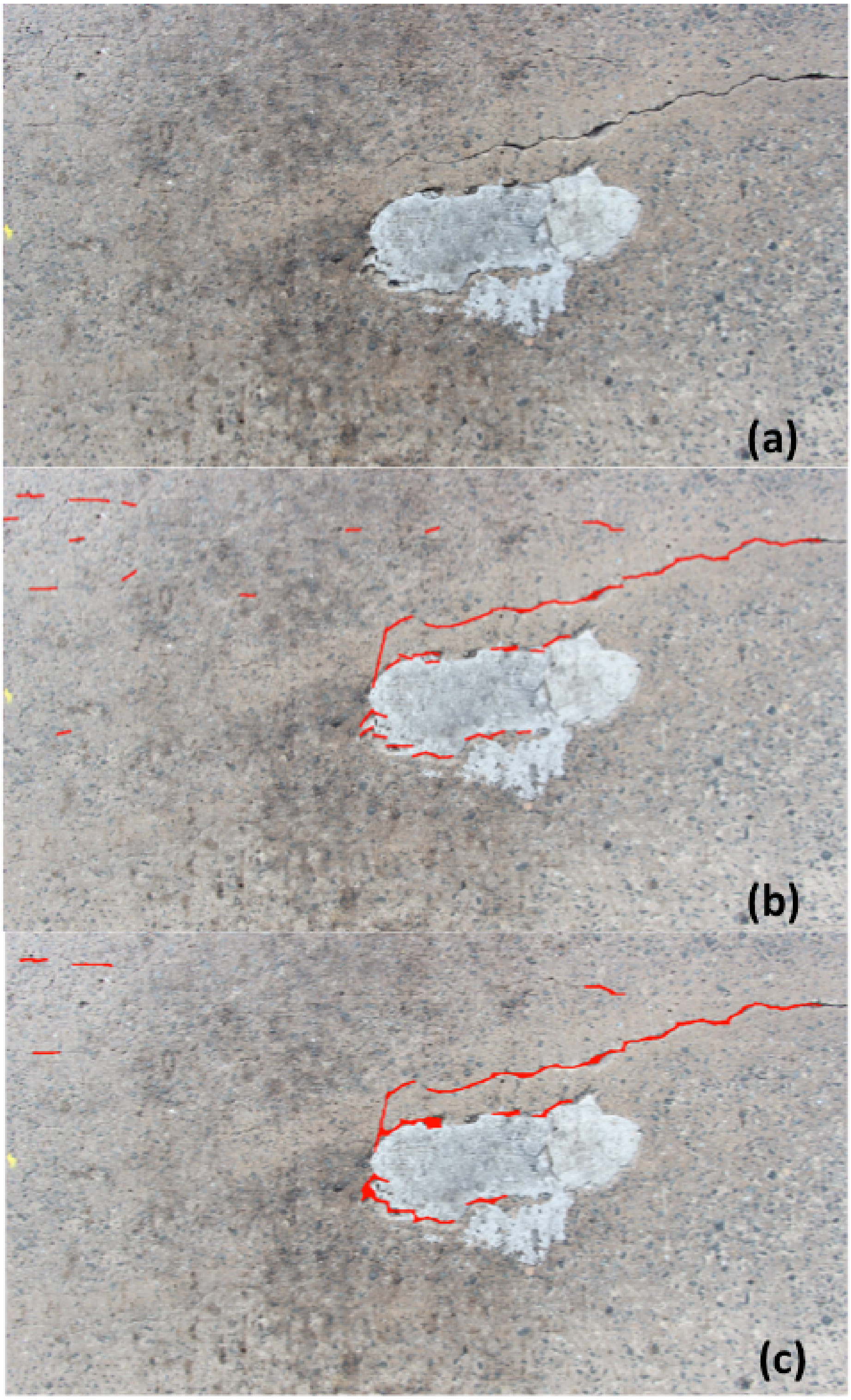


Figure 4.21: (a) Raw image from the Virginia bridge deck (b) Image showing the detected cracks (c) Most of the false positives are removed by morphological operations where curves with total pixels less than a threshold are removed. The image corresponds to a 4 ft × 3 ft section on the bridge.

4.4 Use of Laplace Pyramids

Digital images are in general both scale-variant and highly non-stationary in space. Our images of interest have cracks of varying sizes, as thin as a millimeter to as wide as a few centimeters. Uniformity in the representation and processing of visual information over multiple scales is an inherent property offered by visual systems [45] [46]. Pyramids are a classic coarse-to-fine strategy that help us search over translations.

4.4.1 Construction of pyramids

In Gaussian pyramid construction, a series of images are built, which are blurred and scaled down versions of the original image. In each smaller image, each pixel is a local average corresponding to a pixel neighborhood on a lower level of the pyramid.

The original image is first convolved with a Gaussian kernel. The resulting image is a low pass filtered version of the original image. The Laplacian is then computed as the difference between the original image and the low pass filtered image. The first Gaussian filtered image is then down-sampled. This process is continued to obtain a set of band-pass filtered images. The Laplacian pyramid is thus a set of band pass filters. Figure 4.22 illustrates the sequence of steps for obtaining successive level Laplace pyramid images.

4.4.2 Curve Fitting

We first find out the Laplacian pyramid images for a few levels. Then we up-sample the images to the original size before looking for features in them. It is seen that features appear more prominent when we look at the level ‘k’ Laplacian image. This is illustrated in Figure 4.23.

We then look at fitting curves, using RANSAC, to pixels whose pyramid values lie in the lowest 10-15 percentile in a chosen detector window. It is seen (Figure 4.24) that we can detect thinner cracks when curves are fitted to such points (scale-space extrema) as opposed to lowest intensity points as discussed earlier .

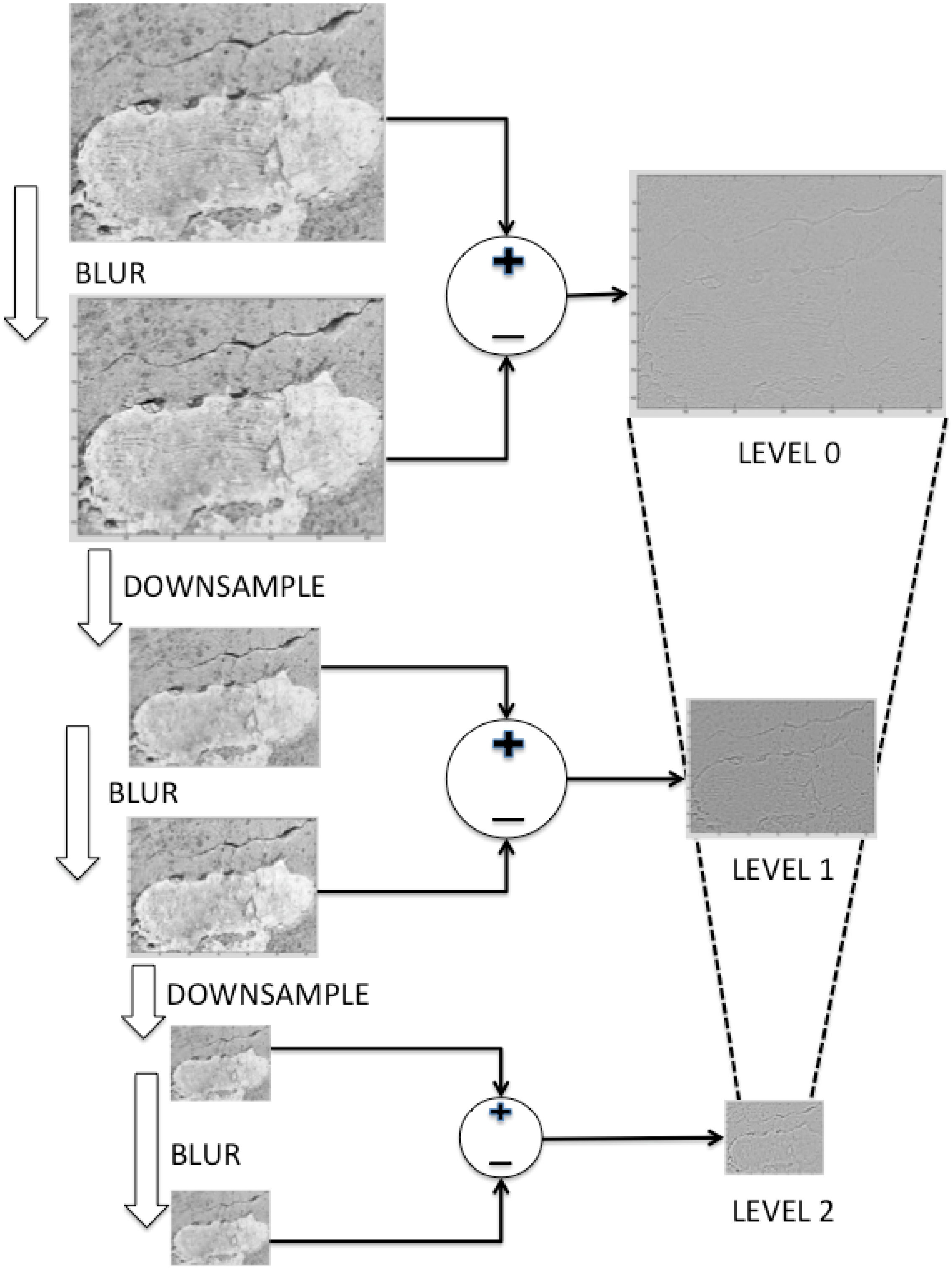


Figure 4.22: Pyramid Structure: This figure illustrates the construction of successive level image pyramids.

