Real Time Rail Line Anchor Inspection: Context of Bangladesh

Omid Chowdhury

ID: 10101031

Supervisor

Rubel Biswas

Department of Computer Science & Engineering

December 2014

BRAC University, Dhaka, Bangladesh

Abstract

Rail inspection is an essential task in railway maintenance. It is periodically needed for preventing dangerous situations and ensuring safety in railways. In Bangladesh it has been seen many train accidents occur due to missing of hooks which attach the tracks to the ground. At present, this task is operated manually by a human operator who periodically walks along the track searching for visual anomalies. This manual inspection is lengthy, laborious and subjective.

This thesis presents a machine vision-based technique to automatically detect the presence of rail line hook using Shi-Tomasi and Harris-Stephen feature detection algorithms collaboratively. This inspection system has been carried out on videos acquired by a digital camera installed on a cart. The combination of these two algorithms has been successful to identify scenarios with attached and missing hooks with accuracy of 85.7%. Hence it can be concluded that the proposed system is robust.

Acknowledgement

My deep thanks to my thesis supervisor Mr. Rubel Biswas and to my co supervisor Mr. Rubayat Ahmed Khan for their generous guidance and continuous support throughout the work.

I am extremely thankful to my parents, family members and friends for their support and encouragement.

Finally I thank BRAC University for giving me the opportunity to complete my BSc. Degree in Computer Science and Engineering.

Table of Contents

1		**Introduction**	**6**
	1.1	**Proposed System**	**8**
	1.2	**Thesis Outline**	**9**
2		**Previous Work**	**10**
3		**RGB VS Gray scale Overview**	**11**
	3.1.1	**RGB Color Model**	**11**
	3.1.2	**RGB Images**	**12**
	3.2	**Gray scale Images**	**14**
	3.3	**RGB to Gray scale Conversion**	**15**
4		**Technical Overview**	**16**
	4.1	**Overview of Harris Stephen**	**16**
	4.2	**Overview of Shi &Tomasi**	**17**
5		**System Design**	**19**
	5.1	**Tool**	**19**
	5.2	**Methodology**	**19**
	5.3	**Data Collection**	**21**
	5.4	**Creating Data Set**	**23**
	5.4.1	**Pre Processing**	**23**
	5.4.2	**Corner Point Detection Using Shi-Tomasi**	**24**
	5.4.3	**Corner point Detection Using Harris-Stephen**	**25**
	5.4.4	**Training**	**26**
	5.5	**Matching**	**26**
	5.5.1	**Threshold**	**28**
6		**Experimental Results**	**30**
7		**Accuracy and Comparison**	**31**

8	**Conclusion and Future Work**	**33**
9	**References**	**34**

List of Figures

1	**Hook Present**	**6**
2	**Hook Absent**	**6**
3	**RGB Model**	**11**
4.1	**RGB Image**	**12**
4.2	**Red Channel**	**13**
4.3	**Green Channel**	**13**
4.4	**Blue Channel**	**14**
5	**Grayscale**	**15**
6	**A Visible Hook Plate**	**21**
7.1	**Unclear Hook**	**22**
7.2	**Unclear Hook**	**22**
7.3	**Unclear Hook**	**22**
8	**Samples of Training Images (After Pre Processing)**	**23**
9	**Possible Shi-Tomasi Features and Valid Points**	**24**
10	**Possible Harris-Stephen Features and Valid Points**	**25**
11	**Feature Matching**	**26**

12.1	**Matched Harris-Stephen Feature Points**	**27**
12.2	**Matched Shi-Tomasi Feature Points**	**27**
13.1	**True Positive Images**	**28**
13.2	**True Negative Images**	**29**
13.3	**False Positive Images**	**29**
13.4	**False Negative Images**	**29**

List of Tables

1	**Experimental Results of Shi-Tomasi and Harris-Stephen Feature Detectors**	**30**
2	**Comparison**	**32**

1. Introduction

Rail line hooks or anchors are the metallic components that join the lines with the sleepers. In Bangladesh we find a pair of such hooks joining the lines every 3.5 (approx) feet. Absence of this component results in derailment. During the period of 2009-2010 there were 403 of such case [1]. The conditions of these hooks are inspected manually by a railway employee travelling along the track. Therefore it is very much time consuming and non reliable as the quality of work differs from person to person. In order to speed up the process, provide constant good quality and to minimize the human errors, my paper aims to find a solution through a system that will monitor the anchors using video processing and determine whether they are missing or not. Figure 1 and Figure 2 show conditions of attached and missing hook respectively.

Figure 1: Hook Present

Figure 2: Hook Absent

Although different machine vision methods have been built up in the field of railways, not a great deal of work have been done focusing particularly on rail anchors. Proposed system of

Rubayat Ahmed Khan and Samiul Islam in 2014[2] has detected rail line anchors from still images using a combination of one feature point detection method, Shi-Tomasi[3] and one blob detection method, SURF[4]. In my paper I have combined two feature detection techniques, Shi-Tomasi and Harris-Stephen[5], to detect the presence of anchors from video.

The steps I have followed in my propose system are video acquisition, frame segmentation, preprocessing, detection, extraction and matching. Preprocessing is very important as it is in this stage where different types of noise are removed and the focused area is enhanced. After preprocessing came the frame segmentation process, interest point detection, feature extraction and matching. A video consist of frames. Videos are segmented into frames and detection process is carried out on specific frames of interest. Features of an image are pieces of information which are extracted by applying neighborhood operations. Neighborhood operations contain a square or circular neighborhood of size N with a function f and centered at p. This matrix is iterated over every pixel of the image and a value is calculated at the centre using the function f. The features I have used to detect hooks are corner points.

Corners refer to points where edges intersect. Feature detectors include BRISK[6,7], Harris-Stephen, Shi-Thomas, etc. which are used to detect corners. In my paper I have used Harris-Stephen and Shi-Thomasi algorithms collaboratively to identify the presence or absence of anchors. This pair of detectors (Harris-Stephen, Shi-Tomasi) is applied over a set of sample/training images and on the video frames that contain anchors.

1.1 Proposed System

The purpose of my thesis is to detect rail line anchors from videos using the different feature detection algorithms mentioned above. Videos were acquired using digital camera from two locations in Dhaka city – Khilgaon and Malibagh.

