Summer 2009

Vehicle positioning using image processing

Armadeep Kaur
Missouri University of Science and Technology, kaura@mst.edu

Department:

VEHICLE POSITIONING USING IMAGE PROCESSING

by

AMARDEEP KAUR

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2009

Approved by

Steve E. Watkins, Advisor
Randy H. Moss
Ray A. Luechtefeld

ABSTRACT

An imaging-processing approach is described that detects the position of a vehicle on a bridge or similar environment. A load-bearing vehicle must be carefully positioned on a bridge for quantitative bridge monitoring. The personnel required for setup and testing and the time required for bridge closure or traffic control are important management and cost considerations. Consequently, bridge monitoring and inspections are good candidates for smart embedded systems. The objectives of this work are to reduce the need for personnel time and to minimize the time for bridge closure. An approach is proposed that uses a passive target on the bridge and camera instrumentation on the load vehicle. The orientation of the vehicle-mounted camera and the target determine the position. The experiment used pre-defined circles as the target, a FireWire camera for image capture, and MATLAB for computer processing. Various image-processing techniques are compared for determining the orientation of the target circles with respect to speed and accuracy in the positioning application. The techniques for determining the target orientation use algorithms based on the centroid method, template matching and color of the target object. Timing parameters are determined for each algorithm to determine the feasibility for real-time use in a position triggering system. The development can be combined with embedded sensors and sensor nodes for a complete automated procedure. As the load vehicle moves to the proper position, the image-based system can trigger an embedded measurement which is then transmitted back to the vehicle control computer through a wireless link.

ACKNOWLEDGMENT

With the completion of this thesis, I accomplish another milestone towards my self-evolution and realizing the proximity of the horizon which at one time looked far off. I would like to take this moment to thank some people close to me and remember numerous others who have contributed so much into my life. First and foremost, I am thankful to my parents Mr. Rashpal Singh and Mrs. Satwinder Kaur for providing me with all the support and encouragement on what I decided for myself. I am also thankful for the love and support of my younger siblings Simardeep Kaur and Arashdeep Singh which has always boosted me to set forth examples for them to look up to me. My stay in Rolla would not have been as easy and successful without the wonderful support of my fiancé Raj Kanwar Singh who with his own commitments made sure to give me all the time and encouragement that I demanded. I appreciate his time and support.

I am really thankful to my advisor Dr. Steve E. Watkins for his immense support and the knowledge imparted by him, which not just helped in my thesis but also in my overall development as a student. I am also thankful to Dr. Ray Luechtefeld and Dr. Randy Moss for the valuable courses they have taught me, their guidance on my thesis and for serving on my committee.

I would like to thank the Missouri University of Science and Technology for supporting me as a teaching assistant and as a student leader, both of which roles seem important in my evolution. Finally, I would like to thank my fellow graduate students who always gave a helping hand when I needed one.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Page

LIST OF TABLES

Page

1. INTRODUCTION

Regular health monitoring and field testing of civil engineering structures is very necessary because such infrastructure and its maintenance are central to a country's economy. Bridges constitute a significant percentage of this infrastructure and play an important role in transportation. Conventional methods of field monitoring include visual inspections, chain drag systems, and others (explained in next section). These types of inspections include different inspections for each or a few components (for example, inspections for monitoring the surface, wooden components, inside metal components etc.) of the bridge. Performing just one type of inspection does not provide data regarding the whole bridge. These inspections consume a lot of time, are usually targeted on some specific locations across the bridge and do not provide a detailed health scan of the whole bridge. For the bridges that are occupied at all times or are used frequently, a field monitoring system which is not expensive and is time efficient would provide management benefits.

"Smart" structures provides an intelligent solution to these problems. A smart structure is the one that can adapt to its changing environment and is able to communicate with the outside world (machines or human). A smart structure has several embedded sensors or sensor motes on different locations across the entire structure. These sensors act as data centers in the structure. These sensors collect data about the external as well as internal conditions of a structure. This data can then be retrieved as and when needed by the help of proper instrumentation. For collection data of a smart bridge,

instrumentation may be mounted on a load vehicle and that vehicle is moved across the bridge. At a particular location (per measurement requirements), the vehicle is stopped, sensor nodes are triggered using instrumentation mounted on the vehicle, and data from the sensors is received. Positioning of the vehicle plays a vital role in this type of testing and data collection. Data can only be retrieved if the triggering instrumentation is placed on a right location above the sensor. For achieving an optimal positioning of the vehicle, an approach using infrared has been suggested [1].

Image processing is an active field of research when major industrial applications are concerned. Techniques like image detection or pattern recognition are very commonly used in the research related to image processing. Image processing can also be used for load positioning. Using an image processing technique makes it very easy to customize an existing algorithm to work under different conditions. MATLAB is a user friendly tool and the in-built ‘help’ feature can act as a guide for a beginner. Anybody who understand the basic logic behind an existing algorithm can spend some time and manipulate it according to the required conditions. Since, the cost of the MATLAB tool is one time, experimentation is not limited. These features make image processing a competitive candidate for utilization in health monitoring systems.

This thesis proposes an approach for vehicle positioning using image processing techniques. By adding a passive target object to the image scene at the location of load positioning, it can be detected later using image processing to position the vehicle. In this approach, vehicle would be positioned at a position normal to the center of the target object. Three image processing techniques are tested and compared. These three

approaches are based on centroid location of the target object, color detection of the target object, and template matching. This work proposes a cost-effective and fast method to position the vehicle. Purpose of the work presented here is to integrate an already existing triggering system[1] with image processing to achieve the desired goals of a cost-effective and time efficient load positioning system.

2. LITERATURE REVIEW

Infrastructure is an important part of the progress of any civilization. Work has been going on since the beginning of civilization to make improvements in existing things and to come up with new techniques to create something more advanced and efficient. This section is an overview of the related work done in the field of "smart" structures [2], structural health monitoring and assessment, image processing, and/or other related fields. A "smart" structure is one that senses its environment, assess its own health conditions, adapts to the changing environment, decide for itself and communicates on a regular basis with the outside world (human or another system) [2-5]. It has in-built sensors, an embedded sensing node, and a triggering system. Embedded instrumentation helps in health monitoring data acquisition on a regular basis without cumbersome set-ups. The monitoring done is to the exterior a well as interior of the structure. Various sensing techniques have been established that can be used to collect data that is helpful in finding the internal as well as external condition of a bridge. This data is then processed to find out the parameters like estimated life time of the bridge, load that it can carry and other structural health related parameters. Various terms are defined in this section that set the basis of the work presented later.

2.1. SMART BRIDGES

The management of bridges as part of the transportation infrastructure is a significant expense. They must be monitored, maintained, and repaired throughout their

service life. Technologies that can contribute to keeping these structures in adequate condition address a national concern [6]. By integrating advances in materials with sensor instrumentation, better adaptive structures can be made. Various approaches have been exploited to make these man-made structures "smart" [2-4], [7], [8]. It is required that a structure should possess an ability to sense in order for it to be called as a "smart" structure [2]. The most important purpose of the structure via sensing is to adapt to changing conditions around it. To convert a passive load-bearing structure into a smart structure involves in-situ sensor instrumentation.

The ability to monitor and interpret the condition of structures accurately can facilitate more rapid adaptations to conditions. Low-cost instrumentation and monitoring approaches are needed to provide quantitative structural health thus improving service life of new as well as old structures [3]. The main area of the research related to bridges revolves around the enhancement of the basic load bearing functions [2]. Load-induced strain measurement is an important parameter for assessing bridge condition. While working towards this development, two different approaches need to be taken. One is for those structures that are already in existence and are a part of the economic development going around the globe. And the second is for those structures that are to be built yet. For both of these approaches, interfacing between the structure and sensing modules becomes very important.

2.2. SENSORS AND INSTRUMENTATION

Visual inspection is the most commonly used method to evaluate and assess the condition of a bridge traditionally [9], [10]. Some common tools like probes, knife, wire brushes, scrapers, binoculars, flashlight etc. are used for the purpose of these type of inspections. Though an inspector's vision ability is the main tool, he/she is not required to have a minimum visible acuity. There are five type of inspections that are usually carried out. These are: initial inspections (performed on new bridges to notice changes in condition such as erosion, regrading of slopes, etc.), routine inspection (regularly scheduled inspections to assess bridges for any possible damages or deteriorations), damage inspections (performed in case of a damage caused by natural or unnatural conditions), in-depth inspections (carried out as follow-up inspections to an initial or damage inspections to further identify the deficiencies), and special inspections [9]. According to a study conducted at the Federal Highway Administration (FHWA) Nondestructive Validation Center (NDEVC) [11], there is no standard procedure that is followed for visual inspections. Thus, inspections conducted vary from person-to-person and so do the results. In addition to visual inspections, other techniques like chain drag systems (manual, electrochemical, and automated) are used to find delaminations in a structure [10]. Several other tests are used that are structure or component specific and only give assessment on one or fewer components of a bridge (for example, assessment of wooden components, iron components, bridge surface, etc.) . These include tests to detect steel corrosion (electrochemical and physico-chemical techniques), electrical impedance testing of wood components, ultrasonic testing of structural timber

components, digital radioscopy etc. All these tests are used to assess only one (mostly) component of a bridge. Assessment and monitoring of a bridge using different techniques for different components can be a very time and resource consuming process. This time lag can also delay any action that needs to be taken immediately if the extent of damage can pose immediate threat to public safety. Instrumentation that can address several problems at one time at low cost is required for these inspections and "smart" structures are capable of addressing all of these problems over a long period of time.

Embedded sensing systems are being developed that can last the lifetime of structures. Optical fibers are capable of sensing a large variety of physical effects and are insensitive to electromagnetic interferences [2]. They can easily sense physical effects like stress, strain, temperature, pressure, degree of cure etc. [2], [3]. Fiber-based Extrinsic Fabry-Perot interferometric (EFPI) strain based sensors are being used successfully for making point measurements in bridge structures [3-8]. Optical fiber based sensors are classified into two categories namely intrinsic and extrinsic sensors based on the sensing region of the fiber [12], [13]. If the sensing takes place within the fiber, the sensor is referred to as an intrinsic sensor. If sensing occurs outside the fiber, the sensor is referred to as an extrinsic sensor. In this case, the optical fiber is only used as a channel to transmit the data. Fiber optic sensors are very popular due to their robustness, small size, light weight, low loss, immunity to electromagnetic interferences etc [4], [13], [14]. Also, optical fibers can be used both for sensing as well as transmitting data [14].

Interferometric fiber sensors can be successfully used to monitor the earth's movement and strain on high tension wires [3]. The Fabry-Parot technology can be used

to measure strain with a precision of +/- 0.01 με [8]. Fiber Bragg grating sensors are another type of optical sensors that are very effective intrinsic sensors [14]. Their working principle lies in wavelength shift resulting from changing strain [14], [15]. Multiplexing of several Bragg grating sensors is also very easy. These strain-measuring sensors are being used for the functions like health monitoring, performance monitoring and warning system inside a smart structure [1,4]. The usage of these types of structures is limited because of time constraints, testing constraints in data collection and interpretation and field robustness, etc.

Field monitoring is an important aspect of a bridge or similar structures. Significant research has been done to reduce set-up time, to provide significant information out of load test and to reduce closure time to traffic [1]. Autonomous triggering instrumentation is an important development towards this direction. Personnel intervention in this system is minimal. The instrumentation system once set-up determines the correct location for load positioning on the bridge and then collects the required data too. Infrared triggering is used to determine the correct load position.

The field of smart structures research has emerged as an interdisciplinary field [2], [4]. The reason for this development is the need to enhance various elements of a structure, its geometry and health characteristics to make it smart. A smart material needs special building materials, advanced sensing techniques, artificial intelligence, adaptive systems, and other advanced instrumentation [2], [3]. These things can help developing a small structure that provides cost and performance effective solutions over the conventional passive structures. Image processing is an active field of research when

major industrial applications are concerned. Techniques like image detection or pattern recognition are very commonly used in the research related to image processing.