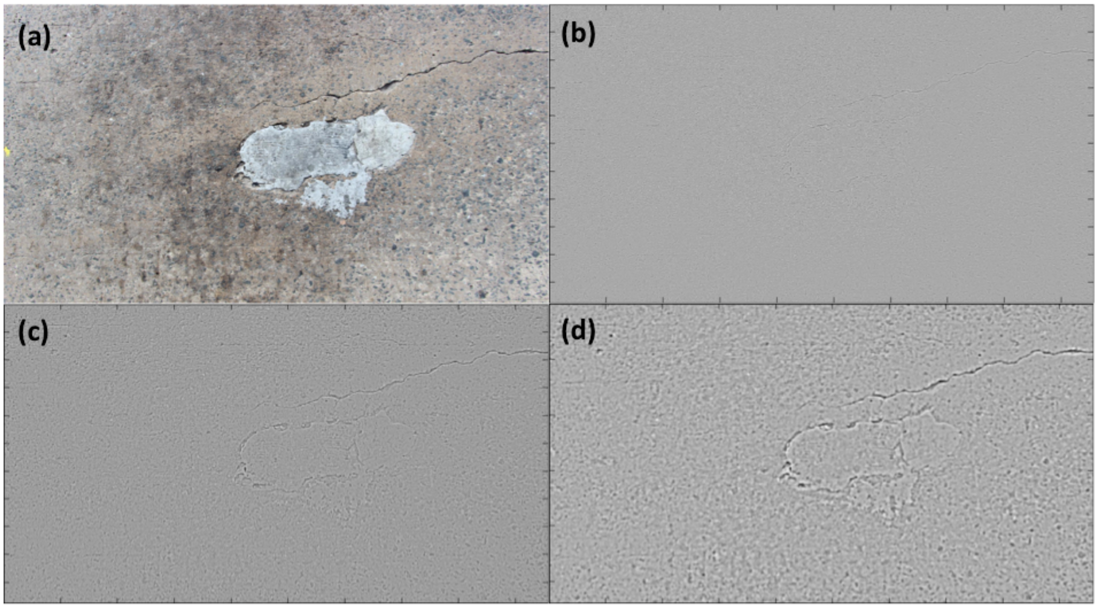


Figure 4.23: (a) Original Image (b) Level-0 Pyramid (c) Level-I Pyramid (d) Level-II Pyramid. All the pyramid levels are upsampled to the same size as the original image. It is seen that some cracks become more prominent in the coarser pyramid levels. Laplace pyramid enhances edge features, which exist at a characteristic scale.

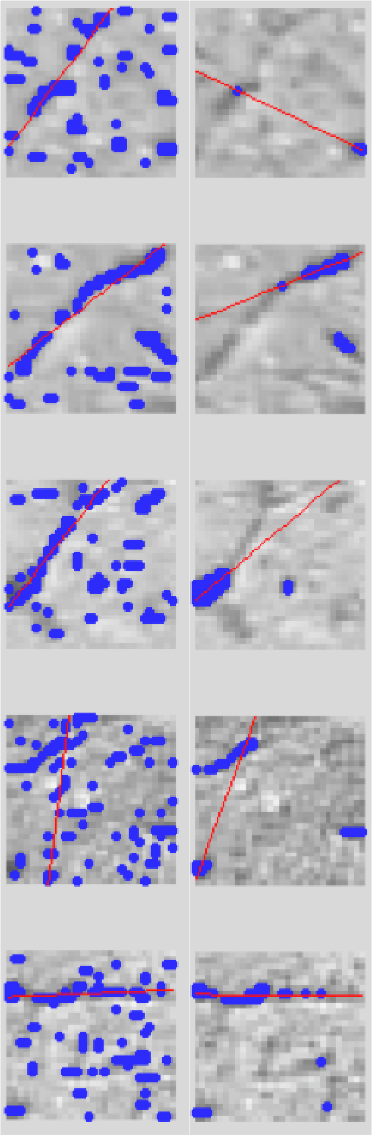


Figure 4.24: Patches on the left show the curves fitted to the points which have the lowest Laplacian Pyramid values. Patches on the right show the curves fitted to the points of minimum intensity. These patches have very fine cracks. It is seen that curves fitted to pixels having lowest Laplacian Pyramid values perform better than the other curves in detecting finer cracks.

4.4.3 Classification using laplace pyramid features

Apart from the features (F1-F5) discussed earlier in Section 4.3.4, the following new features are considered as inputs to the classifier:

- Maximum of laplace pyramid values (in the detected curve) across three levels (F6)
- Minimum of laplace pyramid values (in the detected curve) across three levels (F7)
- Mean of laplace pyramid level 1 values along a detected curve (F8)
- Mean of laplace pyramid level 2 values along a detected curve (F9)
- Mean of laplace pyramid level 3 values along a detected curve (F10)

Feature Vectors	Accuracy (%)	Precision	Recall	F-score
F6	53.2	0.55	0.38	0.46
F7	58.2	0.63	0.404	.491
F6,F7	60	0.629	0.488	0.55
F8,F9,F10	61	0.663	0.448	0.535
F6,F7,F8,F9,F10	64.6	0.748	0.44	0.554
F1-F10	91.6	.93	.9	.915

Table 4.29: Comparison of SVM classifier performances with different Laplace pyramid feature vectors

Classifier	Accuracy (%)	Precision	Recall	F-score
SVM	91.6	.93	.9	.915
Adaboost	93.7	0.89	0.904	0.897
Random Forest	94.7	0.911	0.9	0.905

Table 4.30: Classifier performance on the validation set data using F1-F10. It is evident from the table that a combination of the new features with the old ones results in an increase in accuracy for all three classification algorithms.

The next set of experiments involved training and testing our classifier on different bridge datasets. For this purpose, we constructed training and test sets for the bridges in both Virginia and California. Each of the two bridges had 500 samples with equal positive and negative instances. We first evaluated the classifiers with features F1-F5 and then F1-F10. The results are presented in Table 4.31 and Table 4.32.

Classifier	Accuracy (%)	Precision	Recall	F-score
SVM	51.6	0.786	0.044	0.083
Adaboost	68.8	0.857	0.528	0.653
Random Forest	73.7	0.864	0.408	0.554

Table 4.31: Classifier performance with Virginia bridge dataset as the training set and the California bridge dataset as the test set with F1-F5 as features.

Classifier	Accuracy (%)	Precision	Recall	F-score
SVM	64.8	0.603	0.864	0.711
Adaboost	83.5	0.646	0.868	0.741
Random Forest	62.4	0.589	0.792	0.676

Table 4.32: Classifier performance with California bridge dataset as the training set and the Virginia bridge dataset as the test set with F1-F5 as features.

Classifier	Accuracy (%)	Precision	Recall	F-score
SVM	50	-	-	-
Adaboost	67.1	0.902	0.148	0.254
Random Forest	64.5	0.883	0.332	0.483

Table 4.33: Classifier performance with Virginia bridge dataset as the training set and the California bridge dataset as the test set with F1-F10 as features.

4.5 Classification using Radon Transforms as feature vectors

A new set of feature vectors considered for building the classifier consists of radon transforms of the intensity images [47]. Radon transform of an image f(x,y) along a radial line for a specific set of angles, is the projection of the image along those angles. The resulting projection is the line integral of the pixels in each direction. It maps cartesian rectangular co-ordinates to polar co-ordinates.

Classifier	Accuracy (%)	Precision	Recall	F-score
SVM	79.6	0.723	0.96	0.825
Adaboost	90.8	0.723	0.952	0.822
Random Forest	76.3	0.685	0.676	0.672

Table 4.34: Classifier performance with California bridge dataset as the training set and the Virginia bridge dataset as the test set with F1-F10 as features.

Mathematically, radon transform is defined by

$$\rho = x \cos\theta + y \sin\theta, \tag{4.6}$$

where ρ and θ are the parameters in the cylindrical co-ordinates. The radon transform R is given by

$$R(\rho, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y)\delta(\rho - x\cos\theta - y\sin\theta)\, dx\, dy \ , \tag{4.7}$$

where $\delta(.)$ is the Dirac Delta function.

In order to effectively calculate radon transforms, we first need to segment the original image and then detect the edges [48]. Figure 4.25 justifies the selection of radon transforms as a possible feature vector. The radon transforms are viewed as images. Strong peaks in the images give an idea about the presence and the type of cracks in the regions.

Figure 4.25 illustrates that apart from detecting the presence of cracks, radon transforms may be used to classify the crack into different categories like vertical, horizontal and oblique ones.