Videos were segmented into frames and training samples were chosen from there. At first the each frames was cropped using a 154 x 123 window to remove stones on both horizontal sides. The RGB frames were then converted to grayscale. The principle reason for this conversion was to reduce a 3 channel color (R = red, G = green, B = blue) to a 2 channel color (black and white). There are 256 gray levels in an 8 bit storage with the intensity of each pixel ranging from 0 – 255 [8]. On the other hand in 8 bit storage the intensity of each pixel of a RGB image is 24 bits [8].Therefore processing a grayscale frame will take significantly less space and time. The second reason was the image parameter of the detectors requires a grayscale image.

After the conversion was done the first step was to detect corners using the pair. The detectors detect interest points and return respective objects. The next stage was to extract the features of the points found in the first step. Features include the descriptors of the points and their corresponding locations. The final step was matching. The feature descriptors of the frames of the test videos were matched with the descriptors of the samples. The number of matches had to be greater than a particular threshold to determine the presence of a hook.

1.2 Thesis Outline

Section 2 describes the previous work done on rail components. Section 3 describes grayscale and RGB images and the conversion. This is followed by technical review, section 4, where the mechanism of the algorithms we have used is described. Section 5 describes the whole procedure, how we have detected hooks using the algorithms mentioned above. Section 6 and 7 deals with the result and the comparison and finally concluded with our future work in section 8.

2. Previous Work

Many machine vision techniques have been developed which deal with railway components but not many are there that focus directly on hooks. Rubayat Ahmed Khan and Samiul Islam[2] have already done detecting hooks using Shi-Tomasi and SURF algorithms. They have done it on still images.

The paper from the IBM T.J. Watson Research Centre [9], worked on railway components like anchors/hooks, plates, etc. It has detected hooks based on Adaboost classifier. Multiple classifierswere run simultaneously but the detection results were selected based on one classifier which had the highest detection in the last 50 frames. 50 was the threshold used in this paper.

Paper [10] by Yohann Rubinsztejn, University of Manchester, proposed to detect rail anchors by using Viola-Jones object detection framework. This framework uses integral image technique to compute the Haar Wavelet features, used Adaboost as the learning algorithm and lastly it uses a cascade of classifiers. A set of positive and negative images were used to construct the dataset. Then a set of feature template was built from it. The training set along with the feature template was fed to Adaboost. This was how the detector was created. Then it was applied on rail images to detect true and false instances of anchors.

Paper [11] also worked on a number of railway components and detection of hooks was a part. In this paper correlation was used. Although correlation is neither scale nor orientation invariant it could be applied using filters at multiple orientation and scales and then merging the correlation results. Of many correlation techniques Optimal Tradeoff Maximum Average Correlation Height (OT – MACH) technique was used. A set of images were collected and categorized into classes and were then trained using the algorithm. This method was used for detecting grounded hooks. In order to find the missing ones two approaches were taken. Firstly the OT_MACH algorithm itself but it was not very accurate and secondly by measuring the average interval and deviation between anchors. When anchors were detected the gap was smaller than the gap which arose due to the presence of a missing anchor.

In my paper I am proposing to detect anchors using combination of two feature detecting algorithms. I have excluded blob because the hook scenario consists of intersection of physical edges and researches have proved that corner detectors do work better on such circumstance.

Also, I believe that using features would be a simpler technique rather than using different kinds of classifiers and training algorithms. The algorithms I have used are scale and orientation invariant. Moreover I am showing a quantitative analysis of accuracy of the different algorithms we have used.

3. RGB vs Grayscale Overview

3.1.1 RGB Color Model

The RGB color model is color model where red (R), green (G) and blue (B) colors are merged together to produce a range of colors. As 3 colors define the RGB model, the geometric representation is a three dimensional cube. Each node of the cube represents a particular color. Each color has three components – red, green and blue and each component ranges from 0 – 1. Therefore any color is represented by R, G or B where the presence of a component is denoted by 1 and the absence is denoted by 0. White is represented by (1, 1, 1) whereas black is represented by (0, 0, 0).

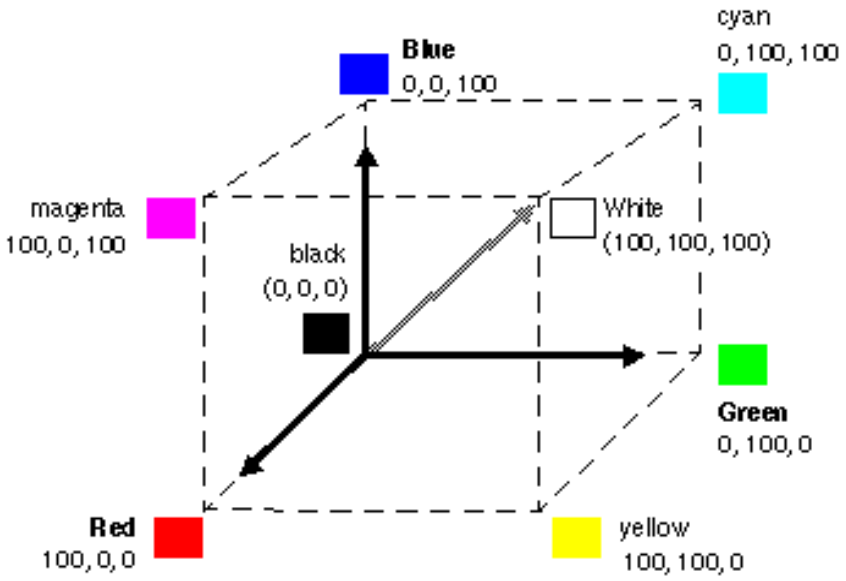


Figure 3: RGB Color Model

3.1.2 RGB Images

RGB images consist of three channels and they are red (R), green (G) and blue (B). In a 24-bit image each channel contains 8 bits. The whole RGB image is composed of three images one for each channel. Each image stores pixel with brightness ranging from 0 to 255. The figure 4.1shows RGB image with all the 3 components present, figure 4.2only has the red channel on, figure 4.3 and figure 4.4 show green channel and blue channel respectively.