2.3. IMAGE PROCESSING

Image processing can also be used for load positioning. By adding an object to the image scene at the location of load positioning, it can be detected later using image processing to position the vehicle. This work proposes a cost-effective and fast method to position the vehicle. Techniques for circle detection can be helpful in this type of approach. Extraction of a circle from an image can be used for location finding industrial applications [16]. Several techniques used for circle detection include detection using the Circle Hough transform (CHT) [16], gradient pair vectors [17], weighted minimum mean square error (MSE) estimator [18], and customized Hough transforms etc. [18], [19], [20].

Hough transform for circles is a very known method to detect circular shapes [17]. The circle hough transform use radius as the basis of locating a circle [16]. Work presented by Duda and Hart in 1975 consist of using parameterization of circles into an equation and transforming a match (figure point) in a parameter space defined by another equation [16], [19]. If the object in consideration is of the known radius, the algorithm is relatively fast and requires less space [17], [16]. For an object with unknown radius, large storage space is required and processing speed of the algorithm is compromised. These limitations arise from the fact that the algorithm has to be run for all possible values of the radius. Also, with a significant noise in the background, there is high probability of

false detection [16]. In this type of algorithm, better resolution means more computational time. There has been a significant amount of work reported on customized circle hough transforms. Many properties that were used for customizing the transform include parallel property of circles [21], randomized selection of pixels [22], local geometrical properties [23], and others. But these experiments have not been successful in providing solutions for parameters like accuracy, consistency, speed, space constraints, and probability estimation all at once [17], [21-23].

Another approach used weighted mean square error (MSE) as a parameter to find circles [18]. In this case, pixels that are more helpful in finding the circle are weighted more. Computational speed of this process is faster than Hough Transforms but this method only works for certain known parameters and assumptions. It works for the images with only one circle, noiseless background, and assume that the pixels should be detected easily. Algorithm is based in weighted MSE, thus would produce a weighted average of all the circles in an image in case more than one circles are present.

Adaptive randomized Hough transform is another form of the Hough transform where faster computational speed and higher detection rates have been achieved via using a moving window for the purpose of circle detection. A square window is moved across the image and a number of sub-images are created. But this method is also dependent upon a known value of the maximum limit of radii. Large signal to noise ratio can be attained using this algorithm.

3. BACKGROUND

The work presented in this thesis describes the development of a vehicle positioning system using image processing. This proposed system is designed to be used during field monitoring applications of highway bridges. The integration of image-processing techniques with triggering instrumentation for load testing is the ultimate desire. The approach is proposed as an alternative to other triggering methods [1] with the same requirements for low cost, relatively fast operation, and ease of use. This system should not require specialist knowledge and can be set-up with minimal personnel intervention. The proposed load positioning/triggering system uses very simple algorithms to determine the correct load position. The camera for taking images is mounted on a vehicle which moves across the bridge. Images are taken continuously and are processed using the image processing toolbox of MATLAB. The proposed algorithm returns the deviation of the vehicle from the desired position and also gives the direction in which the vehicle should be moved in order to place the load at the correct position. Testing is done in a laboratory environment to test the feasibility and efficiency of the system.

3.1. DEFINING PROBLEM

The problem addressed in this thesis is to position a vehicle at a desired location (mid-span, quarter-span etc.) on a bridge. The vehicle serves as a known load. Once the vehicle is in position, instrumentation triggers a set of embedded sensors and collects data

for health monitoring. The approach uses image-processing techniques. The proposed solution should be able to provide a better method for strain/load testing of highway bridges than conventional load testing. It should provide an accurate positioning for the vehicle with minimal personnel intervention, an easy-to-use set-up and instrumentation, and relatively low cost. Also, the total testing time as well as traffic closure time should be minimized. The instrumentation includes sensors, sensor nodes, and a target on the bridge and a triggering and processing unit on the vehicle. A passive target is preferred to minimize the cost of bridge hardware. The passive target object (a circle) is attached on the railing of the bridge in consideration. This target object is taken as a reference to position the vehicle. Different parameters used in image processing algorithms are the circular shape of the target object, the centroid of the target object, and the color of the target object. Different algorithms that are used for the experiment in consideration will be explained in later sections.

Sensors and sensor nodes in the set-up described are pre-developed and embedded beneath the bridge surface. Embedded instrumentation helps in health monitoring data acquisition on a regular basis without cumbersome set-ups. The positions of these embedded sensors are known. The position of the passive target is to locate the load bearing vehicle right above the embedded sensors/sensor nodes. Hence, a typical load test will locate the vehicle at mid-span [24]. The data from strain and deflection sensors can be related to bridge health. The approach must accommodate vehicles in different lanes and bridges of different widths. Different separation distances between the vehicle and

bridge railing have been taken into account to analyze the effectiveness and accuracy of all of the proposed algorithms on all range (of dimensions) of bridges.

3.2. TRIGGERING INSTRUMENTATION: SMART MONITORING

The proposed process is a part of a bigger smart system used to perform load testing on bridges. This smart system has in-built strain sensors, an embedded sensor node, and a triggering system to perform measurements [1]. Figure 3.1 shows a brief explanation of the overall testing protocol.

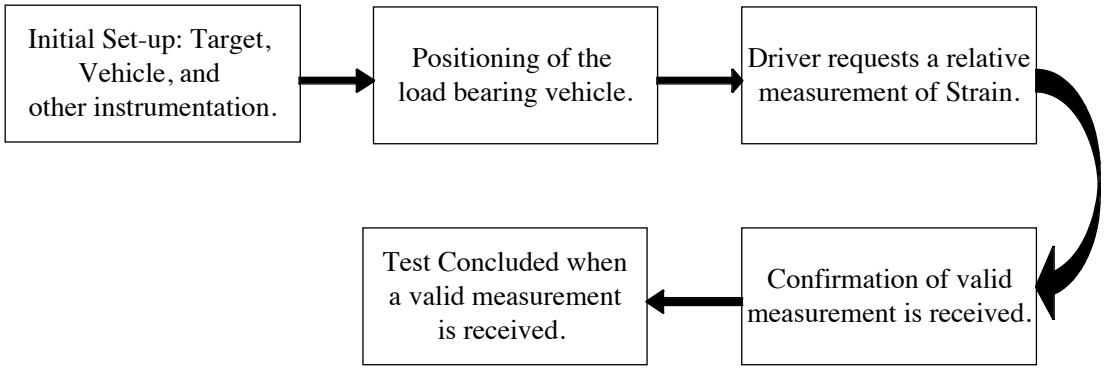


Figure 3.1 - Overall testing protocol.

After initial set-up of instrumentation, vehicle, and target object, the vehicle is positioned at a desired location on the bridge using image processing and a passive target. Once the vehicle is positioned at the desired location, the driver requests desired data readings. Upon getting a valid measurement, the system confirms that a valid

measurement has been received and the process is repeated for another point of interest if required. Figure 3.2 shows an actual load test set-up with a H20 weighted truck [7].

Figure 3.2 - Load test environment with H20 weighted truck.

The set-up including trigger system and the position changes in the vehicle relative to the target object is shown in Figure 3.3. This figure shows the vehicle position and an estimated detection range when, a) the camera is in normal position to the target object, and b) the camera is at an angle such that the target appears to be an ellipse. A compatible control unit consisting of triggering instrumentation is mounted on a weighted

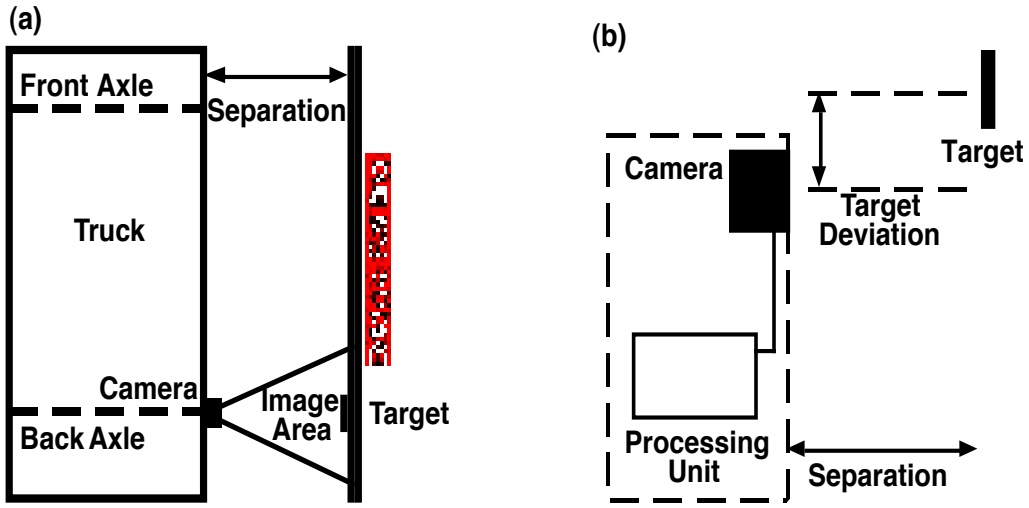


Figure 3.3 – Triggering system. (a) Camera is in normal position to the target. (b) Ellipse detection.

vehicle that is acting as a load [1]. This load is placed at the desired position on a bridge. Circular targets are attached on the railing of the bridge. These images act as passive markers of the location where the load vehicle needs to be positioned. When the vehicle is normal to the markers, the load is correctly positioned. Image processing techniques are used to locate the target with the help of a camera mounted at a side of the vehicle. The expensive triggering element is the camera which can be used on multiple structures. The only triggering component on the bridge is the passive target. The driver of the vehicle moves the vehicle across the bridge, and images are captured and analyzed. This analysis will tell the driver where to halt the vehicle. Once the vehicle is positioned at the desired location, the embedded sensing process is automatically performed [1]. The whole procedure can be completed by one person or by a team of few people. This can be done with minimal disruption of traffic and in a limited time interval.

Figure 3.4 illustrates the deviation of the vehicle from the desired position and the relative position of camera and the target object. S is the separation (in inches) between