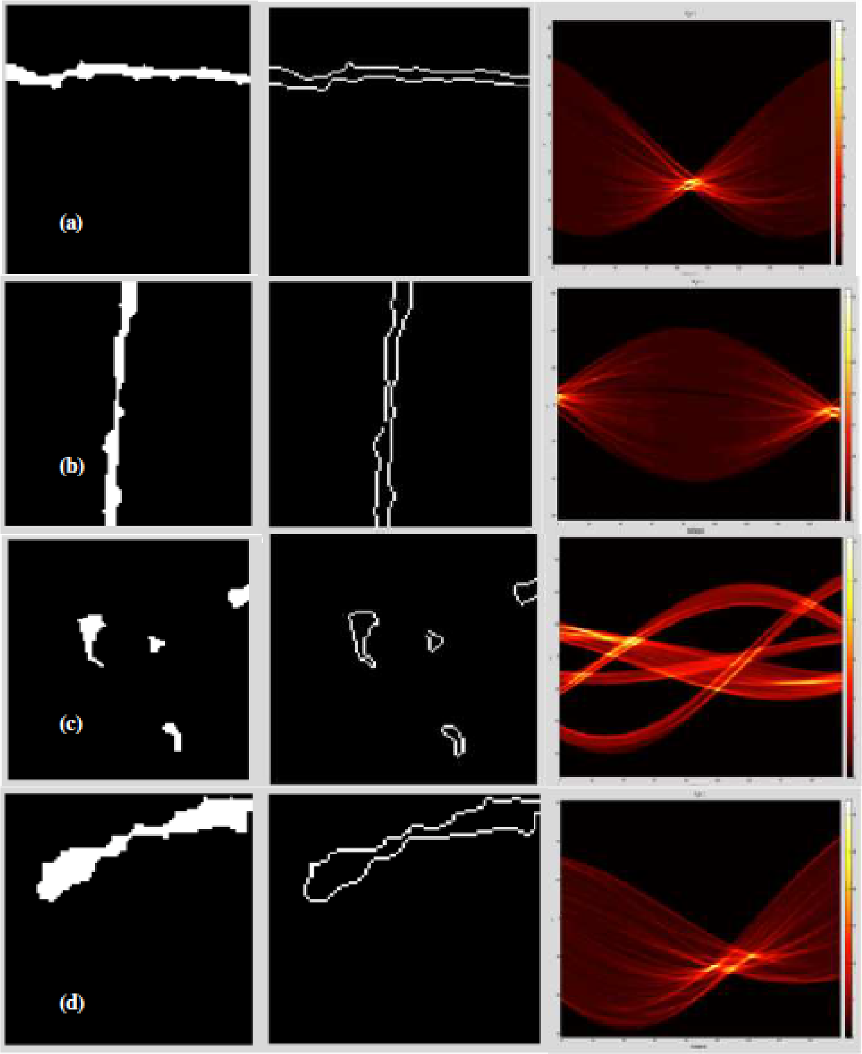


Figure 4.25: (a) The strongest peaks in R corresponding to an angle of 90 degree which shows the presence of horizontal cracks. (b) The strongest peaks in R corresponding to an angle of 179 degree and 0 degree, which shows the presence of vertical cracks. (c) The strong peaks are present at many angles, which shows that the artifacts are noise and not real cracks. (d) Strongest peaks concentrated around 120 degrees which denotes oblique cracks.

4.5.1 Feature vector extraction

Figure 4.26 enlists the steps followed in order to generate the feature vectors for classification. The original image is first converted to grayscale after which median filtering is applied. Then the original image is subtracted from the filtered image. Gaussian filtering operation is applied followed by morphological closing operation. Unconnected components having fewer pixels are then removed.

We need to compute the projection of an image matrix along specified directions. The radon transform of an image for 0-179 degrees has 180 columns in total. The maximum magnitude of the radon transforms of each column is computed to form a 180×1 vector. In Figure 4.25(a), an element around the 90^{th} entry would have the maximum value. Similarly, in Figure 4.25(b), an element around the 1^{st} and the last entry would have the maximum value. The block size considered here is 100×100.

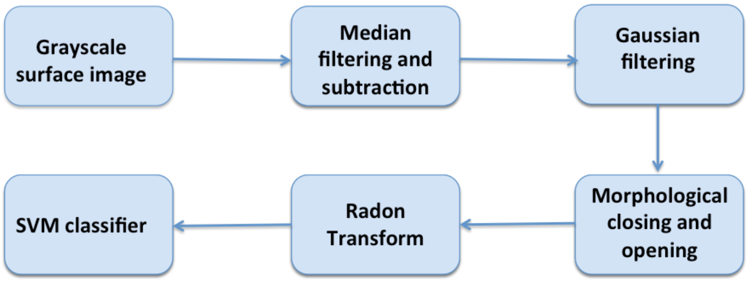


Figure 4.26: Crack-detection algorithm using radon transform features.

4.5.2 Classification results using Radon Transforms

The validation set consisted of 400 blocks with equal number of positive and negative samples. The accuracy was found to be 90%. The false positive and true positive rates were found to be 10% and 87% respectively.

Label	0	1
0	180	20
1	26	174

Table 4.35: Confusion Matrix for classification using radon transform features (with SVM)

4.6 Classification using area of segmented regions as feature vectors

The second column in Figure 4.25 shows the segmented images of 100 ×100 square patches. Another feature vector is constructed considering only number of white pixels in each block. The number of white pixels in the regions of interest is the area occupied by the same. It is clearly seen from Figure 4.27 that white pixels occupy a larger area in blocks that have cracks in them as compared to blocks that do not have cracks or blocks that predominantly have noise in them. Using SVM classifier, and areas of segmented regions as feature vectors, the accuracy was found to be 94 % .

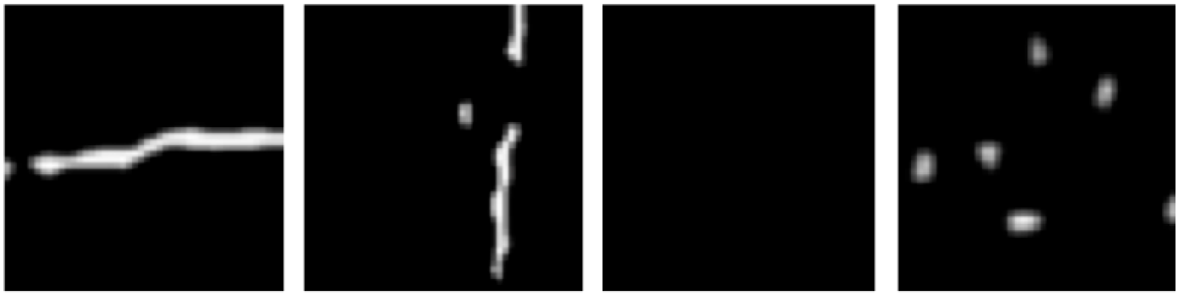


Figure 4.27: The four patches correspond to four regions: horizontal crack, vertical crack, no crack and noise. The area of the white regions in the patches is used as a feature for classification.

4.7 Histogram of Gradient Orientations

Both cracked and uncracked regions of interest are considered and the gradient magnitudes of all the pixel points are computed. Experimentally it is found that crack regions occur in the top ten percentile of the gradient magnitudes. These are now the new points of interest. The next step is the computation of gradient orientations of the points of interest.

Let I be the image whose gradients along the x and y directions need to be computed as

$$\nabla I = \frac{\partial I}{\partial x}\hat{\imath} + \frac{\partial I}{\partial y}\hat{\jmath}, \tag{4.8}$$

where, $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$ are the gradient magnitudes along the x and the y axes respectively denoted by I_x and I_y. The gradient magnitude $|\nabla I| = \sqrt{I_x^2 + I_y^2}$. Once the gradient magnitudes along the required two co-ordinate axes are calculated, the gradient orientation θ of each pixel point can be calculated as

$$\theta = \tan^{-1} \frac{I_y}{I_x}. \tag{4.9}$$

After obtaining the gradient orientations of all the points of interest, they are binned into histograms that span 0 to 360 degrees. The entire histogram is divided into twenty bins, each encompassing 18 degrees.