Figure 4.1 RGB Image

Figure 4.2 Red Channel

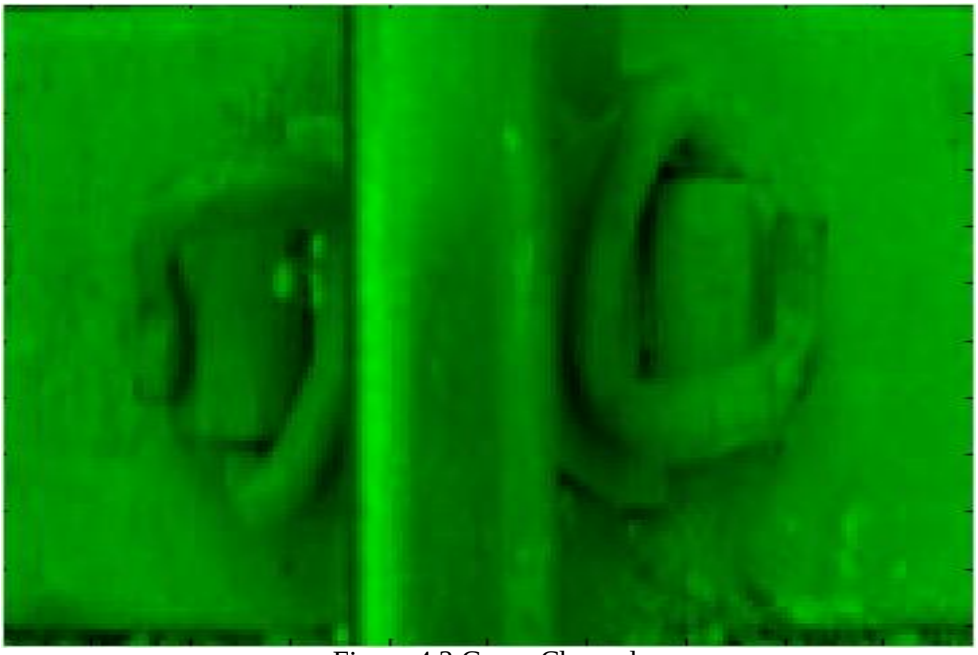


Figure 4.3 Green Channel

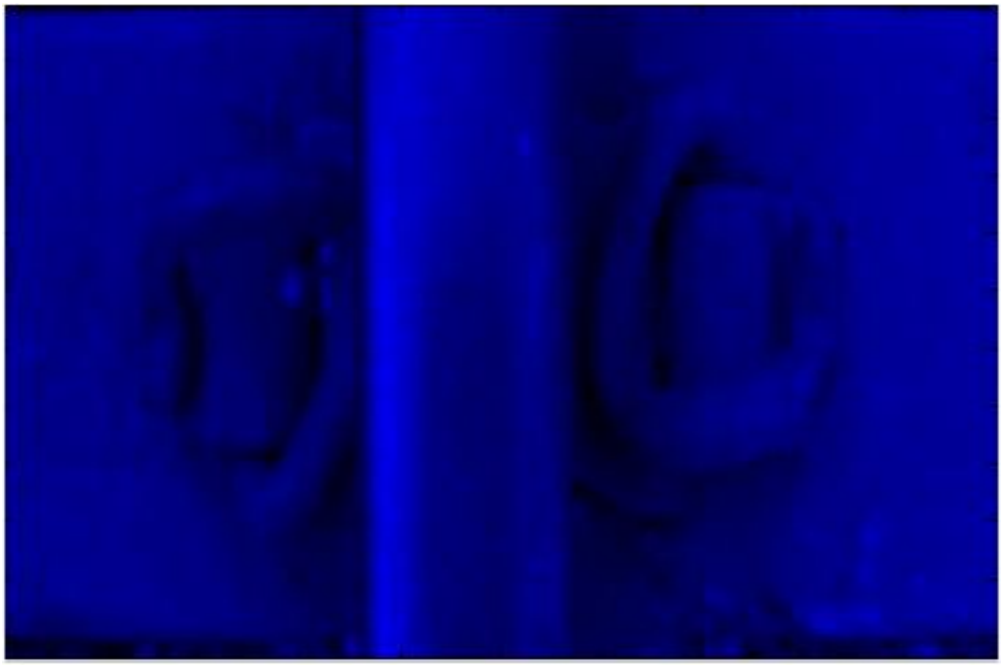


Figure 4.4 Blue Channel

3.2 Grayscale Images

Grayscale images are images whose pixels carry only the intensity information. It does not carry any color component. The intensity ranges from black (darkest) to white (brightest). Grayscale images are also called monochromatic. The intensity information is represented in 3 ways.

1. Total black is represented by 0 and white is represented by 1. The different intensity gray colors between them are represented as fractions.
2. Percentage representations where 0% denotes black and 100% denotes white. The colors between then are denoted by integers in percentage.
3. Pixel depth representation. For an 8 bit per pixel image 0 is black and 255 is white and for a 16 bit per pixel image 0 is black and 65535 is white.

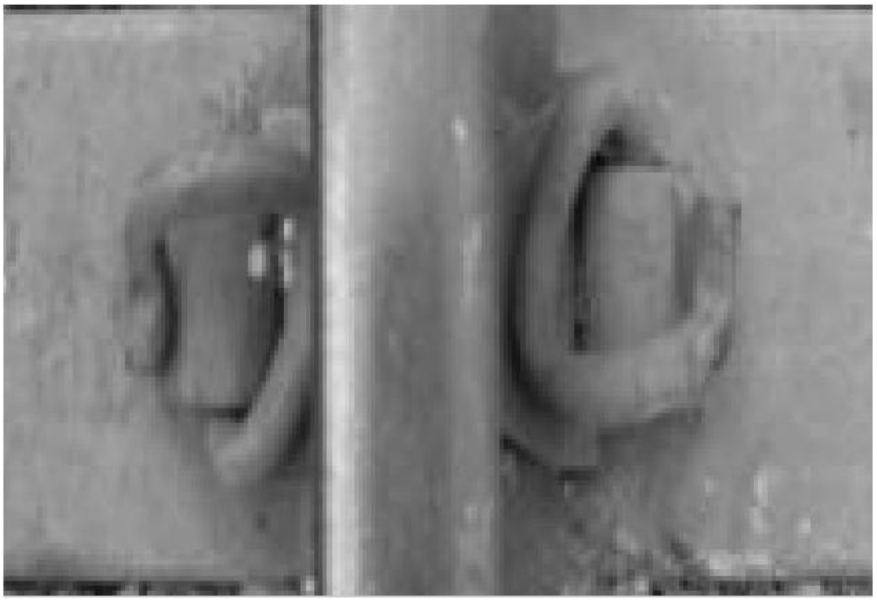


Figure 5 Grayscale Image

3.3 RGB to Grayscale Conversion

The main reason for this conversion is to reduce space and processing time. RGB consists of three channels whereas grayscale consists for one. The conversion is done by calculating the weighted sum in a linear RGB space.