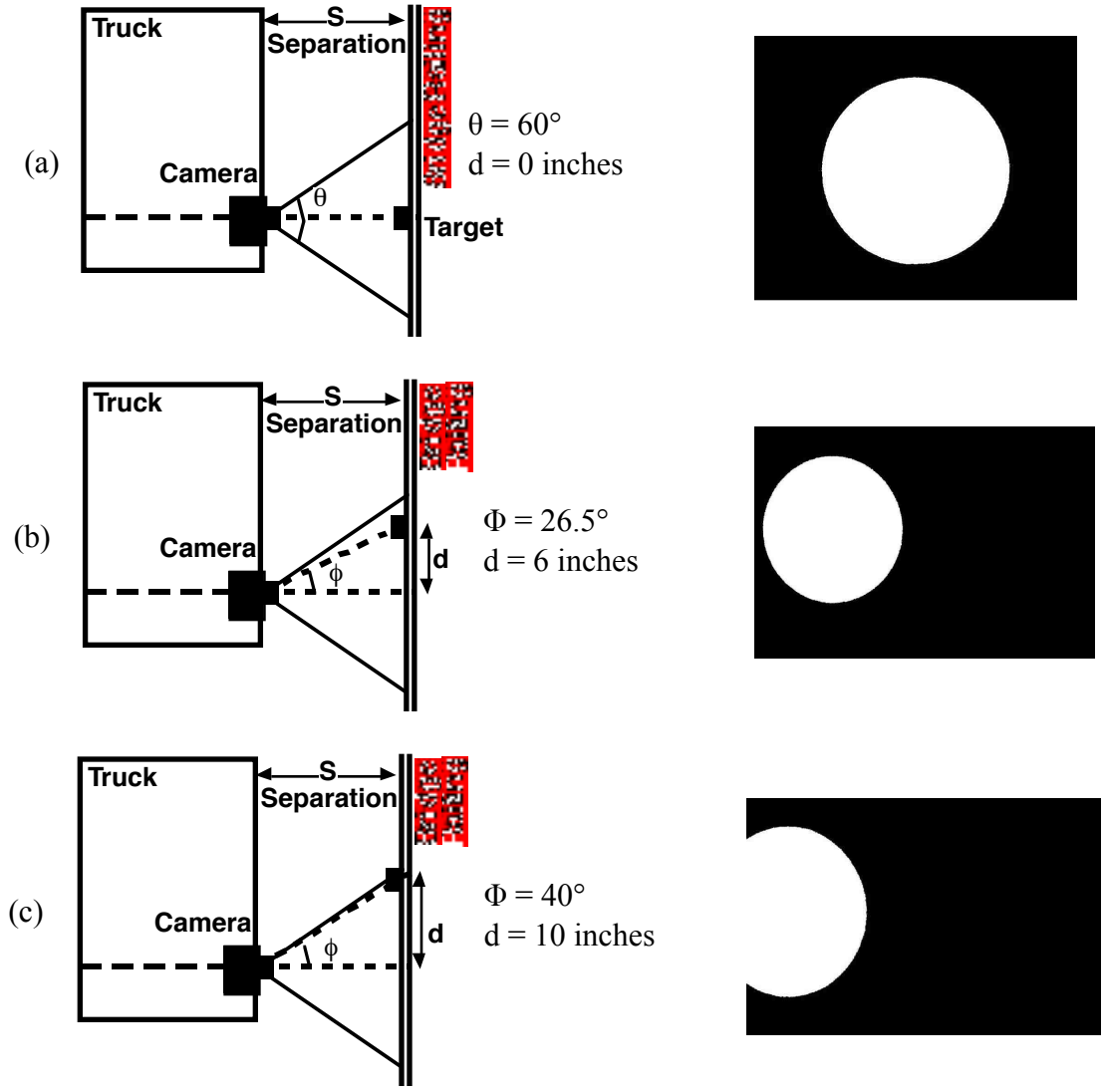


Figure 3.4 - The relative position of the camera and the target object. (a) The vehicle is at the desired position and the extracted target object at this position is given. (b) The vehicle is at 6 inches from the target object and the extracted target object at this position is given. (c) The vehicle is at 10 inches from the target object and the extracted target object at this position is given.

the target object and the vehicle, d is the deviation (in inches) between the desired position and the actual position. θ (in degrees) is the angle of detection (described in detail in Section 5.3), and Φ (in degrees) represents the angular deviation. Figure 3.4 (a) shows the position of the camera relative to the target when the vehicle is positioned at the desired position. In Figure 3.4 (b), the relative position of the camera and the target object is shown when there is a deviation of 6 inches between the actual position of the vehicle and the desired position. Figure 3.4 (c) gives the relative position when the vehicle is positioned 12 inches away from the target object.

Consider a circular target image mounted on the side of a bridge. As shown in Figure 3.3, the separation distance S is the distance between the target and a load vehicle traveling in the specified lane. The deviation between the target perpendicular and the camera perpendicular shows how far the load is off position. For the camera at some specific deviation, the target image will appear elliptical. Then the eccentricity of this resulting target image is :

$$e = (1-(w/h)^2)^{1/2}$$

where 'w' and 'h' are the maximum width and height of the target object, respectively. If w = h, the image is a circle which means that the vehicle is at the desired position, i.e. the image matches the circular target. For the laboratory work in this investigation, separation distances S of 12, 24, and 36 inches (30.48 cm, 60.96 cm, and 91.44 cm respectively) are used. Note that for a deviation of 13.3 inches (33.78 cm) with an S of 12 inches (30.48 cm), the eccentricity 'e' is approximately 0.8.

Three methods for performing the required image processing are examined. The first method is Centroid Location. The target object in the source image is identified and its centroid is calculated and compared to the calibrated center of the image, i.e. the desired position in the image. The second method is Color detection. The target object is identified through color discrimination and the eccentricity is calculated directly. The third method is based on Template Matching. The image of the target circle is compared to a stored target pattern. The deviation distance is calculated from the best match.

The Centroid Location method should work well in the presence of noise, but it may be very sensitive to orientation of the camera. The range of detection might change depending upon surrounding conditions, and selection of camera lens. The Color method calculates the eccentricity directly and is insensitive to camera orientation, but it is dependent on having a clearly defined color difference between the desired target and other features in the image. The Template Matching method should be able to handle small variations in camera orientation and separation distance, but it may be sensitive to noise in a field application. The research questions to compare the methods and to determine feasibility are:

- to determine the maximum angle (see Figure 3.3) for which each method can detect the target and produce an accurate measure of deviation distance,
- to determine the relative speed of computation for each of the methods, and
- to determine whether a method is feasible to be used in actual health monitoring applications.

3.3. EXPERIMENTAL SET-UP USING IMAGE PROCESSING

The experiment to collect the comparison data was set-up in a laboratory environment. An Imaging Source Firewire CCD Bayer camera was connected to a PC. A CCTV lens was used with the camera to capture the images. IC capture (version 2.0) software was used to take the images. The images were taken in sequence multiple times. In this set-up, the target object was kept still and the camera was moved across the target horizontally. The camera was used in two different modes - automatic and manual. In automatic mode, a time delay can be specified and camera takes a picture after that specified time interval. In the manual mode, after doing the initial settings, an image can be captured by clicking the mouse of the computer. The gap between each image taken was varied between 1 second to 20 seconds in auto mode. The run time mentioned in Appendix A is based on manual mode. These images were later analyzed using MATLAB. Figure 3.5 shows the arrangement of the camera relative to the target object. In this set-up, camera is at 36 inches (91.44 cm) from the target object.

The target object plays an important role in this process. To detect and extract the target object from an overall image, the shape of the target object should not be common to the shapes found in the background. A circle was chosen to be a target object. Detection of any other shape is difficult because the outside environment surrounding bridges has line features in abundance. Almost every other shape consists of line(s). Target detection and extraction becomes tedious. Circular shapes are not very common to be found in the vicinity of bridge surroundings (specially on the railing of a bridge). As the vehicle is moving, the target object can be interpreted as an ellipse and the

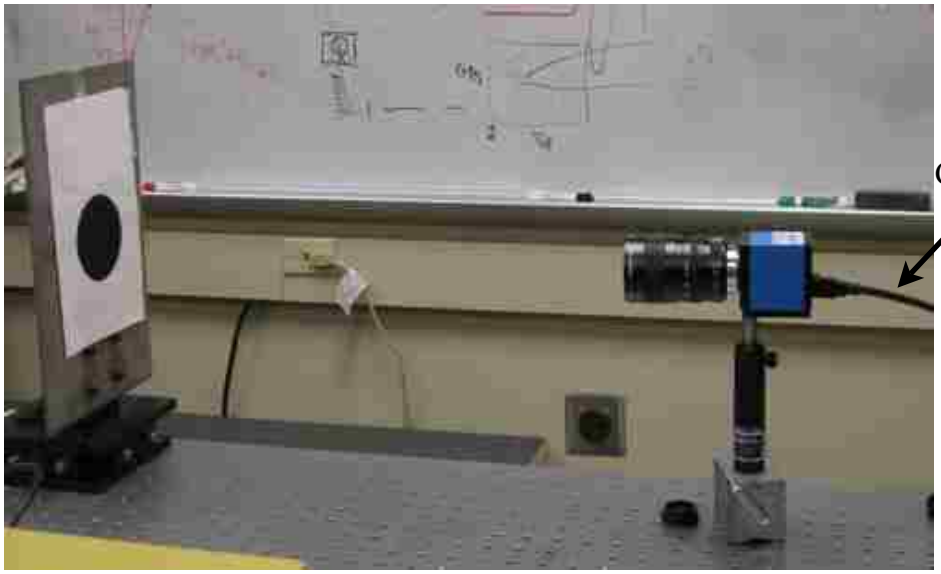
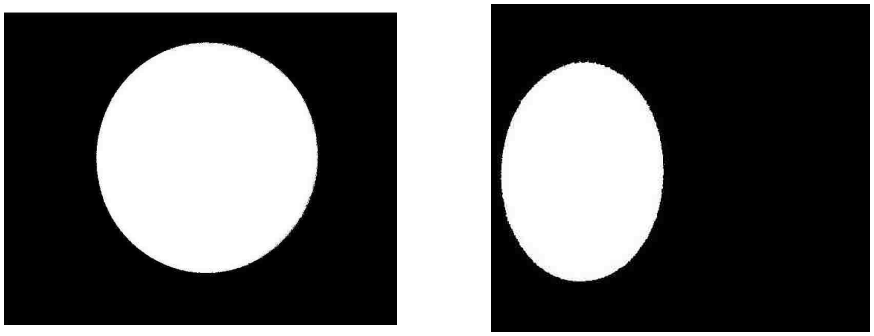


Figure 3.5 - Laboratory set-up.

eccentricity will be related to deviation distance. Figure 3.3 shows the vehicle position and an estimated detection range when, a) the camera is in normal position to the target object, and b) the camera is at an angle such that the target appears to be an ellipse. For the laboratory work, all three proposed methods are tested within 60° of vision of the camera.

Figure 3.6 shows two extracted targets. These target objects are extracted from actual images taken in an outside environment. Figure 3.6(a) shows a circle and Figure 3.6(b) shows an extracted ellipse. The actual target color is black. During the processing, the target object is denoted a white color and all the other objects in the source image are made a part of the background by denoting them a black color. Varying lighting conditions or surface of the paper used to print the target might have caused distortion of the ellipse. Eccentricity calculated for these two extracted shapes was 0.192 and 0.8612 respectively. After analyzing the images, MATLAB codes return the position of the



(a) (b)

Figure 3.6 – Extracted target images. (a) Circular target detected. (b) Elliptical target detected.

vehicle and the deviation between the centroid of the target object and the vehicle. This deviation gives the distance that the vehicle needs to travel either in forward direction or backward direction. If deviation is returned as negative, then the vehicle needs to be moved backward. If there is no deviation that means the vehicle is normal to the plane of the target object and this is the desired location on the bridge where the load needs to be positioned. More images can be found in Appendix B.

4. VEHICLE POSITIONING

This section discusses the working of various algorithms to detect a passive target on a bridge railing. A vehicle was positioned across the bridge on a desired location by bringing it normal to the center of the target object (circle). Different approaches used for the purpose include detecting the centroid of the target object, detection based on the color of the target object, calculating eccentricity of the target object, and template matching. Block diagrams are also provided for an overview of each method.

4.1. METHOD I - USING CENTROID

The centroid method consists of finding the centroid of the target in the image. The target object is an ellipse or a circle depending upon the deviation of the vehicle from the desired location. The desired position in this case is normal to the target object. More accurately, it is normal to the center of the target object. In this set-up, the camera is moving horizontally, hence, the vertical components are not significant for positioning of the vehicle. But, the vertical component gives us valuable information regarding the motion of the vehicle. The vertical component should remain constant along the horizontal axes. If it does not stay that way, it would indicate an error. Also, if the bridge surface is not very smooth, the motion of the vehicle might result in an erroneous center of the image. The centroid of the target is calculated using horizontal and vertical projections and the area of the target. The horizontal position of the centroid of the target is compared with the horizontal position of the center of the image. Based on the

difference between these two parameters, the camera and the vehicle are either moved forward or backward to the desired position on the bridge. The largest object in the image was selected as the target before calculating the centroid.