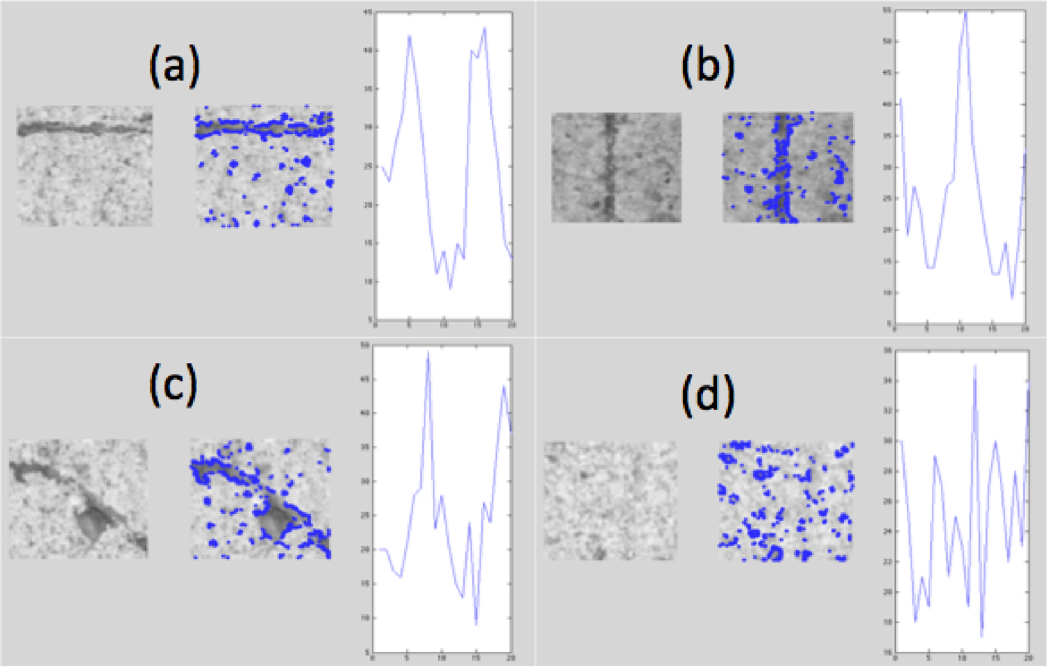


Figure 4.28: Blue dots are the pixel points whose gradient magnitudes lie in the top ten percent of the region. The figure in the third column shows the histogram of gradient orientations of the pixel points marked in blue. (a) Horizontal crack (b) Vertical crack (c) Oblique crack (d) No crack

Figure 4.28 shows the histogram of gradient orientations of four patches, namely horizontal crack, vertical crack, oblique crack and uncracked regions. When the crack is horizontal, peaks are observed around the 5^{th} and 15^{th} bins corresponding to 90 and 270 degrees. In case of a vertical crack, peaks are observed around the 0^{th}, 10^{th} and 20^{th} bins corresponding to 0, 180 and 360 degrees. For an oblique crack, two peaks are observed at other angles. On the contrary, when there are no cracks, peaks form at random locations. The location of the peaks hence becomes an inherent feature for the classifier. The feature vector consists of the binned histogram values in the form of 20×1 vectors.

4.7.1 Classification with histogram of gradient orientation features

The validation set consisted of 700 blocks with equal number of positive and negative samples. The accuracy was found to be 78.7%. The false positive and true positive rates were found to be 12.86% and 70.29% respectively. The number of validation blocks in the latter two methods is lesser than the first because the detector windows used in the first method is smaller (15×15) as compared to the ones (100×100) used in the other methods.

Label	0	1
0	305	45
1	104	246

Table 4.36: Confusion Matrix for classification using histogram of gradient orientation (with SVM)

Feature Vectors	Classifier	Accuracy
Histogram along ROIs	SVM	75
F1-F5	SVM	89
F1-F5	Adaboost	91.2
F1-F5	Random Forest	92
F1-F10	SVM	91.6
F1-F10	Adaboost	93.7
F1-F10	Random Forest	94.7
Histogram of gradient orientations	SVM	78.7

Table 4.37: Comparison of methods

Chapter 5

Deck-mosaicing: Methods and Results

The high-resolution surface images need to be stitched to form a coherent spatial mosaic of the deck. This would form a comprehensive tool, allowing the bridge inspector to analyze the surface condition visually, without being physically present on-site.

5.1 Image stitching

The stitcher has three major components:

1. Obtaining matched points
2. Estimation of homography H
3. Warping to align images

5.1.1 Obtaining matched points

Given two images that have a certain degree of overlap between them, the stitching algorithm first needs to identify matches between the two images. Extraction and matching of features between images is thus the first step. We can consider SIFT, SURF or ORB features for this purpose [49] [50] [51]. ORB features can be extracted in lesser time as compared to the other two. The percentage of correct matches is maximum for SIFT features. When overlap between successive images was less, SIFT was found to be more robust. In this project, we make use of SIFT features. We refer to matches as *keypoints*. There is overlap only between successive images. So during the stitching process, we need to check for keypoints only between images *n-1*, *n* and *n+1*. Since SIFT features are rotation and scale invariant, our system can handle images with varying orientation and zoom. Figure 5.1 shows the SIFT matches between two successive surface images. From Figure 5.1, we see a number of *improper* matches.

These matches should be removed in order to avoid erroneous results while calculating homography.

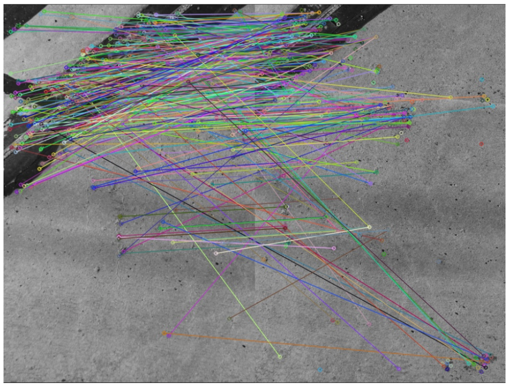


Figure 5.1: SIFT Matches between two successive images captured by the robot before RANSAC.

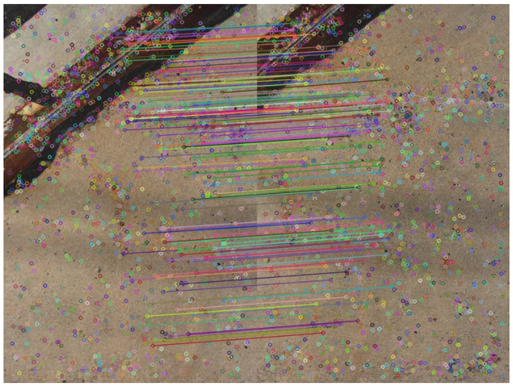


Figure 5.2: SIFT Matches between two successive images captured by the robot after RANSAC.

RANSAC is then used to select a set of inliers from amongst the matches. These inlier points are then used to calculate the homography [52]. The inlier matches are shown in Figure 5.2.

5.1.2 Calculating homographies

As seen in Figure 5.3, homography relates points in the source frame to the points in the destination frame. In other words, it relates the pixel co-ordinates in two images.

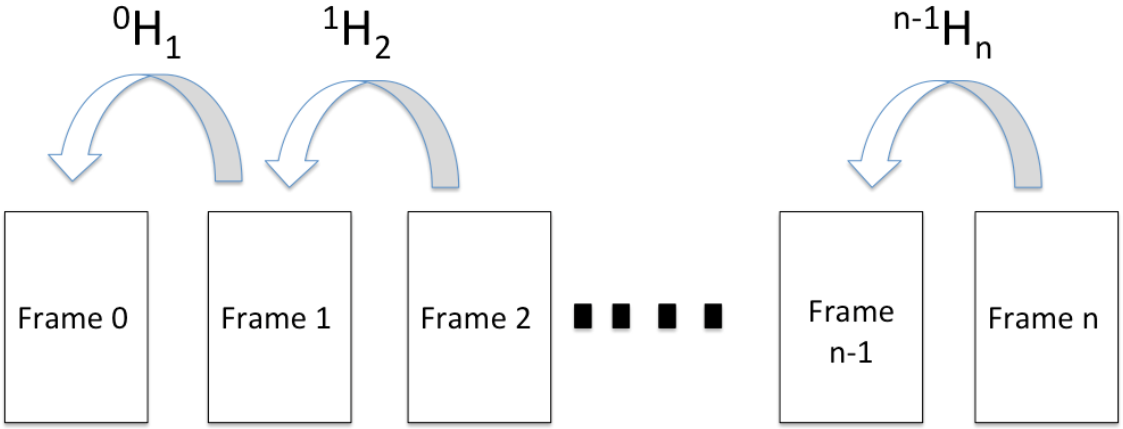


Figure 5.3: Inter-frame homographies

The homography H is a 3×3 matrix given by

$$\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}.$$

The i^{th} keypoint in the j^{th} frame when transformed to the 0^{th} frame $^{0j}P_i$ is related to the i^{th} keypoint in the j^{th} frame by

$$^{0j}P_i =^0 H_j \times^j P_i, \tag{5.1}$$

where j (= 0,1,2,3....M) is the frame number, i is the point pair index and ^0H_j is the concatenation of homographies given by equation 5.2. The homography between frame j and frame *0* is the concatenation of the intermediate homographies given by

$$^0H_j =^0 H_1 \times^1 H_2 .... \times^{j-1} H_j. \tag{5.2}$$

Significant improvement is obtained by running the inter-frame homography estimation using source points that are already transformed to the 0^{th} coordinate frame. Using ^0H_n, points are reprojected from frame n to *0*. Then the points in frame *n+1* and the reprojected points (from n to *0*) are used to obtain $^{n+1}H_0$.

This is how we calculate the homographies from frame n (n varies from *0* to M) to frame *0*. This results in the reduction of cascade error.

5.1.3 Warping

For a pixel in the destination co-ordinate frame, the homography matrix is used to find out the corresponding location of pixel in the source image. The source pixel-coordinate (x_{src}, y_{src}) is related to the destination pixel-coordinate (x_{dst}, y_{dst}) by

$$[x_{src} \quad y_{src} \quad 1]^T =^1 H_0 \times [x_{dst} \quad y_{dst} \quad 1]^T. \tag{5.3}$$

To find out the value of $I(x_{src}, y_{src})$, we make use of bilinear-interpolation in the source image [53]. This value is then taken up by $I(x_{dst}, y_{dst})$.