The linear Y is calculated by .2989 R + .5870 G + .1140 B in MATLAB, the toolbox we have used in our work.

4. Technical Review

4. 1 Overview of Harris-Stephen

The Harris-Stephen[5] algorithm was developed to recognize image sequences from a moving camera by extracting and tracking image features. Matching images using just edges work when the camera is still but when it comes to a camera in motion just edges are not enough. The algorithm was developed based on two image sequences which contained corners and edges. Edge matching worked poor due to difference in fragmentation in sequences. Therefore the solution was to detect both corners and edges from an image. To detect corners the Moravec[12] corner detector was modified.

Moravec corner detector operates by using a local window in the image. The window shifts by small amount in various directions and calculate the average change in the intensity. Moravec algorithm has the following cases:

i) If the intensity of the image within the window is consistent then the shifts will result in small change.

ii) If the window finds and edge, moving the window along it will bring a small change and moving it perpendicular will bring a large change.

iii) If the windowed image is a corner or an isolated point, then every shift will result in a large change. Thus a corner can be detected when the change is large.

This is the mathematical representation

$$E_{x,y}= \sum_{u.v} w_{u,v} \mid I_{x+u,\, y+v} - I_{u,v}\mid^2$$

Where w is the image window, (x,y) are the shifts. The Moravec detector has the following drawbacks.

i) Due to the discrete number of shifts at every 45 degrees the response is anisotropic.

ii) As the window is rectangular and binary, the response in noisy.

iii) The detector responds edges more as the minimum of E is taken into account.

The Harris Stephens algorithm solves this problem.

i) Through performing analytical expansion about the shift origin all possible shifts can be covered.

ii) Using a Gaussian circular window can reduce noise

iii) Reformulating the corner measure can avoid the edge response

Based on the third solution, the change E, can be written as,

$$E(x,y) = (x,y)\ M\ (x,y)^T$$

Where M is a 2x2 symmetric matrix.

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}$$

If α and β are the Eigen values of M then they will be proportional to the principal curvatures of the local autocorrelation function [3]. The following cases can be drawn.

i) If both α and β are small then the images within the window is of consistent intensity.

ii) If α is high β is low, it indicates an edge.

iii) If α and β are high it indicates a corner.

The measure of the corner and the edge, the response, was used to select isolated corner pixels. The corner response is indicated by,

$$R = Det - kTr^2$$

Where $Det = \alpha\beta$ and $Tr = \alpha + \beta$. R is positive in corner detection and negative in edge detection. A corner is detected if the corner region pixel is an eight way local maxima. An edge detected if the response of an edge pixel is negative and local minima in either x or y direction.

4.2 Overview of Shi-Tomasi

Shi -Tomasi [3] detector detects corner points by measuring the feature dissimilarity between the first frame and the current frame in a motion. The idea of the algorithm is when the divergence of the features of the two frames is very large they are abandoned. The paper proved that pure translation is not adequate but linear warping and translations are. This algorithm also detects features which are best for tracking.

When a camera moves, the points in an image change in a very complex way. The amount of motion is called the displacement at a point X**(**x,y). X is any point in an image with the coordinates x and y. The displacement vector is represented in the following way

$$\sigma = DX + d$$

where D is the deformation matrix represented by

$$D = \begin{bmatrix} dxx & dxy \\ dyx & dyy \end{bmatrix}$$

d is the translation of the feature window's centre. Xmeasured with respect to the window's centre. During motion a point Xin the image I gets into a new point AX+ d in a new image J where is 1 + D:

$$I(X) = J\ (AX+ d)$$

Given two images I and J and a window size in I, tracking is finding out the 6 parametres in D. Although small window size is preferred but a small window size means tracking is harder as the changes noticed will be very less. While tracking the deformation matrix D is likely to be small so therefore it is safe to set D to the zero matrix. Regardess of the methods used for tracking every information of an image is not contained in all parts of an image. In order to solve this problem researchers have proposed to track corners or windows with a high spatial frequency content. But the problem with these points are they are based on an uninformed idea hence they are not assured to be the best for tracking.

Z is a symmetric matrix derived while computing image motion in [14].

$$Z = \begin{bmatrix} g2x & gxgy \\ gxgy & g2x \end{bmatrix}$$

If both eigen values of Z is large it can represent a corner. If both eigen values are small it means a coarse intensity profile within a window and one small and one large value means an undirectional texture pattern.

A feature with a high texture can be a bad feature. For exmaple, a scenarion in an image might appear in such a way that shows an edge but in not in reality. The measure of the dissimilarity by this algorithm will identify this issue.