Using this method, there are three assumptions taken into consideration. First, the position of the vehicle should be exactly normal to the center of the target object. This condition will ensure that the vehicle is at the desired position to trigger the sensors and retrieve data related to strain measurements. The second assumption is that the target object should be the largest object in the image. This condition is required because the algorithm extracts the largest object and calculates the centroid for the extracted object. The last assumption is regarding the motion of vehicle. It is assumed that the surface on which the vehicle is moving is even and thus, the vertical component of the centroid is constant along the x-axis of the centroid. The centroid of the target object was found by using the following formula [25]:

$$i = \frac{\sum_{r=1}^{w} r \cdot h_r}{A} \qquad \& \qquad j = \frac{\sum_{r=1}^{h} r \cdot v_r}{A}$$

where A = Area of the object
v_r = Vertical Projection of the object
h_r = Horizontal Projection of the objectss
i = Vertical position of the centroid
j = Horizontal position of the centroid

Figure 4.1 describes the process of target object detection using the centroid location method. After reading the image, center of the image is found. Then, thresholding values are set for the image to create a binary image consisting only of black and white colored pixels. After this process, Eight - connectedness [20] , [26] is applied

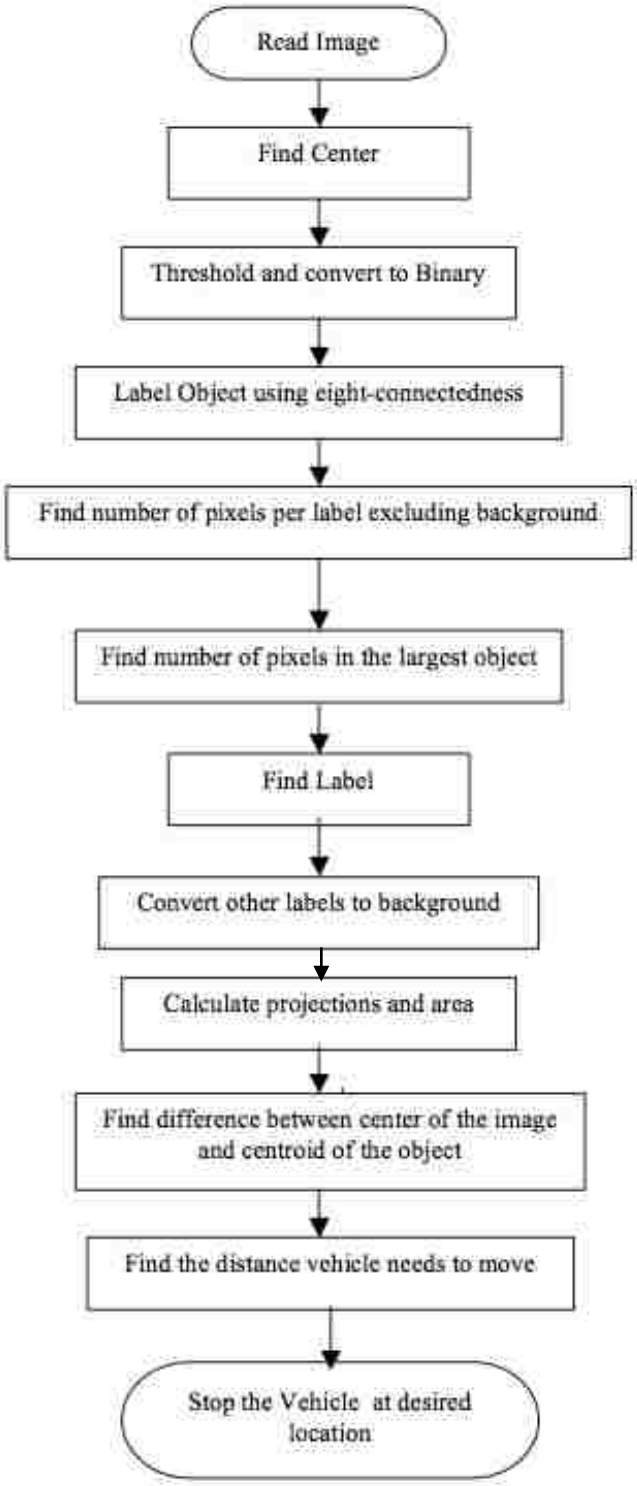


Figure 4.1 - Flow diagram of the process of vehicle positioning using 'Centroid' method.

and all of the objects in the image are labeled. The largest object is found by counting the number of pixels in all of the labeled objects. The target object is extracted as the largest object in the image. Once the target object is found, all of the other objects are made a part of the background. The centroid of the extracted circle is calculated using horizontal projection, vertical projection, and area measures. The difference between the calculated centroid (of the target circle) and the center of image indicates the distance that the vehicle needs to move in order to be correctly positioned. If the difference is negative, the vehicle is moved backwards.

This method is convenient in a way that we only need to consider one parameter, that is the horizontal position of the centroid for the results. Vertical position may be used to monitor the vehicle motion and errors. Ideally, the vertical component should remain the same along the horizontal axis. Otherwise, the correct centroid might be missed. Field implementation may use this parameter as a quality check, e.g. To verify whether the correct object has been identified. The centroid method is independent of the color and the background of the image. But this method is only convenient once the target object is extracted. Labeling using eight-connectedness might result in objects bigger than the target object. This will lead to a false detection since the current implementation only identifies the largest object. Techniques like eight-connectedness [20] and finding higher density were tried for the purpose of extraction but most of the time; connectivity resulted in creation of bigger objects than the target object and smaller objects shown to have a higher density. Any noise in the image poses a potential possibility for great deviations in the result.

Extraction of the target object is the biggest concern using the centroid method. The target object should be the largest object in the image to make sure that the largest object extracted after the process of labeling is only the target object and not any other object. Also, the centroid method only works when either the vehicle is at the position normal to the center of target object or if the vehicle has moved ahead of the target. This method does not work for the images that are taken from a point (on the bridge) before the center of the target. This method only work when either the vehicle is at the desired location or have moved ahead of the desired location. This method does not give an instant result, but the average time taken by this method is considerably less (7.99 seconds) and thus makes this method a good option.

4.2. METHOD II - USING COLOR

The color method uses the color of the target object as the primary parameter of detection. With the other two methods, it is very difficult to isolate the object in consideration (it is the largest object in the set-up used). To resolve this problem, a specific color for the target image was used. By using color, the object can be isolated based on its pixel's particular color value. An RGB (Red, Green, Blue) format of the target image is converted into an HSV (Hue, Saturation and Value) format of that image. The Hue values are used to isolate the object in consideration. During the process, a range of pixels is selected rather than selecting the pixels with a particular value. Two different approaches were used after labeling objects in the source image. In the first approach, after the target object is isolated, its maximum horizontal as well as vertical

dimensions are calculated. Semi-major axes of the target circle are found and are used to calculate the eccentricity. After isolating the object, its eccentricity is calculated. When the vehicle is perfectly normal to the target image, eccentricity will be 0. Otherwise, depending upon the variation of eccentricity, vehicle is moved towards the desired location. (Greater the deviation, greater is the distance between the vehicle and the desired location.) This method takes care of lighting issues. Figure 4.2 describes the process of target detection and extraction using color as a parameter (Approach 1).

In the second approach, an inbuilt MATLAB function 'regionprops' (for 'Eccentricity' property) was used to calculate the eccentricity. This function calculated the eccentricity of the assigned object. For this to be accomplished, it was required to find the target object and attach it to this function. Once, the target object is attached to the function, eccentricity is calculated most accurately.

By using color, extraction of the target becomes easier and the background can be completely eliminated from the calculations based on color selection. This method provides a great accuracy. But, selection of the color needs to be done very carefully. Varying the values of hue even by a small number can result in varying the results enormously. Because of the ease of extraction, even an ellipse with its eccentricity close to 0 can be viewed as a circle that might pose difficulty in calculating very small distances.

It is assumed, using the color method, that the color chosen for the target object is unique from the background so that the detection of the target object is easier. This method uses eccentricity as a measure of deviation from the desired location. Even if

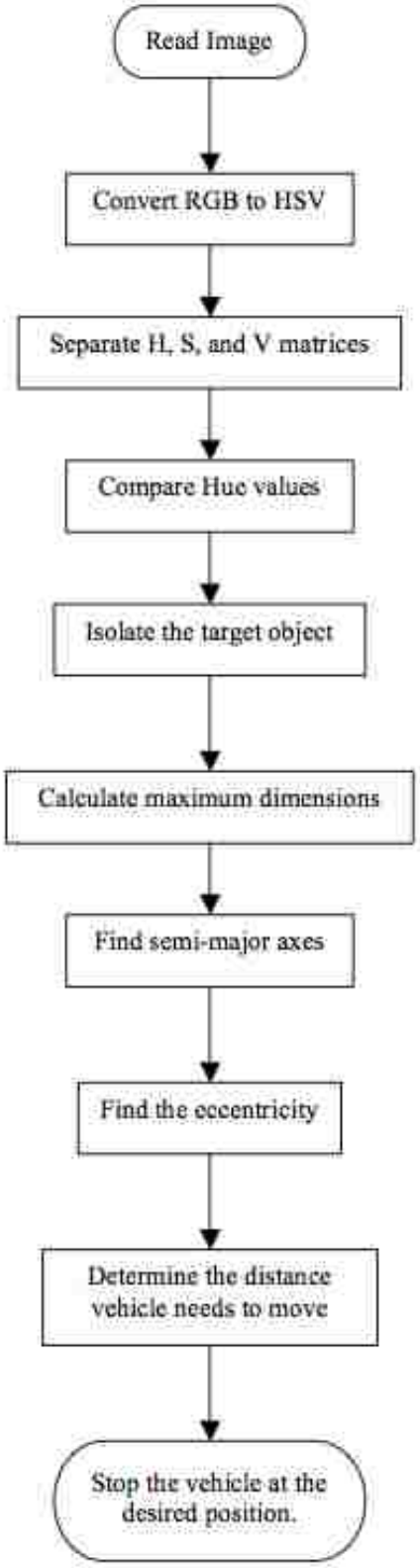


Figure 4.2 - Flow diagram of the process of vehicle positioning using 'Color' method.

there is a very small difference (e.g. 0.1 cm) in the desired and actual position, eccentricity value would suggest that the current position is not the desired position. In actuality, even at this deviation, instrumentation will be able to trigger the sensors and collect the data.

4.3. METHOD III - USING TEMPLATE MATCHING

Template matching is an effective method for identifying an object in a large group of objects. It is a digital image processing technique in which sub-parts of an image are found by matching that image with an image template. The actual source image (to be matched) is larger than the template image. In the process, the template image is moved to all of the possible positions in the source image and a numerical index is computed based on which the extent of matching is found on a certain position. This numerical index is termed as Euclidian distance. Formula used for the calculation is:

$\mathbf{d} = [\sum [f(x) - t(x-y)]^2]^{½}$, where 'd' is Euclidean distance, 'f' is the original image and 't' is the template [27].

The template used for matching with the source image is a circle as the target object is also a circle. The process can be very slow if the template has to be matched with every object in the source image. To eliminate this problem, the source image is converted into a binary image where one value is denoted to the target object and the other to the background. All of the objects other than the target object are made a part of the background. This result is achieved by thresholding the source image.