5.2 Error in stitching

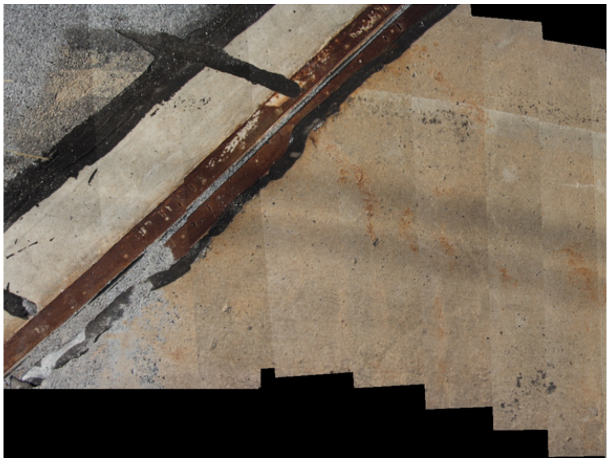


Figure 5.4: Improper stitching due to reprojection and distortion error.

As we see from Figure 5.4, the stitched image has a certain error associated with it. The error in the stitching has two components

1. Re-projection error
2. Distortion error

5.2.1 Reduction of reprojection error

Despite the selection of proper inlier points, there is a small degree of residual reprojection error that needs to be minimized. This error, though small for two frames, increases when homographies are concatenated. The reprojection error Err_{reproj} is given by

$$Err_{reproj} = \sum_{j=0}^{M} \sum_{i=1}^{N_j} ({}^{0(j-1)}P_i -{}^{0j} P_i)^2, \tag{5.4}$$

where N_j is the number of keypoints in the j^{th} frame and M is the number of frames.

The homography matrix needs to be reworked so that the term in equation 5.4 is minimized.

5.2.2 Reduction of distortion error

The geometric distortion induced by the homographies needs to be quantified. We see from Figure 5.5(a) that there is a perspective distortion.

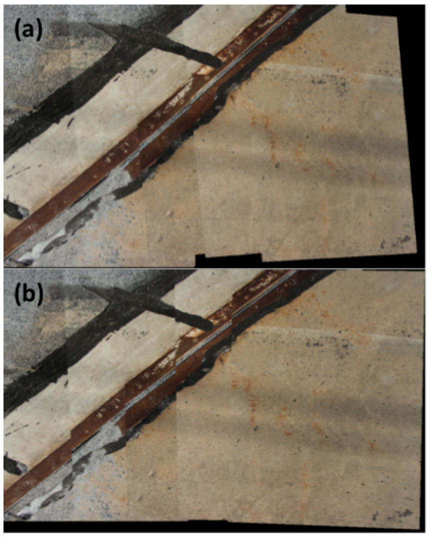


Figure 5.5: (a) Stitched images before non-linear optimization step.(b) Stitched images after non-linear optimization step. There are eight images stitched together in this figure.

To ensure that the homographies do not introduce any such distortion, the original rectangular shape needs to be preserved [54]. The distortion term Err_{dist} that needs to be minimized can be formulated as

$$Err_{dist} = ||H[1,0,0]^T - [1,0,0]^T||^2 + ||H[0,1,0]^T - [0,1,0]^T||^2. \tag{5.5}$$

Total error Err_t is the sum of both the errors given by

$$Err_t = Err_{reproj} + \alpha Err_{dist}. \tag{5.6}$$

Thus, the final optimization step is necessary to reduce the total error and find out the proper value of the tuning parameter α that leads to the best result. Levenberg-Marquardt algorithm is used for this non-linear optimization step. Figure 5.5 shows the outputs before and after non-linear optimization. The results for three different values of α are shown in Figure 5.6.

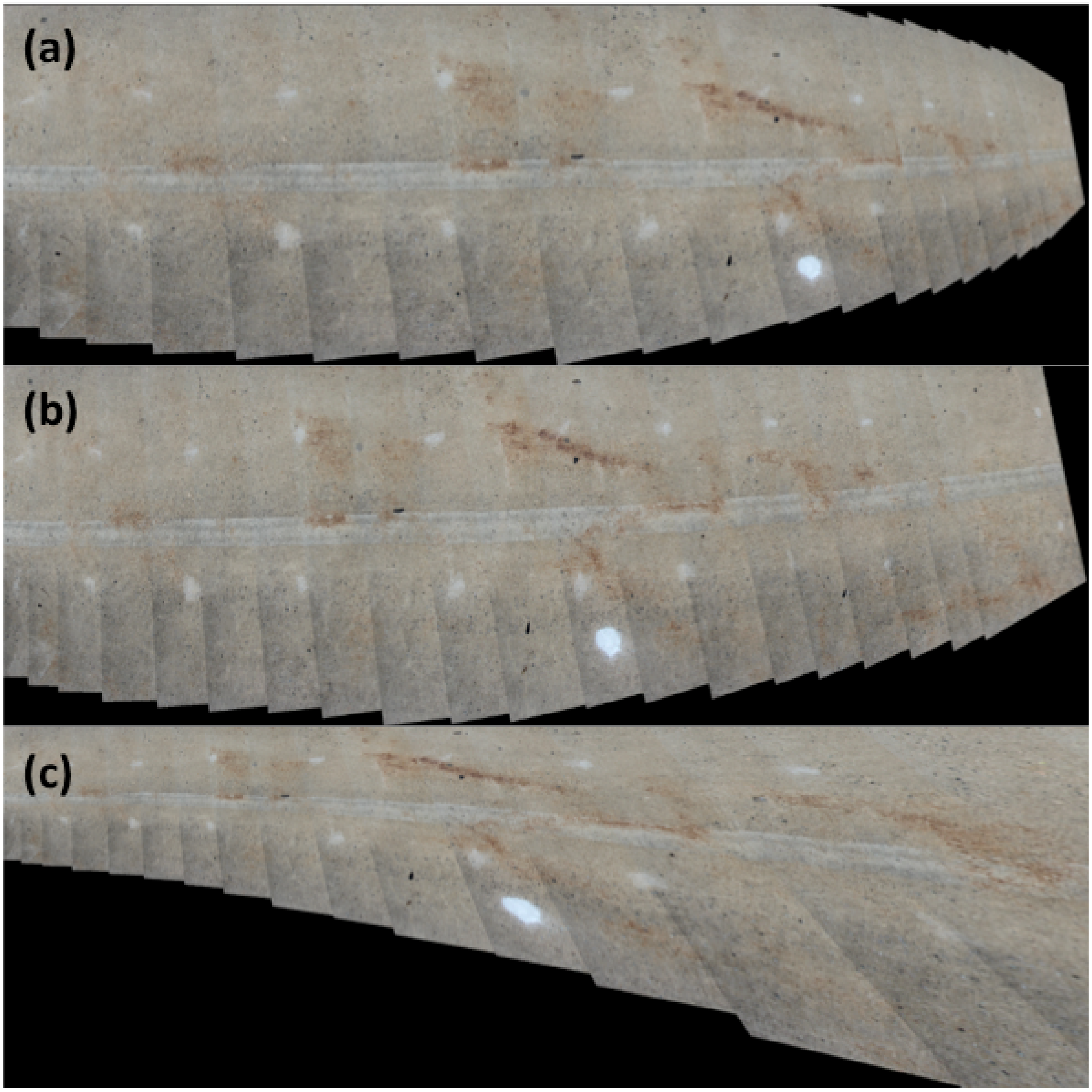


Figure 5.6: Result of 21 images stitched together with different values of α. (a) $\alpha = 10^3$ (b) $\alpha = 10^5$ (c) $\alpha = 10^7$.

5.3 Panorama Tools Graphical User interface

Panorama Tools Graphical User interface (PTGUI) is an automated photo-stitching application [55]. Its user-friendly interface and batch-processing mode makes it convenient to stitch the photos captured using the robot cameras.

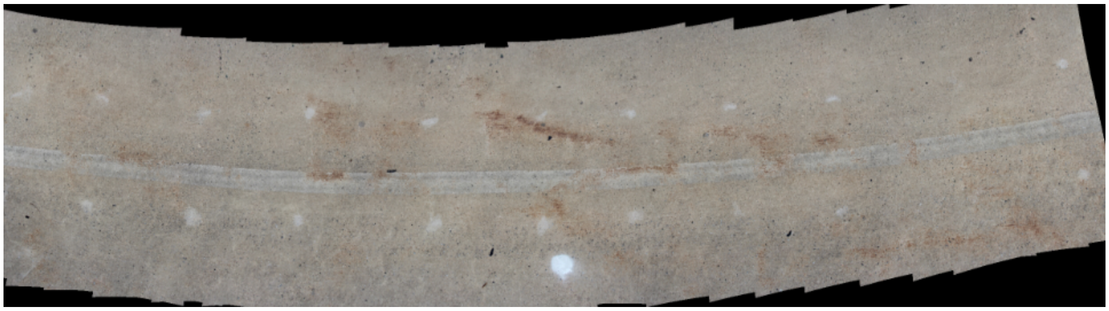


Figure 5.7: Images stitched using PTGUI photo-stitching application. It carries out optimization in all frames as opposed to the two-frame optimization discussed in Section 5.2.