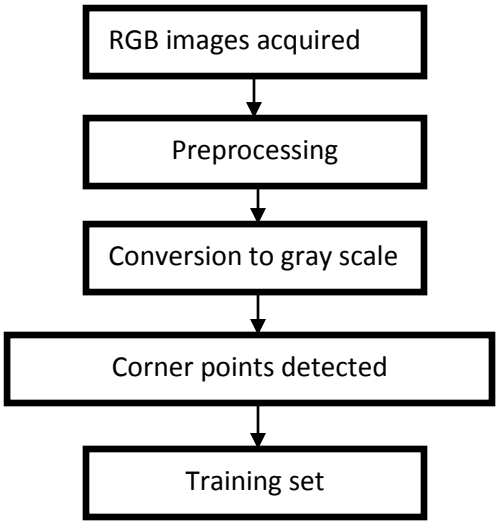
5. System Design

5.1 Tools

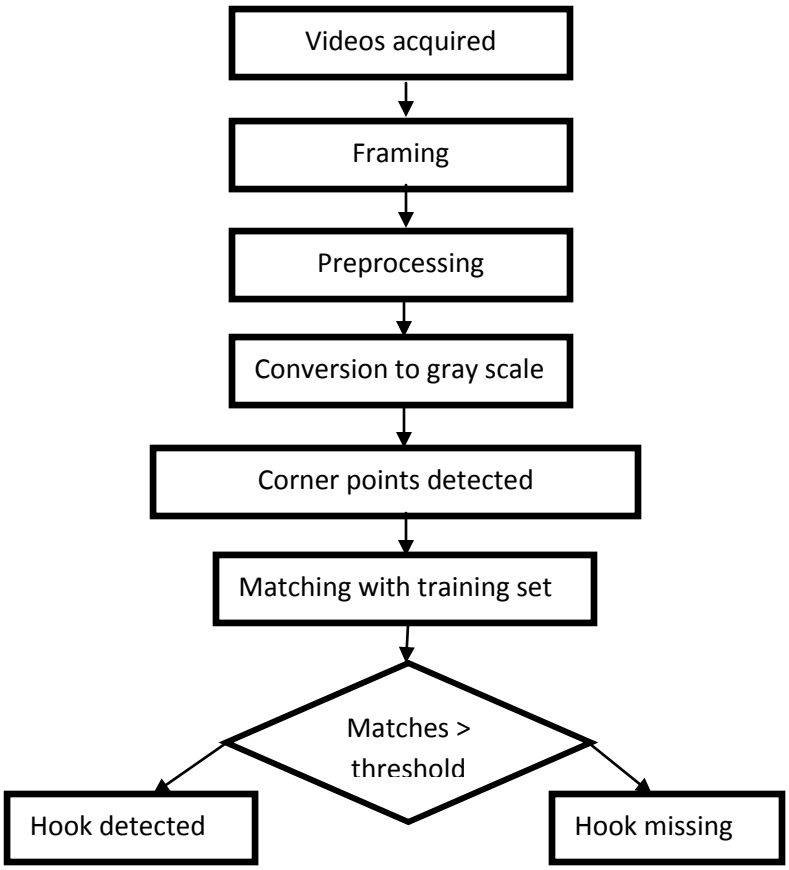
To build the proposed system I have used MATLAB R2013A.

Algorithms used are Shi- Tomasi corner detector and Harris-Stephen corner detector.

5.2 Methodology



Training Steps



Testing Steps

5.3 Data Collection

For this very experiment I could not find any universal image set. Therefore I had to go to the rail tracks physically and collect the training images and the videos. The pictures were taken with an eight mega pixel digital camera. In the papers I have studied so far related to rail way components [9], [10] and [11], all have used a vehicle with cameras mounted on it that ran on the rail line to take the images. I have built a manual cart where the camera was set at a height of 1.5 feet perpendicular to the hooks. One and a half feet is a decent height. This is because increasing it would make the object of interest move further away plus noise like stones would be more visible in the image frame. Lower than one and a half feet would bring the object closer but as we are using a regular digital camera blurring became an issue.

For this paper I have only considered totally visible hook plates and the lighting condition was daylight as shown in Figure 6.

Figure 6 A Visible Hook Plate

In the rail tracks I have discovered many scenarios where the hooks are completely or partially covered by stone layers or other objects. The following figures show such scenarios. I have not dealt with such situation in this paper.

Figure 7.1 Unclear Hook

Figure 7.2 Unclear Hook

Figure 7.3 Unclear Hook

I have taken the pictures and the videos from two different locations in Dhaka city and they are Khilgaon and Malibagh. The next task was to create the data set.

5.4 Creating Data Set

The steps required to create the data set were mentioned in the section 5.2. I have created the data set using 200 still true positive images.

5.4.1 Pre Processing

In the pre processing phase every image obtained were resized to 200 x 200 pixels. Reducing the image size to 200 x 200 decreases processing time and space. Then they were cropped by a window of 154 x 123. The reason behind the cropping was to reduce the stone noise which was visible in the image frame. The following diagram shows the image before and after cropping.

Figure 8 Samples of Training Images (after preprocessing)

5.4.2 Corner Point Detection Using Shi-Tomasi

The images were converted into grayscale and the reasons behind it have been mentioned earlier in section 3.3. Feature points of each training image were extracted using the Shi-Tomasi feature detector. The algorithm returns cornerPoints object. The cornerPoints object stores information about the feature points detected from the image. The pieces of information are the location, metric which explains how strong the detected features are and lastly the count which describes the number of point detected. After the points have been detected the corresponding feature vectors called the descriptors are extracted along with their associated valid points. Figure 9 shows the possible corner points (left) and the valid corner points after extraction (right). The descriptors are returned as binaryFeatures object. binaryFeatures object has the benefit to be used to match features extracted from different data. I have used this property to match which will be discussed later. The pieces of information contained by this object are the descriptors represented as M x N input matrix. The matrix contains M binary feature vectors stored in N containers of Uint8 class; NumBits which explain the number of bits per feature and lastly NumFeatures which describes the number of descriptors. The associated valid points are of the cornerPoints type as the input. These points contain the location of the descriptors. The descriptors are computed from an area around the interest point. If the area lies outside the image or on the edge of the image they are not considered as valid.

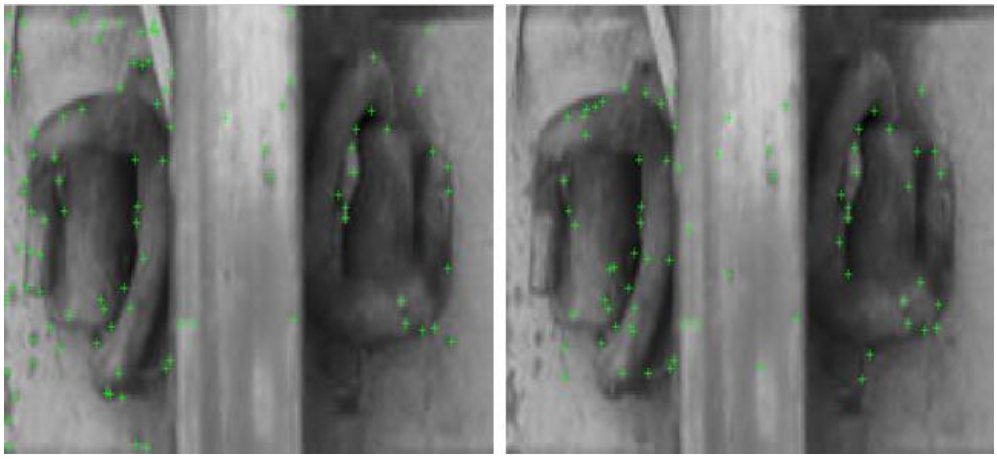


Figure 9 Possible Shi-Tomasi Feature Points (Left Column) Valid Feature Points (right column)

5.4.3 Corner Point Detection Using Harris-Stephen

The images were converted into grayscale and the reasons behind it have been mentioned earlier in section 3.3. Feature points of each training image were extracted using the Harris-Stephendetector. The algorithm returns cornerPoints object. The cornerPoints object stores information about the feature points detected from the image. The pieces of information are the location, metric which explains how strong the detected features are and lastly the count which describes the number of point detected. The figure 10 shows the Harris-Stephen feature points detected and the valid points.