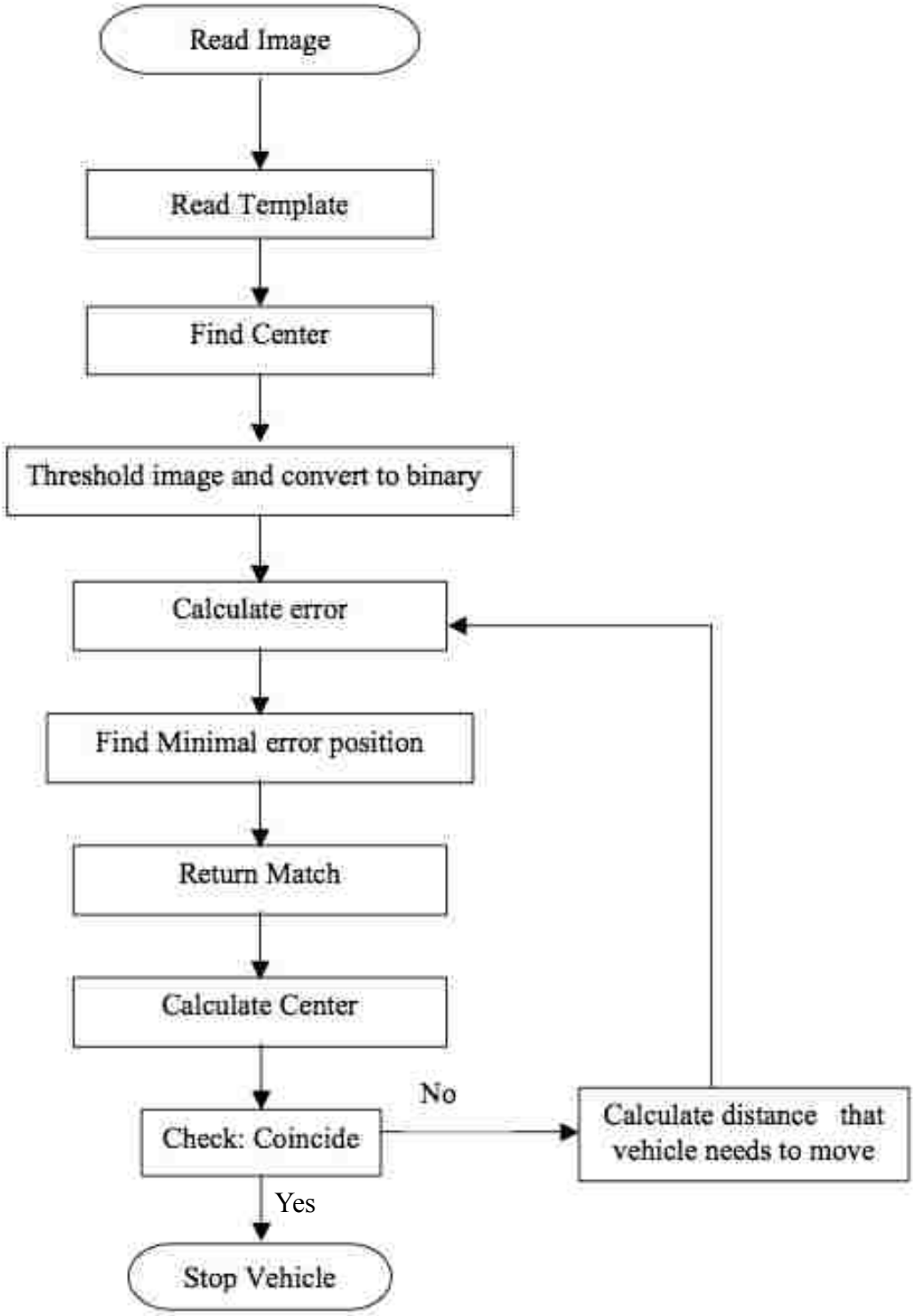


Figure 4.3 - Flow diagram of the process of vehicle positioning using ‘Template’ method.

Figure 4.3 explains the working of Template Matching algorithm. The calculation involves the template being shifted across the image to different offsets; then the superimposed values at this offset are subtracted and squared and the resultants are added together. This process is repeated for each location. The minimum numerical index is calculated and the location where the value of Euclidian distance is equal to the minimum error is returned. This location is taken as the location where the template almost or completely matched. After finding this location, the center of the template and the center of the source image are calculated and compared for verification. If both of the centers coincide, the vehicle is said to be perfectly normal to the plane of the target and is at the required location. If the two centers do not coincide, then the difference between their horizontal coordinates is calculated. This difference gives us a measure of the distance that the vehicle is required to be moved in order for it to be perfectly normal with the target plane. The difference can be negative or positive thus, giving the direction in which the vehicle should be moved.

Though template matching gives a very accurate position of the vehicle on bridge, it comes with its costs. The method is very slow since it involves matching each part of the image to a template. To obtain accurate results, the source image needs to bigger than the template image. Value of the Euclidean distance can be affected by noise in the image and usual properties of an image like brightness and contrast. The template that is used is a circle. The target object will appear an ellipse at a certain deviation. Thus, the template will not match. The deviation might be of few pixels which would mean that the position of the vehicle is still under the detection range mentioned in [1].

5. RESULTS

This section discusses the results that were recorded for different set of images using three different algorithms. Results discussed here are based upon speed (run-time) of an algorithm, range of detection, and flexibility. Both success using a certain method as well as complexities encountered have been discussed here. Probable solutions have been suggested for the encountered difficulties. Corresponding data can be found in Appendix A. Data regarding a particular image, total run time, and the result of processing the image in consideration has been mentioned. Total run time is the total time of the process from the moment an image was captured until the processing of that image was accomplished. Time taken for clicking and saving an image was timed by a stop watch or it was pre-set in the timer of the camera to take images at a certain time interval. Time taken for the processing of an image was given by each algorithm itself. An inbuilt MATLAB function was used to calculate this time. Hence, the run time should be viewed as a relative measure, which might change with different implementations, different settings of camera, different processing speed of the computing machine, usage of different software tool for writing the codes (C, C++) and any changes made in the algorithms might result in longer or shorter run time.

A few difficulties that were encountered during the initial set-ups and that were common to all of the methods are as following:

- Different settings of the camera had to be used at different distances ‘S’ between the target image and the camera/vehicle.

- Different adjustments resulted in additional time while taking images at different distances.
- For the camera used in the work described, images taken at a distance of 60 inches (152.40 cm) or greater were more clear and bright. Images taken within the ranges of 36 - 50 inches (91.44 - 127.00 cm) had smeared edges of the target circle but not noticeable. Images taken at a distance of 24 inches (60.96 cm) or below had noticeable smears in the edges of the target circle.
- It was very difficult to come up with a single radius of the target circle that worked at all of the tested distances. This limitation was due to the fact that combination of camera setting, distortion in the image and detection angle was different for each tested distance 'S'.

Most of the problems above can be solved by using cameras that have automatic focus, and brightness adjust settings. This arrangement can save time and would increase image quality that would further help in detection of the target object. Strain testing, and health monitoring procedures use pre-planned distances, so it is not difficult to determine a proper radius for the target object beforehand.

5.1. EXPERIMENTAL PARAMETERS

The experiment to collect the comparison data was set-up in a laboratory environment. An Imaging Source Firewire CCD Bayer camera was connected to a PC. A CCTV lens was used with the camera to capture the images. Table 5.1 summarizes the camera settings used. IC capture (version 2.0) software was used to take the images. The

images were taken in sequence multiple times. In this set-up, the target object was kept still and the camera was moved across the target horizontally. These images were later analyzed using MATLAB. The distance 'S' between the target object and the camera was varied from 12 inches (30.48 cm) to 60 inches (152.40 cm). The radius of the circle used as target object was varied between 0.415 inches (1.054 cm) to 2.5 inches (6.35 cm). Automatic shutter speed of 1/21 seconds was used throughout. It can be set to increased or decreased as per the requirement.

Table 5.1 - Camera settings and related adjustments.

Parameter	Setting
Camera Resolution	1024 by 768 pixel
Camera shutter speed	1/10000 to 30 s (manual)
Focal length of camera lens	12.5 to 75 mm (varied)
Pixel Size	4.65 μm by 4.65 μm
Camera to target distance	30.48 cm to 152.40 cm (varied)

5.2. CENTROID METHOD

The centroid method worked very well for the images that had a target circle with smeared edges, and for the images that were not very bright. It also worked well for the images taken in an outside environment at or around a bridge. Data for a few of the several images taken and tested has been presented in Appendix A. Distance between the

camera/vehicle and the target object, deviation between the desired location and the camera, centroid of the target object, and total time taken for the whole process (image capturing and image processing) has been given in the appendix A.

Total time taken by this method varied dependent upon the size of the target object in the source image. This method resulted in slow processing when the target object was smaller compared to the source image. It worked faster when the target object was relatively bigger than other objects in the source image. With eccentricity greater than 0.3, it took longer for the processing. If the deviation between the vehicle and the desired location was very small, processing time was much smaller. Also, if an automatically timed sequence of images are taken, it becomes easy to detect whether an image has been taken closer to the centroid. With larger deviations, processing time was relatively large. Average total time taken by this method was 7.99 seconds. This average time does not include some of the outliers that took above 150 seconds for the processing. In these cases, either the circle to be extracted was too small as compared to the source image, or there was significantly high noise in the background. These cases only comprise 3% of the total data set.

Since this method only considered the horizontal component of the centroid to calculate the deviation, eccentricity did not pose any problems. Images having target circle with an eccentricity of 0.3 or smaller were able to achieve a fast processing. Eccentricities of up to 0.66 were detected in the extracted circles. Another important aspect to be noted here is the detection range. This method worked well for a detection range of approximately 30º at one side which made the total detection angle view to be

approximately 60º. At a distance ‘S’ of 12 inches (30.48 cm) and a deviation of 6.5 inches (16.51 cm) from the horizontal component of the centroid of target circle, the viewing angle at camera was 28.66º. The complete target object was not detected in the source image beyond this range. This detection range might vary for a different camera and different radii of target object. This detection range worked well for smaller to bigger detectable radii.

This method worked well in case of noisy images that were clear enough for the purpose of circle extraction. This method did not work if the target circle that was to be detected did not lie completely within the source image. Even if the edges of the target circle were too close to the edges of source images such that they were about to merge into each other, this method failed to calculate accurate deviation. In order for this method to work, it is important that the target object clearly and completely lie within the source image. This method only worked when the vehicle/camera was completely aligned with the centroid of the target object or when the vehicle/camera moved ahead of the desired location such that it had to move backward in order to be at the desired location. But even with this limitation, this method is very effective since it can be used for any deviation until the target circle is completely within the view of the camera.

5.3. COLOR METHOD

The color method worked very well in different lighting conditions and with the images that had too much noise. Choosing a color different from the colors present in the surroundings of a bridge was not very difficult for the testing conditions used in this

work. In real field testings, it should be easy to find out one such color. Using a color specific algorithm made the detection and extraction of the target object very easy. This algorithm calculated eccentricity at the current position. A value of '0' was returned at the desired position i.e, when the camera was in the direct line of sight of the centroid of target object. At the positions other than this, values for eccentricities varied depending upon the deviation.

5.3.1. Approach I. If deviation is too large, value of the eccentricity returned was not accurate. This error was due to the fact that for a larger deviation, smaller the width of the target object as compared to its height. This situation lead to detection of a line segment with a width of '0' and height equal to the height of target object. This method only worked well for the images with bigger target object. The reason behind this result lies in the fact that all of the objects in source image were labeled and eccentricity was calculated for the biggest object in the image. For the images with smaller target object and probable bigger objects present in the surroundings, results were misleading.

Total time taken by this method depended upon the deviation of camera from the desired location. More the deviation, slower the processing was. Average time taken by this method was 6.13 seconds. This method was the fastest amongst the proposed three. Maximum allowable deviation for this algorithm was also largest amongst the proposed method. This algorithm can detect the target object and calculate eccentricity within the range of 90º. This range was the angle of view of the camera used. Table 5.2 lists the maximum allowable deviation for this algorithm as 60º because within this range, it was

tested that, target object came out to be the largest object. Also, using this range made sure that the camera was close to the railing on which target object was attached and

Table 5.2 – Comparison of methods in laboratory tests.

Methods	Maximum angle of detection	Relative speed of computation
Template Matching	40°	Very Slow
Centroid Location Method	60°	Moderate
Color Method (Approach I)	60°	Fast
Color Method (Approach II)	60°	Slow

larger portion of the source image consisted of the target object.

5.3.2. Approach II. This method was accurate for all type of images. It worked well for images that were very noisy or had smeared edges. It worked well for all of the tested values of distance 'S'. Maximum allowable deviation for this method was also 60º because it can only calculate eccentricity for a complete circle. Thus, source image should contain complete target object.

Average time taken by this method was much longer than approach I and the 'centroid' method but this method was still faster than the template matching algorithm. If number of objects in the source image were less, processing was faster. For the images with larger number of objects with comparable sizes, processing was slower. The reason

was that the labeling of objects took more time. In this case, it was difficult to tell that whether the eccentricity calculated was of the desired object. So, to get the best results out of this approach, the target object must be the biggest object in the source image.