Chapter 6

Image Unwrapping and Crack-density Maps

6.1 Panoramic Image Unwrapping

Standard cameras, including video cameras, have a limited field of view. To enhance the field of view, keeping the viewing point constant, we make use of an omnidirectional camera mounted on the robot [56]. The robot collects 360 ° images of the surroundings every 10 feet of travel. Images are taken at various positions of the bridge so as to enable the detection of location of a particular feature of interest with respect to the global scene. The MATLAB program constantly scans the folder on the robot computer. As soon as a new image is encountered, it is transmitted to the base station computer and unwrapped.

The unwrapped image is then displayed in a browser with pan and zoom functionalities in both auto and manual mode. One such raw and unwrapped image is shown in Figure 5.1. This image has been collected on the Virginia bridge. When unwrapped, the center of the image gives us an idea about the location of the robot on the bridge. In future, it can also serve the purpose of assisting robot navigation as it gives a full visual representation of the scene from the robot's perspective.

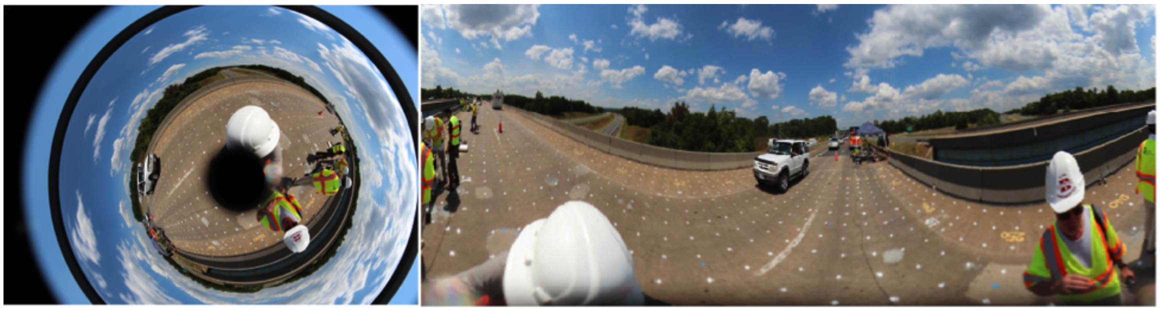


Figure 6.1: Raw and unwrapped panoramic image.

6.2 Crack Density Maps

A pixel-by-pixel computation is required to find out crack density regions in an image. The crack density maps give us a detailed overview about the surface degradation of the bridge deck. Algorithm 6.2.1 gives the pseudocode for developing a density map from a crack-labeled image as shown in Figure 4.21. The integral image at (x, y) is computed by calculating the sum of the pixel values above and to the left of (x, y). Integral image $int(x, y)$ is given by

$$int(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y'), \tag{6.1}$$

where $i(x, y)$ is the input image.

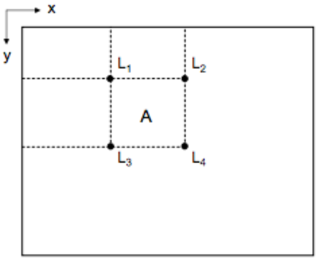


Figure 6.2: Region A is computed using the following four array references: L4+L1-(L2 + L3).

```
Algorithm 6.2.1: DENSITY($Crack_{image}$)

$Crack_{image}$ to $Crack_{binary}$;
$I_{filt} = (Crack_{binary}$ * 2D Gaussian filter);
comment: Appropriate zero-padding required;
$Crack_{integral} = \text{IntegralImage}(I_{filt})$ ;
for $i \leftarrow 1$ to $num_{rows}$
  do for $j \leftarrow 1$ to $num_{cols}$
       do $Crack_{map}(i,j) = Crack_{integral}(i-30, j-30)+$
          $Crack_{integral}(i+30, j+30) - Crack_{integral}(i-30, j+30)$
          $-Crack_{integral}(i+30, j-30)$;
return $(Crack_{map})$
```

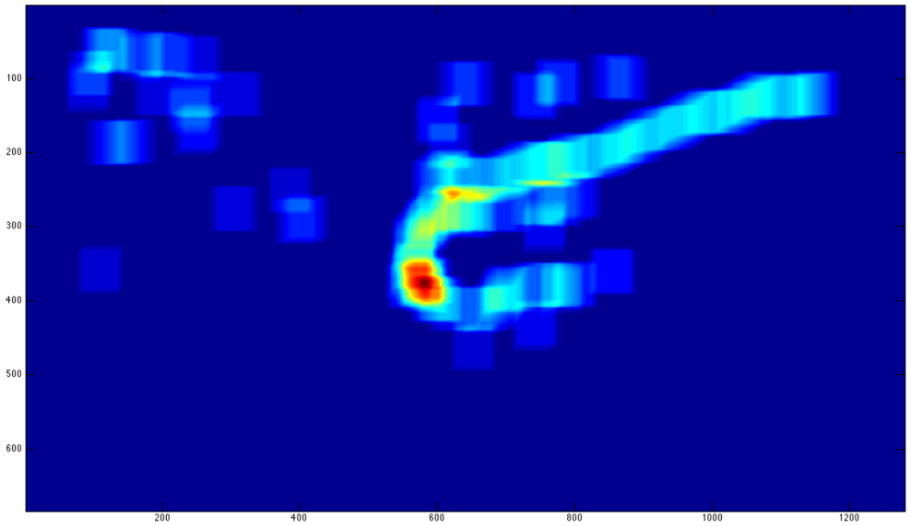


Figure 6.3: Crack density map of crack-labeled image shown in Figure 4.21. This colormap can show various levels of degradations indicated by the colors. For example, red color implies that the pixel is associated with the most number of cracks whereas a dark blue would indicate no cracks.

Chapter 7

Conclusion and Future Work

In this thesis, we have proposed an automated bridge inspection method for detection of cracks on concrete surfaces. We discussed novel ways of constructing appropriate feature vectors that would enable the classifier to distinguish between *good* and *bad* regions on the bridge in complicated datasets. Additionally, our analytical results prove that detection rate is high even when there are predominantly unwanted artifacts surrounding such cracks. Cracks as thin as 2mm in width can be detected by our algorithm. Future work involves training and testing our system on a larger database that would include images from more bridges and with different types of cracks. This system would contribute in making surface health monitoring of bridges more robust, streamlined and infallible.