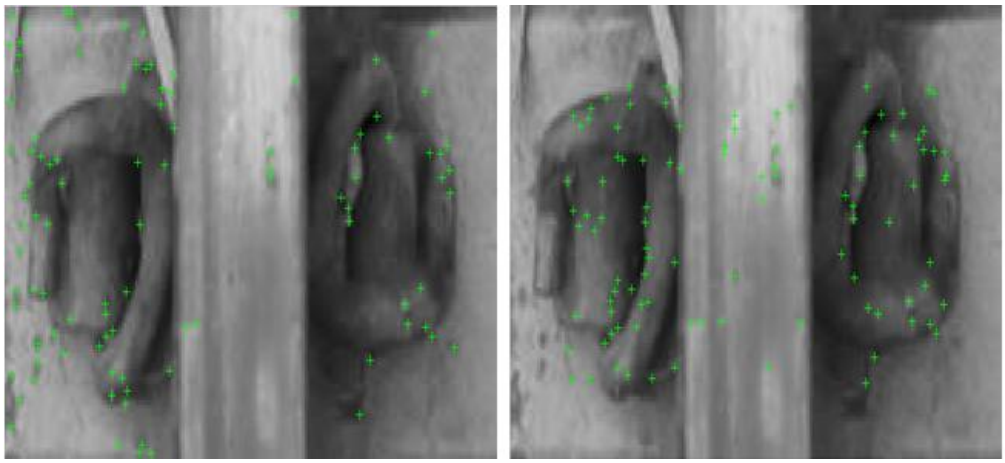


Figure 10 Possible Harris-Stephen Feature Points (Left Column) Valid Feature Points (right column)

After the points have been detected the corresponding feature vectors called the descriptors are extracted along with their associated valid points. The descriptors are returned as binaryFeatures object. binaryFeatures object has the benefit to be used to match features extracted from different data. I have used this property to match which will be discussed later. The pieces of information contained by this object are the descriptors represented as M x N input matrix. The matrix contains M binary feature vectors stored in N containers of Uint8 class; NumBits which explain the number of bits per feature and lastly NumFeatures which describes the number of descriptors. The associated valid points are of the cornerPoints type as the input. These points contain the location of the descriptors. The descriptors are computed from an area around the interest point. If the area lies outside the image or on the edge of the image they are not considered as valid.

5.4.4 Training

For this paper I have not used the conventional way of training data using classifiers like Adaboost, SVM or Neural Network. The descriptors and the location I have extracted using Shi-Tomasi and Harris-Stephen from all the images have been kept in individual cell arrays and these have been stored in the hard drive. The training, that is, the extraction of the descriptors from the training images is conducted once. If new training images are added the whole procedure will have to be run in order to extract a new set of descriptors.

5.5. Matching

At first I have acquired videos of rail tracks using my manual cart keeping the velocity as constant was possible. The videos contain conditions with both grounded and missing anchors. After acquisition I have segmented video into frames. Due to my manual cart its velocity was not totally constant. So I have arranged the frames into a uniform order. It takes 113 frames to go from one hook to the next. As the number of frames between two anchors is fixed, I have carried out the detection process only on those frames where the anchors were best visible.

If x =113, I have started detection from x-5 frames till x+5 frames. This has been done to increase the reliability of detection. From frames x-5 to x+5, feature points are detected using Shi-Tomasi and Harris-Stephen techniques separately. After detection, their corresponding features descriptors have been extracted. For each frame from x-5 to x+5, the extracted features are matched with the Shi-Tomasi and Harris-Stephen features of all the train images respectively.

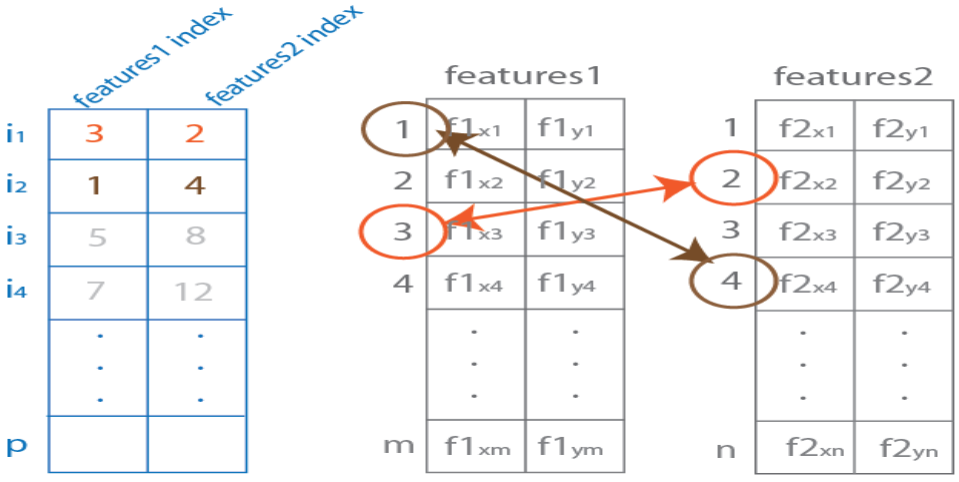


Figure 11 Feature Matching

The matrices named Features 1 and Features 2 in Figure 11 are the feature vectors of a train image and a particular frame respectively. When a particular feature matches between these two matrices, their corresponding indices are stored in a new matrix with the first column storing the index of the Features 1 matrix and the second column storing the index of the Features 2 matrix. As we are using two detectors, eventually there will be two matrices (one for each detector) storing the indices of the matched features. The number of rows of the new matrix measures the number of matches. The measure of matches of each detector is summed up and it must satisfy a threshold in order to conclude that an anchor has been detected. Figure 12.1 and figure 12.2 show the matched feature points between a taining image and a test image for Harris-Stephen and Shi-Tomasi respectively. Figures 13.1 – 13.4 illustratte the presence and the absence of rail anchors.