5.4. TEMPLATE MATCHING

Template matching worked well for the images that contained clearly visible target object. In terms of results, it was the best method since it only considered the camera and the target object aligned when there was a 0 deviation or when the template matched the target object completely. Best results were given by this method when used for processing the images with a bigger target object. When template used for matching was smaller than the actual target object, this method worked very well. Also, it gave better results when the template was extracted from the source image of actual target object. Data for a few of the several images taken and tested has been presented in Appendix C. Distance between the camera/vehicle and the target object, deviation between the desired location and the camera, and total time taken for the whole process (image capturing and image processing) has been given in the appendix.

Total time taken by this method varied depending upon the size of target object and the template used for matching. Processing was relatively faster when the size of target object and template were comparable to each other. For the cases where template was extracted from the source image itself, processing was faster. This improvement was because the template was a the target object itself and thus, matching process became easier. While doing so, some additional time added to overall process but it still saved

significant time in the processing later. Overall, this process was very slow because the template seek a match at all of the possible positions in the source image. If source image was bigger in size, had noise, or target circle in the source image had smeared edges, this method took very long for the processing. Average time taken by this method was approximately 17 minutes which was considerably large as compared to other methods. For some of the images that were taken in outside environment with a large distance 'S', processing using this method took more than an hour.

Since, this method worked best when target object was bigger than the template, detection range for this method was lower as compared to other methods. This method worked well for a detection range of approximately 20º at one side which made the total detection angle view to be approximately 40º. Using this detection range, and smaller distance 'S' (12 inches for best results) between target and the vehicle/camera made sure that the target object was bigger in the source image. Using this setting lead to the fact that most of the source image comprised only of the target object. This adjustment helped in easy and fast template match.

Since, template used for this method is of a perfect circle, it was difficult to get a match for an elliptical target object. In some cases, where eccentricity was very close to 1, it worked but the processing time was very large. For this method to work, it was required that the target object detected in the image was a perfect circle or as close to it as possible.

5.5. COMPARISON OF THE ABOVE MENTIONED METHODS

All methods provided accurate position information within their maximum angle of detection as shown in Table 5.2. Template Matching worked very well with the ideal target images created, but it was the most limited in terms of maximum angle of accurate detection. For the desired position, it gave no error for the Euclidean distance for in-laboratory set-up where there was no noise or very minimal noise. In field environments, there would be significant noise in the background. With images captured by the camera in outside environment, this method may not work very well. Also in field environment, movement of the vehicle results in change of size and shape of the target object which will affect the quality of the results. This method is computationally very slow. Unlike template matching, Centroid Location gave very accurate results and was not limited in terms of maximum angle of detection. It is very easy to consider horizontal parameters and calculate the deviation. It had a moderate computational speed. Figure 5.3 shows a comparison between run times of the centroid method and the template matching method. The Centroid Location method should work well for complex images, taken in outside environment such as a bridge, despite of the background noise. But this method poses the difficulty of target object extraction. Target object is not always the biggest object in the source image. Using color feature of the target object gave the most encouraging results. The approach I of the color method worked best in terms of processing speed. Figure 5.4 shows a comparison between the run times of the color method (Approach I) and the centroid method. Like the Centroid method, the Color method should also work

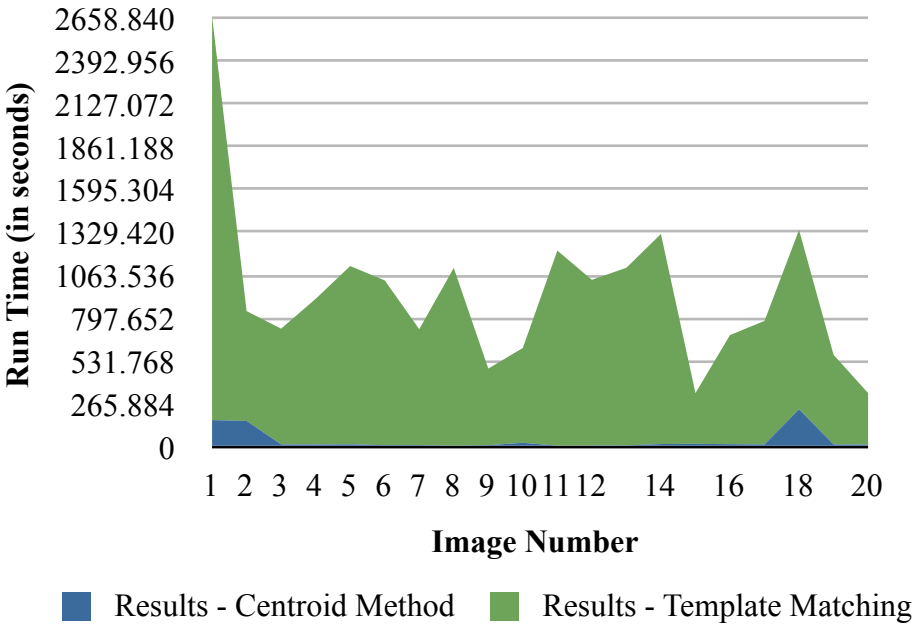


Figure 5.3 - Comparison between run time of centroid and template matching methods.

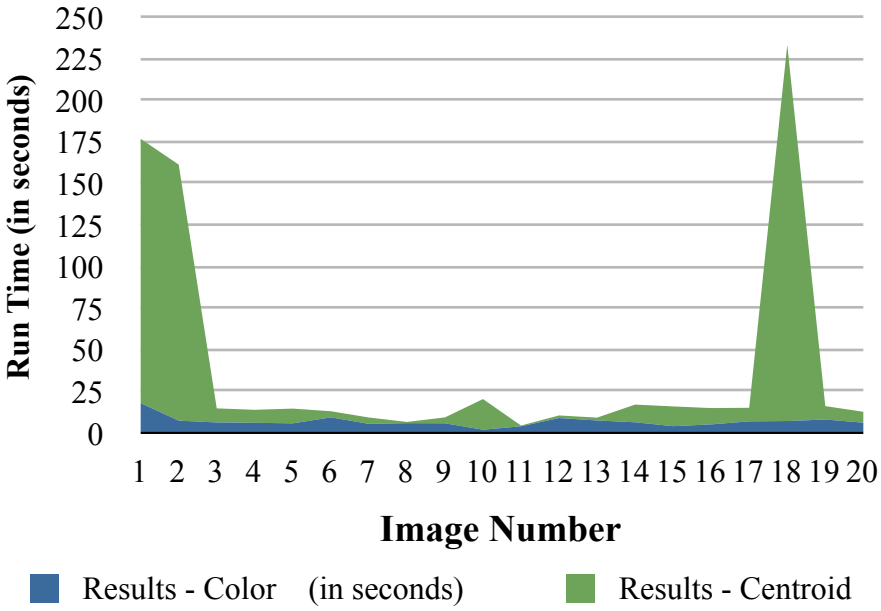


Figure 5.4 - Comparison between run time of centroid and color method (Approach I).

very well for images taken in outside environment. It was not limited in terms of the maximum angle of detection in the laboratory test and was computationally fast. The Color method was very effective in extracting the target object even in changing lighting conditions. For a field environment, the selection of an appropriate color and contrast for target object will be the main concerns.

6. CONCLUSION

The problem addressed in this thesis was to position a vehicle at a particular location (mid-span, quarter-span etc.) on a bridge to trigger a set of embedded sensors and collect data. Important discussion parameters were minimal personnel intervention, an easy-to-use set-up and instrumentation, minimized set-up and thus traffic closure time, and a low cost. The intent was to provide a better method for strain/load testing over conventional load testing and health monitoring methods by integrating an already existing triggering system with image processing. A passive target was used to position the vehicle using three image - processing algorithms. It is very easy to customize these algorithms and change them to meet expectations related to speed, better results in terms of positioning, or a compromise between both as and when needed. Image processing is a growing field in industrial applications but has not been used to a large extent in structural health monitoring systems. Image processing techniques can solve various speed and accuracy related questions while maintaining a low cost. Also, image processing tools like MATLAB are very user friendly and do not require much expertise to start working with. As these tools are software applications, there is no risk involved in multiple simulations. Thus, selected algorithms may be optimized for field monitoring applications.

A smart monitoring system is possible that includes sensors, sensor motes, and a triggering instrumentation. Sensors collect data about the external as well as internal condition of a structure. The triggering instrumentation can be used to retrieve the stored

data from the sensors. This triggering instrumentation is mounted on a vehicle that can be positioned above a sensor or sensor node to retrieve the data. An image processing approach has been used in the presented work to position the vehicle.

The experiment to collect the comparison data was set-up in a laboratory environment. An Imaging Source Firewire CCD Bayer camera was connected to a PC. A CCTV lens was used with the camera to capture the images. IC capture (version 2.0) software was used to take the images. The images were taken in sequence multiple times. The target object was a circle. In this set-up, the target object was kept still and the camera was moved across the target horizontally. These images were later analyzed using the Image Processing toolbox in MATLAB.

The proposed image-based processing is very simple and does not require any advanced operator expertise. It is a potential candidate to trigger an automated load test for a smart bridge. Each method described in Sections 4 and 5 had advantages and disadvantages. For the images that are not noisy, the Template Matching method worked best. However, this method is not effective when speed is a major concern. Using the Centroid Location method gave the most accurate results, but it suffered from target isolation and extraction. The best method overall was the Color method due to its computational speed and its ability to handle different lighting conditions. Despite these advantages, the Color method may be sensitive to noise in a field environment. The target color must be easily recognizable in the field surroundings. The selection of a method will depend on specific test conditions. Within the detection ranges at various separations, mentioned in reference [1], all of the above mentioned methods work well.

Future investigation should consist of conducting field testing to gather actual data and to study the behavior of proposed algorithms in actual field environment. This testing would help making the proposed algorithms more usable and practical. An image-processing system should be integrated as a trigger for the load test and tested in a field environment. Another possible algorithm could be based on area calculations rather than centroid calculation. In this case, principle of working will be equating the areas of two halves of a circle. In this algorithm, there would be no need to consider eccentricity feature of the target object. Also, since it would be based on area, it is expected to work very well even for the target object with smeared edges. Detection of the target object and finding the centroid will be considered as two different problems. Best methods would be found for both of these problems separately and then would be combined later. It is possible that the centroid of the target object is missed if there are any bumps on the path of the vehicle. Possible vertical deviation would be analyzed. Field parameters including lighting and scene clutter should be investigated for their influence on system performance.

APPENDIX A

DATA TABLE FOR CENTROID METHOD

The data table given below summarizes the results of 20 images processed using the 'Centroid' method. For each mentioned image, the distance between the target object and camera, actual deviation from the desired location, calculated deviation, and run time (calculated as described in Section 5) are mentioned.

S.No.	Image Name	'S' (in inches)	Actual Deviation (in inches)	Calculated Deviation (in inches)	Run time (in seconds)
1	Testimage1	12	0	0.001	158.84
2	Testimage2	24	6	5.778	153.85
3	Testimage3	60	6	5.400	10.46
4	Testimage4	60	10	9.098	9.81
5	Testimage5	24	8	8.100	10.86
6	Testimage6	24	10	11.091	5.69
7	Testimage7	24	6	5.775	5.81
8	Testimage8	12	0	0.074	1.73
9	Testimage9	12	8	7.133	4.53
10	Testimage10	60	10	10.176	18.40
11	Testimage11	12	8	8.103	2.44
12	Testimage12	36	12	12.001	3.56
13	Testimage13	48	12	11.890	3.71
14	Testimage14	12	8	7.253	10.63
15	Testimage15	12	10	10.001	11.86
16	Testimage16	36	0	0.011	9.80
17	Testimage17	60	12	10.786	10.16
18	Testimage18	60	0	0.038	226.00
19	Testimage19	24	10	9.998	8.05
20	Testimage20	24	14	13.908	8.53

APPENDIX B

DATA TABLE FOR COLOR METHOD

The data table given below summarizes the results of 20 images processed using the ‘Color’ method, Approach I. For each mentioned image, the distance between the target object and camera, value of the eccentricity calculated and run time (calculated as described in Section 5) are mentioned.