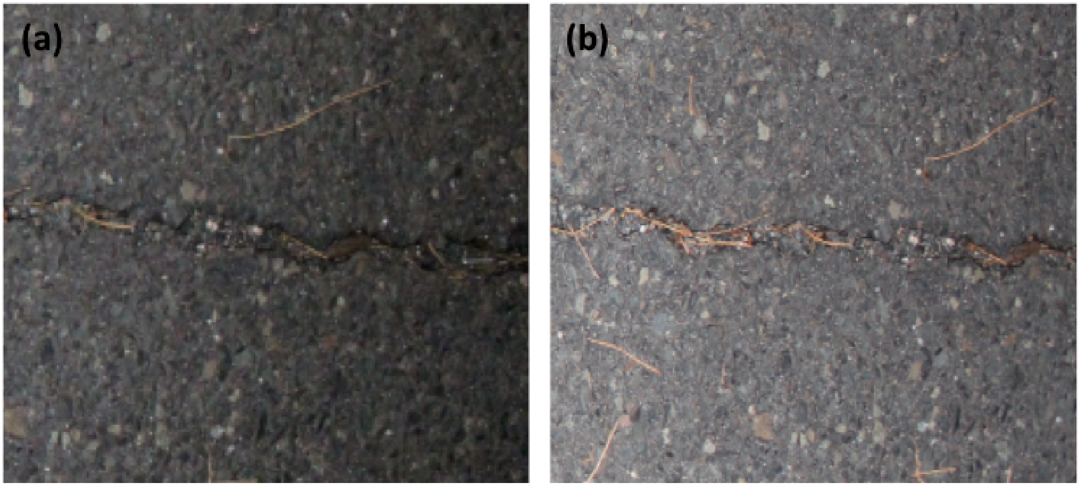


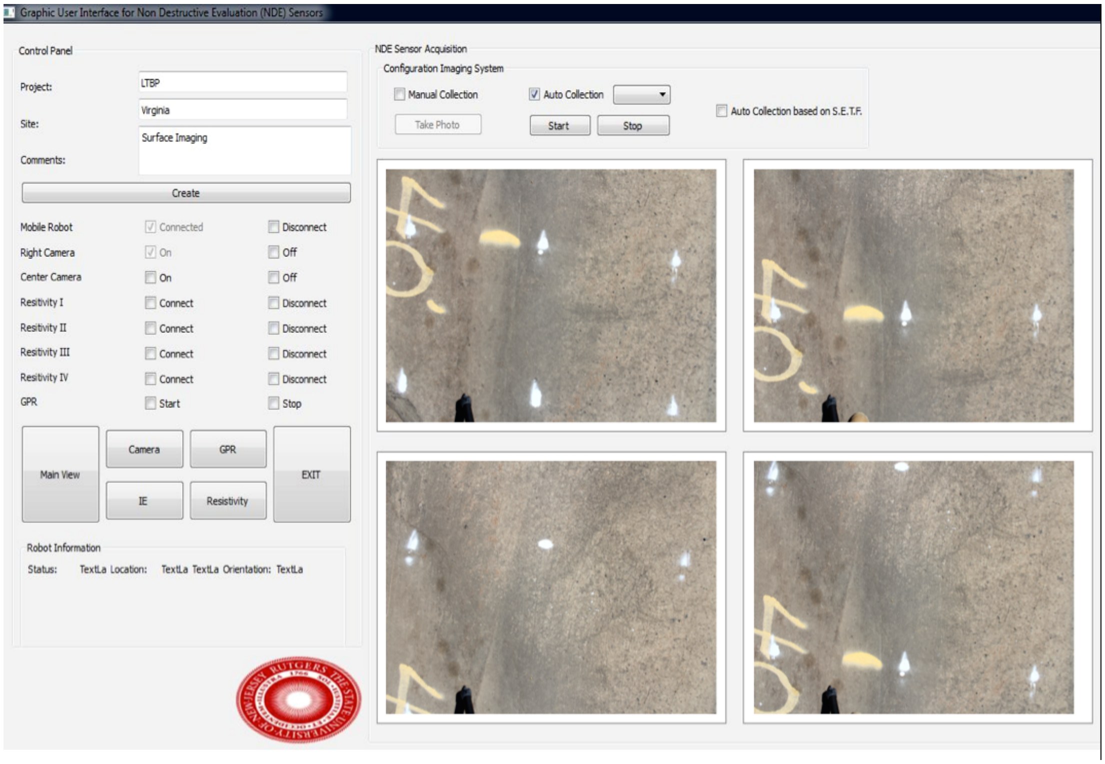
Figure 7.1: Since the background i.e the cracks is relatively distant as compared to the foreground, the change in appearance of background is small. The foreground appears brighter with the flash on.

We would also like to try out a *flash vs no-flash* approach to differentiate between concrete surface regions. Images of the bridge deck are to be collected with and without flash. A true crack does not reflect much light and remains dark even in flash conditions. This helps us differentiate cracked regions from normal ones. Another approach would be making use of bright sunlight and shadow environment to simulate the flash and no-flash conditions.

Appendix A

Graphical User Interface

A.1 Image Collection


Graphic User Interface for Non Destructive Evaluation (NDE) Sensors

Control Panel

Project: LTBP

Site: Virginia

Comments: Surface Imaging

Mobile Robot ☑ Connected □ Disconnect

Right Camera ☑ On □ Off

Center Camera □ On □ Off

Resitivity I □ Connect □ Disconnect

Resitivity II □ Connect □ Disconnect

Resitivity III □ Connect □ Disconnect

Resitivity IV □ Connect □ Disconnect

GPR □ Start □ Stop

Robot Information

Status: TextLa Location: TextLa TextLa Orientation: TextLa

NDE Sensor Acquisition

Configuration Imaging System

□ Manual Collection ☑ Auto Collection □ Auto Collection based on S.E.T.F.

Figure A.1: GUI for image collection. *Courtesy*: Ronny Lim

Figure A.1 shows the GUI for image collection. The images shown here are captured by one of the surface imaging cameras during the motion of the robot. The GUI can show for successive images in the same window. It also allows the user to toggle the camera between *on* and *off* modes at any given time.

A.2 Panoramic Imaging

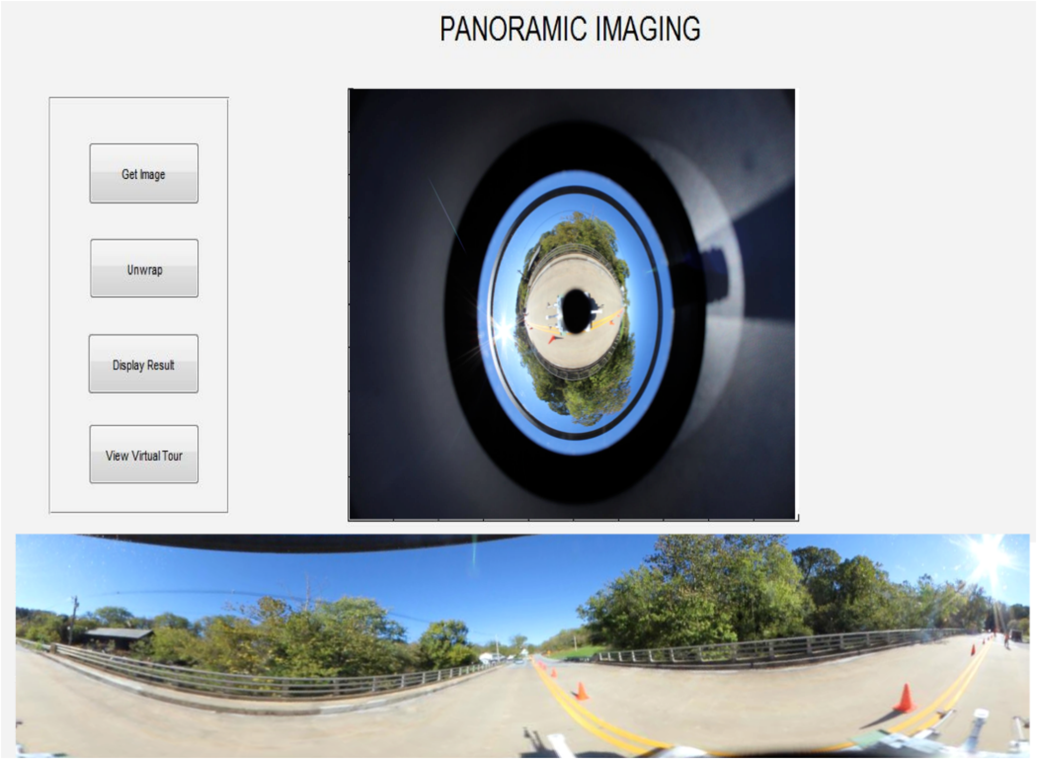


Figure A.2: GUI for Panoramic Imaging.

Figure A.2 shows the GUI for panoramic imaging. It allows the user to transfer a 360-degree image from the robot to the base-station at any given time. The unwrapped image can be viewed in the same window. It also provides the functionality of viewing the virtual video tour of the unwrapped scene when required

Appendix B

Lighting Issues

B.1 Illumination requirement

For photography in the night, the illumination levels need to be sufficient. We make use of digital LED video light Cowboy Studio CN-160 for this purpose. Two such light sources can be fixed to the robot using L-bracket holders. The operating voltage is 7.2-9V and it makes use of AA batteries.

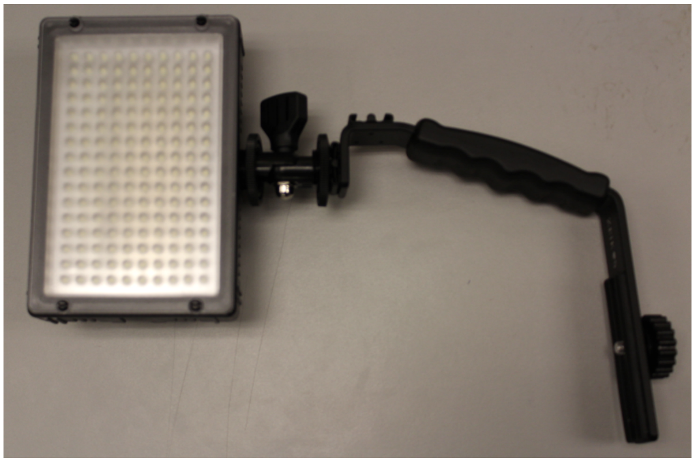


Figure B.1: CN-160 LED light with L-bracket holder.

B.2 Shadows in Images

The image-collection procedure can be carried out at any time of the day. When it is sunny, we encounter the problem of shadows. Shadow-removal is a long-standing problem and we would like to work on this aspect of the problem as well.

A part of the problem was encountered using high-power flashes attached to the

Canon cameras. Two Canon Speedlite 600EX-RT external flash lights were used. Flashes, however, did not eliminate the shadows completely and there was still intensity variation in the images.

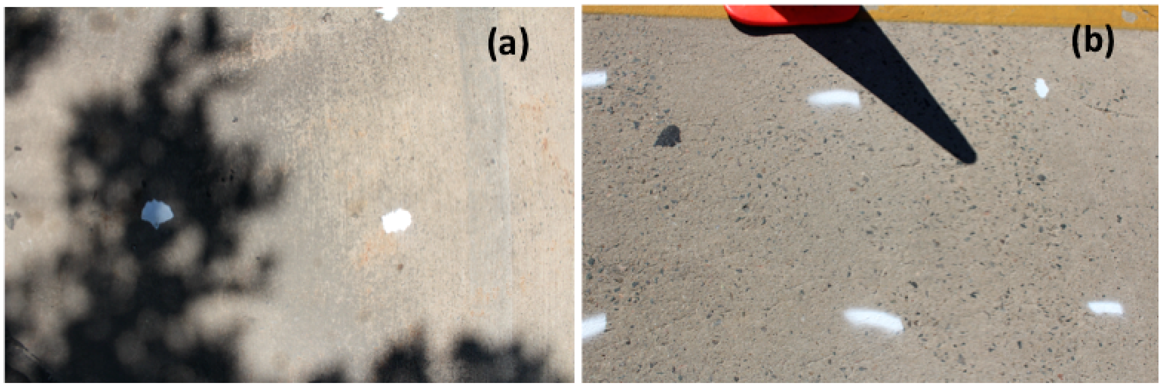


Figure B.2: Images with shadows. (a) Shadows of the surrounding leaves. (b) Shadow cast by the cones.