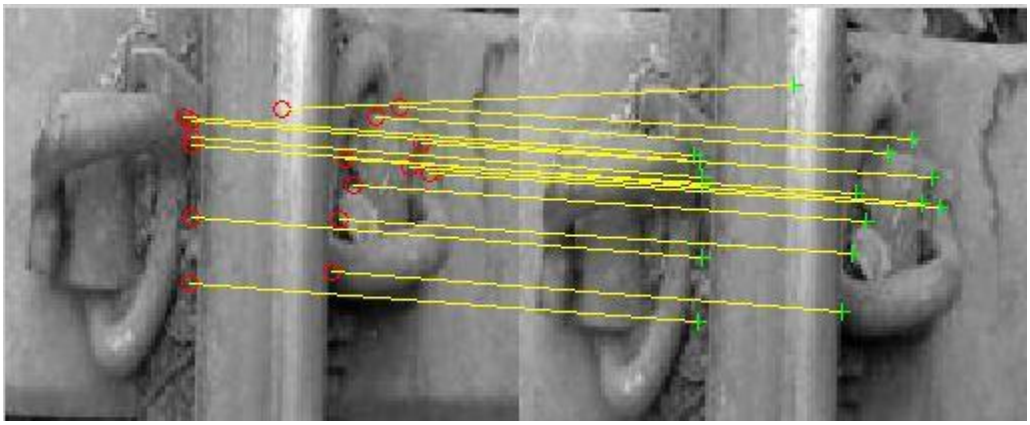


Figure 12.1 Matched Harris-Stephen Feature Points

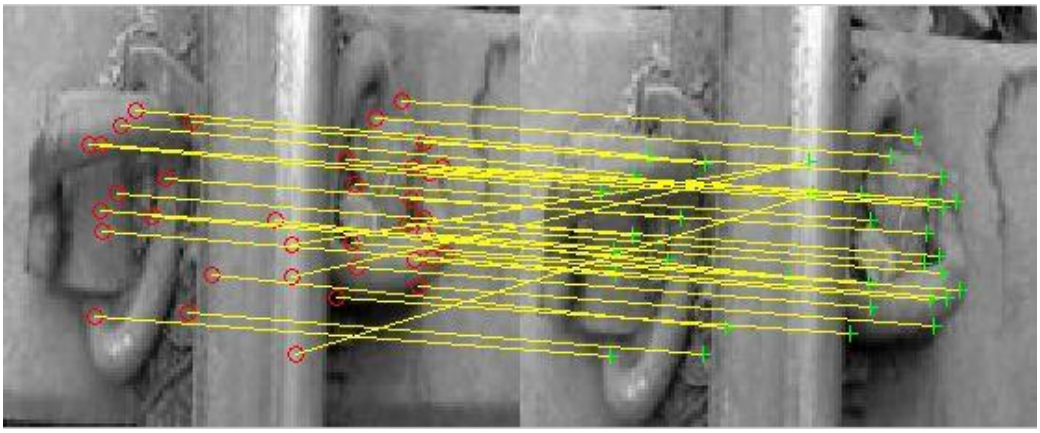


Figure 12.2 Matched Shi-Tomasi Feature Points

5.5.1 Threshold

The number of corners detected for each algorithm is different and hence the number of matches. From the videos used, the maximum number of matches for Harris–Stephen found is 55 and for Shi–Tomasi it is 169. For Harris –Stephen, $\frac{1}{6}$ ×55=9.167 and for Shi–Tomasi, $\frac{1}{6}$×169 = 22.167. Summing up the results we get, 9.167 + 22.167 = 37.334. Flooring the sum we get 37 and this is the threshold we have considered for our proposed approach. The formula for finding the threshold is generalized as follows:

$$T = \lfloor [\frac{1}{6} (Max_{Harris - Stephen} + Max_{Shi - Tomasi})] \rfloor$$

where T is the threshold, $Max_{Harris - Stephen}$ is the maximum number of matches found through the implementation of Harris–Stephen algorithm and $Max_{Shi - Tomasi}$ is the maximum number of matches found through the execution of Shi–Tomasi algorithm.

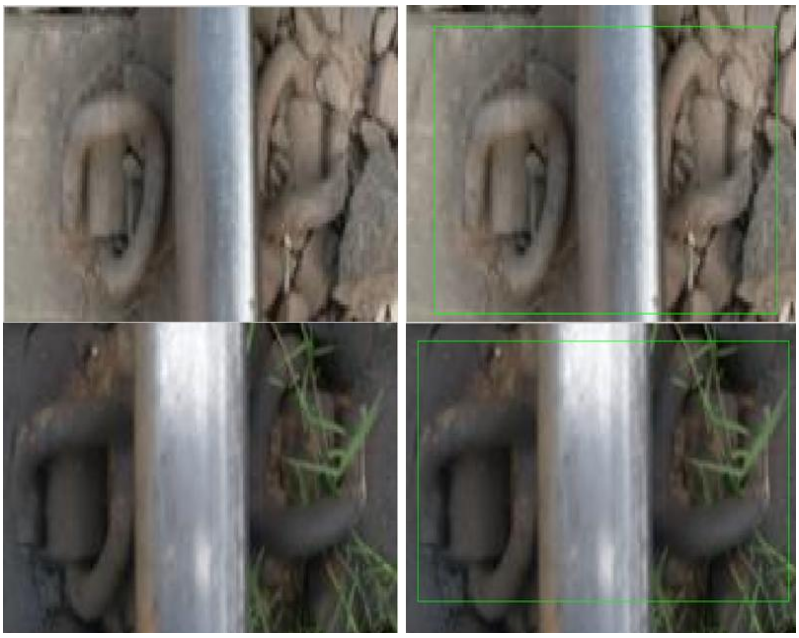


Figure 13.1 True Positive

Figure 13.2 True Negative

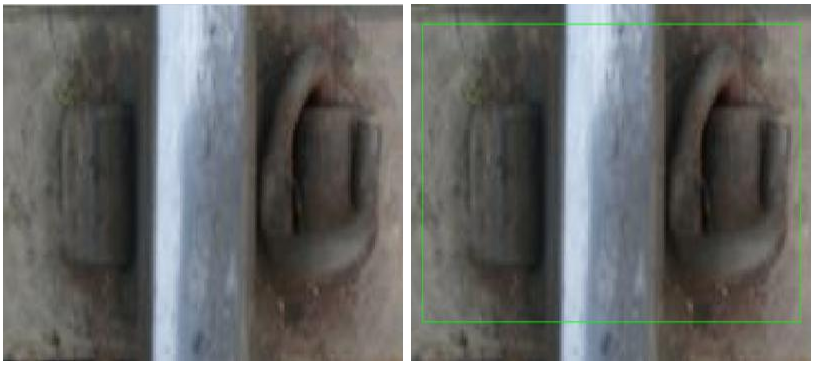


Figure 13.3 False Positive

Figure 13.4 False Negative

6. Experimental Results

The experimental results of both the techniques are shown in table 1.