S.NO.	IMAGE NAME	‘S’ (in inches)	Eccentricity Calculated	Run time (in seconds)
1	Testimage22	12	0.8603	16.92
2	Testimage23	12	0.1778	6.42
3	Testimage24	60	0.1481	5.32
4	Testimage25	60	0.1343	5.11
5	Testimage26	60	0.1332	4.79
6	Testimage27	60	0.0847	8.37
7	Testimage28	60	0.3140	4.48
8	Testimage29	60	0.0000	4.89
9	Testimage30	60	0.1212	4.82
10	Testimage31	24	0.0000	0.97
11	Testimage32	12	0.1120	3.03
12	Testimage33	36	0.3112	7.97
13	Testimage34	48	0.5000	6.51
14	Testimage35	12	0.4135	5.43
15	Testimage36	12	0.6740	3.09
16	Testimage37	36	0.0230	4.16
17	Testimage38	12	0.1145	5.98
18	Testimage39	12	0.1677	6.17
19	Testimage40	24	0.0015	7.09
20	Testimage41	24	0.0000	5.20

APPENDIX C

DATA TABLE FOR TEMPLATE MATCHING METHOD

The data table given below summarizes the results of 20 images processed using the ‘Template Matching’ method. For each mentioned image, the distance between the target object and camera, actual deviation from the desired location, calculated deviation, result in terms whether the template matches with target object, and run time (calculated as described in Section 5) are mentioned.

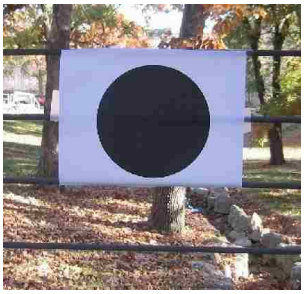
S.No.	Image Name	‘S’ (in inches)	‘Match’ or ‘DoesNotMatch’	Actual Deviation (in inches)	Calculated deviation (in inches)	Run time (in seconds)
1	Testimage1	12	DoesNotMatch	2.00	1.977	4770.0
2	Testimage2	24	DoesNotMatch	2.00	2.003	680.0
3	Testimage3	60	DoesNotMatch	2.00	2.109	716.7
4	Testimage4	60	DoesNotMatch	6.00	6.001	902.1
5	Testimage5	24	DoesNotMatch	6.00	6.000	1,104.0
6	Testimage6	24	DoesNotMatch	10.00	9.998	1,020.5
7	Testimage7	24	DoesNotMatch	12.00	12.001	718.0
8	Testimage8	12	Match	0.00	0.012	1,100.4
9	Testimage9	12	Match	0.00	0.001	474.3
10	Testimage10	60	DoesNotMatch	10.00	10.025	587.0
11	Testimage11	12	DoesNotMatch	12.00	11.687	1208.2
12	Testimage12	36	DoesNotMatch	6.00	5.013	1,025.9
13	Testimage13	48	DoesNotMatch	6.00	6.038	1,100.5
14	Testimage14	12	DoesNotMatch	12.00	11.908	1,301.0
15	Testimage15	12	DoesNotMatch	10.00	9.280	314.0
16	Testimage16	36	DoesNotMatch	12.00	12.000	675.4
17	Testimage17	60	DoesNotMatch	12.00	10.897	765.0
18	Testimage18	60	DoesNotMatch	12.00	12.005	1,110.5
19	Testimage19	24	DoesNotMatch	12.00	10.680	555.3
20	Testimage21	24	Match	0.00	0.301	320.0

APPENDIX D

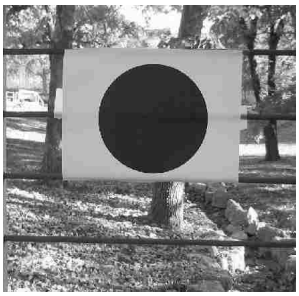
ORIGINAL AND RESULTANT IMAGES AT VARIOUS STAGES

This appendix contains original images and images as a resultant of certain operations used in the proposed algorithms.

CENTROID METHOD:



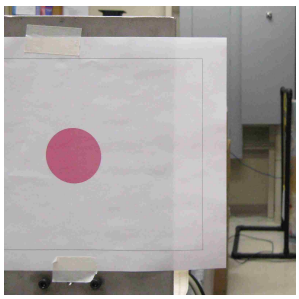
Original Image fed as an input to the algorithm.



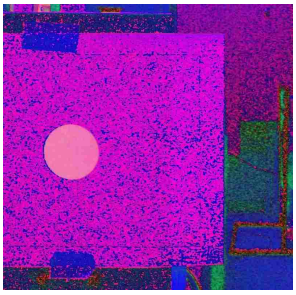
This image is a resultant of ‘rgb2gray’ operation which converts a true color image into a gray level based intensity image.

This is the binary image that results after thresholding operation. In this case, target object is assigned a value of 255 which results into a white color and all other objects in the background are assigned a value of 0 which results in their black color.

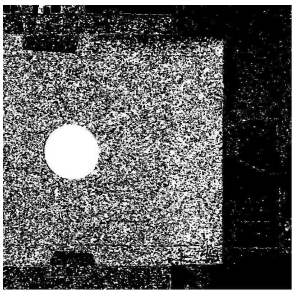
COLOR METHOD:



Original Image with a bright colored target fed as an input to the algorithm.

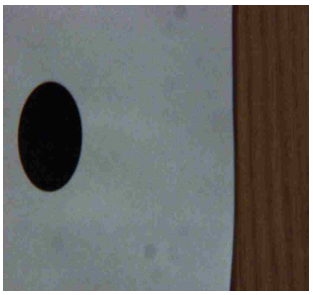


This image is a resultant of ‘rgb2hsv’ operation which converts a true color image into a an image based on Hue, Saturation and Value of the pixels in the image.

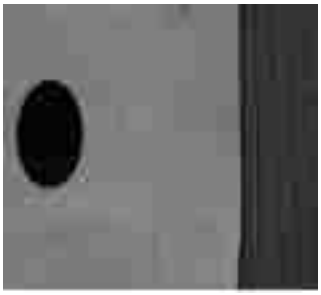


This is the binary image that results after thresholding operation. In this case, target object is assigned a value of 255 which results into a white color and all other objects in the background are assigned a value of 0 which results in their black color. Noise due to lighting conditions is visible in this image. Respective color algorithm was able to provide accurate measurements for the noisy images also.

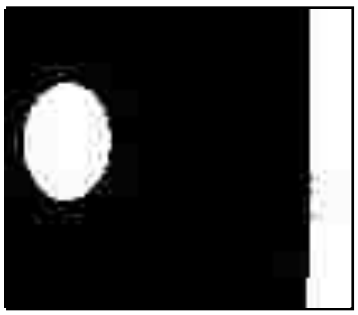
TEMPLATE MATCH:



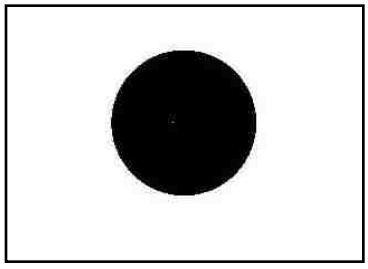
Original Image fed as an input to the algorithm



This image is a resultant of ‘rgb2gray’ operation which converts a true color image into a gray level based intensity image.



This is the binary image that results after thresholding operation. The target object is assigned a white color and all other objects in the background are assigned a black color.



This is the template used for matching with the target object in the original image.

NOTE: Black colored boundaries in the above two images are made for the purpose of presentation only and are not a part of the image.

APPENDIX E

MATLAB CODES FOR PROPOSED ALGORITHM USING CENTROID METHOD

This appendix contains the MATLAB code implementation of the ‘Centroid’ method. A brief description of the commands used is also given. This algorithm was fed an image in JPEG format as an input and the resultant output was centroid location (pixel position), center location of the source image (pixel position), resultant deviation and the total time taken by the algorithm to process the input image.

```
tic                          % Initiates timer.
x = imread('testimage.jpg'); % Image named ‘testimage.jpg’ is read and saved in ‘x’.
x = rgb2gray(x);             % ‘rgb2gray’ converts true color image into gray level
                             % based intensity image.

[w,h]=size(x);               % Record the dimensions of image ‘x’ in [w,h].
for i=1:1:w                  % This ‘for’ loop thresholds the image and convert it to a
for j=1:1:h                  % binary image (White = 255 and Black =0).
if(x(i,j)>100)
x(i,j)=0;
else
x(i,j)=255;
end
end
End

[L,num]=bwlabeln(x,8);       % Returns an object ‘L’ of the same size as ‘x’ with labeled
                             % connected objects.
for i=1:1:num                % This ‘for’ loop labels the largest object.
a(i)=length(find(L==i))
end;

b=max(a);
maxlabel=0;
for i=1:1:num
if (a(i)==b)
maxlabel=i;
end
End
[c,d]=find(L~=maxlabel);     %  This ‘for’ loop is used for making all of the  for
i=1:1:length                 %  objects, other than the biggest object (target object),
```

```
s=c(i);                 %  a part of the background.
g=d(i);
x(s,g)=1;
End

subplot(3,1,3)
imshow(x)               % Pops up the image 'x'.
centerx=w/2             % Calculates the  x component of the center of the image.
centery=h/2             % Calculates the y component of the center of the image.
a=zeros(h,1);           % This set of statements is used to calculate horizontal and
b=zeros(w,1);           % vertical projection of the target object.

for row=1:1:w
for col=1:1:h
if (x(row,col)>50)
a(row)=a(row)+1;
end
end
End

for col=1:1:h
for row=1:1:w
if (x(row,col)>50)
b(col)=b(col)+1;
end
end
End
area=0;                 % Calculated the area of the target object.

for i=1:1:h
area=area+a(i);
End
ibar=0;                 % This set of statements calculate the centroid of the target
for i=1:1:h             %  object using projections and area calculated earlier.
ibar=ibar+i*a(i);
End

ibar=ibar/area
jbar=0;
for i=1:1:w
jbar=jbar+i*b(i);
End
jbar=jbar/area
```

```
dev=abs(centery)-jbar
toc                         % Stops the timer initiated earlier.
t=toc
t;                          % Gives the total time taken for the processing of test
                            %  image using  this code.
```

APPENDIX F

MATLAB CODES FOR PROPOSED ALGORITHM USING COLOR METHOD (APPROACH I)

This appendix contains the MATLAB code implementation of the ‘Color’ method. A brief description of the commands used is also given. This algorithm was fed an image in JPEG format as an input and the resultant output was the value of eccentricity of the target circle and the total time taken by the algorithm to process the input image.

```
Tic                                 % Initiates timer.
[x,map]=imread('testimage.jpg');    % Image named ‘testimage.jpg’ is read and
                                    % saved in ‘x’.

x=rgb2hsv(x);          % Converts the input matrix representing intensities of red,
                       % green, and blue, respectively to an output matrix
                       % representing hue, saturation, and value, respectively.

r=x(:,:,1);
g=x(:,:,2);
b=x(:,:,3);
[w,h]=size(r);

r1=0.9156;
g1=0.4392;

for i=1:1:w
for j=1:1:h
 if(r(i,j)>(r1-0.075) && r(i,j)<(r1+0.08))
    r(i,j)=1;
 else
    r(i,j)=0;
end
end
end
subplot(3,1,3)
imshow(r);

[L,num]=bwlabeln(r,8);      % Returns an object ‘L’ of the same size as ‘x’ with labeled
                            % connected objects.

for i=1:1:num
   a(i)=length(find(L==i));
```

```
end

e=max(a);                  % Finding the label with maximum number of pixels
                           % i.e. the largest object
maxlabel=0;                % Initialization

for i=1:1:num              %  This 'for' loop is used for determining the label of the
                           % largest object
if(a(i)==e)
    maxlabel=i;
end
end

for i=1:1:w
for j=1:1:h
  if L(i,j)==maxlabel;
     L(i,j)=1;
  else
     L(i,j)=0;
end
end
end

height=0;
for i=1:1:w
for j=1:1:h
    if (L(i,j)==1)
       height=height+1;
break;
end
end
end

width=0;
for i=1:1:h
for j=1:1:w
  if (L(j,i)==1)
     width=width+1;
break;
end
end
end
```

```
dimdiff=height-width          % Measures the difference in the width and height of the
target object.
ecc=sqrt((height./2).^2-(width./2).^2)./(height./2);     % This formula is used for
                                                         % measuring eccentricity
ecc=abs(ecc)
figure(1)
Toc                           % Stops the timer initiated earlier.
t=toc
t;                            % Gives the total time taken for the processing of test
                              %  image using  this code.
```