References

[1] The geography of transport systems. [Online]. Available: http://people.hofstra.edu/geotrans/eng/ch7en/conc7en/shareemploytrspusa.html

[2] LTBP. [Online]. Available: http://cait.rutgers.edu/ltbp

[3] Xianqi-He, Ziqiang-Zhu, Guangyin-Lu, and Qunyi-Lu, "Bridge management with gpr," in *Information Management, Innovation Management and Industrial Engineering, 2009 International Conference on*, vol. 3, Dec., pp. 325–328.

[4] Seekur robot. [Online]. Available: http://www.mobilerobots.com/ResearchRobots/Seekur.aspx

[5] M. Segraves, "NBC-Washington," February 2013. [Online]. Available: http://www.nbcwashington.com/news/local/One-of-a-Kind-Robot-Inspects-DC-Area-Bridges-193127901.html

[6] B. Wan, T. McDaniel, and C. Foley, "What's causing cracking in new bridge decks?" Wisconsin Highway Research Program. [Online]. Available: http://wisdotresearch.wi.gov/wp-content/uploads/WisDOT-WHRP-project-0092-09-06-brief.pdf

[7] B. Wan, C. Foley, and J. Komp, "Concrete cracking in new bridge decks and overlays," Marquette University, Tech. Rep., March 2010.

[8] M. Saadeghvaziri and R. Hadidi, "Transverse cracking of concrete bridge decks: Effects of design factors," *Journal of Bridge Engineering*, vol. 10, no. 5, 2005.

[9] C. French, L. Eppers, Q. Le, and J. Hajjar, "Transverse cracking in bridge decks," University of Minnesota, Tech. Rep., january 1999.

[10] A. Qader-I., O. Abudayyeh, and M. Kelly, "Analysis of edge-detection techniques for crack identification in bridges," *Journal of Computing in Civil Engineering*, vol. 17, no. 4, pp. 255–263, 2003.

[11] I. Abdel-Qader, K. Ahmed, and O. Abudayyeh, "Linear structure modeling and pca algorithm for bridge crack detection," in *Electro/Information Technology Conference, 2004. EIT 2004. IEEE*, Aug. 2004, pp. 138 –142.

[12] X. Tong, J. Guo, Y. Ling, and Z. Yin, "A new image-based method for concrete bridge bottom crack detection," in *Image Analysis and Signal Processing (IASP), 2011 International Conference on*, Oct., pp. 568–571.

[13] J. H. Lee, J.-M. Lee, H. J. Kim, and Y.-S. Moon, "Machine vision system for automatic inspection of bridges," in *Image and Signal Processing, 2008. CISP '08. Congress on*, vol. 3, May, pp. 363–366.

[14] R. Lim, H. M. La, Z. Shan, and W. Sheng, “Developing a crack inspection robot for bridge maintenance,” in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 6288 –6293.

[15] H. Oliveira and P. Correia, “Automatic crack detection on road imagery using anisotropic diffusion and region linkage,” in *18th European Signal Processing Conference (EUSIPCO-2010)*, 2010.

[16] T. Yamaguchi, S. Nakamura, and S. Hashimoto, “An efficient crack detection method using percolation-based image processing,” in *Industrial Electronics and Applications, 2008. ICIEA 2008. 3rd IEEE Conference on*, June 2008, pp. 1875 –1880.

[17] T. Yamaguchi and S. Hashimoto, “Automated crack detection for concrete surface image using percolation model and edge information,” in *IEEE Industrial Electronics, IECON 2006 - 32nd Annual Conference on*, Nov. 2006, pp. 3355 –3360.

[18] H.-G. Sohn, Y.-M. Lim, K.-H. Yun, and G.-H. Kim, “Monitoring crack changes in concrete structures,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 20, no. 1, pp. 52–61, 2005. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8667.2005.00376.x

[19] S. K. Sinha and P. W. Fieguth, “Automated detection of cracks in buried concrete pipe images,” *Automation in Construction*, vol. 15, no. 1, pp. 58 – 72, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580505000452

[20] P. Wang and H. Huang, “Comparison analysis on present image-based crack detection methods in concrete structures,” in *Image and Signal Processing (CISP), 2010 3rd International Congress on*, vol. 5, Oct. 2010, pp. 2530 –2533.

[21] K. C. P. Wang and W. Gong, “Automated real-time pavement crack detection and classification,” Highway IDEA Project 111, Tech. Rep., May 2007.

[22] A. Khanfar, M. Abu-Khousa, and N. Qaddoumi, “Microwave near-field nondestructive detection and characterization of disbonds in concrete structures using fuzzy logic techniques,” *Composite Structures*, vol. 62, pp. 335–339, 2003.

[23] H.-G. Moon and J.-H. Kim, “Inteligent crack detecting algorithm on the concrete crack image using neural network,” in *ISARC*, 2011, pp. 1461–1467.

[24] Z. Chen, R. Derakhshani, C. Halmen, and J. T. Kevern, “A texture-based method for classifying cracked concrete surfaces from digital images using neural networks,” in *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 31 2011-Aug. 5, pp. 2632–2637.

[25] S. Choi and S. Shah, “Measurement of deformations on concrete subjected to compression using image correlation,” *Experimental Mechanics*, vol. 37, pp. 307–313, 1997. [Online]. Available: http://dx.doi.org/10.1007/BF02317423

[26] Y. G. SEO, “A comprehensive study of crack growth in asphalt concrete using fracture mechanics,” Ph.D. dissertation, North Carolina State University, 2003.

[27] P. Prasanna, K. Dana, N. Gucunski, and B. Basily, “Computer-vision based crack detection and analysis,” in *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring.* International Society for Optics and Photonics, 2012, pp. 834 542–834 542.

[28] S. Alpert, M. Galun, B. Nadler, and R. Basri, “Detecting faint curved edges in noisy images,” in *Computer Vision–ECCV 2010.* Springer, 2010, pp. 750–763.

[29] F. Oloumi, R. M. Rangayyan, P. Eshghzadeh-Zanjani, and F. Ayres, “Detection of blood vessels in fundus images of the retina using gabor wavelets,” in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE.* IEEE, 2007, pp. 6451–6454.

[30] 0-360. [Online]. Available: http://0-360.com/

[31] Canon-Rebel T3i. [Online]. Available: http://www.usa.canon.com/cusa/consumer/products/cameras/slr_cameras/eos_rebel_t3i_18_55mm_is_ii_kit

[32] AVT-MANTA. [Online]. Available: http://1stvision.com/cameras/AVT/Manta-G-125-B-C.html

[33] USAF chart. [Online]. Available: http://en.wikipedia.org/wiki/1951_USAF_resolution_test_chart

[34] GoPano. [Online]. Available: http://www.gopano.com/products/videowarp-director

[35] Media Server. [Online]. Available: http://www.adobe.com/products/adobe-media-server-family.html

[36] CAIT. [Online]. Available: http://cait.rutgers.edu/

[37] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[38] ROC curve. [Online]. Available: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

[39] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition).* Wiley-Interscience, 2000.

[40] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[41] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[42] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-.* MORGAN KAUFMANN PUBLISHERS, INC., 1996, pp. 148–156.

[43] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[44] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: An update,” SIGKDD, 2009.

[45] P. Burt and E. Adelson, “The laplacian pyramid as a compact image code,” *Communications, IEEE Transactions on*, vol. 31, no. 4, pp. 532–540, 1983.

[46] Image Pyramids. [Online]. Available: http://docs.opencv.org/doc/tutorials/imgproc/pyramids/pyramids.html

[47] E. Salari and D. Ouyang, “An image-based pavement distress detection and classification,” in *Electro/Information Technology (EIT), 2012 IEEE International Conference on*, May, pp. 1–6.

[48] R. Gonzalez and R. Woods, *Digital Image Processing.* Prentice-Hall, 2002.

[49] D. G. Lowe, “Object recognition from local scale-invariant features,” 1999.

[50] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” *Computer Vision and Image Understanding (CVIU)*, vol. 310, no. 3, pp. 346–359, 2008.

[51] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: An efficient alternative to sift or surf,” in *Computer Vision (ICCV), 2011 IEEE International Conference on*, Nov., pp. 2564–2571.

[52] OpenCV Homography. [Online]. Available: http://opencv.willowgarage.com/documentation/python/camera_calibration_and_3d_reconstruction.html#findhomography

[53] Bilinear Interpolation. [Online]. Available: http://en.wikipedia.org/wiki/Bilinear_interpolation#Application_in_image_processing

[54] R. Garg, “Unstructured image mosaics,” PhD Thesis, 2012.

[55] PTGUI. [Online]. Available: http://www.ptgui.com/

[56] S. Nayar, “Catadioptric omnidirectional camera,” in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 482–488.