Image	Condition	Shi-Tomasi Features	Harris-Stephen Features	Result
1	T	78	35	✓
2	T	78	22	✓
3	T	28	8	✗
4	F	38	12	✓
5	T	82	36	✓
6	T	105	44	✓
7	F	22	10	✗
8	T	122	16	✓
9	T	112	24	✓
10	F	152	30	✓
11	T	87	42	✓
12	F	36	8	✓
13	T	93	38	✓
14	T	129	24	✓
15	T	169	37	✓
16	T	169	52	✓
17	T	109	55	✓
18	T	146	38	✓
19	F	169	55	✓
20	T	19	9	✗

Table 1 Experimental Result of using Shi-Tomasi and Harris-Stephen Feature Detectors

7. Accuracy and Comparison

I have conducted my detection technique on three different videos. Video one contained five true positive and two true negative cases, the second one contained four true positive and two true negative cases and the last one contained 6 true positive and one true negative case. True positive image accuracy and true negative image accuracy have been calculated using the following formula.

For video 1,

$$\frac{true\ positive}{total\ \ positive} \times 100\% = \frac{4}{5} \text{ x } 100\% = 80\%$$

$$\frac{true\ negative}{total\ negative} \times 100\% = \frac{1}{2} \text{ x } 100\% = 50\%$$

The overall accuracy is calculated as follows.

$$\frac{true\ positive + true\ negative}{total\ number\ \ of\ images} \times 100\% = \frac{5}{7} \text{ x } 100\% = 71.4\%$$

The F1 score is calculated as follows.

$$\frac{2TP}{(2TP+FP+FN)} \times 100\%$$

Where TP denotes true positive, FP denotes false negative and FN false negative.

$$\text{F1} = \frac{2\times4}{(2\times4)+2} \times 100\% = 80\%$$

For video 2,

$$\frac{true\ positive}{total\ \ positive} \times 100\% = \frac{4}{4} \text{ x } 100\% = 100\%$$

$$\frac{true\ negative}{total\ negative} \times 100\% = \frac{2}{2} \text{ x } 100\% = 100\%$$

The overall accuracy is calculated as follows.

$$\frac{true\ positive + true\ negative}{total\ number\ \ of\ images} \times 100\% = \frac{6}{6} \text{ x } 100\% = 100\%$$

The F1 score is calculated as follows.

$$\frac{2TP}{(2TP+FP+FN)}\times 100\%$$

Where TP denotes true positive, FP denotes false negative and FN false negative.

$$\text{F1} = \frac{2\times 4}{(2\times 4)}\times 100\% = 100\%$$

For video 3,

$$\frac{true\ \ positive}{total\ \ \ positive}\times 100\% = \frac{5}{6}\ \text{x}\ 100\% = 83.33\%$$

$$\frac{true\ \ negative}{total\ \ negative}\times 100\% = \frac{1}{1}\ \text{x}\ 100\% = 100\%$$

The overall accuracy is calculated as follows.

$$\frac{true\ \ positive\ + true\ \ negative}{total\ \ number\ \ \ of\ \ images}\times 100\% = \frac{6}{7}\ \text{x}\ 100\% = 85.7\%$$

The F1 score is calculated as follows.

$$\frac{2TP}{(2TP+FP+FN)}\times 100\%$$

Where TP denotes true positive, FP denotes false negative and FN false negative.

$$\text{F1} = \frac{2\times 5}{(2\times 5)+1}\times 100\% = 90.91\%$$

The experimental analysis is shown in a tabular form.

Video	% True Positive	% True Negative	% Accuracy	% F1
1	80	50	71.4	80
2	100	100	100	100
3	83.33	100	85.7	90.91
Average	87.78	83.33	85.7	90.3

Table 2 Comparison Table

8. Conclusion and Future Work

In this paper the cart I have used was completely manual. In the future I am planning to design an automatic cart for our system which will run along the rail track and collect input data, process it in real time and store the information with the exact GPS location for future maintenance work. Along with the GPS receiver, I am planning to install a gyroscope and an accelerometer sensor in that cart which will let us know whether there is a height mismatch between both sides of the lines or not.

9. References

1. http://www.railway.gov.bd/train_accidents.asp
2. Automatic Detection of Defective Rail Anchors and Measurement of Rail Line Expansion Joint Gaps, Rubayat Ahmed Khan and Samiul Islam, BRAC University 2014
3. Good Features to Track, JianboShi, Cornell University, Carlo Tomasi, Stanford University, IEEE Conference on Computer Vision and Pattern Recognition, June 1994
4. SURF: Speeded Up Robust Features, Herbert Bay, Tinne Tuytelaars, Lue Van Gool, Katholieke Universiteit Leuven
5. A Combined Edge and Corner Detector, Chris Harris and Mike Stephens, Manchester 1988
6. BRISK: Binary Robust Invariant Scalable Keypoints, Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart, Autonomous Systems Lab, ETH Zurich
7. A Comparison of Keypoint Descriptors in the Context of Pedestrian Detection: FREAK vs. SURF vs. BRISK, Cameron Schaeffer, Stanford University CS Department
8. A Theory Based on Conversion of RGB image to Gray image, TarunKumar,AssistantProfessor, Computer Science and Engineering Department, Vidya College of Engineering, Meerut (U.P) & Karun Verma, Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala (Punjab)
9. Enhanced Rail Component Detection and Consolidation for Rail Track Inspection, Hoang Trinh, Norman Haas, Ying Li, Charles Otto, SharathPankanti, IBM T. J. Watson Research Centre, 19 Skyline Dr, Hawthorne, NY 10532
10. Automatic Detection of Objects ofInterest from Rail Track Images, YohannRubinsztejn, University of Manchester
11. Visual Inspection of Railroad Tracks, PavelBabenko, University of Central Florida. 2006
12. Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, Hans Moravec, Stanford University, 1980