APPENDIX G

MATLAB CODES FOR PROPOSED ALGORITHM USING COLOR METHOD (APPROACH II)

This appendix contains the MATLAB code implementation of the ‘Color’ method using region property (‘eccentricity’ in this case) calculations. A brief description of the commands used is also given. This algorithm was fed an image in JPEG format as an input and the resultant output was the value of eccentricity of the target circle and the total time taken by the algorithm to process the input image.

```
tic                              % Initiates timer.

[x,map]=imread('testimage.jpg');      % Image named ‘testimage.jpg’ is read and
                                      % saved in ‘x’.
x=rgb2hsv(x);                    % Converts the input matrix representing intensities of red,
                                 % green, and blue, respectively to an output matrix
                                 % representing hue, saturation, and value, respectively.
r=x(:,:,1);
g=x(:,:,2);
b=x(:,:,3);
[w,h]=size(r);

r1=0.9156;                       % Thresholding and conversion to binary image based on
g1=0.4392;                       % the color of the target object.
for i=1:1:w
for j=1:1:h
if (r(i,j)>(r1-0.075)&&r(i,j)<(r1+0.08))
   r(i,j)=1;
else
   r(i,j)=0;
end
end
end

subplot(3,1,3)
imshow(r);

[L,num]=bwlabeln(r,8);           % Returns an object ‘L’ of the same size as ‘x’ with
                                 % labeled connected objects.
for i=1:1:num
   a(i)=length(find(L==i));
```

```
end

e=max(a);              % Finds the label of an object with maximum number of pixels.

maxlabel=0;            % This 'for' loop calculates the label of largest
for i=1:1:num          % object in the source image.
 if (a(i)==e)
   maxlabel=i;
 end
end

for i=1:1:w
for j=1:1:h
if L(i,j)==maxlabel;
      L(i,j)=1;
else
    L(i,j)=0;
end
end
end

for i=1:1:w                    % This 'for' loop measures Eccentricity for
for j=1:1:h                    % maximum labeled region in the label matrix L.
If L(I,j)==1;
STATS= regionprops(L,'Eccentricity') % An inbuilt MATLAB function that
                                     % calculates eccentricity.
End
End

STATS                          % Displays the calculated eccentricity.

figure(2)
subplot(3,1,3)
imshow(L)
toc                            % Stops the timer initiated earlier.
t=toc
t;                             % Gives the total time taken for the processing of
                               %  test  image using  this code.
```

APPENDIX H

MATLAB CODES FOR PROPOSED ALGORITHM USING TEMPLATE MATCHING METHOD

This appendix contains the MATLAB code implementation of the ‘Template Matching’ method. A brief description of the commands used is also given. This algorithm was fed an image in JPEG format as an input and the resultant output was a text message to indicate whether the template matched (‘Match’) or not (‘DoesNotMatch’) and the total time taken by the algorithm to process the input image.

```
tic                          % Initiates timer.

x=imread('testimage.jpg');   % Image named ‘testimage.jpg’ is read and saved in ‘x’.
x=rgb2gray(x);               % ‘rgb2gray’ converts true color image into gray level
                             % based intensity image.

subplot(3,1,1)
imshow(x)                    % Pops up the image ‘x’.

temp=imread('template.jpg'); % Template image is read.
temp=rgb2gray(temp);         % Template image is converted into gray level image.
[wtemp,htemp]=size(temp);    % Size of the template image is saved into [wtemp,htemp].
temp1=zeros(wtemp,htemp); % A matrix of the dimensions of template is created to store
                             % errors at each location.
[w,h]=size(x);               % Record the dimensions of image ‘x’ in [w,h].

E=zeros(w-wtemp,h-htemp);
for i=1:1:w                  % This ‘for’ loop thresholds the image and convert it to a
for j=1:1:h                  % binary image (White = 255 and Black =0).
if(x(i,j)>100)
  x(i,j)=0;
else
  x(i,j)=255
end
end
End

for m=1:1:wtemp              % This ‘for’ loop thresholds the template image and
for n=1:1:htemp              % convert it to a binary image (White = 255 and Black =0).
if(temp(m,n)>100)
 temp(m,n)=0;
else
```

```
 temp(m,n)=255;
end
end
end
flag=0;                         % This flag is set to '0' when template matches an object in
                                % the source image.

for i=1:1:(w-wtemp)             % This set of 'for' loops calculates the error at all points in
for j=1:1:(h-htemp)             % the image and save the center information where there
for p=1:1:wtemp                 %  is a match and record (i,j) into (y,t).
for q=1:1:htemp
E(i,j)=E(i,j)+(double(x(i+p-1,j+q-1))-double(temp(p,q)))^2;
end
end
end
End

for i=1:1:(w-wtemp)             % This 'for' loop search for the perfect match of
for j=1:1:(h-htemp)             % the template in the source image.
if(E(i,j)==minE)
y=i;
t=j;
calccenterx=y+abs(wtemp./2)
calccentery=t+abs(htemp./2)
end
end
End

for h=1:1:(w-wtemp+1)
for v=1:1:(h-htemp+1)
temp1=x(h:h+wtemp-1,v:v+htemp-1);
if(isequal(temp,temp1)),
flag=1;
 v=h-htemp+1;
 h=w-wtemp+1;
 else
 temp1=zeros(wtemp,htemp);
end
end
end
if(~flag)
   sprintf('DoesNotMatch')      % Displays 'DoesNotMatch' when the template
Else                            % does not match the target object.
```

```
   sprintf('Match')          % Displays ‘Match’ when the template matches.
   h=h
   v=v
end
toc                          % Stops the timer initiated earlier.
t=toc
t;                           % Gives the total time taken for the processing of test
                             %  image using  this code.
```

REFERENCES

[1] Steve E. Watkins, Theresa M. Swift, and James W. Fonda, "Development of Autonomous Triggering Instrumentation," *Smart Structures/NDE 2008: Sensor and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems*, Proc. SPIE. **6932(2)**, 693235.1-693235.10 (2008).

[2] W. B. Spillman Jr., "Sensing and Processing for Smart Structures," *Proceedings of the IEEE*, **84(1)**, 68-77 (January 1996).

[3] Eric Udd, "Fiber Optic Smart Structures," *Proceedings of the IEEE*, **84(6)**, 884-894 (June 1996).

[4] Steve E. Watkins, "Smart Bridges with Fiber Optic Sensors," *IEEE Instrumentation and Measurement Magazine*, **6(2)**, 25-30 (June 2003).

[5] A. E. Aktan , A. J. Helmicki, and V. J. Hunt, "Issues in health monitoring for intelligent infrastructure," *Smart Mater. Struct.* (**7)**, 674-692 (1998).

[6] American Society of Civil Engineers, "The 2005 Report card for America's infrastructure," ASCE, (2005). Available WWW: http://www.asce.org/reportcard.

[7] Steve E. Watkins, James W. Fonda, and A. Nanni, "Assessment of an Instrumented Reinforced-Concrete Bridge with Fiber-Reinforced-Polymer Strengthening," *Opt. Eng.*, **46(5)**, 051010 (2007).

[8] J. R. Casas, and P. J.S Cruz, "Fiber Optic Sensors for Bridge Monitoring," *Journal of Bridge Engineering*, **8(6)**, 362-373, (November/December 2003).

[9] Texas Department of Transportation, "Bridge Inspection Manual," Available WWW: ftp://ftp.dot.state.tx.us/pub/txdot-info/gsd/manuals/ins.pdf , revised (2002).

[10] G. Fu, *Inspection and monitoring techniques for bridges and civil structures*, (CRC Press, Woodhead Publishing Limited, Cambridge, England, 2005).

[11] M. E. Moore, B. M. Phares, B. A. Graybeal, D. D. Rolander, and G. A. Washer, "Reliability of Visual Inspection of Highway Bridges," Federal Highway Administration, Report FHWA-RD-01-020, Washington, DC (2001).

[12] P. C. Furrow, R. T. Brown, and D. B. Mott, "Fiber optic health monitoring system for composite bridge decks," *Proceedings of SPIE*. **3988**, 380-390 (2000).

[13] Steve E. Watkins, "Smart Bridges with Fiber-Optic Sensors," *IEEE Instrumentation and Measurement Magazine*, **6(2)**, 25-30 (2003).

[14] R. L. Idriss, M. B. Kodindouma, and M. A Davis, "Multiplexed Bragg grating optical fiber sensors for damage evaluation in highway bridges," *Smart Mater. Struct.,* **(7)**, 209-216 (1998).

[15] B. Culshaw, C. Michie, P. Gardiner, and A. McGown, "Smart Structures and Applications in Civil Engineering," *Proceedings of the IEEE*, **84(1)**, 78-86 (1996).

[16] R. Duda, and P. Hart, "Use of the Hough Transform to Detect Lines and Curves in Pictures," *Communications of the ACM*, **15**, 11-15 (1975).

[17] A. A. Rad, K. Faez, and N. Qaragozlou, "Fast Circle Detection Using Gradient Pair Vectors," *Proc. VIIth Digital Image Computing: Techniques and Applications*, 879-887 (2003).

[18] G. M. Schuster, and A. K. Katsaggelos, "Robust circle detection using a weighted MSE estimator," *International conference on Image Processing (ICIP)*, 2111-2114, (2004).

[19] S. Y. Guo, X. F. Zhang, and F. Zhang, "Adaptive randomized Hough Transform for circle detection using moving window," *Proceedings of the Fifth Conference on Machine Learning and Cybernetics*, 3880-3885, (August 2006).

[20] D. H. Ballard, and C. M. Brown, *Computer Vision*, (Prentice-Hall, Englewood Cliffs, New Jersey,1982), 66-71.

[21] S. Tsuji, and F. Matsumoto, "Detection of ellipses by a modified Hough Transformation," *IEEE Transactions on Computers*, **C-27 (8)**, 777-781 (1978).

[22] L. Xu, E. Oja, and P. Kultanan, "A new curve detection method: randomized Hough transform (RHT)," *Pattern Recognition Letter*, **11 (5)**, 331-338 (1990).

[23] R. Yip, P. Tam, and D. Leung, "Modification of Hough transform for circles and ellipse detection using a 2-dimensional array," *Pattern Recognition*, **25 (9)**, 1007-1022 (1992).

[24] American Association of State Highway and Transportation Officials, "Standard Specifications for Highway Bridges," Sixteenth Edition, AASHTO, Washington D.C., (1996).

[25] B.K.P. Horn, *Robot Vision*, (The MIT Press, Cambridge, Massachusetts, 1986), 53-61.

[26] E. R. Davies, *Machine Vision: Theory, Algorithms, and Practicalities*, (Academic Press, San Diego, 1997), 211-248.

[27] B. O'Neill, *Elementary Differential Geometry*, (Academic Press, New York, 1967), 42-49.

VITA

Amardeep Kaur was born in Hoshiarpur (Punjab), India on January 7, 1985. She completed her Bachelor of Technology in Electronics and Communication Engineering at Punjab Technical University in Punjab, India in May 2006. Amardeep started her Master of Science program with the Department of Electrical and Computer Engineering at Missouri University of Science and Technology in August, 2007. While a graduate student, Amardeep served as the Secretary and then, the Vice-President of Graduate Student's Governing Body (Council of Graduate Students) of Missouri S&T.

With a penchant in Electrical Engineering and liking to know more about her field, Amardeep worked under Dr. Steve E. Watkins to write one paper, proposing an image processing approach for vehicle positioning that can be used in strain testing and health monitoring of "smart" bridges. She received her Master of Science Degree in August of 2009.